B#

ELABORATION AND TESTING OF NEW METHODOLOGIES FOR AUTOMATIC ABSTRACTING

BERNADETTE SHARP BSc, MPhil, MBCS

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

October 1989

UNIVERSITY OF ASTON IN BIRMINGHAM

ELABORATION AND TESTING OF NEW METHODOLOGIES FOR AUTOMATIC ABSTRACTING

BERNADETTE SHARP BSc, MPhil, MBCS

Doctor of Philosophy

October 1989

THESIS SUMMARY

The primary objective of this research was to understand what kinds of knowledge and skills people use in "extracting " relevant information from text and to assess the extent to which expert systems techniques could be applied to automate the process of abstracting. The approach adopted in this thesis is based on research in cognitive science, information science, psycholinguistics and textlinguistics.

The study addressed the significance of domain knowledge and heuristic rules by developing an information extraction system, called INFORMEX. This system, which was implemented partly in SPITBOL, and partly in PROLOG, used a set of heuristic rules to analyse five scientific papers of expository type, to interpret the content in relation to the key abstract elements and to extract a set of sentences recognised as relevant for abstracting purposes. The analysis of these extracts revealed that an adequate abstract could be generated.

Furthermore, INFORMEX showed that a rule base system was a suitable computational model to represent experts' knowledge and strategies. This computational technique provided the basis for a new approach to the modelling of cognition. It showed how experts tackle the task of abstracting by integrating formal knowledge as well as experiential learning.

This thesis demonstrated that empirical and theoretical knowledge can be effectively combined in expert systems technology to provide a valuable starting approach to automatic abstracting.

Key Terms: automatic abstracting, natural language processing, expert systems, computational linguistics

DEDICATION

This thesis is dedicated to my family. A token of thanks.

ACKNOWLEDGEMENTS

I wish to express my deep appreciation to my supervisor Professor F. E. Knowles for his guidance and support during the research and writing of this thesis. I would like to thank Wolverhampton Polytechnic, Aston University and Coventry Polytechnic for allowing me to carry out this research.

I am grateful to Catherine Johns Lewis, Alan Harget, Sobhi Zora and Gabriel Kilo for helpful discussions. I am grateful to many colleagues as well as friends who have taken interest and contributed to the various experiments carried out in my work. I am particularly grateful to many abstractors namely from LISA, RAPRA, Milk Marketing Board and BCIRA who kindly offered their collaboration which, in some cases, lasted over several years.

I am indebted to Henry Carvel and Douglas Owen whose suggestions and comments on various drafts were always generous and constructive. I also wish to thank Sheila Allen and Iloria Whittier, who transformed the barely legible drafts into the present pleasing form.

Finally, my sincere thanks to Denis, my husband, for his support and encouragement, and my children, Timothy and Daniel, who have endured my obsession to complete this research before 1990.

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

INTRODUCTION

Information has become an essential basic commodity for today's economy, industry and education. Access to timely and accurate information is now recognised as being a key factor in the development of successful organisations. This is reflected in the volume of information generated, retrieved and processed, which is unprecedented in history. To give an indication of the growth of information needs, computer databases hardly existed in the 1960s, in the 1970s the number of on-line searches rose into the hundreds of thousands, while in the 1980 these rose into the millions [Stonier, 1989 : 224]. Within the educational sector, this is illustrated by the number of scientific and technical publications, which is growing at an exponential rate. About 8312 papers were abstracted by INSPEC in the field of artificial intelligence since 1978. In 1988, 10220 academic books were published in science and technology. Whilst the growth rate of published literature is increasing, the time available for retrieving the most relevant information is limited.

In order to cope with the growth of literature, information specialists have developed special skills and techniques for storing, searching and disseminating information. The computer has added a new and powerful technology to the practice of information storage and retrieval. Analysis and synthesis of information, though possibly by computers, are still largely carried out by skilled human labour. The availability of online retrieval systems has transformed the role of indexing and abstracting. Indexing facilitates the efficient retrieval of information while abstracting enhances judgements about the relevance of the retrieved information. Abstracts have, as a result, become important surrogates for original articles. Without them, keeping up-to-date would be impossible with the growth of literature. Abstracting is now a key segment of the information industry, and recognised as a costly and labour-intensive process [Borko et al, 1975]. The need to generate automatic abstracting facilities seems therefore extremely crucial to effective dissemination of information. More and more research studies are focusing their attention on information organisation and information processing and on this aspect of it in particular. Recent advances in this area have led to the emergence of new disciplines, namely computational linguistics, artificial intelligence, and cognitive science.

The terms "information processing" and "information analysis" have, in the past, been the concern of many disciplines, namely philosophy, psychology, linguistics, information science and computer science. Although each of these disciplines look at information from different perspectives, they all recognise the common core to be the study of language as a process of communication. Linguists take human language as their special domain of inquiry. Their aim is to understand how linguistic knowledge is stored in the mind, how it is acquired, how it is perceived and used. While sociolinguists tend to concentrate on the social interaction manifested in conversation, psycholinguists are more concerned with issues related to language comprehension. Information scientists are concerned with the processes of organisation, codification, retrieval, dissemination and acquisition of information. Researchers in cognitive science, on the other hand are interested in the mental process and strategies that are used in the perception and production of language, whereas computer scientists, particularly artificial intelligence workers, seek to understand the relationships between language processing and general information processing in order to develop computational models in these domains.

Although the concept of information is central to many disciplines, there is no consensus as to its nature [Walker, 1981], partly because we still do not understand the nature of information [Stonier, 1989 : 228]. Numerous definitions have been proposed in the literature. It can refer to meaningful organised data, the capacity of a communication channel [Shannon et al, 1949]. It can be taken to designate a process of interaction between what is already given and what is new. This is different from the mathematical concept of information which is the measure of its unpredictability [Halliday, 1978]. Derr defines information as "an abstract , meaningful representation of determinations made of objects ... [It] has derivative properties which enable it to communicate, inform, empower and to exist in some findings from an analysis of language " [Derr, 1985 : 497].

Given the lack of consensus on proposed definitions of information, it is reasonable to adopt the definition of most information workers. Information is related to the process of communication of data between two people, each person makes whatever use he/she can of it. When this information is publicly recorded it takes the form of objective knowledge available to everyone [Rowley et al, 1978].

This process of communication involves simultaneous related activities by producer and receiver of communication as illustrated in Figure 1.1 [Winograd, 1982 : 14].

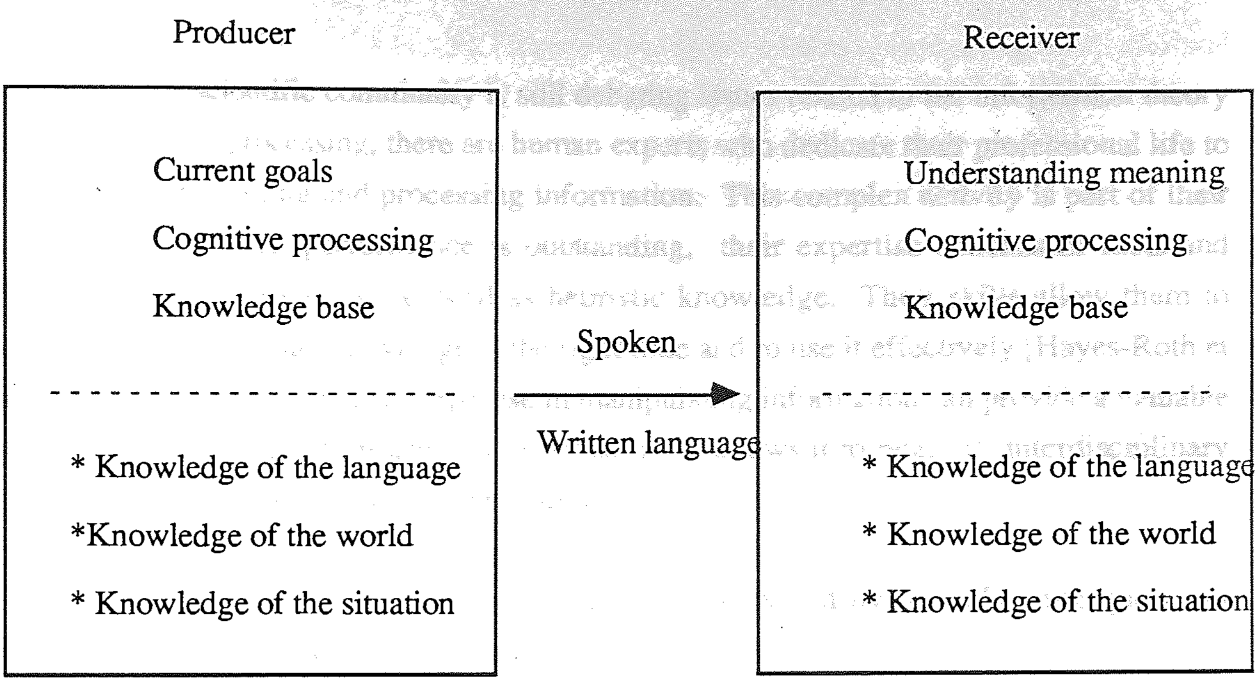


Figure 1.1 Basic model of communication

According to Winograd's model, the producer has communicative goals that include conveying information to the receiver, expressing attitudes, establishing context for subsequent goals. He/she maps the message onto a sequence of utterances. The choice of words, phrases and sentences and patterns of emphasis play an important part in providing the receiver with appropriate cues to infer his/her communicative goals [Winograd,1982: 13]. Complex as these linguistic manifestations are, they are not sufficient to achieve successful communication. Both the producer and receiver of information have to activate two other sets of knowledge for the message to be appropriately conveyed: knowledge about the world and knowledge about the context in which the utterance is performed. Information, language and knowledge are key issues that require a thorough investigation in order to achieve a clearer understanding of the human competence to organise and process information.

Since the study of how humans interpret and communicate information is central to many disciplines, any research into aspects of information processing must integrate the contributions of all the above mentioned disciplines. Each discipline brings to the research a special emphasis, and none of the disciplines encompasses the entire subject of information processing. Thus, research into information processing must be of an interdisciplinary nature. Such an approach however, is too complex and too ambitious to be carried out as a realistic programme of study.

While the scientific community is still debating issues related to the information theory and information processing, there are human experts who dedicate their professional life to organising, abstracting and processing information. This complex activity is part of their daily routine. Their performance is outstanding, their expertise consists of facts and theories found in textbooks as well as heuristic knowledge. Their skills allow them to retrieve the appropriate knowledge at the right time and to use it effectively [Hayes-Roth et al, 1983]. The study of human expertise in manipulating information can provide a valuable approach to the study of information processing. It allows it to retain its interdisciplinary nature and hence make it a feasible approach.

The long term objectives of this research are motivated by the following questions regarding the organisation and processing of information:

1. What knowledge does an expert use when processing information in written texts?

2. Can one represent the above knowledge using computational formalisms?

3. Can one capture human expertise using computational models based on artificial intelligence techniques?

It is believed that by using artificial intelligence techniques one can gain insights about how experts perform complex tasks and learn about their skills in integrating various kinds of knowledge required to achieve such an outstanding performance. Such an approach is a powerful alternative methodology for exploring and testing hypotheses about human processing of information.

The immediate objectives here are to apply the above approach to study how human experts generate an abstract from scientific papers. These objectives are summarised in the following questions:

1. How does a human expert identify key information in a paper to generate an abstract?

   1.1 What kinds of knowledge are required to identify key information?
   1.2 What skills do they use?

2. Could the process of abstracting be simplified so as to be modelled using computational tools?

3. Could one capture experts' knowledge and skills required to extract information using artificial intelligence techniques? Could this extract be used as an adequate basis for abstracting?

Due to the interdisciplinary nature of the subject of the study, this research project has many constraints imposed on it. Initially, scientific papers in the field of computer science are considered. Even within this discipline, only those of an expository type in expert systems domains are to be used for experimentation purposes. As the work is motivated by the desire to understand better how human experts manipulate information, it is necessary to concentrate on one specific aspect of information processing. Abstracting, being one of the important activities in the dissemination of information, is taken as the focus of the present research.

In this thesis, the human expert provides the point of departure and remains at the centre of the process of abstracting techniques. By eliciting the strategies and techniques that abstractors use in generating an abstract, one uncovers the range of knowledge and skills involved in extracting key information from a text. Part of this knowledge will involve linguistic knowledge , that is knowledge required to read and understand a text. Another part will relate to cognitive psychology, that is the mental processes that abstractors undergo when analysing a text, identifying and organising the main ideas and producing an abstract. In attempting to produce a computational model of abstracting, the thesis uses artificial intelligence techniques, namely the expert system formalism, which will serve, on one hand, as a valuable tool to gain insights into how experts perform complex tasks and, on the other, as a way of integrating empirical strategies with theoretical findings from disciplines such as linguistics and cognitive science.

This thesis is organised in seven Chapters. The second Chapter consists of two major parts: the first part shows the important role that abstracts play in the transfer and dissemination of knowledge whereas the second part surveys major contributions to the study of abstracting from various disciplines, namely artificial intelligence, linguistics and cognitive science. Because this thesis is interested in modelling the abstracting process using computational techniques, Chapter 3 analyses some of the difficulties involved in the development of an automatic abstracting systems and describes an alternative approach deeply rooted in the artificial intelligence discipline. This approach is called a knowledge base system and is described in Chapter 4. The development of such a system involves eliciting the knowledge and skills used by the human expert in generating an abstract, finding a suitable representation for computer manipulation and building an engine in the system to apply this body of knowledge on a given text and to produce an extract. These technical stages are described in Chapter 5 while the issues related to the evaluation of this system and its performance are discussed in Chapter 6. Chapter 7 concludes the research and suggests areas for further research.

CHAPTER 2

THE ORGANISING AND PROCESSING OF INFORMATION

THE ORGANISATION AND PROCESSING OF INFORMATION

1. Introduction

This chapter first reviews the importance of information in today's society. It then explores the role of the abstract in the dissemination of information and analyses the contributions of abstractors in the transfer of knowledge from the originator of the information to the ultimate consumer. Because of the information explosion which characterises our society of today the need for effective access to new published material is acknowledged by all its members and this is why many researchers have redirected their efforts towards understanding how people analyse and process documents with a view to automating some of the human tasks. The study of information processing embraces many disciplines and scholars from information science, linguistics, cognitive science and artificial intelligence have undertaken this study from different starting points and for different "sub-purposes". The third part of this Chapter surveys influential work carried out by information scientists in the field of abstracting and explores the contributions of other fields to text understanding and text processing. This chapter does not attempt to produce an exhaustive literature survey; instead, it aims to draw together the key areas in the various disciplines relevant to the study of abstracting.

2. The Importance of Information

Industry, commerce, business and education are characterised by rapid technological and organisational change. Decision-making processes under such conditions become much more complex. Scientists and technologists feel the pressing need to access up-to-date and accurate information; they need to know not only what has appeared but also what is in the process of production. Information is the crucial key to survival, and access to specific information is perceived today as leading to knowledge and, hopefully, often to success and power.

The importance of information within organisations is reflected internally in the establishment of Management Information Systems (MIS), Decision Support Systems (DSS) and Corporate Databases (CD). Externally, it is reflected in the growth of on-line retrieval systems and information services. The expansion of the information industry is an indication of the economic value placed on accurate information. The concept of information is not new but its strategic character is receiving greater attention today as it increasingly assumes a larger and crucial role in the making of decisions that affect both the future and the operational activities of the organisation.

A major problem that faces all information users is the exponential growth in the volume of information [Rowley, 1978: 25]. The number of people generating scientific and technical knowledge is now increasing vastly. According to "Which International Periodical Directory" about 100,000 regular serials are issued in the world each year, of which 60,000 are periodicals.

The information explosion has exerted increased pressure on the community of information specialists. Although storage and retrieval of literature is aided by new technology, the organisation and analysis of information is still carried out by skilled human labour. The work of collecting, indexing and abstracting published papers becomes an immense and complex task if truly effective dissemination is sought. Scientific documentation has a key role in the expansion of human knowledge - if it ceases to fulfil its role, the future development of science is seriously jeopardised.

This situation is aggravated by virtue of the fact that published material can no longer be contained within one single discipline. Researchers working in the area of natural language processing, for instance, have to consult at least ten different information sources, within databases for linguistics, computer science, information science and psychology. The computerisation of information services has certainly made access to published material across disciplines a reality; however, it has imposed another set of constraints on the scientific community. Effective searching of the stored literature requires training in and familiarisation with computer technology. Consequently, this community relies more and more on abstracting and indexing organisations which are becoming truly vital links in the chain of communication and in the transfer of knowledge between the producer of information and the ultimate consumer.

The scientific community has, as a result, become more aware of documentation problems and many researchers have shifted the focus of their attention to solve them. Computer-based approaches to the study of organisations and the use of information are having an important impact. This new area of research offers important tools which stand to assist information specialists to grapple with the substance of their tasks. The general trend is to devise computer systems which model human approaches and to delegate some of the intellectual functions involved to these systems.

3. The Role of the Abstract in the Transfer of Knowledge

The analysis, organisation and processing of information is an important human competence. Abstractors, in the profession of information scientist, dedicate their professional lives to scanning documents, interpreting their contents and generating abstracts. These abstracts are a concise representation of the contents of documents without added interpretation or criticism [Weil, 1970].

Their purpose is twofold: to draw attention to the information contained in the original document and to provide a sufficient basis for a user to decide whether or not the document is relevant and hence offer a criterion for consulting the original [Weisman, 1972]. The function of the abstract helps distinguish it from other forms of terse writing such as a brief, an annotation, a precis, a summary or a synopsis. A brief differs from an abstract in that it is "a shortened restatement of an argument, in outline form, emphasising the principal contention and supporting statements or evidence" [Brown et al, 1971: 4]. An annotation is simply a brief note added to a title; it explains, evaluates or comments on the title. Whereas a precis is a brief statement of the essential statement or facts mentioned in the document, a summary functions as a short statement of the main points of a document with a view to aid the reader's comprehension or recall by showing the conceptual relationships binding these points. It may emphasise some aspects to the exclusion of others. In contrast to an abstract, a synopsis is usually written from a point of view other than that of the original document although it outlines its important elements and features.

By presenting sifted information, the abstractor reduces the amount of time a reader needs to spend in gathering and selecting information. Most abstracts state the purpose, methodology, results and conclusions presented in the original document. Findings and other information incidental to the main purpose of the document may be reported in the abstract but they must not distract attention from the main theme.

Since an abstract must be intelligible to a reader without the need for him/her to refer to the document, completeness, accuracy and length are key elements. An abstract must be self-contained, clear, and concise. For most published papers, an abstract of 250 words is considered adequate even for long documents, such as reports, theses, etc. An abstract should never exceed 500 words.

Abstracts may be classified into three categories: 1) indicative or descriptive abstracts, disclosing significant specific information found in the document; 2) informative abstracts presenting the conceptual content of documents and summarising the essential ideas; and 3) critical abstracts, being heavily editorial as they include critical comments with respect to documents [Harris, 1973]. Although many abstracts are clearly either indicative or informative, in actual practice others are often combined, depending on the type of document being abstracted.

There is a growing consensus that these traditional methods of producing abstracts are no longer meeting today's needs. Many complaints are levelled at the inadequacy of abstracts or even the total lack of them. H.W. Wellish pointed out in his introduction to "Indexing and Abstracting" that in many cases "abstracts were too long and contained data and statements which, in the context of a chronologically arranged bibliography, would have been redundant". Some of the abstracts he investigate were badly written or padded with meaningless statements. He also claims that "in many instances the so-called 'abstract' was written more like an introductory paragraph of an article" (sometimes in the first person!) [Wellish, 1980]. Abstracting from the flood of documents published each year presents a virtually impossible task if it is to be carried out manually, especially if deficiencies of the above sort abound. Only a selection can be abstracted to an adequate standard by manual methods.

The current environment suggests that it will become more and more difficult for researchers to keep up with scientific publications on a given topic. As a result, more and more researchers rely on abstracting journals and abstractors' skills to "select" relevant papers for them. This qualitative change in the role of the abstract in communicating knowledge along with the growth of published material, has spurred research efforts to automate the abstracting process. The following section reviews major studies carried out in this area within information science. Abstracting has also received great attention among researchers in artificial intelligence, linguistics and cognitive science. A survey of influential work relevant to the aforementioned research area is also included. Developments in computer-based procedures for working with concepts of information, knowledge and language are also relevant to such disciplines as philosophy of language and also of science and education. A brief introduction to their contributions to information processing is also given.

4. Contributions to Abstracting from Information Science

Interest in abstracting can be traced back to the early 1950's when computers were no longer used exclusively for mathematical applications. Although machine translation was among the first non-numerical processing applications, the domain of natural language processing (as opposed to computer language processing) was soon expanded to include research in automatic abstracting and text processing. H.P. Luhn was one of the earliest researchers to realise the potential of computer technology for solving information problems [Luhn, 1952]. A few years later, he published a technique for automatic abstracting based on a statistical analysis of documents [Luhn, 1958].

This technique involved a frequency count by computer of all the words in the text of an article, the exclusion of certain common words such as prepositions, articles, pronouns and the production of a list of non-common words ranked by frequency of occurrence. Luhn then argued that keywords have greatest significance when used in clusters. Sentence significance was thus taken to be the value of the most significant cluster in it and the highest scoring sentences were printed out to form the abstract. [Luhn, 1958b].

In the same year, P. Baxendale who was also interested in automatic indexing, experimented on an IBM 650 with three methods making use of so-called representative words in documents . The first method was based on the idea of deleting function words from the text and assigning a frequency count to the remaining content words. Those sentences that scored highly were considered to be the most representative and are thus eligible for inclusion in the abstract. The second method dealt with topic sentences in paragraphs. After an investigation of 200 paragraphs she found that in 85 percent of the paragraphs the topic sentence was the first sentence and in 7 percent the last sentence of a paragraph [Baxendale, 1958: 355]. The abstract was made up of these topic sentences. The third method concentrated on prepositional phrases in sentences, and selected those sentences with high frequencies of these phrases.

Research in automatic abstracting has, by and large, concentrated on attempting to improve Luhn's approach by applying various measures of "representativeness" for words and sentences in the text. Edmunson and Wyllys have suggested comparing word frequency in a given text to its frequency in large corpus as an indication of its importance in the text [Edmunson et al, 1961: 226]. Edmundson has devised three methods regarding sentence selection: the Cue method is a method based on pre-selected list of words, the Title method assigns a higher importance to words if found in title and headings, and the Location method is function of location in the text and type of headings such as "method" or "investigation". Best results were achieved when the three methods were combined.

The Oswald study resembled Luhn's approach in selecting the representative word-types and determining representative sentences. His study appears to be the first to attempt to employ a group of words recorded as a multiterm [Edmundson et al, 1959: 5].

The Ramo-Wooldridge study, directed by Edmundson, was the largest research project undertaken in automatic abstracting at that early period. The research group investigated three main aspects of abstracting: (1) how consistent are people in extracting sentences from texts; (2) what is the indicative probability for a given sentence to be selected for extraction by computer program; (3) what types of relationships can be expressed by sentences and what criteria can be derived for extracting sentences. These aspects were then incorporated into a computer program for experimentation purposes. As Borko pointed out in his book, the results of the Ramo-Wooldridge study justified the assertion that a workable empirical technique for the semantic analysis of texts can be developed [Borko, 1967: 156].

The 1950's and 1960's witnessed a slow development in the field of automatic abstracting, mainly due to the cost and limitations of the hardware available at that time. The advances of computer technology and the reduction in cost of various computing units in the 1970's contributed to the rise in research activities in automatic abstracting. Linguistics proper, statistical approaches to linguistics, and software and hardware technologies had all advanced a great deal. On the hardware side, machinery was available for faster and cheaper access to data organised and stored in computer memory. Prior to 1970 most of the research carried out in machine abstracting of natural language text concentrated on measuring sentence significance using statistical methods as a basis for emphasising the semantic aspect of documents. As Climenson, Hardwick and Jacobson pointed out, "the semantic approach evolved naturally from the traditional interest in semantic classification, descriptive terms, cross-indexing of related subject fields etc. A machine-stored thesaurus, while well suited to the work-matching capabilites of a computer, presents a tremendous intellectual task to its organisers even in a limited subject area. The statistical approach attempts to use no more than the occurrences of word spellings and their relative distances in the document environment. Word endings, synonyms, multiple meaning words, unwanted correlation of words, etc. cause serious problems in the formulation of statistical models, and in their elementary form cannot provide the discrimination necessary for most indexing and abstracting applications" [Clemenson et al, 1961].

During the 1970's a number of papers appeared, recognising the importance of the role of linguistics in the field of automatic abstracting. However, it is important to note that papers on abstracting were considerably less numerous than those dealing with other forms of computational text processing. The development of text-linguistics on the international scene has been characterised by the relative exclusion of other related fields. Many contributions made in this field have not been noticed by fellow text-linguistics experts such as Z.S. Harris [Dressler, 1978]. Studies in the field of text-grammar, and particularly in lexicography, are important prerequisites for the wider application of text-linguistic approaches within computational linguistics, and more specifically for the purpose of automatic text reduction, as Sevbo has pointed out in his paper [Sevbo, 1969].

Early linguistic attempts are represented by Earl's two methods of automatic extracting. The first one analyses the syntactic form of a sentence for choice criteria while the other uses both "... syntactic and frequency criteria to choose words which are in turn used in choosing the extract sentences themselves" [Earl, 1970]. Her experiment was encouraging but, however, inconclusive . A year later Rush, Salvador and Zamora attempted to produce indicative abstracts, making use of contextual inference and syntactic coherence criteria [Rush et al, 1971]. The selection of sentences to form an extract is a complex task. The above-mentioned techniques used some criteria for the selection of such a set of sentences from documents as would provide an adequate representation of them. Some sentences, however, although well-suited to the document in their original context, are not necessarily suitable for inclusion in the abstract because of the new context in which they are placed. The resulting set of sentences revealed problems of disjointedness, incompleteness and lack of cohesion. To improve the readability of automatic extracts, Mathis, Rush and Young described several procedures for the modification of sentences initially selected by Rush, Salvador and Zamora's technique. These procedures use the notions of a structural approach to linguistic analysis based upon Fries's proposed structure of English [Rush et al, 1971].

The difficulties facing the above-mentioned abstracting techniques are great and stem from being based solely on the use of syntactic analysis. Bergon recognised these problems and, in his paper, he argued that a syntactic analysis of individual sentences, without any consideration of at least minimal information on how the relations between sentences are constructed, can hardly be a sufficient basis when solving the problem of how informative the individual text segments are [Bergon, 1971]. Consequently, he investigated, using graph structure, supersyntactic links between sentences in order to study the distribution of essential information in a connected text . In his paper, he claimed that his results were encouraging but required further investigation. A further improvement to this study was carried out by Skorokhod'ko who developed an adaptive method in which the relationship between sentences was dependent on the semantic relatedness of the words with them. This relationship was then used to draw a graphical representation of the text. He gave high weighting values to sentences which are semantically related to large numbers of other sentences and whose deletion could cause serious disruption of the content matter of the document concerned [Skorokhod'ko, 1972].

A modified version of the extraction algorithm of Rush, Salvador and Zamora has been used by Pollock and Zamora to describe current research at Chemical Abstracts Service using primary documents. The abstracting program used in their work relied on statistical criteria as a basis for sentence rejection. Selection and rejection of sentences for abstracts are based on the use of "cue" words. The results suggested that some subject areas are inherently more amenable than others to automatic extraction. A physical chemistry document, for instance, may be replete with complex equations necessary for the understanding of the content and hence would not be conducive to machine abstracting. Pollock and Zamora argued that better results may be obtained if the algorithm is tailored for application to a narrow subject area [Pollock et al, 1975].

The feasibility of using the method of text structural analysis was investigated by Bondarenko. This method breaks the text into a supersyntactic units scheme as a basis for representing the distribution of repetitions in a text to be segmented. The study of these segments reveals the internal structure of a text. In constructing the stringing schemes of the texts analysed, technical texts contained sentences saturated with repetitions of words formed from a common stem. Most sentences were simple sentences but nonetheless fairly cumbersome. Bondarenko recognised in his paper that his method is by no means the ultimate solution of the problem of how to consider the internal structure of a text [Bondarenko, 1976].

In 1977, Taylor and Krulee developed a similar approach in which the meaning of a text is expressed in terms of the semantic networks of Simmons which are based on the case grammar relationships of Fillmore. Their system made use of network techniques borrowed from graph theory. They identified a portion of the original networks namely the largest maximally connected subgraphs. The system then identified the nodes that are most influential within that subgraph and converted the subgraph back into a set of sentences in order to produce the abstract [Taylor et al, 1977].

The theme of indicator phrases discussed by Pollock and Zamora was taken up by Paice with a view to producing indicative abstracts. His method involved four stages. The first stage identified indicators and assigned appropriate weights to them. The second stage attempted to aggregate the resulting sentences and produce a coherent self-contained passage. The last two stages contributed to selecting the highly weighted passage(s) which produced the abstract. Paice agreed that the result obtained so far did make the programming effort well worthwhile and that "progressive refinement" of this technique will need to continue for some time. [Paice, 1981]. This popular approach was also adopted by Blyumenau who based his techniques on the extraction of sentences using fixed lists of word stereotypes such as indicators, connectors and markers. A considerable number of papers were processed in this way with a small amount of post-editing by non-specialists [Blymenau, 1981].

A system based on programs for the analysis of primary document text and the synthesis of abstract text was developed by Karasev in 1978. Key sentences were extracted from the rest and classified by importance. Analysis of the text processed on the basis of a special dictionary of phrases, words and other syntactic structures to which semantic, morphological, syntactic and other characteristics had been assigned. The so-called indispensible key structures were fed into a synthesiser which, under certain conditions, generated the abstract. [Karasev, 1978a]. The implementation of the system was published later confirming the efficiency of his approach [Karasev, 1978b].

More complex methods using algorithms of syntactical analysis with a view to dividing the text into standard kernel-sentences were developed by Nistor and Roman as an extension of some previous papers concerning automated abstracting in the Romanian language. [Nistor, 1978]. These kernel-sentences, modelled on Harris's study are regarded as "information quanta", upon which several operations can be carried out to simplify the text [Harris, 1969]. After these reductions the abstract is recomposed out of the processed ideas. In the same year Arzikulov, Piotrovskii, Popesku and Khazhinskaya concentrated their work on the use of an automatic dictionary and a thesaurus semantic network for the production of Russian abstracts of French scientific and technical documents. A user-computer dialogue was established which resulted in the computer generating an abstract [Arzikulov et al, 1978].

A similar approach was adopted by Berzon and Brailovskii in 1979. In a joint paper, they described an interactive system of automatic abstracting in which users' requests can be answered. This system is based on a "questionnaire aspect" which, by means of "connectedness indicators" reflects semantic and logical-semantic links between the sentences of a connected text [Berzon, 1979].

In an earlier work, Janos established that improvement in automatic abstracting depended on the analysis of a text's semantic structure and its surface structure. She claimed that the theory of functional sentence perspective is particularly promising. This theory represents an analysis of relations between the theme (what is spoken of in the text) and the rheme (what is said about it). Such concepts as theme and rheme would help to arrive at other general descriptions of the structure of any specialised text [Janos, 1978]. This theory was further developed in 1979, in a paper in which she emphasized that the need to establish the information content of a text is more important than a traditional examination of subject-predicate relations [Janos, 1979].

The concept of dynamic abstracting was introduced by Nistor and Roman in 1980. In a joint paper, they used fuzzy sets and their function to measure similarity between the semantic notions and applied them to automatic abstracting. They also used a thesaurus to quantify the distance between a document and a query. There is not a single abstract for a given document. The abstracts for a document are tailored to the needs of user requests, hence the name of their approach [Nistor, 1980].

Prikhod'ko and Skorokhod'ko proposed a method of automatic abstracting based on an analysis of semantic links between the sentences in a text. They agreed that the number of inter-phrase sentence links is one of the most important indicators of significance in a text and can serve as a criterion for the selection of text for inclusion in an abstract [Prikhod'ko, 1982].

Most past and current studies in automatic abstracting of documents have centered on syntactic, semantic or statistical techniques. Early work in this field investigated the concept of word frequency and relied on the detection and use of sentence significance according to statistical criteria. Measurement of sentence significance was based on ad hoc rules. What is needed is a program of systematic experimentation rather than experiments using ad hoc rules as pointed out by Edmundson and Wyllys as early as 1961.

In the sixties the lack of involvement of the rules of linguistics was acknowledged. Attempts to analyse the phrase structure and some of the relations between clauses in the sentence were carried out, but were, however, hampered by the limitations of hardware and software. Because of these limitations most techniques were aimed at solving specific experiments and were never extended to a large number of documents.

A more adequate relationship between the disciplines of linguistics and information science was established in the seventies. The development of linguistics, particularly in the field of semantics, has contributed significantly to natural language information processing research. It is interesting to note that while information scientists were heavily involved in using statistical methods to analyse the content of a document, most linguists have totally ignored statistics in their work [Montgomery, 1982].

As a result, the seventies and the early eighties in particular, witnessed numerous attempts to pursue automatic abstracting using a linguistic approach. The integration between information science and linguistics is particularly illustrated in the work carried out by Nistor and Roman, Salvador, Zamora, Taylor and Krulee, and Bondarenko. In 1969, Harris proposed to describe a language by means of a set of kernel sentences, thereby yielding the sentences of the language. This concept was taken by Nistor and Roman as an aid to generating abstracts from Romanian documents. Contextual inference and syntactic coherence criteria were used by Rush and colleagues in their attempt to produce indicative abstracts. A study of the interconnection of segments of text in order to reveal the internal structure of a text was proposed by Bordarenko, investigated by Berzon and applied by Prikhod'ko and Skorokhod'ko.

The abstracting method described by Paice relies on finding indicators within a text [Paice, 1981: 174]. Some interesting results were obtained but further work was recommended for documents where no indicators could be found. His research aimed at producing indicative abstracts which meant that selection of sentences for inclusion in abstract is controlled by a preset maximum length.

A major acknowledgement was noted in the late seventies, in that knowledge is a central focus of all natural language processing activities. Before one can put knowledge into a system, one needs to acquire and represent that knowledge appropriately, to be able to access and use it during processing of text. Various knowledge representation formalisms were developed by the artificial intelligence community. Some of the formalisms will be discussed in the next section. These contributions have played an important role in the information science as more and more research on automatic abstracting began to explore these formalisms.

Another realisation was that in order to process texts by computers, one must understand how people use and process language: hence emphasis shifted from statistics to linguistics and then on to cognitive science.

The desire to learn about the intellectual process of text understanding motivated the research group of the University of Constance on automatic abstracting. The group is currently working on automatic text condensation system [Kuhlen, 1984]. TOPIC and TOPOGRAPHIC are the two major projects being worked on. TOPIC is an automated text analysis system that transforms German texts into condensates (abstracts, summaries, synopses). TOPOGRAPHIC manipulates these condensates by means of an interactive graphical interface to TOPIC. The two projects emphasise the importance of linguistic knowledge in processing text and apply a frame representation technique to manipulate this knowledge. Procedures for text parsing are determined by the occurrence of significant tokens in the text being analysed. Word experts assist the identification of these tokens. Examples of these tokens are: microcomputer, RAM, Z-80A, HP-125, Printer.

A similar approach is being investigated by the University of Udine with regard to text summarisation. A procedural, rule-based approach is implemented in a prototype experimental system, called SUZY. The system parses a text using an Extended Linear Representation [Fum et al, 1984] and produces a new representation called Hierarchical Propositional Network of concepts and propositions. The system then applies several classes of rules to evaluate the importance of these propositions. Some of these rules derive importance values from the structure of references among conceptual units, some rely on semantic relations such as ISA relations, and others use the metaknowledge approach.

The problem of inference in text understanding is also under study at the University of California. Inferencing from a text requires a great deal of general knowledge on the part of the reader. Norvig argues that past approaches produced new algorithms to process a particular kind of knowledge structure (such as script or plan). This means that it is not possible to incorporate previous results unless they are implemented in a new formalism. He therefore designed a program called FAUSTUS (Fact Activated Unified Story Understanding System) to handle a variety of texts and new subject matter by adding new knowledge [Norvig, 1987].

Rollinger and Schneider from the University of Berlin regard the task of understanding text as a knowledge-based process [Rollinger et al, 1984 : 129]. They adopted the point of view of artificial intelligence that to understand a text, one has to consider two major problems. First, the meaning of a text must be represented in some formal language, and second the surface structure of the text has to be transformed into appropriate expressions of this formal language. This transformation is knowledge-based. Their research efforts are directed towards summarizing a text and answering questions about a text. The semantic representation is still under development at the project base in Berlin and is based on the notation of Kalish/Montague's description of symbolic languages for formal theories.

In summary, in early work in automatic abstracting syntax dominated the research efforts: statistical criteria were used to measure sentence representativeness. Later, the importance of semantics was recognised and incorporated in computer systems. The results obtained were encouraging but most studies recommended further work. Furthermore, some systems were limited to short texts and therefore could not serve to test full texts. Many historical factors contributed to this preoccupation with syntax, including limitations of computer technology and the strong influences of American linguists [Lehnert, 1982 : 35].

Over recent years there have been several studies motivated by the need to understand how people organise and process information from texts. These contributions have helped information scientists to look beyond syntax and semantics and leave behind statistical measures and dwell on the intellectual processes involved in abstracting. Aided by advances in computer technology, in particular VLSI technology and the introduction of fifth-generation programming languages such as LISP and PROLOG, issues such as knowledge-based systems and logic programming have begun to mark out the path of current research in automatic abstracting. This shift manifests contributions from other disciplines and these are to be highlighted in the next sections.

5. Natural Language Understanding and Processing in Artificial Intelligence

Much research in natural language processing is concerned with the study of intellectual faculties via the use of computational models. Unlike other fields of study the history of natural language processing ("natural" as opposed to computer languages) can be traced right back to its beginnings in the middle years of this century. The major studies can be grouped into three main categories:

i) the syntactic information processing era;
ii) the semantic information processing era;
iii) the knowledge system approach era.

Text is one of the basic tools of information transfer. To understand how people extract meaning from text and emulate this extraction by computational models one must build a meaning representation of it. To this end a group within the AI community have dedicated their efforts to converting source language text into a precise internal representation for eliminating ambiguities in the source text and hence making computer processing a reality. Those who favour this approach believe that text understanding presupposes knowledge of its syntactic structure. So parsing has become a major approach in the study of language and information processing. Rules of syntax specify the legal syntactic structures for a sentence and various representations of the structure of a sentence are produced. To formalise the knowledge of this structure grammar must be invoked. A detailed description of these representations is given in Charniak and McDermott [Charniak et al, 1985].

Initially it quickly became apparent that semantics had to play a role and thus syntactic parsers called semantic routines to help out. In some cases, the parser had to be modified to allow understanding of ungrammatical sentences. Computational parsing models are performance rather than competence oriented [Sparck Jones et al, 1985: 19]. The approach adopted by most parsers is to find out how one may actually use rules of syntax to extract meaning.

The years around 1970 witnessed a flowering of semantic information processing and the sowing of cognitive science studies. Winograd produced a program called SHRDLU, based on two analogies. The first is that sentences can be transformed into instruction sequences or programs which could be used to move blocks on a table, for example, or search a database for information and answer questions. The second analogy associated words to a set of programming steps, so each word triggered off a list of program fragments. The SHRDLU parser would attempt a syntactic or semantic approach: if that failed it would back-up and try another interpretation [Winograd, 1972]. SHRDLU handled declarative, interrogative and imperative sentences. It was able to learn word definitions, hold a dialogue and plan the execution of moves.

Another interesting system came out in 1972, which attempted to answer questions from a database containing moon rock sample analyses. Woods and his colleagues used Augmented Transition Networks (ATN's) for parsing sentences. This approach was similar to that of Winograd in finding the syntactic structure of the sentences. Woods introduced, however, the quantification notion based on a predicate calculus technique capable of translating source questions into database queries [Woods, 1972].

NLPQ is a program developed by Heidorn which sets up simulations. This program embodied a model of what a complete simulation should include and asked questions if the information given to do the simulation was not sufficient. The knowledge of what constitutes a complete formulation of a problem was an important addition with regard to natural language systems.

In the mid-seventies, a number of linguists such as Fillmore and Lakoff pointed out the necessity to going beyond syntax in order to understand how people understand and use language. This proposal echoed ongoing work in the AI community especially that of Minsky, Schank and Abelson, and Charniak [Bolc, 1980 : 26].

Schank and Abelson introduced the notion of scripts in 1977. Very little is said about how the scripts are used and which domains are chosen. However, a number of applications using the script framework has developed by Schank and his students, including Lehnert, Cullingford and Willensky. SAM (Script Applier Mechanism) is a program that takes a story as input and transforms it into an internal representation using a large conceptual dependency network from which question answering can occur and summaries can be generated [Schank et al, 1977 : 190]. Schank argued for the existence of primitive concepts and underlying canonical forms in deriving an appropriate meaning representation for sentences. Independent of their surface forms, if two sentences convey the same meaning then they must have the same representation in terms of primitive concepts. Schank called this representation a conceptualisation.

Two other systems were produced by Wilensky in 1978. FRUMP (Fast Reading Understanding and Memory Program) operates by skimming a newspaper story until it obtains sufficient information to select an appropriate script. PAM (Plan Applier Mechanism), on the other hand,was designed to understand stories which involve goals, plans and themes. This program makes a set of inferences from its knowledge base [Wilensky, 1978].

Another framework for representing knowledge was proposed by Minsky in 1975, called frame theory. He suggested that visual images are frequently evoked in understanding sentences. Attitudes, purposes, functions are among other non-visual aspects associated with this understanding. Such aspects can be represented in a hierarchy of frameworks. Minsky suggested that the necessary levels are as follows [Minsky, 1975]:

Surface semantic frames - deep syntactic frames :
action-centered meanings of surface syntactic frames,
words for example.
Surface syntactic frames - mainly verb cases.
Thematic frames - topics, settings, activities.
Narrative frames - stories, explanations, arguments.

This theory was applied by Rosenberg who developed a frame-based system for processing text [Rosenberg, 1977], and later by Sidner to solve problems of reference arising in discourse [Sidner, 1978].

A number of other systems were produced applying the frames or scripts theory. Some of these were geared towards answering questions from databases, thus leading the natural language processing research into the realm of information management systems, and decision support systems. Other systems directed their efforts towards speech understanding and computer aided instruction. As the major emphasis of this chapter is the contribution of AI to the study of abstracting, the review is limited to areas closely related to this task.

One of the difficulties in evaluating the state of the art in natural language processing is that most papers describe what is achieved by these systems. Very little information is given as to the ability of these systems to handle published documents, or as to the constraints of their vocabulary or repertoire. Nevertheless, one can gain valuable insights into the kinds of available computational models and into the type of questions these studies attempts to answer. The development of natural language processing research somehow mirrors the historical approach followed by information scientists. There is a clear move from syntax via semantics and hence to knowledge. Findings in the linguistics and cognitive science have had a profound impact on the progress of these scholars and are hence echoed in their studies. Again, advances in computer technology have eased the implementation of the systems produced by the AI community. Now it is time to explore the work of linguistis and cognitive scientists in their pursuit for text understanding.

6. Linguistics Approach to the Study of Information Processing

Discourse analysis and text linguistics are the two fields within linguistics highly concerned with the linguistic analysis of naturally occurring connected spoken or written discourse [Stubbs, 1985: 1]. The origins of discourse analysis can be traced back to the study of classical rhetoric more than 2000 years ago. By the nineteenth century, rhetoric had seemingly lost much of its importance and the structural analysis of language took pride of place, although fragments of rhetoric survived in the study of literary language [van Dijk, 1985: 2]. Text linguistics is sometimes treated as an English equivalent for the German term Textlinguistik [Kalverkamper, 1981: 16]. To a certain extent, this distinction in the terminology has caused a relative seclusion of research trends in the development of these fields. This section, however attempts to shed light on contributions from the international scene and thereby gain insights , firstly, into the study of information structure in written text and, secondly, into the resources available to writers for indicating the status of information.

The study of information structure within texts was pioneered by scholars of the Prague school before the Second World War, within the framework of functional sentence perspective (FSP). This subject was studied for the first time by Mathesius in 1942 [Janos,1979] and constituted a crucial approach to the description of natural language texts. Mathesius initiated the study of theme, rheme and focus as integral parts of the cohesive mechanisms that integrate sentences into a text [Enkvist, 1978: 180]. They are devices that help signal the progression of the argument and the difference between given or known and new information. This framework was adopted not only by Czechoslovak linguists but also by scholars from the Soviet Union and the GDR.

Many of the insights developed by the Prague school can be traced in Halliday's approach to the structure of texts. He adopted the given/new information viewpoint and related it to intonation. According to Halliday, the speaker organises his text into information units. The speaker has the freedom in deciding the boundaries of each information unit and its internal organisation [Halliday, 1967: 200]. With regard to the manifestations of information structure in syntactic form, Harris observed that for more than 200 centuries the use of the two articles "a" and "the" denotes whether one perceives something as unknown or known [Harris, 1751: 215]. Kress offers a useful expose of the evolution of Malinowki's, Firth's and Whorf's studies into Halliday's theory. Halliday built on Malinowski's definition of meaning as function in context, accepted Whorf's theory with respect to the relation of language and culture and the system theory advocated by Firth [Kress, 1981 : VIII].

The sentence - which Saussure excluded as belonging to the use of language, not to the language itself - had, of course, long since emerged as the central entity of linguistic theory [de Beaugrande, 1985: 42]. It was easier to treat the sentence as an object [Morgan, 1975] than to go beyond the sentence. Various structural methods were attempted but proved harder to develop. The Tagmemic school led by Pike and his associates were interested in relating linguistic structures to larger patterns of human behaviour. They related grammatical features to discourse structure and situational role. The integration of anthropology in their descriptive structural-linguistic approach has provided invaluable documentation on little-known languages. The slot-and-filler approach was adopted as a theoretical base for their theory [de Beaugrande et al, 1981: 19].

American linguists took Harris "notion" of transformations to describe sentence structures not in a single discourse, but rather in an entire language. Chomsky's transformational grammar (TG) marked a retreat away from language in use toward language in the abstract. A set of rules of formation and rules of transformation was used to describe sentences about John and Mary and this constituted the content of linguistic samples for many years to come [de Beaugrande, 1985: 43]!

More and more linguists became increasingly aware of the importance of context in analysing sentences. Fillmore advocated a methodology to determine the meaning and context of an utterance, known as case grammar. He suggested that "any" particular verb or other predicating word assumes in each use a given perspective " on a scene" [Fillmore, 1977 : 74].

It was not until the 1970's that text linguistics gained wide recognition, and this is reflected in the growing number of publications in this field, mainly by German-speaking scholars. The study of text moved from the usual sentence base to hyper-syntax [Palek, 1968], macrosyntax [Gulich, 1970], and text syntax [Dressler, 1970]. An extension of transformational sentence grammar was explored with the view to proposing a text-grammar approach [Kuno, 1978; Petofi, 1971; van Dijk, 1972]. Proposals to view texts beyond the notion of sentence began to emerge. Chafe stated clearly that in order to understand how language communicates thoughts and idea one has to know more about semantic phenomena that cross the boundaries of sentences. The study of human knowledge is acknowledged and given higher priority than questions of syntax. Chafe's main concerns were the questions of how meanings can be formalised in a linguistic theory of semantic structure, how such a formalisation can incorporate features greater than single sentences and how this type of formalism relates to human knowledge [Chafe, 1972 : 42].

van Dijk pursued a different range of considerations. His macrostructure theory is an attempt to study the content of a full text. He argued that in producing a text one begins with a main idea which gradually evolves into detailed meanings manifested in strings of sentences. Various operations are described in order to extract the main idea in a text : deletion (removal of information), generalisation (recasting material in a more general way), and construction (constructing new from given or subsumed material). His theory was developed with close collaboration with Kintsch [van Dijk, 1979].

de Beaugrande and Dressler are leaders among linguists currently advocating the status of the text as a communicative event. They argued that text issues cannot be addressed if research is based on syntax only. The text is "normally composed of cohesive phrases and clauses, and the concepts it represents are normally coherent. The text is distinguished by its "textuality", based not only on cohesion and coherence but also on intentionality, acceptability, situationality, intertextuality and informativity" [de Beaugrande, 1985: 49]. He strongly stresses the need to study "real, naturally occurring texts" rather than invented passages. Narrow analyses of linguistic structures would overlook the dynamics of textual communication and impoverish the results of such studies [de Beaugrande, 1985: 61].

This brief survey of the various directions in the study of language use and text analysis shows the divergence of views and the multitude of approaches. The early approaches focussed on syntax and semantics, a similar progression in views to the disciplines mentioned in the previous sections. A shift towards the integration of human knowledge and realisation of the dynamic aspects of text is clearly reflected in the later trends in discourse analysis and text linguistics. Along with this departure from sentence boundaries there has been an interest in developments in psychology with a view to understanding the intellectual processes that occur when people produce or are exposed to discourse material. Here the gap between linguistics and psychology is slowly narrowing.

7. Cognitive Studies and Text Processing

Traditionally linguists have paid most attention to the structure and function of language in communicating information and knowledge whilst in the field of language psychology scholars focussed on the cognitive processes involved in reading, comprehending, producing and acquiring information. Although most of the work undertaken in cognitive studies is primarily directed towards understanding mental processes for purposes other than abstracting it is believed, however, that these contributions can provide valuable insights into how people organise and process information from texts and they may shed some light on the approaches to the study of information from other disciplines. These contributions are grouped into three main issues : reading, interpreting and comprehending .

It may seem curious to begin with a section on the psychology of reading; however, it is assumed that writers work with a specific type of readers in mind. Currently, there seem to be two schools of thought with regard to reading. One school stresses perceptual processes and argues that reading is a "bottom-up" or "data-driven" activity and the other maintains that reading is rather a "top-down" approach or "conceptually driven" process.

According to this theory one first discriminates horizontal, vertical and oblique line segments. From these, letters are identified, then words are recognised. Strings of words are parsed into phrase constituents and word meanings are retrieved. Eventually, a semantic interpretation of a sentence is produced, followed by the meaning of the text. The other school holds that what people see when they read is determined by their expectations and past experience. Thus, skilled readers sample the text in order to confirm or reject hypotheses about its content [Hartley, 1980 : 11].

Anderson argues that it is simplistic to imagine that reading is either a bottom-up or top-down process. He agrees with Ausubel that the prior knowledge and experience that readers bring to a text can have a marked effect upon their perception and recall of words, sentences, paragraphs and complete text but he goes further by suggesting that text information is interpreted, organised and retrieved in terms of mental structures called "high-level schemata" [Anderson, 1980]. The schema provides a device to the reader in focusing attention to important elements in text. It also aids the retrieval of selected categories of text information. Basically, a schema will contain slots for important information, and may contain optional slots for unimportant information. In this way, information gets encoded precisely because there is a niche for it. Many scholars in artificial intelligence adopted this slot-filler approach in their work, namely Schank and his associates.

Now, in order to examine what information is being processed by the reader, one needs to know what information is presented in the text. Hence many psychologists turn their attention towards the structure of text from the point of view of recall and models of memory. The structure strategy is the dominant reading strategy, pursued by skilled comprehenders. They search out and follow the "text's superordinate relational structure" and focus on the text message and on its relation to other supporting details [Meyer, 1985: 2]. Meyer's findings show that skilled readers process text with knowledge about how texts are conventionally organised and that they use the top-level structure as an organisational framework to facilitate retrieval.

A model of text processing at a microstructure or content level was proposed by Kintsch. He represented the meaning of a text by "text bases" which consist of lists of propositions. These propositions are n-tuples of word concepts formed according to a set of rules which are part of the reader's semantic memory. The formalism for constructing propositions are derived from linguistic theory [Kintsch, 1974]. In a joint work with van Dijk, Kintsch developed his microstructure theory further and argued that the semantic structure of discourse must also be described at the more global macrolevel. "Beside the purely psychological motivation for this approach, ... the theoretical and linguistic reasons for this level of description derive from the fact that the propositions of a text base must be connected relative to what is intuitively called a topic of discourse ... that is, the theme of the discourse or a fragment thereof. Relating propositions in a local manner is not sufficient. There must be a global constraint that establishes a meaningful whole, characterized in terms of a discourse topic" [Kintsch et al, 1978: 366]. These macrostructures are described in terms of propositions and proposition sequences. Semantic mapping rules based on microstructural information as input and macrostructural elements as output aim to show how a discourse topic is related. Three basic macrorules are described: deletion (of propositions that have neither direct nor indirect interpretation condition of subsequent propositions), generalisation (some propositions can be substituted by more general ones), and constructions (some implicit propositions can replace explicit ones). Although their model implies a linear processing of propositions in a text, it does not take account of its layout.

Kieras's is the only study to bring the macrostructure model to bear on the arrangement of text in terms of paragraphs as complexes of sentences. He was concerned with how readers build the macrostructure of a text, which represents the theme or gist of a text [Kieras, 1981]. Kieras stated that technical and expository prose in general does not receive as much attention from cognitive psychologists as story materials. Hence his studies focussed on technical and expository prose. He distinguishes three kinds of information involved in comprehending prose : text grammars, content schemas and content facts. "A text grammar is a schema ... that represents a frequent configuration of textual elements, defined independently of specific content, ... termed textual surface structure" [Kieras, 1985 : 93]. For example, a paragraph may first state the topic, then expand on it and then state a conclusion. A content schema is defined as a configuration of content facts. For example, the concept of an enzyme will have a content schema specifying the common facts about a class of enzymes. Because the content of a technical text is mostly novel to the reader, Kieras suggested that understanding in this type of text is not primarily a matter of matching the content to a known pattern, but rather a case of dealing with the content at the level of individual propositions [Kieras, 1985]. In a earlier paper, Kieras suggested that in abstracting the main ideas from simple technical prose, people read a paragraph and then make up a brief, one-sentence statement of the main idea. He argued that the rules for deriving macrostructure from microstructure for these texts are not clearly explained. A further complication is added as the surface-level aspects of the text and its semantic content appear to be important factors in determining the macrostructure of the text. To study how readers abstract the main idea, he devised an experiment using four passages. One striking result is the treatment of sentence 1 by his subjects as the explicit statement of generalisation and recognition of being a statement of a main idea, hence it is rated as highly important. The overall strategy seems to favour the first sentence of a passage as the candidate main idea, and readers attempt to fit each succeeding sentence into this main idea. If this fails at some point, revision of the first sentence is considered, hence the name of this approach is referred to as the subsuming strategy [Kieras, 1982].

A number of cognitive psychologists focussed their studies on specific aspects of how the topic or main idea is marked and signalled by the writer to the reader. In addition to Kieras' suggestion of a topic sentence, Meyer associates titles with a kind of signalling information that "prematurely reveals information abstracted from the content occurring later in the text" [Meyer, 1975]. As advance organisers, titles facilitate the assembly of knowledge incorporating new material [Ansubel, 1968]. Kozminski argues that "when a title is assigned to a text, it is assumed that the title carries the thematic information in the text. This thematic information is assumed to occupy the highest level in the text base. Therefore, the title propositions are the most superordinate ones in the text base" [Kozminiski, 1977: 485].

On thematization, Perfetti and Goldman argue that "it is the normal state of affairs that the first noun of a sentence is the topic of a discourse" [Perfetti et al, 1974: 78]. This approach seems to agree with Halliday on the definition of the theme at the clause level as the first content word of the clause. With the same approach, van Dijk stresses the importance of certain parts of semantic information. "A language has conventional means to signal the fact that certain parts of semantic information may be given prominence. In this case, this prominence is structural (textual): some part of the sentence (proposition) is more prominent than other parts. The reason for this particular kind of prominence assignment must be sought in the cognitive basis of communication processes : certain information is already known, other information is new for the hearer/reader" [van Dijk, 1979 : 1167].

To be effective in communication writers make use of many typographic cues within the text to signal important information and special issues to readers. Gaining access to the text through components within the text is an issue taken up by both linguists and psychologists. These lines may not assist comprehension, but they help readers find their way around the text. Typographic cues such as italics or bold face, use of headings, type of clauses are often employed by authors to highlight important textual components [Foster, 1979]. Hartley argues that spatial lines such as indentation, page, size are very important [Hartley et al, 1977].

Textual configuration and textual components were also the focus of study for many psychologists. The paragraph, for example, is another marker that has a psychological reality. Hoey claimed that paragraphs are real and must be accounted for in relation to their informal organisation and the overall organisation of the discourse [Hoey, 1985]. At a lower level of text organisation, Cook examined the function of connectives and other devices as indicators of topic continuation [Cook, 1975]. Discourse markers and other signalling devices emphasise directions and relations within text. The macro-markers may include phrases such as "now, getting back to our main point", "the aim of this paper", "the problem here was that", "another interesting development was". The micro-markers may be used for segmentation purposes (e.g. well, now, and), contrasting (on one hand, on the other hand, but), emphasis (of course, in fact, as you know), whereas the macro-markers contain signals to major propositions and important transition points in the text [Chaudron et al, 1986].

This brief survey of theories put forward by scholars in cognitive studies summarises the most relevant findings in the attempt to understand the human process of abstracting. A very interesting development is the trend towards the integration of human knowledge, the structure of discourse and its linguistic realisations to arrive at a better understanding of the cognitive processes that the information seeker employs. This section does not attempt to produce an exhaustive survey of cognitive studies with regard to information processing; it aims at highlighting those works which have contributed directly or indirectly to the research of scholars in the field of linguistics, artificial intelligence and information science.

8. Other Information Processing Approaches to Discourse

Apart from the above mentioned contributions to the study of abstracting two other most interesting disciplines must be cited : education and philosophy. The aspects of comprehension and information extraction have received much attention by educationalists. One major concern to them is issues related to school curricula : educators, in particular, have to promote essential skills in language comprehension on the grammatical, lexical and semantic levels of knowledge. With regard to "reading comprehension skills" Davis reaffirms the existence of eight independent skills mainly "remembering word meanings", "following the structure of a passage", "finding answers to questions answered explicitly or in paraphrase", "recognising a writer's purpose attitude, tone and mood", and "drawing inferences from the content" [Davis, 1968]. Freedle and Caroll suggest that "the structure of a discourse can be regarded as a ready-made plan to help the reader understand it .. " [Freedle et al, 1972: 363]. This is in line with information processing approach to discourse in that the information user must contribute his background knowledge and his presuppositions to understand the message behind a passage. "What is explicitly given in discourse represents only cues to underlying semantic structure" [Freedle et al, 1972: 363].

The concepts of understanding and meaning have also intrigued philosophers. A narrow focus to understanding and comprehension based on linguistic aspects would constrain the theories that lie behind comprehension [Scriven, 1972: 38]. The papers published by Platts attempt to clarify some of the central issues raised by the philosophy of language regarding the connections between language, reality and human knowledge. His initial assumption is that "in rendering linguistic comprehension intelligible a central, ineliminable role will be played by a notion of the strict and literal meaning of a sentence" [Platts, 1972: 2]. Philosophical linguistics is another discipline concerned with semantic relationships between constructed pairs of sentences and with their syntactic realisations. They share with philosophers the notion of truth values of sentences. "If language is to communicate, it is just as necessary to understand reference and predication as it is to understand meaning" [Cooper, 1973 : 70]. Another concern that lies on the boundaries between philosophy of language, logic, linguistics and psychology is the subject matter related to propositions. Various accounts of propositions were suggested, as entities constituting the bearers of truth, the meaning of sentences, the objects or contents of thoughts [Rosenberg et al, 1971].

9. Summary

Issues relating to the importance of information and how people analyse and process information have been a focus of study for many disciplines. Interests in these issues have intensified recently, partly due to the increase in the volume of information, and partly due to the advances in computer technology. "For a variety of reasons it seems very probable that learned journals will become obsolete in the future. Scientific information will be distributed in abstracts and stored in a form available from mechanical processing, so that it can be automatically retrieved" [Wason, 1980: 251]. This is one of the reasons that motivated research into automatic abstracting. Other researchers are concerned with producing models of discourse, cognitive processes that occur during the processing of information, communicating knowledge via language. Some of these aspects of information may have emerged from the need to solve practical problems (as in the case of educationalists, information scientists, and computer scientists), others have been stimulated by the theoretical approach to the study of information as in the case of linguistics.

Regardless of scholars' purposes in their pursuit of the study of information, attempts to pool contributions from various disciplines can only help to integrate the development of these theories and approaches and aid people's understanding of how people manipulate information. This is what this chapter has intended to achieve by focussing on the process of abstracting which is the main emphasis of this research. It is interesting to note the similar transitions that marked the research trends in information science, linguistics, cognitive studies and artificial intelligence. The shift from syntax to semantics to knowledge can be clearly traced. Although most of the papers have some constraints imposed upon their studies insights into their approaches can only help research to escape narrow perspective and compartmentalisation.

CHAPTER 3

A HEURISTC ORIENTED APPROACH TO ABSTRACTING

A HEURISTIC ORIENTED APPROACH TO ABSTRACTING

1. Introduction

The previous Chapter reviewed research in automatic abstracting and concluded that knowledge is a key issue in text abstracting, whether one approaches the studies from an information science perspective, a cognitive science point of view, or with linguistics or computer science goals. This Chapter has three major tasks. It first outlines some of the problem areas related to research in automatic abstracting, it then examines the human abstracting process and finally explains the approach employed by the current thesis in dealing with some of the problems faced by previous research. Although one can trace the foundations of this approach to the field of artificial intelligence, one can also see the seeds of cognitive studies and linguistics. In formulating the methodology, the basic assumption was that text understanding embraces many issues: some of these are linguistic in nature, others are cognitive. Another major issue that receives a great deal of emphasis in this study is the interaction and activation of human knowledge.

When an author writes a scientific paper, for instance, the main objective is to communicate his/her achievements and knowledge to a set of specific readers in mind. The paper is the linguistic realisation of this objective. The choice of words, the syntactic expressions, the structure of paragraphs and the paper as a whole, all these linguistic facets are conditioned not just by the writer's own volition but also by the expectations of assumed readers, and the writer's own experience as a reader. With regard to content, the writer assumes a common shared knowledge, problems and interests with the reader, although they may differ in purposes and value-judgements. In other words, in this process of communication between the reader and writer, three kinds of knowledge are activated:

i) knowledge of the language, its use and function;
ii) knowledge of the situation, that is the shared knowledge of the domain and context;
iii) knowledge of the world, that may comprise common sense, general knowledge, past experiences, strategies....[de Beaugrande,1980]

Having highlighted the framework upon which this research study is based, the following section focusses on some of the problems that mark research into automatic abstracting.

2. Problem Areas in Automatic Abstracting

The research community involved with automatic abstracting shares

some of the problems that face all studies in natural language processing. In particular there are five major areas:

i) the input processing problem;

ii) content analysis, that is, the problem of identifying suitable units as basis for content analysis;

iii) measures of representativeness;

iv) the notion of relevance;

v) the single/multi-disciplinary approach.

2.1 The Input Processing Problem

In the past, it was necessary to type in texts for computer processing. Such a process resulted in a lengthy editing task aimed at purging typing errors. Today, OCR machines such as the Kurzweil Data Entry Machine (KDEM) read texts into the computer. Some texts are more amenable to scanning than others. Advances in computer technology have improved the input processing to a great extent; minor editing tasks, however, are still required to remove incorrectly transcribed characters.

Because of the available input processing facilities people are nowadays able to base their studies on larger corpora than was possible. This, however, poses a different kind of editing problems. A collection of papers from one single proceedings, for example, can produce different standards of paragraphs, headings and sections. Some authors use indentation to mark paragraphs, others leave a blank line. To complicate the task, blank lines are also used to demarcate a list of items within one sentence. Many papers have unclear paragraph boundaries. In terms of headings, some papers number them, others centre these and in some cases formulae can be taken as headings because of the position status adopted in the paper. On some occasions, punctuation is not clear or lost in the publication process. The lack of standardisation in published papers make editing a necessary prerequisite to computer processing.

2.2 Content Analysis

Probably the most difficult task any natural language processing research must undertake concerns the way it processes information. If a scientific paper, a technical document, a report is to communicate knowledge, how can the researcher analyse the linguistic manifestations of this knowledge? Is the information contained in words or in phrases? In clauses or sentences? In a string of sentences? In the theme? Is information packaged into cognitive units such as propositions, concepts? Should one analyse the content of a text in ways different from those who generate them?

The use of computers as a tool for content analysis is today gaining in importance. Within computer-driven content analysis, two main approaches must be cited:

i) dictionary and thesaurus approaches
ii) artificial intelligence approaches.

The former approach places the emphasis on single words or short string of characters, identified in a text, removed from their linguistic environment and classified and counted [Krippendorf, 1980]. The dictionary approach is best exemplified by the General Inquirer project developed by Stone et al. In the General Inquirer, a dictionary is used to identify a word stem and assign a category called "tag" to nearly all content words in a text. After tagging, several forms of analysis can be carried out, including contingency analysis, various frequency counts according to tags, word stems and sentences satisfying given sets of criteria [Stone et al, 1966]. The tagging of words is intended to reflect shared semantic similarities between pairs of words or on synonyms, and is thus presumed to represent an important step to the semantic interpretation of a text. This form of understanding is common to the thesaurus approach. The tabulation of words by their thesaurus headings provides a basis for identifying certain basic concepts, for infering the meaning of a text, and comparing different documents to determine what is new or different [Krippendorf, 1980].

An important development in AI approaches to content analysis is a series of computer programs designed to represent certain types of discourses, namely a restaurant scene and social encounters, as reported by Schank and Abelson. It is seen as a response to the criticism of computational linguistics which has limited people's attention to the understanding of single sentences. The AI approach involves a syntactic analysis followed by semantic analysis component which is then mapped into a logic-of-discourse component. Here, the attention is focussed on discourse beyond the sentential unit and is motivated by the need to look at a text in its totality, thus using scripts, plans, goals, schemas to represent the variety of knowledge that individuals employ in processing information [Schank et al, 1977].

Content analysis has also been recently influenced by work in psychology which stresses awareness of assessment of selectivity in people's perception of reality along with his/her text using a model approach to their study [Carney, 1972].

Regardless of the technique pursued, the content analyst has to address a series of problems. These problems take on a new dimension when the analyst uses computational tools. The technique used must be objective and systematic so that it is replicable. To this end, the rules that govern it must be explicit and applicable to all aspects and units of analysis. For instance, when a word is taken as a unit of information, a definition of what constitutes a word must be clearly stated so as the computer can process hyphenated words (question-answering, self-awareness...), abbreviated words (TESP, IBM, AI...), words enclosed in quotations ("raison d'etre"...), words with different spellings (data-base and database), and words such as "i.e", "etc"., "cf" ... What delineates a sentence is another matter for clarification. Does a dot denote a full stop, a character that describes a list of items, (as in 1. input processing), or part of a numbering system (as in figure 6.1)?

If the analyst is using a thesaurus, problems related to its size and the amount of information needed to process a text in a reasonable amount of time without loss of preset objectives must be addressed. If a scheme representation is employed the key problem is the efficient management of schemas and schema instances in memory organisation. How much knowledge is required in storage for effective processing of a piece of text?

2.3 Measures of Representativeness

A number of criteria have been considered by researchers in automatic abstracting with regard to how to identify and select the most representative sentence in a text for inclusion in the abstract: statistical, syntactical, semantic, positional. The early work in abstracting carried out by the information science community, particularly Luhn and his followers, employed statistical measures based upon the absolute frequency of content words in a text. Some content words appeared with high frequency but were not related to the domain of the text [Borko et al, 1975]. As a result, the high frequency of those type of words biased the measure of representativeness of sentences.

The detection of semantic cues that authors frequently use in sentences received much attention in psycholinguistics. Cues such as "conclusion", "summary", "representation" can mislead the study if extracted out of context. The word "summary" in a paper on summarisation does not nesessarily indicate that the author is about to summarise the article. Also authors may mention conclusion in the context of earlier studies or when referring to a debate. Author's styles of writing are likely to vary from one article to another and may not conform to one immutable standard.

With regard to positional criteria some authors may not always favour the first sentence and permit it to convey the topic discussed in the paragraph. Others may wish to highlight the theme in the third sentence or anywhere else in the paragraph. Some authors can adopt more than one strategy in writing their papers. A generalised approach to topic sentences may not prove to aid understanding. Paice argues that "an extraction technique of this kind does not need to be automated at all, since any punch operator can select sentences on this basis" [Paice, 1977: 136].

2.4 The Notion of Relevance

Another matter closely related to the problems of measuring representative sentences is the issue of relevance. Scientific papers and academic articles are distinguished by a peculiar feature in that one is not certain of the type of audience likely to read the paper, particularly if the subject area of the paper embraces many disciplines. Other papers have a very closely defined homogeneous audience.

An assumption was made at the beginning of this thesis that writing is a communicative event. Within this framework, writers and readers operate on the basis of common knowledge. Writers assume that potential readers have largely the same knowledge of those facts about the world relevant to their meanings as they themselves do. Writers and readers are mirror-image processes [Cooper, 1982]. Writers, however, like speakers make use of a number of linguistic features to stress their main points, to emphasise important findings and highlight new concepts and techniques. Some of these features will be described in the next chapter. Skilled writers make use of these features to present relevant information to their readers and will organise the structure of their papers accordingly. Thus getting to know the strategies used by writers to highlight relevant sections or chunks of their papers is a necessary step towards understanding the relevance from the perspective of the writers.

In the content of automatic text processing, this leads to the following questions:

i) should these strategies be deduced by the system?

ii) should these strategies be given to the system?

The first question requires a system that can learn from its experience: thus the system exhibits a learning mechanism - an AI concern still in its infancy. The second question puts the emphasis on the knowledge engineer to elicit the knowledge from experts and embed it into the system. So far the content and structure of ta published paper underlines the relevance from the perspective of the author: what is the author communicating? and how is this communication achieved?

Relevance, however, is a very fuzzy concept. Information theorists define the amount of relevance of a piece of information as the effect that the piece of information has on reducing the uncertainty associated with a particular event or set of events [Naughton, 1974]. One can distinguish relevant from irrelevant information, more relevant from less relevant information. What writers perceive as relevant information may be less relevant or irrelevant from the perspective of readers. Relevance is a matter of degree, and consequently is very difficult to assess by means of quantitative measures. Sperber and Wilson argue that a spearker/writer who intends to convey relevant information has two related aims: one is to create contextual effect in the audience, another is to minimise the processing effort this may involve [Sperber et al, 1986: 202]. Such an approach places great emphasis on the theory of style.

Turning back to automatic abstracting, the research must deal with the study of human communication to settle the problem of relevance. "To communicate is to claim an individual's attention; hence to communicate is to imply that the information communicated is relevant" [Sperber et al, 1986 : VII]. The problem is how this principle of relevance is applied in the text?.

2.5 The Single/Multi Disciplinary Approach

Early work in automatic abstracting and content analysis was confined to one single-discipline approach. Information scientists were mostly concerned with statistical measures in evaluating how representative each sentence was with regard to the theme of the paper. Very little interaction occurred in their work with text linguists who were also working on issues of central importance to the problem of abstracting. Various references to the significance of linguistics are noted in the literature of the information scientists, in the late sixties and early seventies. Most of these references pointed out the importance of the inclusion of linguistics in any theory elaborated. However, there was a general consensus that linguistic contributions, in particular, to the field of content processing have not yet been conclusive enough to warrant adoption and adaptation [Garvin, 1965: 177]. Kay and Sparck Jones suggested that linguistic research may have failed the information scientist in the past:

"Documentationalists and social scientists interested in perfecting the technique of content analysis are faced with severe linguistic problems but their attempted solutions rarely show the imprint of modern linguistics. The reason is clear, linguists are, for the most part, uninterested in practical problems or even in stating the findings in operational terms so that they could be picked up by someone with less distant aims in view " [Sparck Jones et al, 1973: 159].

On the linguistic front most of the work in text linguistics, discourse analysis and content analysis was originally developed very much in isolation from other disciplines. The formal approach taken by traditional linguists in viewing linguistics as the study of the formal properties of a language has to a large extent contributed to their isolation. It is the functional approach of the discourse analyst who is interested not only in the purpose or function of discourse but also how discourse is processed by receiver and producer [Brown et al, 1983] that brought linguistics a step nearer to psychology. Cognitive psychology is one discipline that aims at describing the components processes involved in people's ability to comprehend text, and analysing how these processes are organised and integrated to support the common facts of text understanding. The study often favours the lexical or sentential levels of analysis to neurological or phonetic levels [Bower et al, 1985].

The co-operation of cognitive psychologists with artificial intelligence researchers has fostered the disciplines of cognitive science. Here computer models are being developed to examine the cognitive and linguistic behaviour of people on tasks such as recognising and recalling short passages [de Beaugrande et al, 1981]. Although most work is still confined to the sentence boundary [Clark et al, 1977], the trend is moving toward texts as objects of investigation.

These various approaches to text understanding highlight the complexity of the operations involved. Taking only a linguistic approach to the study of automatic abstracting is to assume that text is a product divorced from its producer and from its purpose. The study of text is not static. Researchers may study units of sound or patterns of sentences from an abstract standpoint. "But many aspects of texts only appear systematic in view of how texts are produced, presented and received" [de Beaugrande et al, 1981: 15].

Similarly, text understanding cannot be confined to the cognitive processes involved. Memory organisation, inferences, recall are issues that vary greatly depending on the nature of text being investigated. Equally, one cannot model human understanding without any linguistic and psychological base. Communication via text breaks down if it is analysed from one single perspective. The modern trend in text understanding is to place the research at the crossroad of linguistics, psychology, computer science and information science. The integration of various disciplines can provide new insights into the process of text understanding and abstracting. However the complexity of such an approach can lead to a combinatorial explosion, particularly if computers are used to test these approaches. Yet one cannot ignore these interactions. The whole business is a major problem that must, however, be addressed.

3. The Human Abstracting Process

A quick look at the history of human abstracting reveals that people were involved in this process as far back as the Mesopotamian times when the envelopes surrounding cuneiform clay tablets contained a summary of the contents of that tablet. It was in the seventeeth century that abstracting work was formalised as a result of institutionalisation. The first issue of the world's first abstracting periodical, the Journal des Scavans, was issued on January 5, 1665. Many other abstract journals followed covering besides the literature of philosophy, other disciplines such as theology, the humanities and the sciences [Witty, 1973]. With the growth of scholarly and professional literature in the twentieth century abstracting has become a vital function of and a necessary service to the life of organisations. The function of the abstract has remained unchanged, but its value has, however, been vastly increased.

Books on abstracting provide general guidelines to abstractors with regard to the form, content and function of abstracts. A survey of abstracting policies used by producers of bibliographic databases was carried out by Fidel with a view to examining available abstracting guidelines. In the study it was found that database producers addressed the content and form in their specific instructions to abstractors [Fidel, 1986]. The function determines the type of abstracts to be produced: that is, whether abstracts are to be informative or indicative. In actual fact, most abstracts are of the informative type, quite deliberately so in order to optimise recall in computerised retrieval systems. The function of abstracts, too often, dictates their form. Matters related to writing style are related again to enhancing retrieval: as a result, many editors issue clear policies regarding the form of abstracts.

The form of abstracts relates primarily to writing style and length. On the issue of style, the American Chemical Society issues a set of instructions which stresses the importance of abstract presentation. These instructions can be grouped under the following headings [American Chemical Society, 1975]:

i) Sentence Structure:
- Sentences must be clear, concise and complete; technical English must be used; use articles for clarity,
- a sentence should be constructed so that the verb follows the subject,
- the use of phrases such as "the authors studied", "in this work", "a method is described" is to be avoided,
- the simple past tense is preferred to describe what was done,
- the imperative mood should be avoided;
- splitting infinitives should be avoided.

ii) Punctuation, capitalisation and hypens:
- a comma is used to separate places and dates, to connect the last two members of a sequence of three or more,
- capitals are used for proper names and adjectives derived from them,
- a hyphen must be used in compound adjectives involving magnitude as in "a 70-cm column", or acting as a modifier qualifying the second modifier rather than the noun, e.g. water-solvable compound, in words like non- Newtonian, by-product.

iii) The use of symbols and superscripts and subscripts.
iv) The use of weights and measures.
v) The recommended spelling.

The above list illustrates the amount of concern given to establish criteria for judging the adequacy of computer-generated abstracts rather than to produce an exhaustive set of instructions. The length of the abstract can vary from one editor to another. While the American Chemical Abstracts takes the view that the abstract should provide a reasonable balance between brevity and length, generalisation and detail when considering the issue of how long should an abstract be. Other editing organisations maintain that most documents can be abstracted in fewer than 250 words. Storage limitations within computerised retrieval system may require abstracts as short as 100 words.

Issues related to the content of abstract tend to be concerned with the elements an abstract should contain, and the kinds of concepts an abstract should include to enhance retrieval.

The following table indicates which elements to include for each type of document [Weisman, 1972: 231].

Type of Document	Elements of the Abstract				
	Purpose	Scope	Method	Results	Conclusion
Report of an Experiment	X		X	X	X
Literature Review	X	X			X
Exposition of Theory	X				X
State of the Art Review		X			X
Critique	X	X			X
Handbook	X	X			
Textbook	X	X			
Report of an Application	X	X	X	X	X
Case Study	X		X	X	
Proceedings and Other Collections		X			

Figure 3.1 Elements of the abstract.

Many of the in-house guidelines are designed to guide and instruct the abstractor, largely by example, in the details of abstract preparation. The Chemical Abstracts Society draws the abstractor's attention to the content of the abstract and gives a detailed outline of what the first sentence of the abstract should include and what the remaining sentences should consist of. Finsbury Data Services, in a document called Abstracting Rules, places great emphasis on the inclusion of company names, people's titles and positions in text line abstracts [Finsbury Data Services]. With the advent of computerisation and international abstracting services, the need for standards for writing abstracts was recognised and the American National Standards Institute (ANSI) consequently established its own legislation on this matter. The ANSI standards identify the main content elements to include purpose, methods, results, conclusions [Borko et al, 1974]. Most international abstracting societies tend to adhere to these standards; however, these standards are only voluntary and hence cannot be enforced.

In terms of extracting information from the original document for inclusion in the abstract, very few instructions are issued to abstractors. The extraction process is usually left to the abstractor's initiatives. Textbooks tend to provide general guidelines which are illustrated in the following list:

> "Read the original article several times.
> Underline essential concepts.
> Include purpose, scope, methodology, results and conclusions.
> Retain the author's choice of terminology.
> Convey the character, style and content of the original as briefly as possible.
> Exclude your own prejudices.
> Refrain from evaluation or judgement of the original document.
> Arrange the content of the abstract to mirror the original...
> Sign or initial the abstract to indicate responsibility for its content"
> [Brown et al, 1971 :14].

The abstracting community consists of authors of the original document, subject area experts or professional abstractors. One would assume that the ideal abstractor would be the author of the document. Professional abstractors, however, dispute this assumption and argue that authors have too narrow a perspective of their audience, tend to produce too technical abstracts and consequently fail to reach users from other disciplines or with different applications in mind. They may be interested in promoting their papers. Authors may not be the best interpreters of their work. Their abstracts often need to be amended or rewritten by professionals for a wider audience. They may be also interested in promoting their papers.

Many abstracting organisations employ professional abstractors who have received a formal training in abstracting and are not necessarily subject specialists. They often become competent in the field to which they are assigned and produce high-quality abstracts. In contrast to this group of abstractors, there are subject area experts who are professional in their field but not in abstracting. Cleveland and Cleveland express some reluctance in asking the latter group to write abstracts for a number of reasons. They argue that an expert in a given area of mathematics, for example, may not be highly competent in another specialised area of mathematics and therefore may not be able to follow a theorem proof in that area. Another expert may resist a new idea from a young author, or may oppose the author's view on a given topic. His/her views will affect the results of his/her abstracting. However, it is felt that it is easier to train a subject expert in abstracting than to make a professional abstractor competent in a given topic [Cleveland et al, 1983].

Because of the growth in the volume of published literature abstractors may produce up to 20,000 abstracts a year. An expert can abstract a paper in 15 minutes. Most professionals find abstracting a tedious and tiresome activity and tend to involve themselves in the indexing and classification of documents in order to maintain control of the quality of their abstracts. They develop, over the years, a set of strategies which help them identify salient information in documents for inclusion in their abstracts. Some of these strategies are related to their reading skills, others are derived from their knowledge of how people communicate their findings and of the kinds of textual linguistic structure writers employ to highlight important elements of their research.

Effective reading is a fundamental skill for an abstractor that involves comprehension, the art of surveying, skimming, scanning and occasionally critical reading. Many experts argue that reading is accomplished by a series of eye fixations. It is rare for abstractors to read an article straight through from beginning to end. There is a general consencus that abstractors are more inclined to browse at first, and then scan particular passages to derive items of interest to them as rapidly as possible. They may jump backwards and forwards in the text, being guided by the physical layout of the paper. Reading becomes a goal-directed exercise.

Of similar importance is the ability not only to infer the function of an article, which is vital to its comprehension, but also to utilise the organisation of the article and its paragraph structure to locate important passages. One of the very first things abstractors find out is whether a paper aims at describing an experiment or a technique, reporting a study or putting forward a new theory. Much of the information is deduced from the layout, the structure, the use or lack of diagrams and tables. Abstractors claim that significant clues are usually given by the author. Having identified the function of the paper abstractors can use this information to guide their reading strategy through the text. If the article is, for example, argumentative in nature, attention is shifted to locate the arguments and counter-arguments, and then the conclusions. With regard to thematisation, authors tend to use surface-level features to mark the topic of each section or paragraph and this pattern is often consistent throughout the paper. Getting to know the author's style of writing and interests is a great asset in abstracting. Short cuts can be used in such cases to locate important information. Some abstractors argue that the introduction and conclusion in an article is often adequate basis for abstracting. Again some types of papers in particular those published in proceedings, share similar interests and a common pattern that may be dictated by the conference, and as a result abstractors find it easier to abstract the entire set of papers rather than individual papers with a wider coverage of subjects.

Beside understanding the function and the organisation of a paper, some abstractors argue that it is essential to understand the content of the paper and the wider issues and major concepts of the domain. Skimming a paper for important concepts in a given domain can help the abstractor to recognise the key sentences of a passage. If the well-established concepts do not appear in the paper abstractors' knowlege of the domain may be then used to allude to the main ideas.

Abstracting is indeed an art and a skill. It is a great asset to an organisation to have available one or more expert abstractors who deal with the problem continuously and who have the knowledge and experience to understand what the real problems are. They deal with diverse knowledge at various depth and on a daily basis. Their ability to interact and integrate the three forms of knowledge can provide a strong basis to an understanding of the various activities involved in processing and extracting information.

4. A Knowledge System Approach to Automatic Abstracting

The complexity of operations involved in abstracting necessitates an interdisciplinary approach. Such an approach, however, imposes a number of constraints: there are problems that stem from the integration of such large amounts of knowledge and there are difficulties in identifying an acceptable depth for such knowledge. It is believed that a study of how experts perform the task of abstracting can provide thebasis for an interdisciplinary approach and can give insights into the way these forms of knowledge are integrated.

Professional extractors handle and frequently solve information problems for which algorithms do not exist and which are characterised by ambiguity, uncertainty and poor structure. The human ability to identify the information needed to solve such problems is often taken for granted and is not yet fully understood. One of the main objectives of this study is to find out if one can program computers to exhibit similar problem-solving capabilities in the domain of abstracting.

The basic assumption in this thesis is that experts know how to organise and access their knowledge when they are abstracting. They know how much knowledge is required and when and how to use it. This form of knowledge is referred to as heuristic knowledge. Their ability to develop strategies that recognise the effective use of their own cognitive resources is the fundamental key to the problem of abstracting.

These strategies may vary from expert to expert, depending on the type of document to be abstracted, the discipline, readers' requirements, abstracting organisations' policy and the writing style of the document's author. It is reasonable to assume that there is an underlying framework upon which experts base their strategies. It is further assumed that these strategies can be expressed in the form of rules which can be represented and implemented in a computer program.

The current research proposes an approach that aims at investigating whether the process of abstracting can be divided into tasks and subtasks. This is achieved by eliciting the knowledge and skills of expert abstractors. Identifying experts'strategies is the first step towards examining how abstractors apply their knowledge and skills to specific situations. Strategy, here, is described as the purposeful use of resources. Using Moray's definition, strategy implies that "... the agent desires to bring about some ends, and takes steps which seem most likely to achieve that end given the resources available to him" [Moray,1978: 302]. The next step is to express these strategies in a from that can be manipulated by a computer system. A suitable approach to capture experts' knowledge and strategies is the rule base system formalism. The aim of this formalism is to map the acquired knowledge into rules. These rules will then be implemented into the computer system and applied on a set of scientific papers. Such a system, which is also referred to as knowledge-based system in the AI circles, is a valuable tool to the understanding of the abstracting process. Acquisition of knowledge from experts proceeds through a number of stages namely identification, conceptualisation, formalisation, prototyping, creating user interface, testing and knowledge-base maintenance [Hayes-Roth et al, 1983]. The purpose of this research is not to produce a commercial system but rather to use AI technology to capture and apply the expertise required for abstracting a given set of documents. For this reason, the study will emphasise the first four stages only. Each of these stages is outlined below.

In the course of building a knowledge-based system, there is a constant revision which may involve refinement of strategies, reformulation of rules and/or redefinition of concepts. Figure 3.2 indicates these revisions.

4.1 Identification

The first stage is to characterise the important aspects of the system. This involves identifying the participants, the problem characteristics and resources.

4.1.1 Participants identification

Identification and availability of expert help is critical to the building of a knowledge-based system. Since this project requires long-term commitment, the process of acquiring the necessary knowledge from experts is expensive and difficult. It was felt important to identify experts who have a vested interest in this research. Thus a group of five experts was identified, each with differing specialism covering the following fields: information science, polymer engineering, education, patent and iron casting.

The abstracting expertise is distributed among researchers in academic institutions who are also constantly engaged in keeping themselves up-to-date with the published literature and have, as a result, developed effective ways of locating relevant information from documents. Another group consisting of six researchers covering the field of computer science joined the experts group.

The participants agreed to be present over a period of three years and were contacted at various stages of development. Some participants were happy to participate to various experiments in order to assess the extent of knowledge required for abstracting, while others preferred to explain their strategy by means of informal discussions.

4.1.2 Problem characteristics

Having identified the participants, it is important to identify the problem areas. Several iterations of problem definition may be required to arrive at a feasible definition for the resources available. The objective of this stage is to obtain a description of typical problems related to abstracting, to elicit strategies of how to solve these problems and to understand the reasoning that underlies these strategies. The following issues need to be answered before proceeding into the next stage:

1. What are the tasks involved in abstracting?
2. Can these tasks be subdivided into subtasks?
3. What aspects of human expertise is associated with each task and subtask?
4. What strategies are used for each subtask?
5. What is the nature and depth of knowledge required for each subtask?

4.1.3 Resource Identification

Resources are required to elicit the knowledge, implement and test the system. This includes knowledge sources, expert's time, computing facilities and financial commitments. Time is a critical resource as knowledge acquisition may involve many months of intensive activity. This imposes a serious financial constraint as expert time is very expensive. The knowledge utilised in the system does not necessarily have to be restricted to human experts, of course. It may also come from books, published papers and manuals.

Computing facilities are another constraint because appropriate knowledge acquisition tools are important in the development of such systems. Two programming languages, Prolog and SPITBOL, were selected as basic tools for this study.

Prolog is based on the branch of symbolic logic which is called predicate calculus. Its attractive and useful features stem from its declarative aspect. This facility is very important to this research as will be seen in chapter 5. SPITBOL, on the other hand, is a powerful language chosen for its ability to analyse the documents from a linguistic perspective and to store the analysis in a declarative form ready for Prolog manipulation.

4.2 Conceptualisation Stage

The previous sections were concerned with two forms of expert knowledge, that is, specialised knowledge of domain and heuristic knowledge. Another form of knowledge, denoted as linguistic knowledge in this study, occupies an equal powerful status in abstracting. Linguistic knowledge includes the knowledge required to read and understand texts. The activities involved in processing texts are explored in terms of model building. Each text has a text world model composed of concepts and relations in a knowledge space where the textual world is defined as "...the cognitive correlate of the knowledge conveyed and activated by a text in use" [de Beaugrande, 1980]. The conceptual text world model here is based on findings from cognitive science, namely the studies of van Dijk and Kintsch, and from linguistics analyses by de Beaugrande and Dressler.

The key concepts and relations for the three forms of knowledge are made explicit during the conceptualisation stage. Identification of these concepts provides the conceptual base for this system.

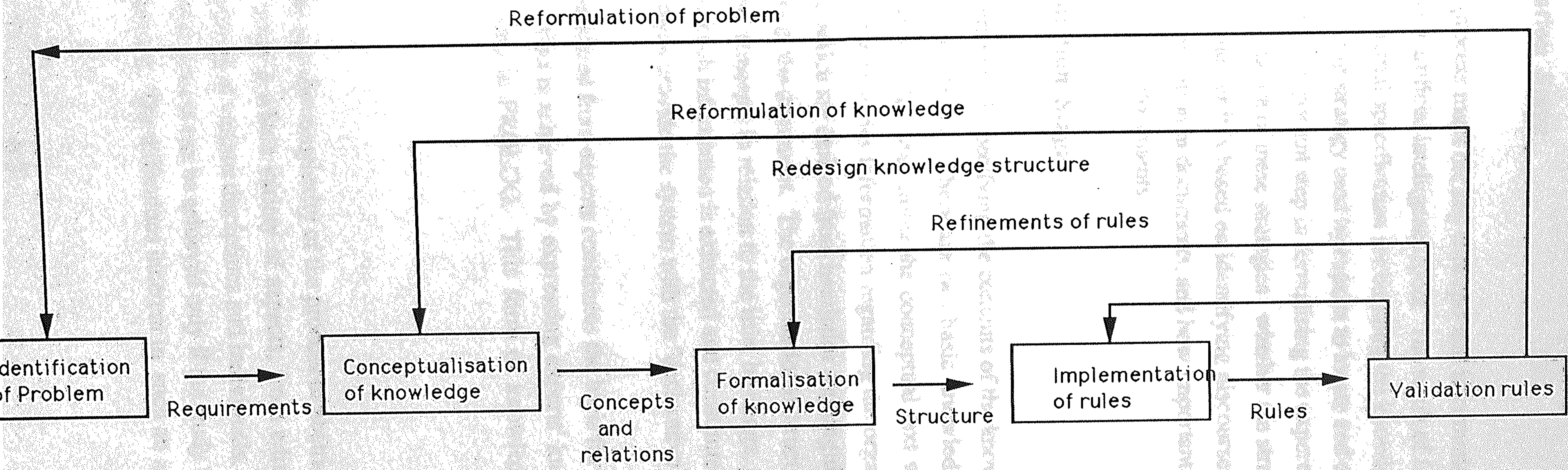


Figure. 3.2. - Stages of Knowledge Acquisition [Hayes-Roth et al, 1983:139].

4.3 Formalisation Stage

The formalisation process maps the key concepts and information flow into more formal representation based on artificial intelligence tools. The result of formalising the conceptual text world model is a partial specification for building the knowledge base for the system. Uncovering the underlying strategy used by experts to locate and extract salient information for abstracting can be an important step in formalising the experts' knowledge. It can also provide clues about the nature of these strategies : whether the strategy relies heavily on the structure of the text, whether it is based on identifying discourse pointers used by authors to signal important information in documents, and how important the knowledge domain is in identifying key issues in documents.

4.4 Implementation Stage

This stage is concerned with specifying the contents of the knowledge base, the inference rules and the control strategies. There are two basic knowledge bases assigned to this system. The first knowledge base stores the conceptual text world model, whereas the second knowledge base provides information regarding the organisational structure of the text. A set of rules, which are derived from heuristics and linguistic knowledge, is applied to identify concepts in the document. The organisational structure of the document gives the position of these concepts in relation to the title, headings, paragraphs and sentences. For each document, such information is extracted and constitutes the text knowledge base. These knowledge bases provide the system with the "declarative type" knowledge.

The knowledge acquired from experts constitutes the basis for the inference rules and the control strategies. This is achieved by expressing experts' knowledge in terms of rules which will be written in PROLOG. This form of knowledge represents procedural knowledge.

5. Conclusion

This thesis addresses the complexity of the process of abstracting by analysing first the problems which cover a whole variety of aspects, from editing text to integrating the techniques and findings of various disciplines and individuals. It is strongly believed that understanding of this process can be achieved only if the study is approached from a multi-disciplinary perspective. The difficulties inherent in such an approach are recognised and the methodology suggested in this chapter offers an interesting way of combining empirical practices with theoretical findings.

CHAPTER 4

PRELIMINARY STUDIES AND ACQUISITION OF EXPERTS KNOWLEDGE

PRELIMINARY STUDIES AND ACQUISITION OF EXPERTS KNOWLEDGE

1. Introduction

The previous chapters have shown that, even though there are many theories and perspectives relating to the study of information extraction and abstracting there is still little concensus regarding how people extract important information from documents. However, it is known that professional abstractors achieve outstanding performance. Their expertise includes an understanding of their readers and writers community, a skill at applying an appropriate reading strategy which enables them to locate effectively important sections in a document, and a private knowledge drawn from their past experience which allows them to extract main ideas in a relatively short time. This private knowledge consists largely of rules of thumb, "little discussed rules of good judgment", referred to as heuristics in the AI community. Heuristics enable experts to devise suitable approaches to problems and deal effectively with ambiguous, incomplete and errorprone information. Such knowledge cannot be acquired from books. Elicitation of this knowledge can help unravel some of the complexities involved in the process of abstracting and throw insights into that process.

This chapter focusses primarily on acquiring knowledge from expert abstractors and practitioners. The expertise to be elucidated may be partly a collection of facts, procedures, judgemental rules and specialised skills related to abstracting. By elucidating expert knowledge it is hoped to decompose the process of abstracting into tasks. Examining these tasks aids in understanding what makes abstracting so complex, and at the same time provides a guide to focus on skills that relate to critical steps in abstracting.

Knowledge acquisition is recognised to be a difficult process. Getting an expert to expresss the knowledge used for solving a problem is the bottleneck in expert knowledge elicitation [Hayes-Roth et al, 1983: 129]. It requires a long series of interviews and discussions with experts in the domain. As Shortcliffe pointed out, the formulation of expert knowledge is no simple matter because experts typically do not structure their decision-making and problem-solving in any formal way, and they may have great difficulties in isolating the steps and describing their strategies [Shortcliffe, 1976].

The approach to knowledge acquisition is carried out in terms of a process model of how to construct an expert system, as explained in chapter 3. It proceeds through several stages. Only the first two stages: the identification and conceptualisation stages are described in this chapter. The first stage attempts to identify the class of problems and the criteria that experts work with as well as the resources available for this study in terms of manpower, time and financial constraints. The second stage uncovers the underlying structure of the abstracting process in terms of tasks and strategies. The remaining stages are closely related to implementation and hence will be the object of study in the next chapter.

2. Problem Identification and Characteristics

In the process of expert knowledge elicitation, a group of five professional abstractors from different abstracting organisations participated in this study. Their subject domains covered the fields of polymer engineering, information science, patent, iron casting and education. A series of interviews were carried out such that the interviewees were relaxed and did not feel threatened.The early interviews were oriented towards defining the tasks involved in the abstracting and then examining if each task could be subdivided into subtasks. The aim of these interviews was to:

1) describe what abstracts should contain;
2) define the process of abstracting;
3) define the expertise required in producing an abstract;
4) identify the background knowledge required to generate abstracts.

3. Conceptualisation

Having identified the aim of the interviews, the following questions were addressed during the preliminary interviews:

1) What is the role of the abstract?
2) How is relevance measured when abstracting?
3) What are the essential aspects of human expertise required to produce an adequate abstract?

There is strong agreement among professional abstractors about the role of an abstract. They stress that abstracts aim at assisting users in deciding whether the article is relevant to their area of interests, and whether to retrieve the original document. Therefore, sufficient information must be provided in abstracts to permit the user to evaluate its relevance. Although needs vary from user to user, abstractors argue that there is a common pattern recognised among the user community. Users must be able to identify fairly accurately and very quickly whether the article contains information pertinent to their work.

On the notion of relevance, the definition of what constitute relevant information is to an extent dictated by the length of the abstract and other economic factors. As these abstracts are produced for insertions into large computer data bases, fewer abstracts would be stored in the file if abstracts were long, resulting in a reduction in the dissemination of information. Short abstracts, however, may not convey enough information for the user to assess its relevance. The general concensus is that an abstract should at least contain the following elements of the paper: purpose, method or technique used, results and conclusion. One abstractor pointed out, nonetheless, the necessity to enclose details such as the name of the polymer, the company name, the type of chemical reactions that may result and any commercial applications. In some organisations indicative abstracts only are produced. What is apparent so far is that abstractors have a clear strategy comprising a list of questions which guide their reading and their search for answers, such as:

i) What is the aim of this paper?
ii) Is the author advocating a new method?
iii) Is the author reporting results?
iv) Is the author describing the technique in this paper or just referring to it?
v) Is the technique claimed as applicable? and in what circumstances?
vi) What are the conclusions?

On the question of the aspects of human expertise essential for abstracting, an important issue emerged relating to the ability to read the paper effectively. Abstractors employ various reading strategies such as scanning, skimming or intensive reading.

Another important issue was raised relating to the role of the experts background knowledge of the topic in the paper. This is known as the abstractor's domain of knowledge. Here two differing views have emerged. The first group argued strongly for the need to be knowledgeable in the domain in which papers are abstracted. These abstractors pointed out that innovative aspects presented in the published paper could only be recognised by readers or abstractors who are experts in their field. Hence knowledge of the field is an important aspect of expertise.

The second group of abstractors stressed the importance of good reading skills. Experts, in their view, can identify relevant issues discussed in the paper even if the paper lies outside of their domain knowledge. They have over the years developed kinds of "hunches" which help them locate and identify important information. They suggested that they rarely read an entire paper to produce an abstract. They stressed the importance of the introduction and the conclusion of the paper. One of the abstracting organisations emphasized in its guidelines the importance of these two sections by stating clearly that these sections could serve an adequate basis for an abstract [LISA, 1984]. This is known anomg abstractors as the short cut strategy.

Abstractors also pointed out that relevant information is often highlighted in the paper by the authors. A number of linguistic features are utilised by authors to signal relevant sections in their paper; these cues guide the reading strategy of these abstractors. Thus, sensibility to the interpretation of authors cues is another aspect of abstracting skill.

The preliminary meetings with the expert abstractors helped to identify the basic skills required to perform abstracting. These skills are grouped into five major tasks as illustrated in figure 4.1.

The results of the previous investigation has led to the following list of strategies applied when establishing the significance of information contained in a given paper:

(1) Note the title. This sets the stage by identifying the theme of the paper from the author's perspective.

(2) Pay special attention to the textual relevance cues to identify statements relevant to the content of abstract.

(3) Consider the name of the author(s) and the source of publication. This will indicate the relevance of the paper within that field.

(4) Note the surface structure of the paper. This will form a useful guide to scanning sections of the paper.

(5) Read the introductory and final sections carefully. These usually provide the most significant leads for relevant issues discussed in the paper.

Having specified the abstracting tasks, it was then possible to uncover the basic concepts involved in each task and to attempt to break each task into subtasks. Further meetings with experts dealt with understanding the nature and depth of knowledge relevant to each task; this was achieved by asking experts to explain the procedures used and justify their reasoning. These meetings revealed the strategies employed by these experts when abstracting. A description of each of these tasks is given in the following sections, and illustrated in figure 4.2.

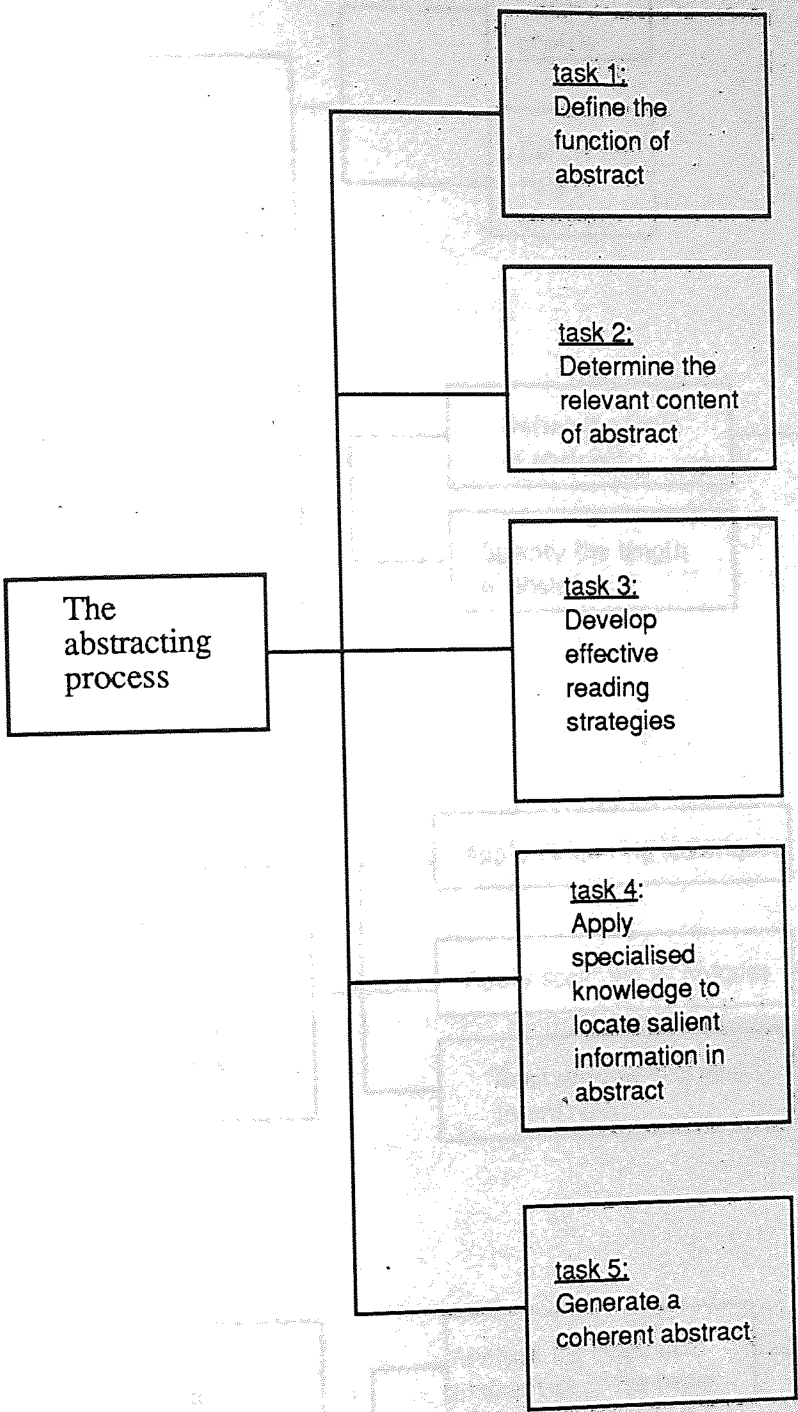


Figure 4.1 Abstracting tasks

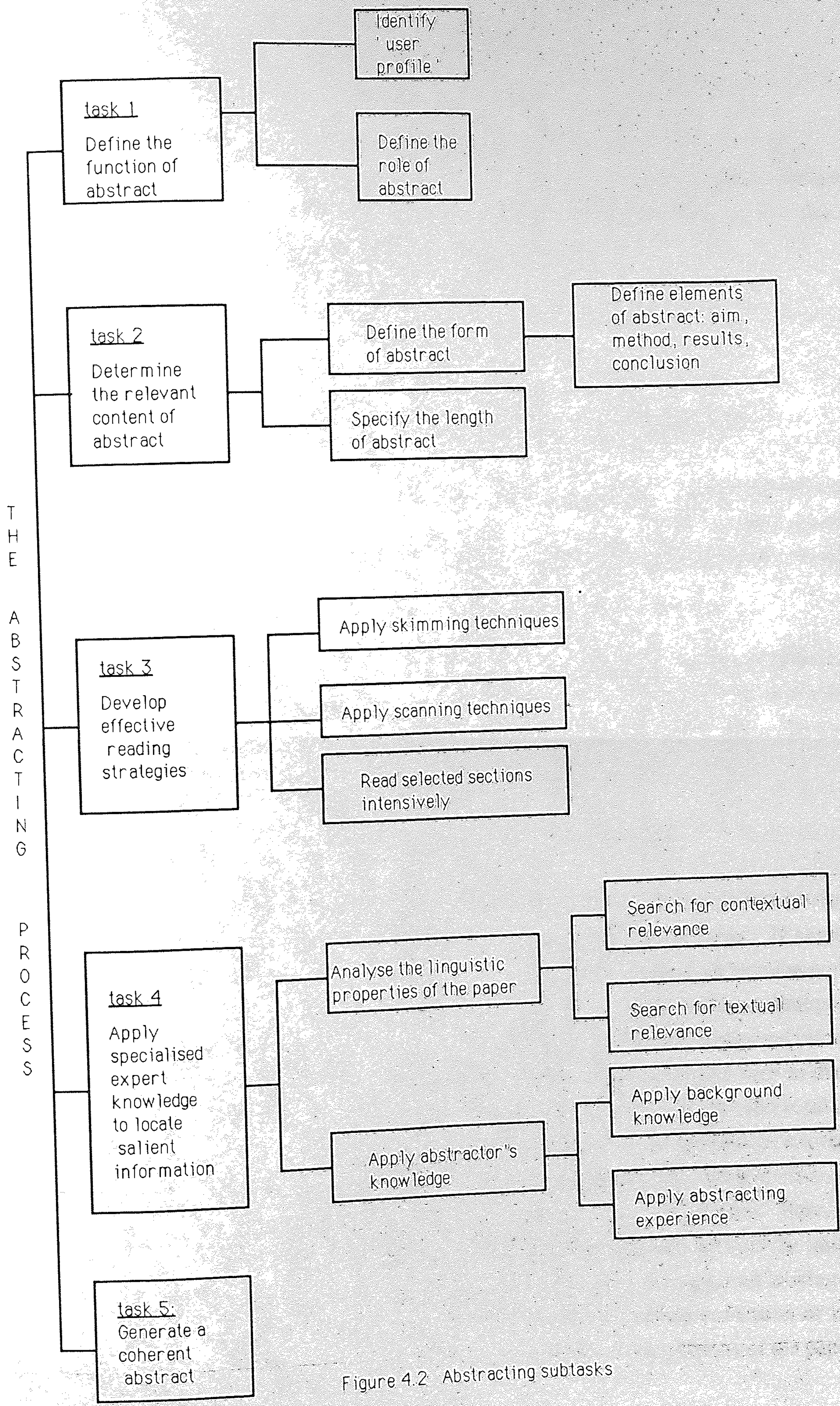


Figure 4.2 Abstracting subtasks

3.1 Task 1: Define the Function of the Abstract

The first task deals with specifying the function of an abstract according to "user profile". Further meetings help draw a strategy that abstractors follow to carry out this task. This strategy can be described as a set of subtasks given below:

i) identify your user,
ii) identify the role of the abstract.

3.2 Task 2 : Determine the Relevant Content of the Abstract

The current trend is to produce an informative abstract to allow users to assess better the relevance of the paper to their interests. Consequently, abstracts should contain the following elements: the aim of the paper, the technique employed, the results and conclusions attained.

The level of details describing these elements is constrained by the length of the abstract imposed by the information retrieval systems, as well as the cost and time available. The average size of an abstract is up to 250 words [Cremmins, 1982: 50]. On average, abstractors spend about 25 minutes to produce an abstract.

3.3 Task 3: Develop an Effective Reading Strategy

Most abstractors agreed that effective readers always have a purpose in mind before they start reading an article and this, in turn, dictates their reading strategy. If someone is looking for specific information, one has certain expectations about its location within the paper. With a specific purpose in mind, abstractors use selective reading techniques from scanning, skimming to intensive reading. By scanning they locate the required information in the paper. For instance, experts may search for specific information such as the use of capital letters to locate the name of a technique, or figures to locate the results of an experiment. The abstractor's eyes move down the page in a zig-zag manner, searching for clues and familiar patterns. Skimming is a more thorough activity which involves a quick preview telling the abstractor what kind of information this paper contains. Here, the title serves as a good basis for previewing. Looking at the headings to identify interesting sections is a common practice for overviewing. One abstractor suggested looking out for headings of sections that reproduce the title - this is an explicit indication of relevant information. Intensive reading is then applied, but only to small portions of the paper.

The following are examples of the skimming strategy used:

i) Note the title, author because these may suggest the theme of the paper.

ii) Rapidly read the introduction or the first paragraph of the paper to obtain a quick overview of the main points discussed.

iii) Read the headings and subheadings to determine concepts and important issues discussed.

iv) Read the last paragraph rapidly as that frequently summarises the main points.

To summarise, task 3 can be subdivided into three main subtasks, as shown in figure 4.2. The first subtask refers to the ability of utilising the title and headings which indicates the topic of the paper. The second subtask relates to the effective use of eye movements in scanning familiar patterns, whereas the third subtask involves intensive readings of selected sections of the paper.

3.4 Task 4: Apply Specialised Expert Knowledge

3.4.1 Analyse the Linguistic Properties of the Paper

When writing a paper, the researcher wishes to communicate his/her work to the outside world and to attract in some way other researchers' attention. Hence the communication process implies that the information communicated is relevant. This assumption of relevance is essential in communication. By relevance, it is understood to be... "the result of an operation by which a reader/hearer, or a method of analysis, assigns some degree of importance to some property of the discourse" [van Dijk, 1979: 113]. The analysis of relevance is made at the level of paragraphs and sentences as well as at the global level of the text. This kind of relevance is termed contextual relevance.

Contextual relevance may be also defined in terms of cognitive set factors [van Dijk, 1979: 25]. These factors include specialised domain knowledge, interest, motivation, etc. It is what abstractors find important, a value judgement based on their knowledge of important issues of the domain of the text, of user's interests and of the aims of an abstract. Abstractors have to reproduce not only what is important in the text, but also what the users may consider important.

Textual relevance is another kind of relevance which is assigned on the basis of textual structures. The text surface structure indicates the level of relevance. For example, some parts of the paragraph are regarded as more important than other parts. Also some parts of the sentences are more important than others. Relevance is also reflected in the hierarchical representation of the text in which high level headings are labelled differently from low level headings.

In other words, abstractors appear to use two kinds of specialised linguistic knowledge when processing a paper: the first kind of knowledge makes use of a list of contextual relevant markers, and the second kind interprets the structure of the paper to infer textual relevance. These two kinds are not carried out exclusively but linked. The next sections explains these two kinds of relevance further.

3.4.1.1 Search for Contextual Relevance

What is implicit in the above section is that we should focus on the fact that writers of a paper seek to communicate relevant information. To achieve their intention, it is in their interest to use a range of different stimuli which would manifest this intention to the reader clearly. The most relevant stimuli are the use of contextual cues, which include discourse markers, the use of graphs and tables and the choice of the title and vocabulary. Abstractors have encountered over the years a number of discourse markers which enable them to focus rapidly on potentially relevant chunks of information. These discourse markers are classified into three main types: graphical, thematic and lexical markers. They scan the paper searching for these markers to locate important sections. Then the abstractors "zoom in" on the sentences containing these markers and evaluate their importance in relation to the abstract. Depending on the type of marker, they associate a different degree of relevance to each sentence. For example, sentences containing concluders will score higher than those containing connectives (See Table 4.3).

1. **Graphical markers:**
- boldness
- capital letters
- italics
- underlining
- framing

2. **Thematic markers:**
- choice of concepts in the title
- frequency of occurrence of concepts
- name of the author(s)
- references at end of paper

3. **Lexical markers:**
- use of "direct relevance" terminology: important, relevant, crucial, now, etc.
- theme markers: the subject/theme/approach/methodology... is, the paper deals with, in the paper we propose a new approach to, our current system consists of, our work investigates, we have chosen to test, in this study we investigate...
- summarisers: in summary, in other words, in brief, to summarise, we have seen so far,...
- concluders: in conclusion, we conclude, the study showed, the results of work point out to the following conclusions,...
- purpose markers: the purpose of this study/ research/ paper/experiment, the aim of this research...
- connective markers: on one hand, on the other, first, second,...,finally, the first, the second..., in addition, furthermore, however,...in the previous section, this section, in the following section/chapter...

Table 4.3 An empirical list of contextual relevance markers in scientific papers

3.4.1.2 Search for Textual Relevance

From the interviews it was revealed that some portions of the text are likely to contain more salient information for inclusion in the abstract than others. A closer examination of these portions of text showed that experts relied on the introduction and conclusion of an article to generate abstracts. They argue that the purpose of the introduction is to set the nature and scope of the problem being investigated, to state the method of investigation and to report the principal results of the investigation. Hence the introduction could provide the first source of information. In some cases, abstractors require to include a summary of the results which are usually described in the conclusion. If these sections do not conform to the standards expected, the introduction may outline the content of the paper and refer to important sections as illustrated by the example below:

"...I shall outline the general characteristics of programs that are necessary for practical software, and I shall show how conventional software engineering attempts to meet these requirements. I shall show that many AI problems appear to require a program development methodology that does not yield programs that exhibit the necessary characteristics of practical software. I shall attempt to delineate the resultant problems as well as sketch out some potential routes to solutions..." [Partridge, 1986: 28]..

As for the remaining sections of the paper, experts concentrate their reading strategy on the first and last paragraphs of each section. It is customary procedure to announce the problem at the beginning of the paper. The main body reports the work and brings in evidence to support the methodology. The final section sums up the evidence brought forth and the conclusions about the problem and its solution.

A similar pattern is reflected in the structure of paragraphs. The tendency is to use the beginning of a paragraph to draw an inference from the previous paragraph and introduce a new topic or a new idea. This is achieved in one or two sentences. Occasionally, writers choose to sum up in the last sentence of a paragraph, in what is called the closing sentence. Thus, experts concentrate their resources on the first and last paragraphs and the first and last sentences of each paragraph (Table 4.4).

Beside initial and final position, many other means are applied to assess textual relevance for abstracting purposes. Paragraphs with more than two decimal numbers (e.g. 4.1.3) are subordinated to those with two decimal numbers (e.g. 4.1). Consequently abstractors disregard these sections as they expect that these sections will contain a level of detail not relevant to abstracting.

In some cases, abstractors focus their attention on the tables and diagrams to obtain an overall picture of the paper. Enumeration is another textual manifestation of the degree of relevance of a given statement, which catches the eyes of the abstractor.

In short, abstractors appear to interpret a great deal of information about a given paper from the structure of textual space. This view should not be taken to imply that meaning is in the text, but that there is an interaction between the properties of a text and the abstractor who is processing that text. What is suggested by the abstractors is that there are linguistic patterns in the text that writers utilise to present information. These patterns help writers ensure that information is appropriately identified by the reader who is the abstractor in this case. Similarly, readers look out for these linguistic conventions to locate important information.

A list of textual relevance cues:
. the introductory paragraph
. the concluding paragraph
. the first paragraph of each section
. the last paragraph of each section
. the positional markers
- the opening sentence of each paragraph
- the closing sentence of each paragraph
. enumeration
. level of headings
. tables and graphs

Table 4.4 Textual relevance cues

3.4.2 Apply Abstractor's Specialised Knowledge

The aim of this section is to elucidate the nature of knowledge necessary to perform abstracting. As a result of the interviews, two types of knowledge were highlighted. The first one referred to a background knowledge about the topic needed when processing a paper. The second one related to the expertise required for locating salient information for inclusion in an abstract. To this end, three experiments were carried out.

The aim of the first experiment was to assess whether background knowledge is crucial to process a given paper and generate an abstract successfully. The paper selected in this experiment is well written and did not contain any technical jargon.

The second experiment was designed to examine the importance of the introduction and conclusion when writing an abstract. As part of this experiment, it was hoped to investigate the criteria used by researchers to ascertain the relevance of a paper to their work.

The third experiment was aimed at identifying the location of the informational constituents in a paper that formed the basis of an abstract.

For the above experiments, it was clear that participants was should have following properties:

1) actively involved in research;
2) able to articulate their thinking processes;
3) willing and free to take part in the experiments which may last over two years;
4) professional in their undertaking of these experiments; and
5) without specific experience in abstracting.

It is hoped that the results of the above experiments will enable a strategy for the production of an abstract to be formulated. The strategy will then be analysed to establish whether it can be expressed in terms of rules and a knowledge base. If so, it is intended to apply this strategy in a computer simulation.

3.4.2.1 Experiment 1 : Relevance of the Background Knowledge

3.4.2.1.1 Description

Participants were selected to abstract a paper in the field of expert systems. Participants 1 and 3 shared common research interests in linguistics. Participant 2 was a research fellow in psychology. Participant 4 was an information scientist responsible for compiling research papers in various aspects of computer science. Participant 5 was a researcher in the field of artificial intelligence. The first three participants had no knowledge of expert systems. Participant 4, however, had a basic understanding of information technology. The fifth participant had strong research interests in expert systems and had specialised knowledge of the domain of the paper. All five participants were asked to read the paper, which was on the topic of knowledge elicitation in expert systems and to produce an abstract. This paper was given to them without an abstract. No constraints were set regarding the length of abstract or reading time. The abstracts produced by the five participants are given in Appendix A.

3.4.2.1.2 Material

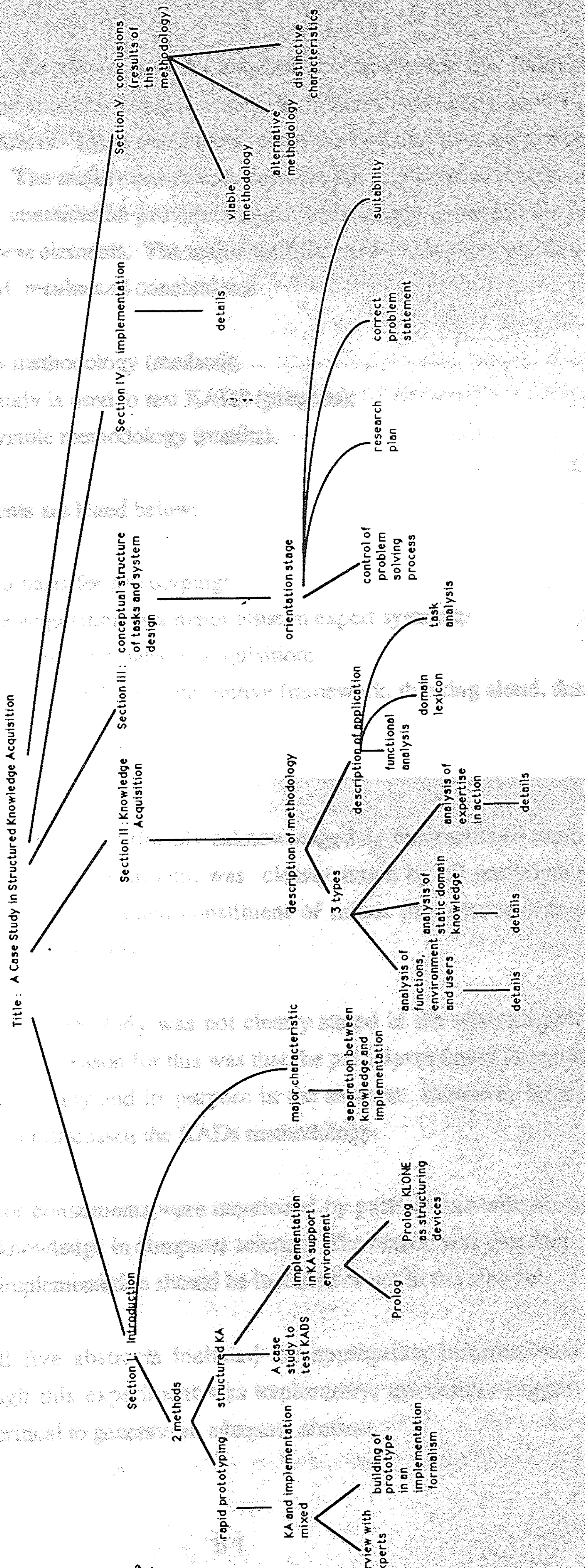
The paper being analysed is published by the international joint conference on artificial intelligence and is titled :"A Case Study In Structured Knowledge Aquisition" [Greef et al, 1985]. The content of this paper is described as a tree structure, shown in figure 4.5.

In the first paragraph of the introduction, the authors stated that a major problem in building an expert system is the methodology used for acquiring expert knowledge. They suggested an alternative methodology called KADS (Knowledge Acquisition Documentation and Structuring) and described its major features in the second and third paragraphs. In the last paragraph of the introduction they highlighted the advantages of KADS methodology. This methodology was presented in terms of a series of knowledge engineering tasks, which were described in detail in section 2. The conceptual structure of these tasks was given in section 3 and the implementation of this methodology on a small prototype expert system was described in section 4. The conclusion (section 5) claimed that this methodology was viable and emphasised its distinctive characteristics.

3.4.2.1.3 Analysis of abstracts

The abstracts generated by the five subjects were examined using three procedures. The first procedure analysed the informational constituents of each abstract, the second related these constituents to the structural organisation of the original document and the third investigated the use of discourse markers in determining textual and contextual relevance of these constituents.

Each abstract was broken into its informational constituents. The function of each informational constituent was to mark off which information the subject was treating as important information, that is, information that the author wished to communicate in his/her paper. The concept of an informational constituent was closely associated with knowledge structure rather than propositions, phrases or words.



KA: Knowledge Acquisition

Figure 4.5 Tree structure

For a case study, the elements of an abstract should include the following details : purpose, method and results. Table 4.6 lists the informational constituents identified in each of the five abstracts. These constituents are classified into two categories: major and minor constituents. The major constituents describe the important elements of an abstract whereas the minor constituents provide either a background to those elements or some details to support these elements. The major constituents for this paper are those describing the purpose, method, results and conclusions:

1) the KADS methodology (method);
2) the case study is used to test KADS (purpose);
3) KADS a viable methodology (results).

The minor constituents are listed below:

4) KADS as a basis for prototyping;
5) knowledge acquisition as a major issue in expert systems;
6) two approaches of knowledge acquisition;
7) characteristics of KADS : interpretive framework, thinking aloud, data,
8) tasks in KADS;
9) implementation details.

Constituents 1 and 3 were uniformly acknowledged as statements of main ideas of the paper. The second major constituent was clearly stated by all participants except the second one. The most mentioned constituent of minor importance was constituent 9 providing implementation details.

The purpose of the case study was not clearly stated in the abstract produced by the second participant. The reason for this was that the participant failed to report the fact that the paper was a case study and its purpose in the abstract. However the participant did mention that the paper discussed the KADs methodology.

Most of the minor constituents were mentioned by participants with no background or little background knowledge in computer science. The reason was that they were not able to assess whether implementation should be included or not in the abstract.

In summary, all five abstracts included the appropriate informational constituents required. Although this experiment was exploratory, the results suggest that domain knowledge is not critical to generate an adequate abstract.

Turning to the issue of relevance, the participants were asked to explain what aids were used in assessing the relevance of the information selected. All participants relied on both textual and contextual cues as shown in table 4.7. The list of these relevance cues are drawn with the help of participants. They all agreed that the title acts as a pointer to the new textual information. It provided the basis for contextual relevance, a basis to interpret the textual information and a structure around which the incoming information can be evaluated. Thus ideas that are central to the title are selected and the remaining ideas were usually overlooked.

Information Constituent in abstracts	Category	Abstracts of Participants					Elements of an abstract
		1	2	3	4	5	
1. KADS methodology.	major topic	x	x	x	x	x	x
2. Case study to test KADS	major topic	x		x	x	x	x
3. KADS as viable methodology (useful, possible)	major result	x	x	x	x	x	x
4. KADS as basis for rapid proto-typing	major result		x				
5. Knowledge acquisition as a major issue in expert systems	minor:back-ground knowledge		x	x	x		
6. The two approaches of knowledge acquisition.	minor background knowledge	x		x		x	
7. Major characteristics of KADS (interpretative framework, thinking aloud data)	minor: details of KADS		x				
8. Tasks in KADS	minor:details of KADS		x				
9. Implementation details(PROLOG, KLONE, statistical analysis (4 wks)	minor topic	x	x	x	x		

Table 4.6 Analysis of Abstracts

The way the writers presented their case study was particularly helpful in assessing the main issues of this paper. The introduction set the problems related to the topic discussed and intorduced breifly the suggested methodology. The remaining three sections described the methodology in more detail and the conclusion summarised the findings. Thus the surface structure matched the participants' expectations. The two informational constituents in the title, case study and structured knowledge acquisition, were strongly suggested as key issues in the paper. The next strategic move was to find out more on the one hand about knowledge acquisition and on the other hand the aims and results of this case study. Their expectancies about events, especially with respect to the structure of scientific papers, helped them in guiding their eye movements while scanning the paper for pertinent information. These findings agree with the abstractor's assumption that it is customary to announce the aim in the introduction, at the beginning of any scientific paper, to describe the case study in the body of the paper and to sum up the findings in the conclusion. Hence all five subjects turned their attention first to the introduction and conclusion, and then read the other sections in search of specific information.

Helpful devices in assessing relevance	Part 1	Part 2	Part 3	Part 4	Part 5
1. Textual relevance cues:					
. structure of the paper	x	x	x	x	x
. level of headings				x	x
. positional markers:					
. the first paragraph in the introduction	x	x	x	x	x
. the last paragraph in the conclusion	x	x		x	x
. enumeration			x		x
. tables and graphs					
2. Contextual relevance cues:					
. concepts in the title	x	x	x	x	x
. frequency of reference to certain concepts as reinforcement of main ideas	x		x		
. author's name as indicator to kinds of research interests					x
. references to situate the study within the field					x
. graphical markers	x	x		x	x
. lexical markers	x	x	x	x	

Table 4.7 Textual and Contextual Relevance Cues

Those participants with no background knowledge in expert systems relied heavily on lexical signalling devices. Some of these devices took the form of a phrase and generally occurred at the beginning of paragraphs as in the following examples:

i) A major problem (introduction:paragraph 1, sentence 1).

ii) A major characteristic (introduction: paragraph 2, sentence 1).

iii) Crucial to the methodology (introduction: paragraph 4, sentence 1).

iv) The results of analysis (conclusion: paragraph 1, last sentence).

Two other features have drawn participants' attention to important concepts. The first feature is the graphical marker. The authors used capital letters to describe the conceptual structure of the methodology (ORIENTATION stage, CONCEPTUAL model, RESEARCH PLAN...). They underlined the special characteristics of their methodology as in the following sentence:

"These properties have nothing to do with the statistical problem-solver but with the modality of consultancy" [De Greef et al, 1985: 392].

The authors of the paper have enumerated their paragraphs in order to classify the KADS tasks. The high frequency of occurrence of key concepts to the KADS methodology such as "knowledge acquisition", "methodology", "implementation" have facilitated the tasks of identifying key issues in the paper.

While these lexical markers were crucial to the comprehension of the paper by participants 1,2,3 and 4, only the graphical markers were acknowledged by participant 5. The key issues mentioned in the paper were known to him. Consequently he was less careful in interpreting the author's linguistic signalling. His reading strategy focussed on using the title as a carrier of the thematic information, then the names of authors and their references as a guidance to the importance of the paper within the domain.

The reading strategy applied by the participants is as follows:. Participant 5 scanned first the introduction and conclusion, then "skimmed" the remaining parts of the paper and concentrated on the diagram produced by the author of the paper. The other participants, however, read the introduction and conclusion intensively, then scanned the remaining sections and ignored the diagram. Participant 1 is the only one who had to read the paper entirely before producing the abstract.

3.4.2.1.4 Results

This experiment was intended to explore the level of background knowledge needed to be able to generate an adequate abstract. The following results, although not conclusive, suggest a number of points, namely:

(1) The first point is concerned with the informative value of these five abstracts. Participant 5, as well as those with no background knowledge in the topic of the paper under study, provided sufficient information to aid other researchers to evaluate the

relevance of this paper. This suggests that possession of a specialised knowledge of the domain is not necessary to produce an adequate abstract. Having an understanding of the field, however, enabled abstractors to include fewer details in their abstract and thus produce a shorter but still adequate abstract.

(2) Participants are able to utilise both the title and the surface structure of the paper to locate salient chunks of information. These seem to contain sufficient clues for the participants to perceive accurately the organisation of the paper as well as its content.

(3) The linguistic discourse markers used by the authors of the paper, whether in signalling major or minor ideas, are perceived as an important aid to understanding the content of the paper. This is particularly evident in the case of participants with no background knowledge of the paper being analysed.

(4) Lexical and positional markers seem to be the most important means for determining the relevance of informational constituents. Participants demonstrated the ability to recognise a range of different stimuli used by the author to highlight salient information. This information was reproduced in their abstracts.

(6) The main ideas can be derived with only shallow semantic knowledge as shown in the abstracts produced by subjects 1, 2, 3 and 4.

The results of the previous experiment has led to the following list of strategies to be employed when establishing the significance of information contained in a given paper:

(1) Note the title. This sets the stage by identifying the theme of the paper from the author's perspective.

(2) Pay special attention to the textual relevance cues to identify statements relevant to the content of abstract.

(3) Consider the name of the author(s) and the source of publication. This will indicate the relevance of the paper within that field.

(4) Note the surface structure of the paper. This will form a useful guide to scanning sections of the paper.

(5) Read the introductory and final sections carefully. These usually provide the most significant leads for the important issues discussed in the paper.

The above results tie in closely with the findings of Kieras' experiments. His experiments, which are conducted in the field of experimental psychology, are concerned with the thematic processes in the comprehension of technical prose and prose memory phenomena. In his experiments, Kieras asked his subjects to read short technical passages and them compose a brief statement to describe the main idea of these passages. The results suggested that .."in picking and producing topical or thematic information...people can make use of the semantic context, even though they do not understand the material deeply at all...Only "shallow semantics" might suffice for a great deal of macrostructuring processing..." [Kieras, 1982: 80].

These strategies will be taken into account when designing a computer program to simulate the process of abstracting.

3.4.2.2 Experiment 2: Minimum Discourse Required for Assigning Relevance and Producing an Abstract

The aim of this experiment was to investigate whether an adequate abstract can be produced from the introduction and conclusion of a given paper. This was undertaken to establish the minimum discourse necessary to extract relevant ideas for inclusion in an abstract. It was hoped that the results of this experiment would be used when designing a computer simulation.

3.4.2.2.1 Overview of the Experiment

Five researchers in Computer Science from three different academic institutions were selected to participate in this experiment. Their research interests covered four aspects of computer science; a description of their general topic as well as their specific topic is given in figure 4.9.

3.4.2.2.2 Procedure

The experiment consisted of four stages. A total of 41 papers reflecting the research interests of the participants were selected.

In the first stage, each of the participants was given the titles of the papers relevant to their study whereas in the second stage, the author(s) and the source of publication of these papers were revealed. Out of the 41 papers the most relevant 15 papers were selected for further study. The introduction and conclusion of these papers were then supplied in stage

3 and participants were asked to produce an abstract for each of the papers based on these sections. In the final stage, the participants were asked to read the whole paper and to generate a second abstract.

At each stage, the participants were asked to assess the relevance of each paper using the following scoring system:

i) Score 1 if the paper is extremely relevant (i.e. the paper discusses issues of specific interest to the participant).
ii) Score 2 if the paper is very relevant in the sense that it addresses related issues.
iii) Score 3 if the paper is of little relevance, i.e. the paper discusses general research issues.
iv) Score 4 if the participant cannot assess the relevance from the given information.
v) Score 5 if the paper is not relevant.

No constraints were imposed regarding the type and length of the abstract to produce. Details of the questionnaires sent to the five researchers are given in appendix B.

3.4.2.2.3 Results

The major issues addressed in this experiment are the changes in the scores as each stage supplied more information to the participants. This is why the rank values are analysed by stages followed by an overall assessment. As the emphasis of this experiment is concerned with how the participants determine relevance, only those papers that were identified by the participants as most relevant were selected for further study. The results of this experiment are reported in table 4.10 and classified into three major categories as shown in table 4.11.

Participant	General Topic	Specific Topic
1	Natural language processing	Parsing (deterministic, chart, bottom-up) and machine translating, Prolog and parsing.
2	Artificial intelligence	Knowledge representation and classification, inference engine, software engineering and artificial intelligence.
3	Artificial intelligence	Knowledge representation and elicitation, production systems. expert systems.
4	Database systems	Semantic modelling, database design, intelligent database systems.
5	Computer graphics	Graphics Kernal Systems, inter-modular communications.

Figure 4.9 Research interests of participants in experiment 2.

In the first stage, 15 papers were declared relevant (i.e. they scored 1,2 or 3), 17 papers not relevant and 9 others were difficult to assess for relevance on the basis of the title only. Some titles contain a general description of a research project as in the following titles:

- The professional Workstation Research Project;
- A Representation for Complex Physical Domain;
- Deterministic and Bottom-Up Parsing in Prolog;
- Issues in the Design of Database Programming Languages.

Others give specific description of the research as in:

- A Case Study in Structured Knowledge Aquisition;
- Representing Procedural knowledge in Expert Systems: An Application to Process Control;
- A Perspective for Research on Conceptual Modelling.

Titles containing general concepts scored, by and large, 3 or 4 whereas titles with specific concepts scored 1,2 or 5.

When the author and the source of publication for each of the papers were revealed in the second stage, twelve out of the 15 papers that were declared relevant, remained relevant, two papers joined the "don't know" category and one paper became "not relevant". Three papers out of 9 papers in the "don't know" category joined the relevant category, and 2 papers were evaluated as not relevant, and 4 others remained unknown. It is also important to note that the 17 papers declared not relevant in the first stage remained not relevant in the second stage. To a large extent, the author and the year of publication influenced the participants ranking procedure rather than the source.

As explained earlier, only the most relevant 15 papers that were identified in the second stage as important (i.e. papers with score ≤ 3), were carried over into stages 3 and 4. The results of these two stages are very interesting. By stage 3, participants were able judge whether a paper is relevant on the basis of the introduction and conclusion. Those papers identified relevant in stage 2 remained relevant in stage 3. The three papers which could not be assessed are now declared relevant.

When the whole paper was supplied to participants in the final stage, only one researcher found that two of the papers identified relevant, did not contain relevant information as predicted in previous stages. This participant argued that the techniques described in the papers were not applicable to the work undertaken by him. In his case, relevance was narrowly defined to the notion of applicability. Out of the 13 remaining papers, seven papers maintained their rank value from stage 3 despite additional information. Five others moved relevance by one degree and two papers went from extremely relevant to not relevant.

In summary, 8 of the 15 papers (53%) identified relevant at stage 1 were still relevant at the end of stage 4. There were 5 papers unassessed in stage 1, three of them were declared relevant at stage 2 and remained relevant in stage 4, and two others became relevant at stage 3 and also remained relevant at stage 4. Finally, 7 out of the 15 papers (47%) were identically assessed in the last two stages despite additional information being provided. These movements are summarised in table 4.12.

The conclusion is that one can infer with a good degree of confidence from the title, introduction and conclusion whether a paper is relevant or not. What is difficult to measure is the extent of relevance since very little details are generally given in these sections. If one is successfully able to evaluate relevance from these sections, can one produce a coherent and informative abstract from these sections? This is the question addressed by the third and fourth stages of this experiment.

Participant		Scores for the papers									
1*	Stages	1	2	3	4	5	6	7	8	9	10
	1	4	1	2	5	5	5	3	5	4	5
	2	3	1	1	5	5	5	5	5	5	5
	3	3	2	1							
	4	3	2	1							
2*	Stages	1	2	3	4	5	6	7	8	9	10
	1	5	5	2	4	2	5	4	5	4	5
	2	5	5	1	4	3	5	5	5	4	5
	3			3	3	1					
	4			5	3	5					
3	Stages	1	2	3	4	5	6	7	8	9	10
	1	1	5	2	1	3	5	2	5	5	5
	2	1	5	2	2	3	5	2	5	5	5
	3	2			1			3			
	4	3			2			1			
4	Stages	1	2	3	4	5	6	7			
	1	5	4	4	5	4	2	2			
	2	5	3	3	5	4	4	4			
	3		2	2		3	2				
	4		2	1		3	1				
5	Stages	1	2	3	4						
	1	4	3	2	2						
	2	4	3	2	2						
	3			3	3						
	4			2	3						

Table 4.10 Experiment 2 : results of the questionnaires

*These participants shared the same set of papers.

Stages	Relevant (Values 1,2,3)	"Don't know" (Value 4)	Not relevant (Value 5)	Total No of papers
1	15	9	17	41
2	15	6	20	41
3	15	0	0	15
4	13	0	2	15

Table 4.11 Experiment 2 : major categories

3.4.2.2.4 Analysis of Abstracts

In addition to evaluating the relevance of the papers, the participants were asked to produce an abstract for each of the most relevant papers at the end of stage 3 and of stage 4. These were analysed and compared against abstracts produced by the authors of those papers when available. Out of the total of 30 abstracts, two sets of abstracts: P1.A and P1.B, and P2.A and P2.B were identical. Also one paper was identified to be relevant to two participants, hence the abstract for this paper is accounted once only (G4.C≡H4.C). All these abstracts are reproduced in appendix C. For the sake of simplicity, abstracts generated by participants at stage 3 are denoted as A-type, those produced at stage 4 denoted as B-type. Authors' abstracts are referred to as C-type.

Due to the nature of these papers, the content of these abstracts is grouped under the following four major elements:

i) background information/problem definition;
ii) purpose;
iii) methodology/technique;
iv) results/conclusion.

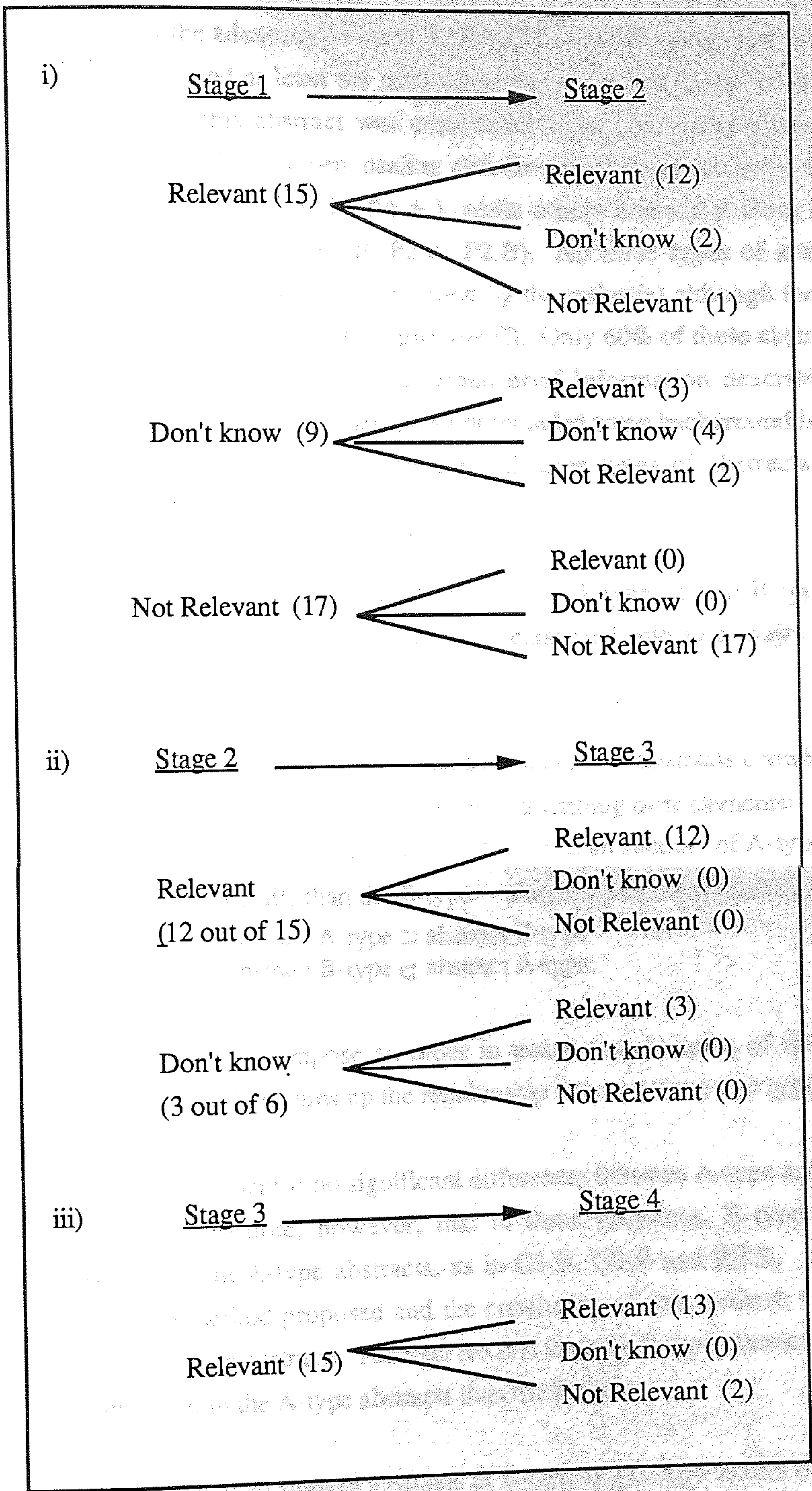


Table 4.12 Movements within Experiment 2.

To specify the adequacy of these 30 abstracts, the following criteria were applied. If an abstract contained at least the purpose of the paper and the technique proposed by the author(s) then this abstract was considered as an acceptable abstract. Each of these abstracts included a statement dealing with the aim of the paper; some abstracts described it explicitly (e.g. H1.A, H7.B, R6.A.), while others inferred it from the structure of the paper (e.g. H4.B, P1.A, P1.B, P2.A, P2.B). All three types of abstracts specified the methodology or the technique proposed by the author(s) although the method was stated implicitly in abstract G3.C (see appendix C). Only 60% of these abstracts reported results or conclusions whereas 70% contained brief information describing the problem as addressed by the authors in their paper or included some background information to situate the paper in the field. In conclusion, all three types of abstracts (A,B and C) were considered acceptable abstracts.

The second step was to compare abstracts of A-type against B-type. The relationship between the two types of abstracts was classified into two major mutually exclusive relations:

i) equivalence (denoted by $\equiv$), i.e. both abstracts contain similar informational constituents describing their elements;

ii) inclusion (denoted by $\supset$), i.e. if an abstract of A-type includes more details than the B-type abstracts, this is expressed as:

abstract A-type $\supset$ abstract B-type

or abstract B-type $\subset$ abstract A-type.

These levels do not impose an order in which the elements of the abstracts are to be arranged. Table 4.14 sums up the relationship between these two types of abstracts.

In most cases, there is no significant differences between A-type and B-type abstracts. It is interesting to note, however, that in three instances, B-type abstracts were less informative than A-type abstracts, as in G1.B, G2.B and R3.B. The A-type abstracts outlined the method proposed and the conclusion of this method; such information was omitted in B-type abstracts. Abstract R6.B is the only B-type abstract that contained details not mentioned in the A-type abstracts than the B-types.

The next step is to analyse abstracts of B-type and C-type to find out how their elements mapped into original documents. A-type abstracts are not included in this analysis because they were generated from the introductions and conclusions only of the papers. Table 4.14 sums up the findings.

Abstract	Background Information/ Problem Definition	Purpose	Methodology	Conclusion
P1.A	x	x	x	x
P1.B	x	x	x	x
P1.C	x	x	x	x
P2.A		x	x	x
P2.B		x	x	
P2.C	x	x	x	x
P3.A	x	x	x	
P3.B		x	x	
P3.C		x	x	x
G3.A	x	x	x	x
G3.B	x	x	x	x
G3.C	x	x	implicit	
G4.A	x	x	x	
G4.B	x	x	x	
G4.C	(see H4.C)			
G5.A		x	x	x
G5.B		x	x	
G5.C	x	x	x	x
H1.A	x	x	x	x
H1.B	x	x	x	x
H1.C	x	x	x	
H4.A	x	x	x	x
H4.B	x	x	x	x
H4.C	x	x	x	
H7.A	x	x	x	x
H7.B	x	x	x	x
H7.C	x	(x) implicit	x	
R3.A		x	x	x
R3.B		x	x	
R3.C		(not available)		
R6.A		x	x	x
R6.B	x	x	x	
R6.C	x	x	x	

Table 4.13 Content of the abstracts (type A, B and C).

One might expect that elements of B-type and C-type abstracts are extracted from sections other than the introduction and conclusion. This is not the case. When the elements of the abstracts were mapped into the original documents, informational constituents relating to the purpose of the paper or research were largely located in the introduction. This accounts for 9 abstracts out of the 20 B-type and C-type abstracts (55%). Only 25% of the abstracts derived the purpose from the conclusion. The remaining 20% of abstracts made use of the middle sections to report the purpose. The introduction provided most of the constituents describing the method, background information and problem definition. In 55% of the abstracts the method can be mapped into the introduction, 25% to the conclusion, and the remaining 20% to the title and middle sections. As one expects, 80% of the abstracts began with an introduction giving some background information and explaining the problem. Finally, when results and/or conclusions were provided by the 10 abstracts, they were extracted mainly from the last section of the paper usually labelled conclusions. This accounts for 5 abstracts. The three other abstracts obtained the result from the section called introduction and the last two from the middle sections of the paper. Another 25% referred to the middle sections to extract the results while the remaining 12% derived authors' conclusions from the last paragraph of the introduction.

In summary, many of the statements related to purpose, method and problem definition were extracted from the introductory section of the paper whereas the results were taken from the conclusion. This experiment suggests that the introduction and conclusion could provide a a solid basis for producing a coherent and informative abstract.

Type A abstracts	Relation	Type B abstracts
P1.A	$\equiv$	P1.B
P2.A	$\equiv$	P2.B
P3.A	$\equiv$	P3.B
G3.A	$\supset$	G3.B
G4.A	$\supset$	G4.B
G5.A	$\supset$	G5.B
H1.A	$\equiv$	H1.B
H4.A	$\equiv$	H4.B
H7.A	$\subset$	H7.B
R3.A	$\supset$	R3.B
R6.A	$\subset$	R6.B

Table 4.14 Abstracts A-type vs abstracts B-type.

Abstract	Introduction	Conclusion	Title	OtherSections
P1.B	method	purpose background info. conclusion		
P1.C	problem definition	method results purpose		
P2.B	purpose method			
P2.C	problem definition purpose			results (section 5) details of the method (sections 2 & 4)
P3.B	purpose method			
P3.C	background info. purpose method conclusion			
G3.B	Problem definition	methodology conclusion		purpose (headings of sections)
G3.C	problem definition	method (implicitly)		purpose (headings of sections)
G4.B	background info			method (section 4) purpose(section 4)
G4.C	background info	purpose	method	details of method (subheadings of section 4)
G5.B	problem definition purpose method			
G5.C	purpose method results			
H1.B	method purpose problem definition	conclusion		
H1.C	problem definition purpose method			
H4.B			method	problem definition and background info (sections 2 & 3) results (section 3) purpose (section 2) conclusion(section 3)
H7.B	problem definition method, purpose	conclusion		
H7.C	problem definition method, purpose results			
R3.B	purpose method			
R6.B		method purpose conclusion		background info (section 2)
R6.C	problem definition	purpose method		

Table 4.15 Mapping of Abstracts B and C to original document.

3.4.2.3 Experiment 3 : The Short-cut Strategy

At the interviews, abstractors suggested that it is worthwhile to look at the introduction and conclusion or summary prior to reading the whole paper because an adequate abstract can be produced from these sections alone. This approach is known as the short-cut strategy among abstractors. The next immediate question that follow is whether these two sections suffice to extract information describing the purpose, method and results or conclusion of a given paper. To this end, experiment 3 was set out to investigate whether this approach could provide the basic rules for extracting information.

3.4.2.3.1 The Data

The corpus consists of ten papers selected from the following fields (table 4.16):

i) two papers from computational linguistics, referred to in this section as CL1 and CL2;
ii) three papers from polymer engineering, referred to as PE1, PE2 and PE3;
iii) three papers from computer science, referred to as CS1, CS2 and CS3;
iv) two papers from information science, referred to as IS1 and IS2.

The selected papers had three pre-conditions. First, they had to be expository, i.e they shared a common purpose of communicating a technique, an algorithm or an approach. Second, they started with an introduction, and they ended with a conclusion or summary. The references of these papers are given in appendix D.

3.4.2.3.2 Analysis

A total of 64 paragraphs were analysed, searching for the three elements that make up an abstract. The results of the analysis are compiled in table 4.17, which shows the purpose, method and results or conclusion for each of the ten papers and shows from where these elements were extracted. As one might expect, the conclusions reached by authors about their work are usually mentioned in the conclusion. However, the author of CS1 included the result of his technique in the introduction as well. In all 10 papers, the purpose of the paper or the work undertaken is explained in the introduction, and announced by means of lexical markers such as:

1. This paper constitutes ... (CL1).
2. This paper introduces ... (CL2).

This paper has introduced ... (CL2).
3. This work is a further step ... (PE1).
4. The following discussions attempt to present .. (PE2).
5. This paper describes ... (PE3, CS1).
6. In our approach we stress attention to ... (CS2).
7. This paper presents ... (CS3).
This paper has presented ... (IS2).
8. In particular, this paper will show ... (IS1).
9. In this paper, we have overviewed ... (IS1).
10. Our goal is to ... (IS2).

In the two papers PE1 and CL1, however, in order to infer the type of method suggested or the algorithm introduced, it is necessary to read the sentence before the one defining the purpose.. In these papers, the authors are primarily concerned with setting up the problem and establishing the area of their work as a way of introducing the purpose of their paper. A further reading of these papers was carried out to investigate if the purpose of their work is explained in the body of the papers; there was no mention of it there either.

The method is introduced first in the introduction but sometimes included in the conclusion as well, as in the papers PE2, CS2 and IS1. There are 16 occurrences of the method in the sample. In 7 out of the 16 occurrences the author again uses lexical markers to announce the method. Example of these markers are:

1. ... we present an algorithm that .... (CL1).
2. This algorithm can be viewed as ... (CL2).
3. The work ... is on the basis of the model ...(PE1).
4. In this paper we have attempted to describe the necessary steps... (PE2).
5. In this paper the ... test ...(PE3).
6. Its main feature is ... (CS3).
7. The concept of ... was utilised... (IS1).
8. We introduce the concept of ... (IS1).

Sometimes the technique is stated with the purpose as in the papers PE2, CS1 and CS3.

It is interesting to note that the introduction and conclusions of these papers share a common structure although they belong to different disciplines, different types of research and their authors cover a wide range of countries, namely Japan, Italy, France, Germany, USA and Canada.

Paper	Introduction			Conclusion			Total (Introduction & conclusion)		
	No of paragraphs	No. of sentences	No. of words	No. of paragraphs	No. of sentences	No. of words	No. of paragraphs	No.of sentences	No. of words
CL1	2	8	188	1	4	155	3	12	343
CL2	2	12	150	5	12	89	7	24	239
PE1	4	8	96	3	4	94	7	12	190
PE2	4	8	537	1	6	70	5	14	607
PE3	4	14	153	3	11	149	7	25	302
CS1	3	18	320	2	6	135	5	24	455
CS2	2	12	103	3	6	123	5	18	226
CS3	8	14	390	1	3	51	9	17	441
IS1	4	13	326	2	6	165	6	19	491
IS2	8	35	755	2	6	140	10	41	895
TOTAL	41	142	3018	23	64	1171	64	206	4189

Table 4.16 Data for experiment 3

3.4.2.3.3 Results

It is not surprising to find elements such as the conclusion in the conclusion or results sections, and the purpose in the introduction of a given paper. What is interesting is to find out that it is possible to get enough information from these sections to make up an abstract. This was suggested by some abstractors and this small experiment indicates that their heuristic approach is a viable strategy.

	Abstract Elements		
Paper	Purpose	Method	Results/Conclusions
CL 1	It should be a practice to publish the algorithms in isolation independent of a particular implementation or system (introduction : paragraph 1) This paper constitutes an effort to initiate such a practice (introduction : paragraph 2).	In response to this problem, we present an algorithm that generates quantifier scopings for English sentences (introduction : paragraph 2).	While this algorithm can sometimes result in a significant saving over the naive approach, it by no means solves the entire quantifier scoping problem,as we have pointed out. (conclusion : paragraph 1).
CL 2	This paper introduces an efficient on-line algorithm, and focuses on its practical application to natural language interfaces (introduction : paragraph 1). This paper has introduced an efficient context-free parsing algorithm, and its application to on-line natural language interfaces has been discussed (conclusion : paragraph 1).	This algorithm can be viewed as a generalised LR parsing algorithm that can handle arbitrary context-free grammars, including ambiguous grammars (introduction: paragraph 2).	It can parse unsegmented sentences...The parser takes about 1-3 seconds CPU time per sentence on a symbolics 3600 with about 800 grammar rules; its response time...is less than a second due to on-line parsing (conclusion : paragraph 3).
PE1	(implicit purpose) Only recently...some effort has been made in the direction of providing...normalised diagrams and computational methods... (introduction : paragraph 3). This work is a further step in this direction (introduction : paragraph 4).	This work is...on the basis of the model suggested by Lord and Williams a normalised injection pressure valve is obtained in the limit of small viscous generation (introduction : paragraph 4). The Lord and Williams model...which refers to an amorphous material filling a rectangular geometry mold cavity has been considered (conclusion : paragraph 1).	It was found that, within this limit, the bulk dimensionless temperature of the flow front does not depend significantly upon them. Also a proper dimensionless inlet pressure accounting of the main effects of the parameters was identified (conclusion : paragraph 1). The results, summarised in Figures 2 and 4, allow the immediate evaluation of the inlet pressure, a comparison with literature (19) data taken on polystyrene (amorphous) is shown in Figure 5. (conclusion : paragraph 2).
PE 2	The following discussion attempts to present some of the key concepts of a model of the complete thermoset injection molding cycle, as they relate to the behaviour of the thermo-setting resin in the cavity (introduction : paragraph 4). In this paper we have attempted to describe the necessary steps required for developing a realistic simulation of the injection molding process for thermosets (conclusion : paragraph 1).	The treatment is an extension of the treatment already developed for thermoplastics, and it emphasises a thin, rectangular mold cavity. (introduction : paragraph 4). In this paper we have attempted to describe the necessary steps required for developing a realistic simulation of the injection molding process for thermosets (conclusion : paragraph 1).	We have shown that numerical simulation of the molding process leads to theoretical predictions which are in reasonable agreement with experimental data (conclusion : paragraph 1).

Table 4.17 Abstract elements and their origins

	Abstract Elements		
Paper	Purpose	Method	Results/Conclusions
PE 3	This paper describes the testing method and the influence of factors such as temperature, pressure, the degree of condensation, and moisture content, on the flowability of phenobolic, melamine, and urea molding compounds (introduction: paragraph 4).	In this paper, the conventional orifice flow test was modified by combining with high frequency (HF) preheating for obtaining the real relation between flowability and temperature under isothermal conditions (introduction: paragraph 4).	A monohole flow test...gave much information on the flow behaviour of thermosetting molding compounds. There was an inflection point which was considered to correspond to a phase transition from solid-like to liquid-like flow.. All compounds greatly increased their flowability... with ureas...the temperature dependence of the flowability decreased. (Conclusion: paragraph 1).
CS1	This paper describes a technique for conflict resolution...whose goal is to capture the reasoning process behind this form of conflict resolution. (introduction: paragraph 3).	Characteristic error conflict resolution integrates two closely related concepts: reasoning about uncertainty and constraint propogation (introduction: paragraph 3).	The characteristic approach is unique in treating each individual knowledge source as a separate entity whose overall reasoning validity may be entirely self-determined. (introduction: paragraph 3). This paper presents a unique means of combining the opinions of divergent experts in a uniform manner using only their own a priori knowledge about their own characteristics (conclusion: paragraph 2).
CS2	In our approach we stress attention to the system theoretical principles of knowledge modelling and on the representation of geometric knowledge. We give an introduction to the knowledge representation language COMODEL (introduction: paragraph 2).	Its main feature is the component oriented description of technical systems and processes covering geometrical and process knowledge by the same means of description (introduction: paragraph 2). COMMODEL is a component oriented approach to the representation of knowledge about technical systems. (Conclusion: paragraph 1).	...use COMMODEL as a technical description language for expert systems, in particular for interpretation of the states of those systems and diagnosis of failures (Conclusions: paragraph 3).
CS3	This paper presents a semi-procedural approach to programming... (introduction: paragraph 6).	An interesting feature of the system is that the degree of procedurality ofa source program can vary between two extremes... (introduction: paragraph 6). The transformations which are used differ from those normally used in the following two respects.... (introduction: paragraph 7). The addition of a new extension corresponds to the addition of a new slice to the program (introduction: paragraph 8).	This paper has presented an attempt to introduce a systematic methodology for the construction of [report generation] programs. (conclusion: paragraph 1).

Table 4.17 Abstract elements and their origins (continued)

	Abstract Elements		
Paper	Purpose	Method	Results/Conclusions
IS1	In particular this paper will show how in the area of database systems the need to represent uncertainty in data and its retrieval is leading to a broadening of the traditional database format. (introduction: paragraph 2). In this paper we have overviewed a variety of conventional and fuzzy set approaches... (conclusion: paragraph 1)	This paper deals with the generalization of databases and retrieval systems using fuzzy set mechanisms. We introduce the concept of a generalized information system to represent the several characteristics that contrast these new systems with their conventional counterparts (introduction: paragraph 4). The concept of generalised information system, parameterized by data homegeneity and semantic information required by queries was utilized... (conclusion: paragraph 2).	This approach can aid in the evaluation and comparison of the rapid developments in this area. It should also be able to furnish guidance towards areas in the spectrum of data imprecision... (conclusion: paragraph 2).
IS2	Our goal is to provide the user with the tools that will allow him to carry out his own translations.. (introduction: paragraph 7). This paper has presented an initial version of the co-word search system for document retrieval. The goal of this system is to help translate his interests into an appropriate search strategy (conclusion: paragraph 1).	Statistical coefficients are at the heart of our co-word search aid system... His entry is obtained by means of an initial formulation of his interests, and a menu... (introduction: paragraph 8).	Describing these patterns..... seems to us to be an efficient and suitable way of producing user-friendly aids for document retrieval (conclusion: paragraph 2).

Table 4.17 Abstract elements and their origins (continued)

The structure of introductions to articles has been the focus of study of many linguists and in particular Swales. Although Swales' research was motivated by pedagogical perspectives, his conclusions are very relevant to this research. His findings, based on scientific papers of expository type, suggest that introductions have a four-move structure [Swales, 1981]:

Move 1: to establish the field.
Move 2: to summarise previous research.
Move 3: to prepare for present research.
Move 4: to introduce present research.

It is in the fourth move that the authors state the purpose of their papers.

In addition, authors feel the need to indicate to their readers the main conclusions of the present research. The reason for that, Swales argues, is because if authors have indicated a gap in the previous research, they are expected by the reader to propose their approach of bridging the gap; if they have suggested a method, they are expected to assess its success or limitations. Thus, Swales' research suggest strongly that introductions alone may provide information describing the purpose of the paper, the proposed method and some indications of its success.

There is so far enough evidence to suggest that introductions and conclusions are the minimum required text to produce an abstract. Consequently, the short-cut strategy could provide a strong basis for abstracting. It is also worthwhile to use those lexical markers which are employed by authors to signal important information to the reader.

The short-cut strategy is a useful empirical strategy that will be implemented in the computerised system.

3.5 Task 5: Generate a Coherent Abstract

The previous task involved techniques and strategies used for a rapid review reading of a paper to identify those chunks that contain salient information for the abstract. This task begins by transforming these chunks into a coherent abstract. Here the abstractor organises and edits the extracted chunks into an acceptable style to conform with the standards of his/her publishing house. Generating an abstract involves writing skills aimed at ensuring that the resulting abstract is coherent, cohesive and informative. Cremmins stipulates the following rules for writing an abstract [Cremmins,1982: 57] :

Rule 1: reread all the information selected on purpose, method and results.

Rule 2: while reading, mentally index the primary and secondary themes described using your own choice of arbitrary terms or phrases.

In addition to these rules, abstractors have a list of hints of the "do" and "do not" for writing good abstracts. Here is a small list of these hints.

Do:

1. Be informative but brief.
2. Use short and complete sentences.
3. Tell what was found.
4. Tell why the work is done.
5. Tell how the work is done.
6. Adapt the abstract to your user profile.

Do not:

1. Comment or interpret the paper.
2. Mention future work.
3. Describe details.
4. Include detailed experimental results.
5. Use jargon.

4. Summary

In this chapter the question of how participants and expert abstractors in particular write abstracts, is addressed. The first part of this chapter identified the tasks involved in the abstracting process with a group of abstractors. These tasks are subdivided further into subtasks with the aim of identifying various methods employed by the experts. For each of these tasks a list of strategies is drawn up and a specific skill is assigned to each task. One of the most important finding is that abstractors adapt their reading strategy to its purpose. The common strategy is to infer from the function and organisational structure salient portions of text and then to scan, and then to read a few sentences intensively whilst skipping sections dealing with details.

Prediction is another skill practiced by the experts. This is the facility of guessing what is to come next, making use of textual and contextual cues and experiential learning. Experts focus primarily on the title to derive the theme of the article, the introduction to extract the purpose and method and some selected statements to situate the work or define the problem if required. The conclusion provides the source of information for reporting results and findings. Abstractors are skilled readers who are able to make use of the organisation of the text to minimise the processing time.

Location of dominant and secondary themes in the paper is achieved through identification of discourse markers spotted by rapid eye movements. This ensures that relevant information outside the anticipated portions of text is captured.

Analysis of abstracts in the second part of this chapter supports the experts' strategy in searching the initial and final sentences of first and last paragraphs for important information. Experiment 2 suggests that some sentences occurring in key positions can be viewed as "privileged" sentences. Experiment 3 confirms the validity of the short-cut strategy.

Furthermore, the results of experiment 1 suggest that the depth of domain knowledge of the article abstracted is not crucial. Participants with no knowledge of the field of computer science were capable of writing adequate abstracts. They were less able to distinguish major themes from secondary themes; nonetheless they conveyed the message of the author(s) of the paper. Shallow knowledge of the domain does not seem to affect the quality of the abstracts significantly.

Finally, this chapter supports strongly the value of discourse markers used by authors to signal important information. The results show that the three classes of markers (graphical, positional and lexical) can be effectively used to guide the reader to the key issues discussed in the document. Graphical and positional markers provide the most rapid route to highlighting portions of the document for further study. The lexical markers are used to identify specific information.

The collection of these discourse markers is not related to the topic of the paper. This is significant within the context of these experiments as it suggests that a limited set of strategies is being used by human abstractors and the strategies are defined independently of the knowledge domain.

CHAPTER 5

INFORMEX - AN INFORMATION EXTRACTION SYSTEM

INFORMEX - An Information Extraction System

1. Introduction

The previous Chapter investigated abstracting tasks and established that the structure of a paper and its linguistic characteristics were far more significant than the knowledge of the specific topic being addressed. This Chapter takes these findings as a strong basis for building an automated system for abstracting. The main purpose of this Chapter is to determine whether that subset of knowledge used in abstracting can be expressed in terms of strategies and represented in formal rules and structured into a computerised rule base system.

The first part of this Chapter is concerned with eliciting expert strategies and then finding a suitable formalism to represent these for computer manipulation. Once the formalism is defined, it can be applied to the textual content of a paper. Equally, the text itself, must be formally represented as a prerequisite to the above manipulation. The second part of this Chapter takes both formalisms and integrates them into a rule base system. This system applies the formal rules to the text and produces a subset of that text to be used as a basis for abstracting. In other words, it extracts the important "chunks" of information of a paper required to generate an informative abstract, hence it is called an information extractor system, INFORMEX.

2. Knowledge Representation

The primary reason for wanting to represent knowledge is so that the computer system can address the problem of abstracting by manipulating the representation necessary to process and analyse the textual content of that paper and the heuristic knowledge applied on the textual content of that paper to generate an abstract.

There are five basic approaches to the knowledge representation which have acquired some degree of acceptability in the field of artificial intellligence {Ringland et al, 1983: 5]. These are : logic, semantic networks, frames, scripts and rule base systems. A detailed description of each of these representation as well as their applicability is found in Gewarter [Gewarter, 1984]. Table 5.1 summarises the main aspects of these representations [Gewarter, 1984].

Nature of Knowledge Acquisition	Advantages	Disadvantages
Logic		
. Logical formulae describing the world. . Knowledge base modified by addition or deletions of formulae. . Statements about objects and properties, situations and relations.	. Precise, flexible, modular. . One representation for each fact. . Inference of new facts from known facts using automated theorem proving techniques.	. Difficulty in representing procedural and heuristic knowledge. . KB difficult to manage. . As facts increase, there is a combinatorial explosion in the possibilities of which rules to apply next.
Semantic Networks		
. World described in terms of nodes and arcs. . Network of nodes and arcs. . KB modified by deletion or insertions of objects and associations	. Important associations made explicitly. . Facts inferred from nodes with direct link. . Can establish inheritance properties.	. No standard terminology about meaning. . Strong organisation needed. . Difficulty in representing Boolean relationships.
Frames, scripts		
. Organised collection of knowledge . Stereotyped events	. Facilitates expectation-driven processing. . Knowledge organised in a way that facilitates recall and inference. . Ability to determine its own applicability in a given situation.	. Not all situations are stereotyped. . New situations not easily accommodated.
Rule-based systems		
. Knowledge expressed in terms of rules. . Global data base used to represent system status. . Inference engine used to select rules for execution.	. Modularity of facts and rules. . Information can be easily added, removed or modified. . Facility for representing heuristic knowledge. . Easy to keep track of changes due to actions. . Useful for controlling the interaction between declarative and procedural knowledge.	. Hard to maintain modularity between rules in large systems. . Hard to follow the flow of control in problem solving. . Poor separation of knowledge and control when 1) knowledge is large, 2) dealing with sequential information. . Problems with consistency and completeness.

Table 5.1 Knowledge representation formalism

Logic provides a means of capturing and reasoning about the important properties and relationships of knowledge through the use of a symbolic language with precise meaning and uses. Rules are defined to manipulate these symbols and properties. An example of logical repesentation is given below [Gewarter, 1983: 202]:

The shuttle is in orbit <==> IN(SHUTTLE,ORBIT).

The ability to infer new knowledge from a basic set of facts in an important feature in logic. The following example illustrates this.

All extra-terrestrial bodies have no known life.
Harry is an extra-terrestrial body,
Harry possesses no known life.

$\forall$ x, EXTRA-TERRESTRIAL BODY(x) ==> POSSESSES(x, NO KNOWN LIFE).
EXTRA-TERRESTRIAL BODY(harry) ==> POSSESSES(harry, NO KNOWN LIFE).

The semantic network is another attempt at describing the properties and relationships of knowledge by means of a directed graph with the nodes corresponding to facts or concepts and the arcs to relations between concepts. For example, a semantic network that describes the properties of a specific shuttle operated by Captain Kirk is expressed in figure 5.2 .

Alternatively knowledge can be organised into a more structured representation to describe complex situations. Examples of these representations are frames and scripts. They reason about classes of objects by using stereotypical events and situations. They are particularly useful schemes for describing well-defined scenarios and stories.

Figure 5.3 shows how an airplane may be represented using the frame approach. The airplane frame contains slots describing the types range e.g fighter, transport, light plane, the manufacturer such as Boeing, Mc Donnell Douglas, the number of cockpit crew (1 to 3) etc. Attached to each frame, information is found to describe the use of the frame in normal events as well as the default slots values [Gewarter, 1984: 206].

Scripts are frame-like structures used in natural language understanding systems to organise a knowledge base in terms of the situations that the sytem is to understand. They

Missing page(s) from the bound copy

from the given facts, or new facts to be added to the global database. A rule consists of two components, the left component beginning with IF, stating conditions, and the right components beginning with THEN, stating what actions follows if conditions met. An example of rule is given below:

If sentence-A contains information about the aim of the paper
then sentence-A is important.
fi

The inference engine uses the knowledge base to reason about the problem, given the contents of the global database. It consists basically of an algorithm with the following cycle:

. find which rules have their condition part satisfied in the database,
. select one of them to be "fired",
. perform the action(s) of the selected rule.

The action may take the form of an advice, a question or an assertion of new fact(s).

There are two main methods of putting the rules into operation. The first method examines the left part of each rule in turn and fires the rule wherever the conditions for this part are met. This is called forward chaining. The second method focuses on the right part of the rule. The system postulates a conclusion and then determines whether or not that conclusion is true.

Rule base systems distinguish between knowledge about something and knowledge of how to do something. For example, facts and statements in the form of "this paper is written by F. H. Green" are a declarative form of knowledge whereas knowledge about how to abstract a paper is a typical example of procedural knowledge.

The above knowledge representation formalisms are intended for implementation on a computer machine. Representation of knowledge is very important; representation must however do more than just represent knowledge. The ultimate aim is to represent knowledge as close as possible to the way humans appear to represent it.

There are limitations to each of the knowledge representation schemes described in the previous section. The performance of one scheme against another varies greatly depending on the type of problem being dealt with. While logic is successfully used to analyse formal knowledge and apply deductive and inductive reasoning over this knowledge, it cannot adequately represent world knowledge. Knowledge of the world is complex, full of ambiguities. "A representation which is constrained to truth or falsehood is not the real world" [Ringland et al, 1988: 36]. Although semantic networks were developed as a possible psychological model of human associative memory, they have been used to represent sentences. A large amount of computer storage and processing power would be required if they are used to represent any scientific paper. Frames and scripts have been used to answer queries from stories, consisting of simple sentences and dealing with stereotyped sequence of events. Scientific papers depart considerably from stereotypes and their content could not be handled by the frames or scripts formalism.

Rule base systems have been used successfully for the modelling of human cognitive processes in narrow domains such as medical diagnosis, computer system configuration, mechanical fault diagnosis. The rule formulation often maps well into English [Ringland et al, 1984: 114]. The modularity aspect exhibited by these systems allows rules to be independent of each other and permits knowledge to be encoded incrementally. They are also capable of describing factual knowledge as well as strategies and heuristics, a feature particularly important for the purposes of this study. Thus strategies can be expressed in the form of rules and can be added to the system gradually, allowing the study to assess the effect of additional chunks of knowledge on the abstracting process. For these reasons, the formalism of the rule base system is the most viable approach for the representation of expert abstractors'knowledge, and hence adopted in this study.

The computer formalism chosen must be powerful enough to express the expert knowledge and capable in modelling human approach to abstracting closely. The rule base formalism is regarded as being a psychological model of human knowledge and skills. One is able to ..."model adaptive behaviour by using rules which modify rule memory (seen as akin to human long-term memory) and/or the working memory (seen as akin to human short-term memory)" [Ringland et al, 1988: 105]. The rule base system offers a useful formalism to express expert strategies in abstracting, as it makes it possible for the system to start in stages and build incrementally to a significant testable system. Rules can be inserted, modified and validated through the various stages of refinement progress. These properties are recognised as a key to success in modelling human behaviour in the process of abstracting. Thus the formalism adopted in the study is the rule base system as described in section 2 of this Chapter.

3. Application of Rule Base Systems to the Study of Abstracting

The rule base system aims at embodying not just the abstractor's knowledge but also the way he/she represents, utilises and applies that knowledge when performing abstracting tasks. In contrast to traditional computer systems, where the tasks are well defined and their solutions clearly outlined, the system in this study deals with approaches based on judgements and strategies that may not guarantee results. Furthermore, because it incorporates judgemental and heuristic aspects of human knowledge, the system cannot focus on some aspects, as their significance is not clearly discernible at the outset. Consequently, the building of this system starts by embodying all the knowledge required so that it can behave as an expert.

Building such a system is a mamoth task and requires further empirical and theoretical studies of each of the abstracting tasks and subtasks. This is the long term objective of the research. One of the specific objectives of this study is however to design a rule base system to emulate the strategies adopted by the experts and elicited in the previous Chapter to extract information from a given paper which is for inclusion in the abstract. This rule base system is called an information extractor system, INFORMEX. It does not attempt to generate a coherent abstract, which is the object of task 5 in the abstracting process. Furthermore, INFORMEX has no knowledge of the specific topic of the papers being analysed.

4. Formalisation Stage

The study can now enter the third stage in building a rule base system, referred to as the formalisation stage. The emphasis, here, is to represent the knowledge required to extract salient information for the abstract in the form of rules that a computer can process. Two types of knowledge must be formalised. On one hand, there is the heuristic knowledge which consists of expert abstracting strategies, and on the other, the textual content of a given paper so that the heuristic knowledge can be applied. These two types must be compatible for computer manipulation. The textual content is first described followed by the heuristic knowledge.

To the computer, a file storing the content of a scientific paper is nothing but a collection of strings and symbols. In order for the computer to behave as if it "understands" the content of a document, it must be provided with means of analysing the content. These may include criteria of how to identify word boundaries, sentence boundaries, paragraphs, sections..., as well as segmenting the content into meaningful informational constituents.

There is a number of available methodologies for representing text content, derived from linguistics, psychology and cognitive science. This study adopts the procedural model approach, proposed by de Beaugrande to describe the content of the documents to be analysed. This model establishes the content as a network of relations between elements, called concepts in the "text-world" [de Beaugrande, 1980: 43]. This approach is discussed further in the conceptualisation stage. The emphasis, here, is placed on the linguistic features that writers use to encode the information and readers apply to decode the message. As a result, this research takes a pragmatic approach to the study of text and content representation.

Furthermore, the representation formalism sought to describe the text content must be well defined and consistent. For the purpose of this study, it is felt important that the chosen formalism must be amenable to computer processing. de Beaugrande's relational network is used as the point of departure for representing text-content. This model is described in the following section.

4.1 de Beaugrande Text's Model

de Beaugrande argues that in a procedural approach, all levels of language (syntax, semantics and pragmatics) should be described in terms of producers of text (what he call "intentionality") and receivers (the notion of "acceptability") and the communicative settings ("situationality") [de Beaugrande et al, 1981: 31]. Emphasis is thus placed not on the discovery of units and structural patterns of texts, but on the operations which manipulate these during the utilisation of language. The text becomes "...the actual outcome of these operations. Morphemes and sentences function as operational units and patterns for signalling meanings and purposes during communication". [de Beaugrande et al, 1981: 33]. Thus "a text makes sense because there is a continuity of senses among the knowledge activated by the expressions of text" [de Beaugrande et al, 1981: 84]. de Beaugrande defines this "continuity of senses" as the foundation of the notion of coherence, "being the mutual access and relevance within a configuration of concepts and relations. This configuration of concepts is the textual world..." [de Beaugrande et al, 1981: 84]. A "concept is defined as a configuration of knowledge that can be recovered or activated with more or less consistency and unity...Each concept appears in one or more relations to others, e.g. "state of", "attribute of" and so on. These relations constitute the linkage which delimits the use of each concept" [de Beaugrande et al, 1981: 86].

The textual world is thus described as a network of relations between concepts. For example, the text describing a rocket is given in figure 5.6 could be represented into a network as shown in figure 5.7 On one level there is a syntactic procedure which yields the relations. On the other level there is a conceptual network. The concept "rocket" is the "control centre" for this passage, to which a number of attributes are assigned such as "great", "yellow"..., a specification type "v-2", a state "stood", a location "New Mexico". The control centre may be described as a "topic entity" for this passage.

"A great black and yellow V-2 rocket 46 feet long stood in a New mexico desert, Empty, it weighed five tons. For fuel it carried eight tons of alcohol and liquid oxygen."

[de Beaugrande, 1981 : 98]

Figure 5.6 The rocket

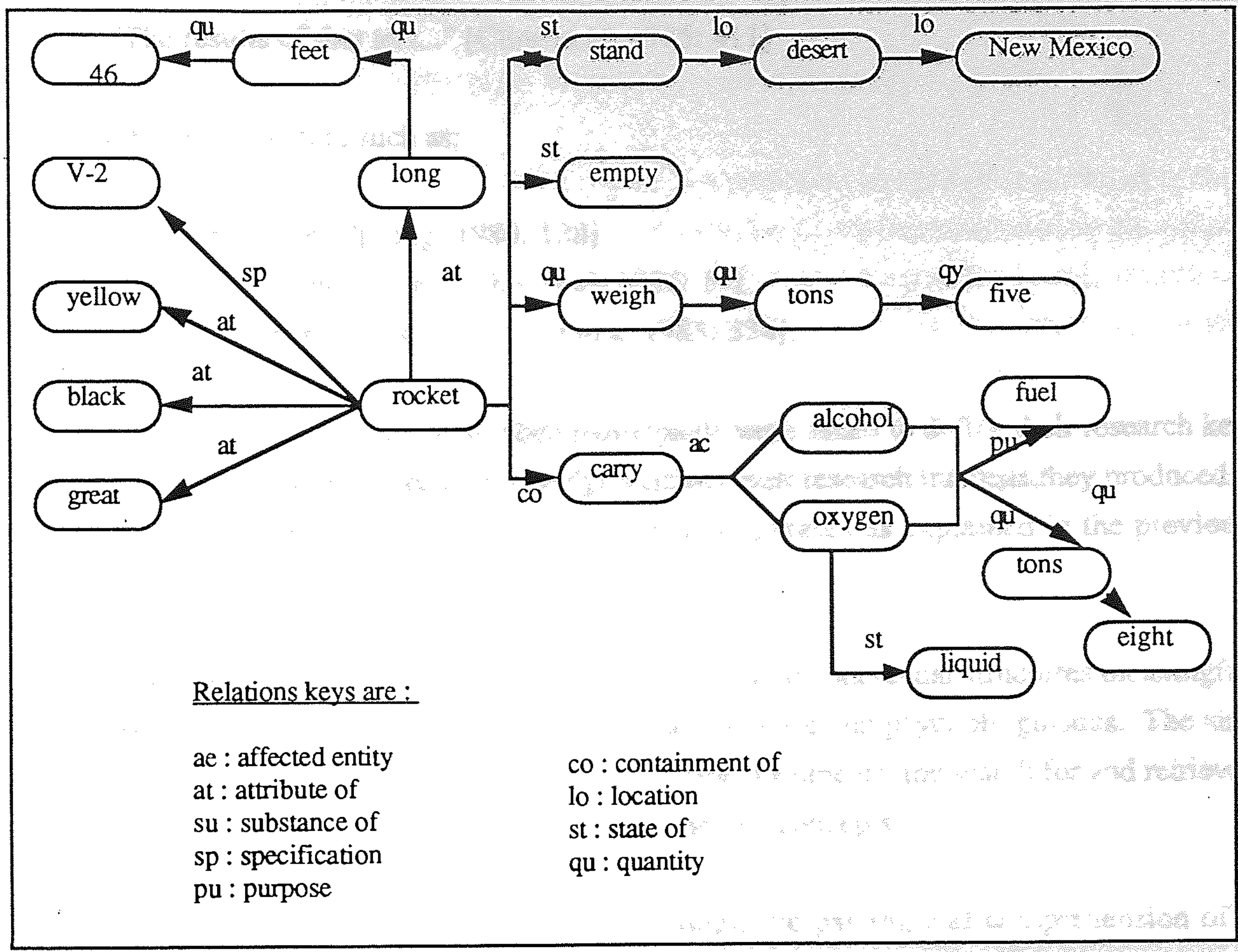


Figure 5.7 Conceptual relational network model

[de Beaugrande et al, 1981 : 100]

de Beaugrande recognises the difficulties related to defining concepts boundaries. However, there are still questions left unresolved in his model. How many units would be needed to constitute a concept? Do concepts have the same units as expressions? Since people communicate by means of expressions, how are units acquired? Concepts boundaries are fuzzy. These questions are not resolved by de Beaugrande's model.

4.2 Heuristic Approach to Text Content Representation

It would be more productive to try defining concepts boundaries as perceived by abstractors when processing texts. At the interviews, abstractors indicated that when they scan a paper, they search for familiar stylistic markers to guide their eye movements. These markers may be contained in the form of a noun phrase such as:

- "The focus of our application..."[Ferrante, 1985: 333]
- "Our approach"...[Dilger et al, 1985: 353]

. "The results of that test..." [Cline et al, 1985: 411]

or in the verb structure such as:

. "To summarise..."[King, 1980: 138]
. "We have shown..."[Bachenko, et al, 1983: 11]
. "We have been discussing..."[Ferrante, 1985: 334].

It is also important to note that when participants were asked to define their research key areas so that the papers selected for analysis reflect their research interests,they produced a set of key concepts expressed in the form of noun phrases as explained in the previous Chapter.

This suggests that scholars tend to organise words into conceptual structures meaningful to others. This is a well recognised phenomena in science and psycholinguistics. The aim of indexing is to generate key concepts that describe documents; the search for and retrieval of these documents in a database is based on these key concepts.

Bower and Cirilo remark, in their study of linguistic parsing and comprehension of a single sentence, that as readers go through a text sentence by sentence, they organise words into surface constituents, extract underlying propositions, set up semantic structures in memory in a cyclically and parallel fashion [Bower et al, 1985]. Some suggest that readers have a "loose-knit" set of heuristic rules to identify these constituents. For instance, Clark and Clark list several heuristics and types of clues used to detect constituents for English syntax [Clark et al, 1977]:

- A function word indicates the beginning of a new constituent;
- Ending of words help decide whether a content word is a noun, verb, adjective or adverb;
- Identify the verb and look for noun phrases that fits its semantic requirement;
- Expect "given" information to precede "new" information unless marked otherwise.

Bower and Cirilo argue that there is psychological evidence for the existence of these heuristics [Bower et al, 1985: 82].

4.2.1 A Conceptual Approach to the Textual Content

The above framework could provide a useful starting point to identify concepts in a text. The text content can be segmented into a configuration of conceptual constituents defined by a collection of strategic rules, some of which are syntactically based, others are semantically derived; some are related to textual relevance cues. To illustrate this, consider the following passage:

"This paper describes a technique for conflict resolution (termed the characteristic error method) whose goal is to capture the reasoning process behind this form of conflict resolution."
[Ferrante, 1985: 331]

Before the conceptual analysis for this passage can proceed, the following strategies are established:

Strategy 1 :	set determiners (the, this, a, an,...) as function words
Strategy 2 :	set prepositions (for, to, on, in, behind...) as function words
Strategy 3 :	set pronouns (who, whose, which...) as function words
Strategy 4 :	set certain symbols (bracket, comma, colon, semicolon...) as function symbols
Strategy 5 :	read a word from the passage

<u>repeat</u>

Strategy 6 :	If a given word is identified as a function word or a function symbol, set the next word at the beginning of a new conceptual constituent

<u>until</u> end of file.

When the above strategies are applied on the passage, the result is the following list of conceptual constituents enclosed in [ ]:

1. This [paper describes].
2. a [technique],
3. for [conflict resolution],
4. [termed]
5. the [characteristic error method],

6. whose [goal is],
7. to [capture],
8. the [reasoning process],
9. behind [ ],
10. this [form],
11. of [conflict resolution].

These constituents could provide valuable information for abstractors who scan a text for useful discourse markers. The first constituent suggests that the author is about to define the purpose of his paper, an aspect very pertinent to abstracting. The second constituent provides the object of the study in this paper: the author is suggesting a technique. The brackets usually indicate that the author is about to include the name, details of this technique, or some references. The constituent "termed" clearly points out to the name of this technique, which is revealed by the following constituent. The information following the pronoun "whose" suggests that the author is enclosing more details which could be deleted or investigated further if desired.

These strategies can be applied to the entire text to identify the conceptual constituents for each sentence. This would yield a conceptual analysis of the text content, called the conceptual text base. Some of these strategies can be expressed in terms of facts (as in strategies 1,2,3,4), others may take the form of rules (as in strategies 5,6).

4.2.2 A Functional Approach to the Textual Content

It is also important to process the textual content as closely related to abstractors' ways as possible. It was found that abstractors are very sensitive to discourse markers used by authors to signal relevant information. They respon d in particular to those markers which provide information regarding the elements of an abstract. Consequently, a functional analysis of each sentence may be carried out, by the computer, determining its relevance for abstracting purposes. Each sentence is scanned for relevant discourse markers and those sentences containing salient information are highlighted in the text, thus yielding a functional analysis model for the text under study. The resulting text knowledge base is denoted as functional text knowledge base.

4.2.3 A Structural Approach to the Textual Content

There is another aspect that must be represented, that is the text structure.. Text cannot be regarded as a collection of related conceptual constituents. These conceptual constituents are well contained in a sentence structure. These sentences are related so as to convey the global message of the text. To the question of how long a stretch of text one actually processes at one time, de Beaugrande supports the view put by psycholinguistics and psychologists [Loftus et al, 1976; Eisenstadt et al, 1975] that "text processing could not afford to run through the participants' vast stores of world knowledge immediately; there must be some ancillary organizational system...In natural language texts, this system is that of syntax... "[de Beaugrande, 1981: 49]. This information describing the syntactic structure must be enclosed in the text representation.

Paragraphing is another aspect that must be considered. It is "...partly a reaction to the organisation of a text, and partly an intentional shaping of it"...The content of a paragraph is intended to be processed as a chunk; and, because the content forms a chunk, the writer is impelled to mark off a paragraph". [de Beaugrande, 1984: 304]. He argues that the most essential strategy in scientific text is to start a new paragraph at a topic shift, when moving from cause to effect. It is also used to mark transition from the opening paragraph(s) to those in the middle and thereby delimiting the main aspects of the text [de Beaugrande, 1984: 309].

Text-content must also reflect the producer's point of view. The organisational structure of the text projects the thoughts and the communication process from the author's perspective. So for each sentence of the text, as well as for the text as a whole, the text knowledge base should include information indicating how each sentence relates to the previous and following sentences but also to the global text. The position of a sentence within each paragraph, and the location of a paragraph within the text has proved to play a significant role in abstracting, as seen in the previous Chapter. Thus, information related to the organisational structure is recognised as important and consequently encoded into the text knowledge base, referred to as the structural text knowledge base. The structural as well as the functional and conceptual text representation yield the global text knowledge base.

4.3 Expert Strategies

The current dominant view of human information processing is that it proceeds in a linear sequential manner throughout a series of stages [Norman et al, 1976]. Although the details of these stages may vary from one information processor to another,..."the general assumption is that processing in task performance proceeds from the perception of cues; the processing of these cues in a short term memory to retrieve action plans from long term memory and then the execution of plans by effective systems responsible for the articulation of sounds or physical movements" [Ringland et al, 1988: 120].

This human information processing paradigm can also be used to describe the strategies adopted by abstractors. One can argue that abstractors process the content of a paper sequentially through the following phases:

i) perceiving textual and contextual relevant cues relevant to the generation of an abstract;
ii) processing these cues to trigger a set of actions; and
iii) executing these actions to produce the abstract.

The scheme proposed to describe the way abstractors process information is the procedural formalism in which strategies are represented in terms of active procedures waiting for patterns relevant to them. Whenever such patterns occur, these procedures trigger and perform a set of actions. These procedures have the following structure:

IF condition-for-trigger ==> THEN do-these-actions.

They consist of rules simulating abstractors' performance.

There are three sets of rules. The first set of rules relates to text processing strategies. It includes a collection of heuristic and linguistic rules describing how to analyse a given text to extract its conceptual, functional and structural properties relevant for abstracting purposes. The second set of rules represents the strategies used by abstractors in interpreting these properties and evaluating the significance of each sentence in abstracting terms. The third set of rules includes knowledge recommending which rules to apply and in which sentences, thus reflecting abstractors' performance and reading skills. These rules are described in detail in the following sections.

5. Implementation Stage

The main concern of this stage is to transform these rules into an executable computer system. Decisions have to be made regarding the specification of control mechanism to operate these rules and details of information flow. This section presents the main characteristics of the rule language and the inference engine.

5.1 INFORMEX, an Information Extractor System

The aim of this system is to apply expert knowledge and expert strategies for extracting relevant information from a given paper which could provide a basis for generating an abstract. For this reason, this system is called an information extractor system, INFORMEX. INFORMEX is a rule base system that consists of three components : a global text knowledge base, a heuristic knowledge base and an inference engine, as illustrated in figure 5.8.

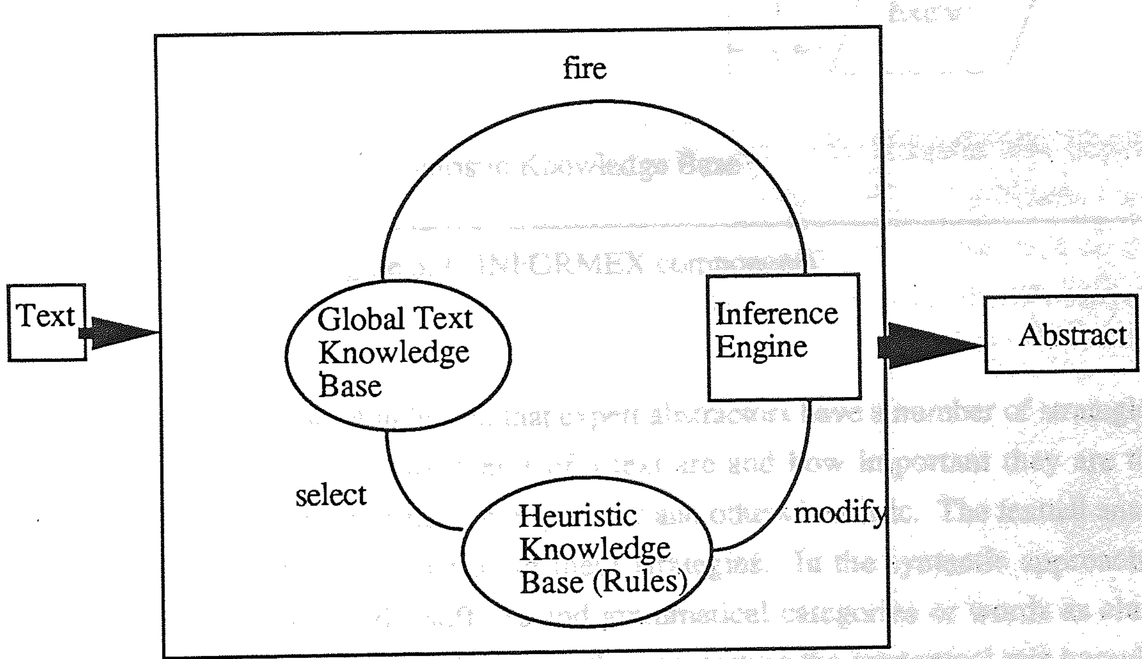


Figure 5.8 INFORMEX - A Rule base system

The global text knowledge base contains information describing the content of the paper being analysed. It consists of three main components: the conceptual model provides the conceptual text knowledge base for the paper, the organisational structure model produces its structural text knowledge base and finally the functional analysis model describes the functional text knowledge base. The global text knowledge base is represented as declarative knowledge.

The heuristic knowledge base includes the strategies elicitated from experts for abstracting purposes. The first module applies the experts' heuristics and linguistics knowledge to analyse a paper and generate the global text knowledge base, mentioned above. The second module examines the content of the global text knowledge base and classifies text sentences by degree of significance. It consists of two modules: the textual analyser and the expert classifier, as illustrated in figure 5.9.

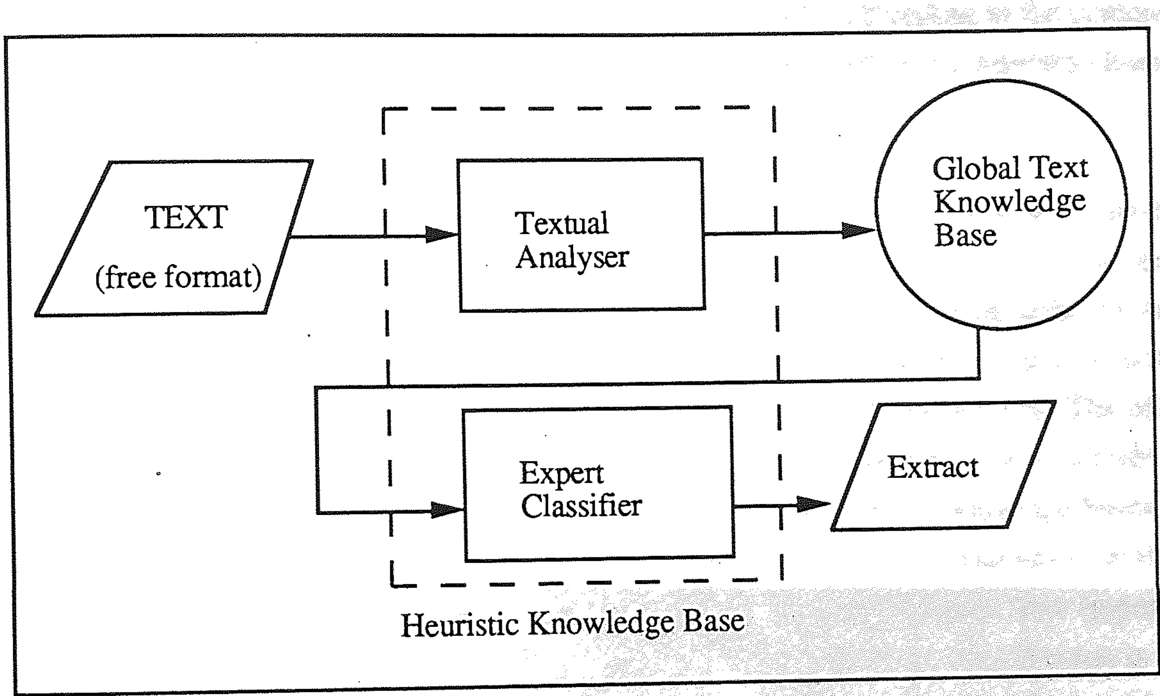


Figure 5.9 INFORMEX components

The findings in Chapter 4 indicated that expert abstractors have a number of strategies by which they infer what the constituents of a text are and how important they are to the abstract. Some of these strategies were syntactic and others heuristic. The textual analyser relies on some flexible combination of these strategies. In the syntactic approach, the strategies use function words, suffixes and grammatical categories or words as clues to identify conceptual constituents of the text, thus generating the conceptual text knowledge base. In the heuristic approach, the strategies make use of specialised linguistic knowledge to search for relevant discourse markers as well as reading skills to locate salient sections of a text, thus producing the functional as well as the structural text knowledge base.

In classifying sentence significance, the expert classifier, like the professional abstractor, does not "read" the paper in a linear fashion. The reading task is guided by a set of rules which is based partly on the physical layout of the paper and partly on the short cut strategy. These rules guide INFORMEX in deciding what rules to invoke and in what order of sequence they should be activated. These rules are referred to as metarules.

The inference engine determines what information is needed to activate the rule at hand, to get this information from the global text knowledge base, to employ the content of this knowledge base to draw inferences and record these inferences in the knowledge base. It consists mainly of an algorithm for pattern matching that examines the premise of a rule (i.e. the if part of a rule) to find whether or not this rule holds, according to the content of the text knowledge base. The process followed in INFORMEX is the forward chaining reasoning.

The overall structure of INFORMEX can now be described. A text is first scanned by the Kurzweil document scanner and stored in free format. It is then edited by a program called the textual editor which checks using rules for inconsistencies regarding the layout of the text and removes ambiguities related to the use of some punctuation symbols such as the full stop and the ampersand &. A list of these rules is given in figure 5.10 The edited text is then ready for processing by the textual analyser to produce the conceptual, structural and functional text knowledge bases. These three text knowledge bases are grouped together to build the global text knowledge base. The expert classifier , assisted by the metarules examines the global text knowledge base and evaluates how important each sentence of a text is with respect to abstracting. The results of its classification is then recorded into the global knowledge base. Those sentences which are identified as highly important will be selected for inclusion in the abstract. The functions of each of the two modules in INFORMEX are outlined below.

5.2 The Textual Analyser Module

The textual analyser is divided into four submodules (figure 5.11):

a) the syntactic analyser program isolates sentences and words in the text;
b) the conceptual analyser program examines each word to generate the conceptual model of this text at a sentence level;
c) the structural organiser program records the layout of the text and stores it in the structural text knowledge base;
d) the functional analyser program scans each sentence for discourse markers relevant to the content elements of an abstract, and records its findings into the functional text knowledge base.

The best way to illustrate the functions of the textual analyser is through an example. Consider the following extract:

"1. Introduction.

Conflict resolution is the integration of alternate opinions about the state of the world into a single coherent view. This approach is often performed by a separate entity (knowledge source) specifically designed for this purpose." [Ferrante, 1985: 331].

```
set     SPAT       (etc/i.e/i.e/i.e/fig./Fig./e.g/e.g./eg./cf./al.)
set     SSYMP      (,/;/:/"/(/)/[/]/*/')
repeat
        read LINE
        rule E1 :  if LINE contains pattern in SPAT
                   then remove dot(s)
                   fi
        rule E2:   if (digit found) and (next char is dot) and
                   (next char is digit)
                   then replace dot by apostrophe
                   fi
        rule E3:   if LINE contains pattern on SSYMB
                   then replace pattern by ' pmarks '
        rule E4:   if LINE contains pattern '&'
                   then replace pattern by 'and'
                   fi
until no more lines.
```

Figure 5.10. Editing Rules

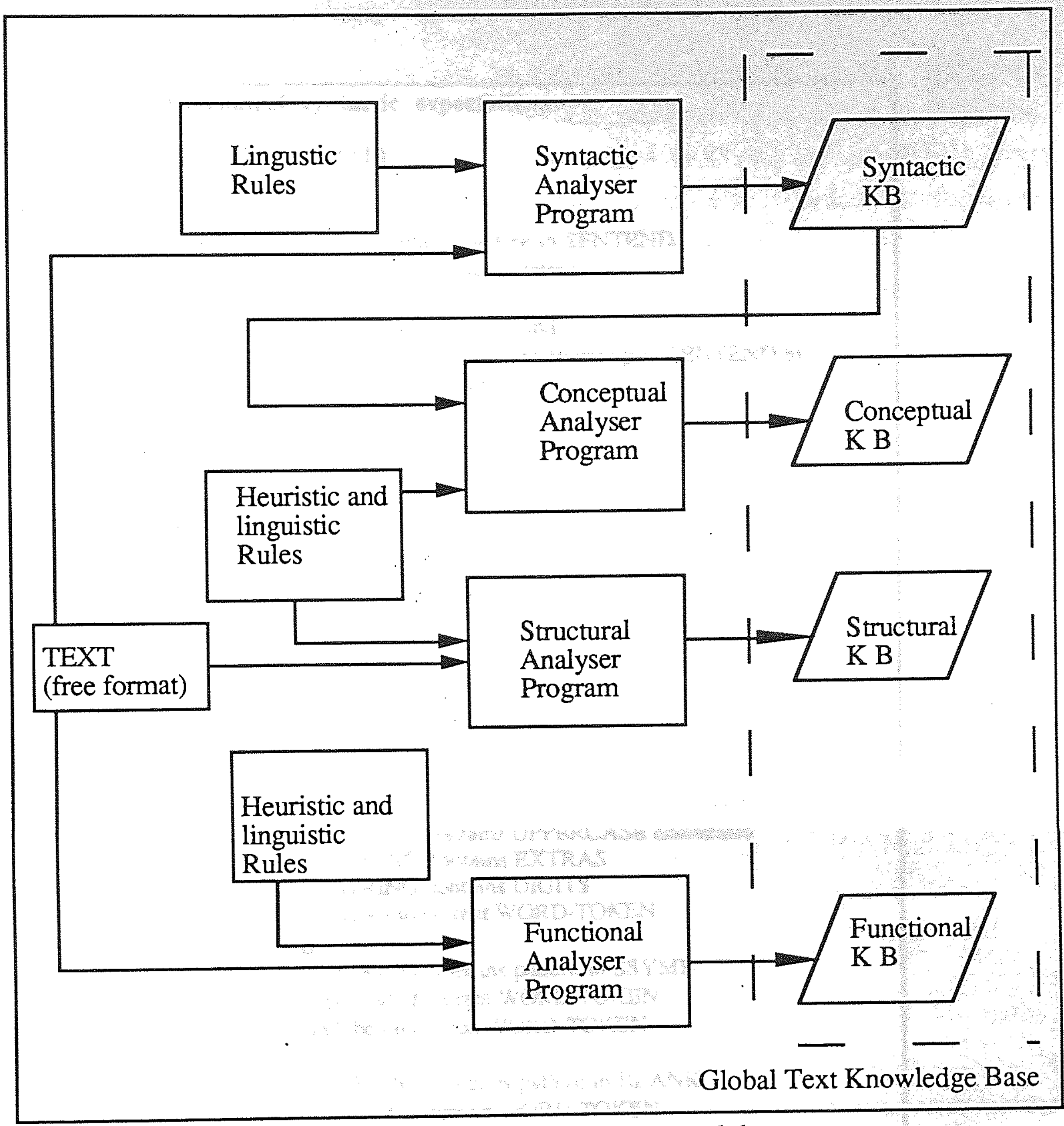


Figure 5.11 The textual analyser module

5.2.1 The Syntactic Analyser Program

The extract above is first analysed to delineate sentence boundaries and isolate words within each sentence. To this end, a list of syntactic rules is compiled for the syntactic analyser and given in figure 5.12 The syntactic analyser scans the text line by line looking for symbols that denote an end of a sentence. It recognises three symbols: the full stop (.), the interrogation mark(?), and the exclamation mark(!). All characters enclosed between these end-of-sentence symbols are attached to a parameter called SENTENCE_TOKEN.

a) To control syntactic expectations:

```
set    SENTEND ( . 1 ? 1 ! )
repeat
read       LINE
           rule  S1:  if LINE contains pattern in SENTEND
                      then  end of a sentence found
                      fi
           rule  S2 :  if end of a sentence found
                      then assign all characters found up to SENTEND to
                      SENTENCE-TOKEN
                      fi
until no more lines
```

b) To determine lexical boundaries:

```
set BLANK ( )
set LOWERCASE (abcdefghijklmnopqrstuvwxyz)
set UPPERCASE (ABCDEFGHIJKLMNOPQRSTUVWXYZ)
set EXTRAS ( ' - )
set DIGITS (0123456789)
set SSYMB (, ; : " ( ) [ ] *)
repeat
scan       STRING

           rule L1 :  if  STRING contains LOWERCASE characters
                      or STRING contains UPPERCASE characters
                      or STRING contains EXTRAS
                      or STRING contains DIGITS
           then  add  STRING to current WORD-TOKEN
                      fi
           rule L2 :  if  STRING contains pattern in SSYMB
                      then  end of current WORD-TOKEN
                      and  begin of next WORD-TOKEN
                      fi
           rule L3  :  if  STRING contains pattern in BLANK
                      then  end of current WORD-TOKEN
                      and  begin of next WORD-TOKEN
                      fi
           rule L4  :  if  STRING contains pattern in SENTEND
                      then  end of current WORD-TOKEN
                      and  begin of next WORD-TOKEN
                      fi
           until no more strings.
```

Figure 5.12 Rules to control syntactic and lexical boundaries

Having identified a sentence in the text, the syntactic analyser scans it searching for word patterns. A word is defined a sequence of characters, some of which may be in upper case, some in lower case, others a mixture of the two cases. A word may also contain a hyphen to account for hyphenated words such as data-base. It may also include apostrophes and numerical characters. This sequence of characters are grouped into a three compound pattern made up of ucs, lcs and extras giving what constitutes a word-forming pattern as described below:

lcs = (abcdefghijklmnopqrstuvwxyz)
ucs = (ABCDEFGHIJKLMNOPQRSTUVWXYZ)
extras = (' -)
digits = (0123456789)
word-pattern = ucs lcs extras

Certain characters such as punctuation marks (, ; : " . ? !) and a blank space aid in isolating words from the text, and as a result they are not included in the definition of a word. Isolating words in a text is carried out on the basis of the above pattern matching procedure, expressed in the form of rules characterised by:
PATTERN,
ACTION.

5.2.2 The Conceptual Analyser Program

The second step is to break the text into its conceptual constituents; this is the function of the conceptual analyser. A set of cognitive and linguistic rules are applied on each sentence to generate a conceptual model for the text under analysis. These rules are specialised strategies used by abstractors when reading a text (Figure 5.13).

```
set PMARKS                    (SENTEND/SSYMBOL)
set ENUM_SYMBOL               (1)/2)/.../ i)/ ii)/.../ a)/ b)/.../(a)/(b)...)
set DETERM                    (THE/A/AN/THIS/THESE/THOSE)
set RELAT-P                   (THAT/WHO/WHOM/WHICH/...)
set POSS-P                    (HIS/HER/MY/OUR/THEIR/...)
set DISTRIB-P                 (EACH/EVERY/EITHER/...)
set INDEF-P                   (ANY/SOME/ALL...)
set PERS-P                    (I/YOU/HE/SHE/...)
set PRONOUNS                  (RELAT-P/POSS-P/DISTRIB-P/INDEF-P/PERS-P)
set PREP                      (IN/AT/WITHOUT/ON/UNDER/INTO/TILL/...)
set CONJ                              (AS/BECAUSE/WHILE/BEFORE/BUT/AND/...)
set ADJ                       (ONLY/MOST/MANY/ALL/EVEN/TOO/JUST/..)
set VERB                      (CONSIST/CONSISTS/REVEAL/REVEALS
                              /COMPRISE/COMPRISES/DESCRIBE/
                              DESCRIBES/...MAY/MUST/CAN/COULD/.../
                              HAS/IS/BE/BEING/...)
set STYLISTIC-MARKER          (THIS CASE STUDY/OUR INTENTION/OUR
                              PURPOSE/THE RESEARCH/WE AIM...)
set EXEMPLIFIER-DELIMITER
                              (LIKE/EG/IE/FOR EXAMPLE...)
set ADV                       (FURTHERMORE/MOREOVER/FIRST/
                              SECOND/QUITE/MAINLY/MOSTLY...)
set ADV                       (word with suffix LY)
repeat
scan SENTENCE
repeat
begin
scan  WORD
rule C1 : if WORD belongs to DETERM) of (WORD belongs to PRONOUNS)
          or (WORD belongs to CONJ) or (WORD belongs to PREP)
          or (WORD belongs to ADV) or (WORD belongs to VERB)
          or (WORD belongs to ADJ)
          then WORD is a GRAMMATICAL-DELIMITER
          fi
rule C2 : if WORD belongs to ENUM-SYMBOL
          then WORD is a GRAPHICAL-DELIMITER
          fi
rule C3 : if WORD belongs to STYLE-LIST
          then WORD is a STYLISTIC-MARKER
          fi
rule C4 : if WORD belongs to EXEMP-LIST
          then WORD is an EXEMPLIFIER-DELIMITER
          fi
rule C5 : if WORD is a GRAMMATICAL-DELIMITER
          then WORD is a CC-DELIMITER
          fi
rule C6 : if WORD is a GRAPHICAL-DELIMITER
          then WORD is a CC-DELIMITER
          fi
rule C7 : if WORD is a STYLISTIC-MARKER
          then WORD is a CC-DELIMITER
          fi
rule C8 : if WORD is an EXEMPLIFIER-DELIMITER
          then WORD is a CC-DELIMITER
          fi
rule C9 : if WORD is PMARKS
          then WORD is a CC-DELIMITER
          fi
rule C10 : if WORD is a CC-DELIMITER
          then end of current CONCEPTUAL-CONSTITUENT
          and beginning of next CONCEPTUAL-CONSTITUENT
          fi
          until end of sentence
          end
until no more sentences.
```

Figure 5.13 Rules to identify conceptual constituents in a text

The aim of the conceptual analyser program is to group words in the text into cognitive units referred to as conceptual constituents. First, the text is processed at the level of a sentence. Each sentence is examined and broken into conceptual constituents bounded by delimiters denoted cc-delimiter. Four major categories of delimiters are identified:

i) graphical delimiters, which refer to punctuation, mathematical symbols, enumeration topography;

ii) grammatical delimiters, which include pronouns, relative pronouns, determiners, conjunctions, adverbs, models, prepositions, quantifiers and a set of verbs;

iii) exemplifiers, which govern ways of illustrating a statement, an approach or a technique using the following terms such as, as, eg, ie, cf, like;

iv) special phrases outlining a sequence of ideas by using terms like firstly, secondly, on one hand, on other hand, well...

When a word is read from a sentence, the conceptual analyser examines it to determine whether it is a cc-delimiter. If it is a cc-delimiter, then this indicates the beginning of a new constituent and the end of a previous one. Each subsequent word is attached to that new constituent until a new delimiter is found.

In the analysis of scientific papers in Chapter 4, abstractors pointed out that authors often use a set of verbs to highlight important conceptual constituents. For example, the verb "reveal" may point out to some results as in the phrase:

this study reveals...

Other verbs such as contain, consist, provide, involve, show, refer, assume...are very common to scientific papers. These verbs are thus used as a means to identify potentially important conceptual constituents.

The output of the conceptual analyser is a set of conceptual constituents, which make up the text. Thus the conceptual model for the passage extracted from Ferrante, is expressed as follows:

concept (1,[Introduction]).
concept (2,[Conflict, resolution]).
concept (3,[integration, of, alternate, opinions]).
concept (4, [state, of, the, world]).
concept (5, [single, coherent, views]).
concept (6, [approach]).
concept (7, [performed]).
concept (8, [separate, entity]).
concept (9, [knowledge, source]).
concept (10, [designed]).
concept (11, [purpose]).

Three comments are worth mentioning here. First, the conceptual constituents are stored into a form of declarative statements, introduced by the predicate called concept. Secondly, constituents are stripped from their delimiters for two main reasons: not to burden the processing time of these constituents by subsequent components and not to overload the computer memory. However, two closely related concepts linked by a determiner (e.g. the integration of alternate options), these two concepts are combined to form one constituent. It is assumed that the determiner "of" always connects two concepts and hence the resulting conceptual constituent is considered to be atomic.

There are instances where the first concept may be a verb followed by the delimiter "of", as in the following phrases:

i) ...consist of...
ii) ...comprise of...

Unless the verb is identified as a cc-delimiter, the conceptual analyser program treats the verb and its object as one single conceptual constituent.

Each conceptual constituent is assigned a unique identity number. This identity number serves as a pointer to locate the position of a given constituent locally within the sentence and paragraph, and globally within the text. For example, the conceptual constituent "conflict resolution", described as concept no 2 by the predicate concept, has five additional coordinates stating its specific position in the text. These coordinates are stored in the predicate called textconcept, as given below:

textconcept (1, 1, 1, 2, 1, 2).

The first coordinate identifies the heading number, in this case, it refers to heading 1. The second coordinate locates its position within this heading; here the value 1 indicates that this constituent belongs to the first paragraph. The third coordinate locates its position within the sentence; this constituent belongs to the first sentence of this paragraph. The fourth coordinate stores its position within the text: this constituent belongs to sentence 3 of this text. Note that the first heading is referred to as sentence 1 in this passage. The fifth coordinate situates the location of this constituent within the sentence; so this constituent occupies the first position in this sentence. Finally, the sixth coordinate denotes the conceptual identity number thereby linking this constituent to the predicate concept which reveals its content.

5.2.3 The Structural Analyser Program

One of the major findings in Chapter 4 was that the significance of the textual space structure in locating salient information for abstracting. The structural analyser program tries to capture the structure of the text. It distinguishes title from headings, locates the beginnings and ends of paragraphs, stores their position with respect to their headings and records the position of the sentences that make up these paragraphs.

In addition, this program compiles statistical information revealing the total number of words, sentences, paragraphs, headings, and conceptual constituents. For the above passage, the following clauses are generated.

summary (1,1).	Sentence 1 contains only 1 conceptual constituent .
summary (2,4)	Sentence 2 contains 4 conceptual constituents.
summary (3,6)	Sentence 3 contains 6 conceptual constituents.
stats (3, 36, 1, 11)	This passage consists of 3 sentences, 36 words, 1 paragraph and 11 conceptual constituents.

To identify the title, headings and paragraphs, the structural analyser applies a set of rules which are summarised in figure 5.14. This program provides the system with a knowledge base that describes the position of each conceptual constituent in relation to the sentence, paragraph, section and the text as a whole.

```
repeat
        read SENTENCE
rule ST1 :      if SENTENCE begins with the SYMBOL (~)
                and SYMBOL followed by NUMERIC-CHARACTER
                then SENTENCE is HEADING.
                fi
rule ST2 :      if SENTENCE is HEADING
                and NUMERIC-CHARACTER is set to zero
                then HEADING is TITLE
                fi
rule ST3 :      if SENTENCE is HEADING
                and NUMERIC-CHARACTER is greater than zero
                then HEADING is SECTION-HEADING.
                fi
rule ST4 :      if SECTION-HEADING found
                then record its order of occurrence
                and initiate local section parameters.
                fi
rule ST5 :      if SENTENCE is indented
                and SENTENCE follows SECTION-HEADING
                then SENTENCE indicates beginning of paragraph 1
                and initialise local paragraph parameters
                and update local section parameters
                fi
rule ST6 :      if SENTENCE is indented
                and SENTENCE does not follow SECTION-HEADING
                then SENTENCE indicates beginning of paragraph
                and update local section parameters
                and initialise local paragraph parameters
                fi
    until no more sentences
```

Figure 5.14 Rules to determine title, headings and paragraphs

5.2.4 The Functional Analyser Program

Throughout the study, it is maintained that any text cannot be investigated in isolation from the communicative function. What writers write and how they convey their knowledge depends greatly on what they expect their readers to do with their paper. With regard to abstracting, a list of discourse markers was heuristically drawn up and listed in figure 4.3. This list is formalised and expressed in terms of rules to aid the fourth submodule of the system to scan each sentence of the text and determine its function in relation to abstracting. This is the first task of the functional analyser program.

Seven major functions are defined: introduction, methodology, objectives, discussion, results, conclusions and background. The functional analyser is to scan each sentence for discourse markers to determine its function. Each functional category has a sequence of word patterns. The program examines each sentence searching for the occurrence of these patterns. If a sentence is found to contain a specified pattern, it is assigned the corresponding functional category. The category "background" is given to any sentence whose pattern matching fails against the first six categories. To obtain the function of a given sentence the functional analyser program applies a list of rules, details of which are given in figure 5.15.

For example, the knowledge base produced by the rules of functional analyser for the above extract is described as follows:

function (1, introduction)
function (2, background)
function (3, methodology)

The second task of the functional analyser is to examine each sentence for any special markers used by the author(s) to introduce specific characteristics about their study. Examples of these special markers were given in section 4.2.3.2 in Chapter 4. Those sentences containing such special markers are selected and noted by the predicate "status" and expressed as follows:

status (sentence_no, theme_marker)

Thus, clauses headed by the predicates "function" and "status" make up the functional knowledge base of the text.

```
set INTRO-PAT          <-- (introduction/introductory/overview/)
set METHOD-PAT         <-- (methodology/techniques/approach/
                              formalism...)
set CONCLUSION-PAT     <--  (conclusion/conclusions/concludes/conclude)
set OBJECTIVES-PAT     <-(aim/goal/pbjectives/concerns/intended/...)
set THEME-MARKERS      <-- (thispaper/thisresearch/thiscase study/our/we/.)
set EQUATION           <--  (=)
repeat
        scan SENTENCE
        rule f1 :      if SENTENCE contains pattern in INTRO-PAT
                       then SENTENCE includes introductory comments
                       fi
        rule f2 :      if SENTENCE contains pattern in METHOD-PAT
                       then SENTENCE includes methodology statements
                       fi
        rule f3 :      if SENTENCE contains pattern in CONCLUSION-PAT
                       then SENTENCE includes conclusion statements
                       fi
        rule f4 :      if SENTENCE contains pattern in OBJECTIVES-PAT
                       then SENTENCE includes objectives statements
                       fi
        rule f5 :      if SENTENCE contains pattern in THEME-MARKERS
                       then SENTENCE has special emphasis from author
                       fi
        rule f6 :      if SENTENCE contains pattern in EQUATION
                       then SENTENCE includes formulae
                       fi
        until no more sentences.
```

Figure 5.15 The functional analyser rules.

5.3 The Expert Classifier Module

The previous module applied heuristic and linguistic rules to capture the structure of the text and describe its content in terms of conceptual constituents pertinent to the process of abstracting.The textual analyser has transformed a free-format text into a text knowledge base in declarative form. This knowledge base contains the necessary information for the expert classifier to evaluate the significance of sentences.

The expert classifier module contains two kinds of knowledge. The first kind is heuristic abstracting strategies which enable INFORMEX to determine the degree of importance of each sentence in the text. A typical set of strategies used by the expert classifier is given below:

Strategy 1 : Use the function characteristics (i.e. examine the predicates "function" and "status") of a given sentence to evaluate its importance.

Strategy 2 : If the function of this sentence indicates that it contains conceptual constituents describing the purpose of this paper,then this sentence is important.

Strategy 3 : If a sentence contains conceptual constituents identified as relevant lexical markers (such as theme markers, summarisers, concluders), then this sentence is important.

Strategy 4 : If a sentence occurs in the first paragraph and occupies an opening sentence position, then this sentence is important.

Strategy 5 : If the function of this sentence is identified to provide background information, then this sentence is not important

To explain how the expert classifier works, assume that the system is processing sentence 12 which is described by the textual analyser as being a heading, containing conceptual constituents describing background information. The corresponding text knowledge base describing this scenario is given below:

status (12, heading).
function (12, background).
text concept (2, 0, 0, 12, 1, 50).
concept (50,[background, study]).

The expert classifier applies strategy 1 and infers that this sentence is a heading and that it contains conceptual constituents providing background information. It then proceeds to apply the following four strategies. Strategies 2, 3 and 4 fail. Strategy 5 is applicable and as a result the expert classifier asserts that this sentence is not important. The next normal set of actions is to move to the next sentence in the text knowledge base and evaluate its importance.

However, a human expert encountering such situation proceeds to a different course of actions. He/she anticipates that the information likely to follow under this heading will describe the preliminary study undertaken before the actual work got started with a view to set directions for the ultimate project, or historical background reporting progress of past study and setting the scene for the current work.

So, in addition to heuristic abstracting strategies, the expert classifier makes use of reading techniques as another kind of knowledge. From Chapter 4, it was found that abstractors developed specialised reading strategies for quickly locating specific information, combining skimming and scanning methods. Intensive reading applied to those sections identified as important as a result of skimming and scanning. These strategies are also formalised and integrated into the expert classifier. They provide INFORMEX with a guidance of when and in what sequence to apply these heuristic abstracting strategies.

These strategies are added to the expert classifier and referred to as metarules.

Metarule 1 : look for a heading

Metarule 2 : <u>If</u> heading found
<u>then</u> apply strategy 1 and strategy 2
<u>fi</u>

Metarule 3 : <u>If</u> strategy 2 fails
<u>then</u> apply strategy 5

fi

Metarule 4 : If strategy 5 succeeds
then select all sentences that belong to this heading
and assign zero importance to all these sentences and apply
metarule 2
fi

Metarule 5 : If strategy 2 succeeds
then select all sentences that belong to this heading and apply
strategy 4.
fi

In assigning importance, the expert classifier distinguishes four degrees: high, medium, low and zero. High importance is assigned to those sentences which are identified as functionally important and positionally relevant. Sentences which contain conceptual constituents describing the purpose of the study and methodology and reporting conclusions of the paper are considered functionally important. Sentences announced by the author(s) by means of theme markers are also described as functionally important. Sentences which occur as the first or last sentence of the first or the last paragraph of a heading recognised functionally important are referred to as positionally relevant.

Medium importance is attributed to those sentences which belong to important headings and occur in the following positions:

i) between the first and last sentence of the first paragraph;
ii) between the first and last sentence of the last paragraph;
iii) between the first and last paragraphs if sentence announced by a theme marker.

Low importance is given to those sentences which belong to headings identified as important but are neither functionally nor positionally relevant. Finally, zero importance is assigned to all sentences belonging to a heading dscribed as not important.

A full design of the expert classifier is given in figure 5.16.

i) Task-specific rules:

rule 1 : if a sentence belongs to the first paragraph of a given heading
and it is the first sentence in this paragraph
then it is an opening sentence
fi

rule 2 : if a sentence belongs to the last paragraph of a given heading
and it is the first sentence in this paragraph
then it is an opening sentence

fi

rule 3 : if a sentence belongs to the first paragraph of a given heading
and it is the last sentence in this paragraph
then it is a closing sentence

fi

rule 4 : if a sentence belongs to the last paragraph of a given heading
and it is the last sentence in this paragraph
then it is a closing sentence

fi

rule 5 : if a sentence contains any of the following functional characteristics:
(introduction, objectives, methodology, conclusion, results)
then assert sentence functionally relevant

fi

rule 6 : if a sentence contains any of the following functional characteristics:
(background)
then assert sentence not functionally relevant

fi

rule 7 : if a sentence contains conceptual constituents recognised as lexical markers
then assert sentence is functionally relevant

fi

rule 8 : if a sentence is an opening sentence
then set SENTENCE <-- IMPORTANT

fi

rule 9 : if a sentence is a closing sentence
then set SENTENCE <-- IMPORTANT

fi

rule 10 : if a sentence is identified as functionally relevant (i.e. check the predicate 'function')
then set SENTENCE <-- IMPORTANT

fi

rule 11 : if a sentence contains lexical markers(i.e. check the predicate 'status')
then set SENTENCE <-- IMPORTANT

fi

rule 12 : if a sentence is a heading
then evaluate relevance

fi

rule 12.1 : if a heading is identified as functionally relevant
then IMPORTANCE (HEADING) <-- HIGH
and read intensively all sentences belonging to that heading
and evaluate each sentence

fi

rule 12.2 : if a heading is not functionally relevant
then skip all sentences belonging to that heading

fi

rule 12.3 : if a sentence belongs to the first paragraph of an important heading
and a sentence is an opening sentence
then set IMPORTANCE (SENTENCE) <-- HIGH

fi

rule 12.4 : if a sentence belongs to the first paragraph of an important heading
and a sentence is a closing sentence
then set IMPORTANCE (SENTENCE) <-- HIGH

fi

rule 12.5 : if a sentence belongs to the last paragraph of an important heading
and a sentence is an opening sentence
then set IMPORTANCE(SENTENCE) <-- HIGH

fi

rule 12.6 : if a sentence belongs to last paragraph of an important heading
and a sentence is a closing sentence
then set IMPORTANCE(SENTENCE) <-- HIGH

fi

rule 12.7 : if a sentence belongs to first paragraph of an important heading
and a sentence is not an opening sentence
and a sentence is not a closing sentence
and a sentence is functionally relevant
then set IMPORTANCE(SENTENCE) <-- MEDIUM

fi

rule 12.8 : if a sentence belongs to last paragraph of an important heading
and a sentence is not an opening sentence
and a sentence is not a closing sentence
and a sentence is functionally relevant
then set IMPORTANCE(SENTENCE) <-- MEDIUM

fi

rule 12.9 : if a sentence does not belong to first paragraph of an important heading
and a sentence does not belong to last paragraph of an important heading
and a sentence is functionally relevant
then set IMPORTANCE(SENTENCE) <-- MEDIUM

fi

rule 12.10 : if a sentence belongs to first paragraph of an important heading
and a sentence is not an opening sentence
and a sentence is not a closing sentence
and a sentence is not functionally relevant
then set IMPORTANCE(SENTENCE) <-- LOW

fi

rule 12.11 : if a sentence belongs to last paragraph of an important heading
and a sentence is not an opening sentence
and a sentence is not a closing sentence
and a sentence is not functionally relevant
then set IMPORTANCE(SENTENCE) <-- LOW

fi

rule 12.12 : if a sentence does not belong to first paragraph of an important heading
and a sentence does not belong to last paragraph of an important heading
and a sentence is functionally relevant
then set IMPORTANCE(SENTENC) <-- HIGH

fi

rule 12.13: if a sentence does not belong to first paragraph of an important heading
and a sentence does not belong to last paragraph of an important heading
and a sentence is not functionally relevant
then set IMPORTANCE(SENTENCE) <-- LOW

fi

rule 12.19 : if a sentence belongs to a heading identified as not important
then set IMPORTANCE(SENTENCE) <-- ZERO

fi

ii) Mata-level rules:

Metarule 1 : if a sentence is identified as a title
then assert IMPORTANCE(SENTENCE) <--HIGH

repeat
scan knowledge base for first heading

```
metarule 1 :  if a sentence is a heading
              then evaluate its importance

              fi

metarule 2 :  if a heading is functionally relevant
                  then a heading is important

              fi

metarule 3 :  if a heading is important
                  then process first paragraph
                  and process last paragraph
                  and process in-between paragraphs

              fi

metarule 4 :  if first paragraph
                  then apply rule 12.3
                  or apply rule 12.4
                  or apply rule 12.7
                  or apply rule 12.10

              fi

metarule 5 :  if last paragraph
                  then apply rule 12.5
                  or apply rule 12.6
                  or apply rule 12.8

              fi

metarule 6 :  if in-between paragraph
                  then apply rule 12.9
                  or apply rule 12.11
                  or apply rule 12.12
                  or apply rule 12.13

              fi

metarule 7 :  if a heading is functionally not relevant
                  then apply rule 12.14

              fi

metarule 8 :  if heading not important
                  then assert all sentences under this heading not important
                  IMPORTANT(SENTENCE) <-- ZERO

              fi

until no more headings.
```

Figure 5.16 The expert classifier rules

6. Languages and Inference Engine in INFORMEX

This section is concerned with mapping the formalised knowledge into a suitable environment with symbol manipulation. Two well known AI programming environments are available: LISP and PROLOG. While LISP is a powerful list processing language with recursive function capability for describing problems and processes, PROLOG is a logic oriented language, based on predicate calculus logic. It is specially well suited for describing problems in terms of objects and relations between objects. Prolog solves a problem by pattern matching. A Prolog program "..consists of a group of procedures, where the left side of a procedure is a pattern to be instantiated to achieve the goals on the right side of the procedure".

PROCEDURE : PATTERN ---> GOALS [Gewarter, 1984: 38]

These features are extremely useful for manipulating knowledge expressed in the forms of rules. It is this powerful structure that makes Prolog an attractive and excellent candidate for representing and manipulating expert knowledge and strategies. Gewarter claims that Prolog is surprisingly efficient and its compiled version is faster than compiled LISP [Gewarter, 1984: 40]. This property is considered vital to the success of INFORMEX .

While Prolog is an ideal language to represent and manipulate the expert classifier module in INFORMEX, it lacks behind other programming languages with regard to processing texts. For this reason, any work associated with processing text is carried out in SPITBOL. This language offers extensive and powerful pattern matching facilities which could be efficiently used by the textual analyser module to process and generate the text content analysis for INFORMEX, as described in figure 5.11. The following sections show how the two modules, the textual analyser and the expert classifier, are implemented and illustrate by means of an example the functions of the inference engine in INFORMEX.

6.1 SPITBOL Implementation of the Textual Analyser Module

Most of the work involved by the textual analyser module involve scanning a given text for the occurrence of a particular string or sequence of strings. The textual editor program, for example, scans the text for removing the full stop from a collection of words such as i.e. because the full stop plays a crucial role in identifying sentences within the text. The main task of the conceptual analyser is to search for the occurrence of delimiters within a sentence to identify the beginning and end of a conceptual constituent. Two main functions are assigned to the syntactic analyser: it has to identify words in a text and isolate sentences. This is achieved by pattern matching as well. Similarly, one of the functions of the structural analyser program is to scan the layout of the text for patterns that determine headings, paragraphs and sections. Finally, the functional analyser submodule looks for phrases and words that may indicate information relevant to abstracting.

There is a common structure to those four submodules programs. Each submodule is assigned a group of patterns relevant to their respective function. This causes the SPITBOL program to scan a line, or a sentence and a word for these patterns. If a match is found, this is recorded as a factual statement into the appropriate text knowledge base. This factual statement is expressed in Prolog notation for a later Prolog manipulation by the expert classifier.

To illustrate the tasks of the textual editor, consider the following sentence:

Original sentence :
This paper describes the techniques given in fig.1 for representing knowledge in Prolog.

One of the textual editor patterns is specified in SPITBOL as follows:

exemplifier-pattern = " etc. / i.e / ie. / fig. / Fig. / eg. /e.g. / cf. / & / "

The textual editor reads the sentence above and searches each string for the occurrence of any of the patterns contained in the exemplifier pattern, until it finds an occurrence; it then removes its full stop. This is repeated until all the occurrences of the pattern have been dealt with. The sentence above is transformed to the following sentence:

Textual editor output :
This paper describes the technique given in fig 1 for representing knowledge in Prolog.

A further transformation is achieved by the syntactic analyser who uses the following pattern to isolate words:

```
Letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
+                   "abcdefghijklmnopqrstuvwxyz"
+                   "' - "
```

Note that the apostrophe and the hyphen are taken as part of a word. Using the following instructions, the syntactic analyser reads a line and isolates one word at a time and assigns it to a word:

```
wpat = break (letters) .  gap span (letters) . word
```

Similarly, it uses the following patterns to detect a sentence:

```
sentend = " . ! ? "
sentence = (break (sentend)   any (sentend)   break ( ' ')) . sentence
```

The rule S1 is implemented in SPITBOL in the following way:

```
line sentence =                              : F(readline)
sentcount = sentcount + 1
```

The terms break and span are built-in functions in SPITBOL. A full description of these functions as well as other features available in SPITBOL is given in "Computers in Linguistics" [Butler, 1985].

The syntactic analyser also compiles the number of words and sentences for statistical purposes.For example, for the above sentence, the output is:

sentence no : 1
This paper describes the techniques given in fig 1 for representing knowledge in Prolog.
(No of words = 14, Total no of words in text = 14).

Each sentence in a text is analysed further by the conceptual analyser program and broken into conceptual constituents using the rules given in figure 5.13. This is achieved by scanning the sentence for the specified delimiter patterns. If an occurrence of this pattern is found then this indicates the end of the current conceptual constituent and the beginning of a next conceptual constituent (Rule C10). This rule is expressed in SPITBOL in the following way:

```
nextword     sentence wpat =
        ident (delimiter < word > , word)                      :S(newlist)
        concept-constituent = concept-constituent gap word  : (nextword)
newlist concept =                                              :(nextword)
```

The if-part is described by :

ident (delimiter < word > , word)

If the condition is satisfied, that is the word is a cc-delimiter, then a newlist is created by initialising the parameter concept. If the condition fails, then the word is attached to the current open list. The output for this program for the above sentence is expressed in declarative form described by the clause concept suitable for Prolog processing.

The structural analyser submodule ensures that whenever a heading, a paragraph and a sentence is identified, global parameters are updated while appropriate local parameters are initialised. Thus each conceptual constituent can be easily located and recorded using Prolog syntax structure.

Finally, the functional analyser program examines the sentence looking for discourse markers, discussed in the previous Chapter. For example, the phrase "this paper" is recognised as a special marker by the following rule and recorded as a theme marker:

```
      sentence markers                                          : F (backg)
      output = "status"   ( " sentcount " , theme-marker ).  "  : (next sentence)
backg       output = " function ( " sentcount " , background ).  "  : (next sentence)
```

For each functional category, a set of word patterns is declared at the beginning of the program. If a pattern is found, the functional analyser assigns the appropriate function. For instance, if a sentence contains any of the following patterns:

introduction/overview/introductory

the program stores this information as follows:

function (sentence-no, introduction).

If a sentence does not contain any of the specified pattern, its function is set to background:

function (sentence-no, background).

These four programs analyse each sentence in the text and provide INFORMEX with a set of declarative clauses which describe the content and make up the global text knowledge base to be processed by the expert classifier. For efficiency purposes, the syntactic analyser, the conceptual analyser and the structural analyser are incorporated into one single program. Thus as soon as a sentence is identified, words can be isolated to check for delimiters, conceptual constituents are created and the text structure analysed. All four submodules use rule based approach to detect a given pattern. The general syntax structure is illustrated below:

(insert pattern here) : (action 1) S (action 2)

If pattern matching succeeds, action 2 is executed else action 1. A listing of the four submodules is given in Appendix E.

6.2 Prolog Implementation of the Expert Classifier Module

This module is entirely implemented in Prolog. As explained, there are two kinds of rules. The first kind of rule is directly related to expert heuristics in identifying conditions that make a given sentence important for abstracting purposes. The second kind of rules, termed metarules guide INFORMEX in applying these rules. These rules have a left side and a right side separated by the symbol (:-) to mean "if". The left side of the rule is a single predicate expression usually with variables as arguments. The right side of the rule is a set of conditions with multiple predicate expressions combined with the comma (,) to express an AND relation, semicolon (;) to describe an OR relation as well as the not symbol. For instance:

```
assert-high (Sentence-no): -   textconcept (H, P, Sp, St, Cs, Ct),
                               assert-high(H).
```

This states that the predicate assert-high is true if the conditions textconcept and assert-high succeed. Prolog allows also expressing rules without a right side predicateas in the following fact:

```
member (X,[X|L]).
```

This is particularly useful if one is examining if an item X is part of a list L. In other words, facts are just a special case of rules in Prolog.

To explain how a rule is implemented in Prolog, consider rule 12.3 which states that:

if a sentence belongs to first paragraph of an important heading
and a sentence is an opening sentence
then set IMPORTANCE(SENTENCE) <-- HIGH
fi

To show how Prolog implements this rule, the following definitions must be first explained. A sentence belonging to the first paragraph has the second and third arguments or coordinates set to 1 as in:

```
textconcept (Heading_no, 1, 1, Sentence-no, _, _, _,).
```

A highly important heading is recorded as an assertion:

```
assert-high (Heading-no).
```

The Prolog description of the above rule can now be written:

```
assert_high(Sentence-no) : -  textconcept(Heading_no, 1, 1, Sentence_no, _, _),
                              assert_high (Heading_no).
```

6.3 The Inference Engine in INFORMEX

The inference engine is the active component of the system. It selects rules from the heuristic knowledge base which match the content of the global text knowledge base and perform the associated actions.

Each rule expresses a relationship between the condition (the If part) and its consequent (the Then part). In general the conditional part in INFORMEX is expressed as a pattern. If a pattern matching is found in the global text knowledge base, then the rule is fired. The flow of control in firing a rule is determined entirely by the inference engine. To fire a rule, the inference engine performs the following operations:

i) determine which rules have satisfied the IF part condition (pattern matching stage);
ii) select one rule with a satisfied condition;

iii) perform the consequent (the Then part) associated with the selected rule (action stage);

iv) go back to step i) .

Consider rule 12.13 in the previous section. To satisfy the goal that a given sentence-no is highly important, it has to satisfy the two parts of the If statement. The first part describes typical characteristics of an opening sentence. It examines the first part and searches the global text knowledge base for sentences that contain such patterns. The first textconcept clause that matches these requirements is selected and as the heading-no is not known, it instantiates this argument to match the selected one. Now the value of Heading-no is known, it can attempt to satisfy the second part of the If condition. Again it searches the text knowledge base for a clause that confirms that Heading-no is highly-important. If such a clause is found, then the goal succeeds and this sentence is asserted highly important. The rule is complete and the inference engine adds the new assertion to the global knowledge base, making it available to the expert classifier if required. This pattern matching mechanism is incorporated into Prolog. If the pattern matching fails as Prolog searches through the rules, then it automatically backtracks to its previous choice rule and searches for an alternative rule. The search process is guided by the metarules held in the expert classifier.

The inference engine is held by the metarules to prove some goals and it tries to find a way of instantiating variables in the predicates as necessary. It gives goal order top priority and then treats rule and fact order of equal importance after that, using the knowledge base order as a way of assigning priorities to individual rules and facts. This is a goal-directed reasoning or backward chaining.

7. Summary

This Chapter had three specific objectives. The first was to find a suitable text linguistic formalism to represent the content of text under study which can be amenable for computer manipulation. The second objective was to transform the abstractors' strategies and knowledge into a rule base system which could then be applied to analyse the characteristics of the text as well as its content. The third goal was to develop and implement a computer system that integrates the above objectives and generates an extract. This was achieved by building a rule base system, called INFORMEX, which is capable of applying expert strategies and skills by means of computer programs onto a corpus of text to select those sentences containing relevant information for abstracting purposes.

CHAPTER 6

ANALYSIS OF RESULTS AND DISCUSSION

Analysis of Results and Discussion

1. Introduction

The previous Chapters investigated the process of abstracting and suggested that a simplified approach of abstracting could be implemented in a computer system using a rule base formalism. The computer system, called INFORMEX, was outlined in Chapter 4. Its main objectives were to capture the knowledge and skills abstractors use in abstracting leaving aside the knowledge of the domain of the papers being processed. INFORMEX is used as a tool to investigate the viability of this simplified approach.

This Chapter attempts to analyse the performance of INFORMEX. The results produced by the two modules of INFORMEX and the extracts, in particular, are carried out from the two standpoints of artificial intelligence and linguistics. Issues related to consistency, usability and completeness of INFORMEX are addressed. Standards of textuality namely informativity, cohesion and coherence, are also raised. The overall performance of INFORMEX as well as the universality of this approach and the value of the rule base system formalism are also discussed.

2. The Data

The data used by INFORMEX were five scientific papers selected from the proceedings of the ninth international joint conference on AI. These papers are listed below. Full references to these papers are given in Appendix F.

Paper 1: The Characteristic Error Approach to Conflict Resolution.
Paper 2: COMODEL: A Language for the Representation of Technical Knowledge.
Paper 3: Checking an Expert System Knowledge Base for Consistency and Completeness.
Paper 4: An Expert Adviser for Photolithography.
Paper 5: Fault Diagnosis through Responsibility.

The size of the corpus is summarised below.

Paper no.	No. of words	No. of sentences	No. of paragraphs
1	2804	128	26
2	1983	116	30
3	1749	82	12
4	1863	108	24
5	2343	145	32
Total	10742	579	124

Table 6.1 INFORMEX corpus

Although these papers were published in the same proceedings there is no consistency in the structure and the style of the manuscripts. For instance, headings are centred and bold typed in paper 4, underlined in paper 3, announced in Arabic numerals in paper 1. In some papers, indentation is used to mark the beginning of a paragraph, in others a blank line separates paragraphs from each others. Furthermore, papers 2 and 3 use indentation only for listing a set of items or describing an algorithm. Paper 4 has a totally different pattern: all paragraphs to the exception of the first ones in each section are indented and there are no blank lines to separate them.

Before analysing these papers by computer programs, the five papers were first "read" by the KDEM scanner and then stored into a computer file. As each paper had a different type font and layout, some manual editing was required to resolve ambiguities. In most papers, it was difficult for the scanner to differentiate between the letters f, t and i, and the letters e and c. Some symbols such as o and •, which are used in some papers for enumeration purposes, were "read" by the scanner as a full stop or the letter o.

A further problem was encountered in the processing of these papers, relating to punctuation. Some authors use a full stop for listing their points as illustrated below:

"Most quasiformal theories in psychology include the following assumptions and analysis.

1. Sentences are segmented into idea units...
2. Idea units within sentences or clauses are structurally related." [Graesser et al, 1985 : 110]

Others use the colon symbol (:) as shown in the following extract:

"Our approach has the following basic features:
1. Component oriented description.
2. Qualitative and qu ntitative representation of the properties of physical objects.
3. System theoretical taxonomy for classifying the variety of components and theory connections (component types, interaction types).
4. Uniform description of the function and the geometric properties of components." [Dilger et al, 1985: 353]

Because of the significance of the full stop in INFORMEX, these inconsistencies in the punctuation were detected and removed manually. Other types of inconsistency, which can be automated, were removed by means of a computer program. To ensure that punctuation symbols were processed correctly, they were replaced by the term "pmarks". Because of the significance of the dot as a sentence terminator, words such as cf., e.g. and fig. had their dot removed. Also the ampersand (&) was replaced by the conjunction "and". The results of the textual editor program are given in Appendix G.

In summary, the data was edited before releasing it for processing by INFORMEX. The purpose of the editing was to remove ambiguities and inconsistencies in the editing, which was carried out manually, ensuring that:

i) title and headings were numbered using one common system, and
ii) paragraphs were clearly marked through indentation.

Second, some textual editing was carried out to resolve the difficulties in scanning certain letters and symbols manually and to remove other potentially ambivalent symbols by computer program. Finally, because INFORMEX could not handle algorithms, figures, footnotes, and tables, these were deleted from the original papers.

3. Evaluation Issues

Having developed a prototype system, INFORMEX, and some rules for applying experts' strategies, one is faced with the inevitable question: does it work? In just the same way that a conventional computer program needs to be validated, so does this expert system. In contrast to conventional programs whose algorithms rest on well established business practices and well defined steps, INFORMEX, is based on heuristics and strategies which are born out by practice but have never been subjected to rigorous examinations. Furthermore, human expertise is seldom evaluated objectively. Consequently, one could not apply the same evaluation criteria to computer systems and human experts.

There is another equally important question concerning the difficulties in validating a system like INFORMEX. The problem is: when can one pronounce a judgemental system to be correct? The most obvious attempt at validation is to compare the results of INFORMEX against abstracts produced by abstractors or authors of the paper. This suggests, however, that there is a unique solution, a concept which may succeed in conventional computer systems but does not apply in expert systems. The abstracts produced by the participants in Chapter 4 demonstrated clearly that there is no such view of a unique abstract to a paper. Not only the content of an abstract varies from one abstractor to another, but also the style and the format.

This problem of validation is a difficult issue that is still being debated by the AI community. Because the technology of expert systems is still in its infancy, it will be a long time before some sort of consensus of opinion is reached. Gasching et al highlight these difficulties in their paper on "Evaluation of Expert Systems : Issues and Case Studies".

"Existing techniques for evaluating these systems are few and primitive. Certainly many criteria, like correctness, efficiency or friendliness, used to evaluate computerized or human systems also apply to expert systems. But expert systems are unique in that they contain human expertise and are thus most often compared and evaluated with respect to human performance. In this context it is not always clear whether a correct solution... is one that a human expert would give, one that a group of experts would agree upon, or one that represents the ideal solution. No one knows how to evaluate human expertise adequately, let alone how to evaluate the expert systems that attempt to recreate that expertise" [Gashing et al, 1983: 277].

At this stage of expert systems' evolution, the evaluation of INFORMEX, however difficult, must take place even if the nature of the evaluation may become judgemental. In the absence of a well defined method of validation, one is compelled to use personal judgement. In this case, caution and explicitness are required. Caution is necessary because many factors can invalidate the results produced by an expert system. Explicitly examining the results against the specific goals of the system for which it was designed, is one way of ensuring that judgement is not biased or misguided.

The main objectives of INFORMEX were to examine whether adequate abstracts could be produced by applying rules which are partly based on experts' reading strategies and the short cut strategy and partly making use of the linguistic properties of the paper being abstracted. The performance of INFORMEX must be evaluated in the light of these objectives. That is, in evaluating the performance of INFORMEX, the following issues must be addressed:

i) consistency,
ii) usability, and
iii) completeness.

The requirement for consistency demands that INFORMEX applies similar strategies to similar situations. The task, here, is to examine for each paper the sentences selected by INFORMEX to make up an abstract and to investigate the criteria of their selection. The requirement for usability stipulates that all sentences selected by INFORMEX contain relevant information. A sentence is defined to be relevant if it contains information which describes the purpose of the paper, the methodology or technique used, the results or conclusions reached by the author(s). Finally, the requirement for completeness states that the knowledge base is sufficiently wide in its coverage to allow INFORMEX to meet successfully its specific goals of extracting relevant sentences for inclusion in the abstract.

4. Evaluation of INFORMEX

This section evaluates the performance of INFORMEX in terms of the requirements discussed in the previous section. The main function of INFORMEX is to analyse the sentences of a given paper individually using the heuristic rules with the aim of classifying these sentences into four degrees of importance: high, medium, low and zero. Only those sentences which are assigned a high degree of importance are selected for inclusion in the abstract. Sentences with medium importance provide further details to

support the selection criteria of the most important sentences. Low and zero degrees of importance are attributed to those sentences with little or no relevance to abstracting. For each experimental paper, the results are first summarised and then examined to check the consistency of the selection criteria, the usability of these highly classified sentences and the completeness of the knowledge of rules.

4.1 Summary of Results

The results of the two modules, the textual analyser and the expert classifier are listed in appendix H and summarised in table 6.2 given below. On average, sentences which are classified as highly important, represent 11% of the total sentences, sentences evaluated as of medium importance constitute another 9% and finally sentences which score low or zero importance amount to 80%.

Category of importance	Paper 1 Sent.	%	Paper 2 Sent	%	Paper 3 Sent.	%	Paper 4 Sent.	%	Paper 5 Sent.	%
high	17	13	11	9	10	12	11	10	13	10
medium	14	11	8	7	14	17	8	7	8	5
low/zero	97	76	97	84	58	71	89	83	124	85
Total	128	100	116	100	82	100	108	100	145	100

Table 6.2 Summary of INFORMEX results

These results suggest that, on average, 89% of the content of a paper can be disregarded and that abstract can be obtained from the remaining 11%. Figure 6.3 shows that the length of a paper does not bias the result of the high and medium categories of importance. Paper 3, for example, which is the shortest paper in the sample scores the highest percentage (17%) in the "medium category", the second highest percentage (12%) in the "high category" whereas paper 5, which is the longest paper, scores low percentage (5%) in the "medium category" and a low percentage (10%) in the high category. On the low/zero category, however, with the exception of paper 1, one can state that the longer the paper the higher the percentage of the "low/zero category". Paper 1 is the only paper whose second heading was identified as an important one and consequently sentences were identified as highly relevant and were enclosed in the high category instead of the low/zero category. This, as a result, increased the number of sentences in the high and medium categories.

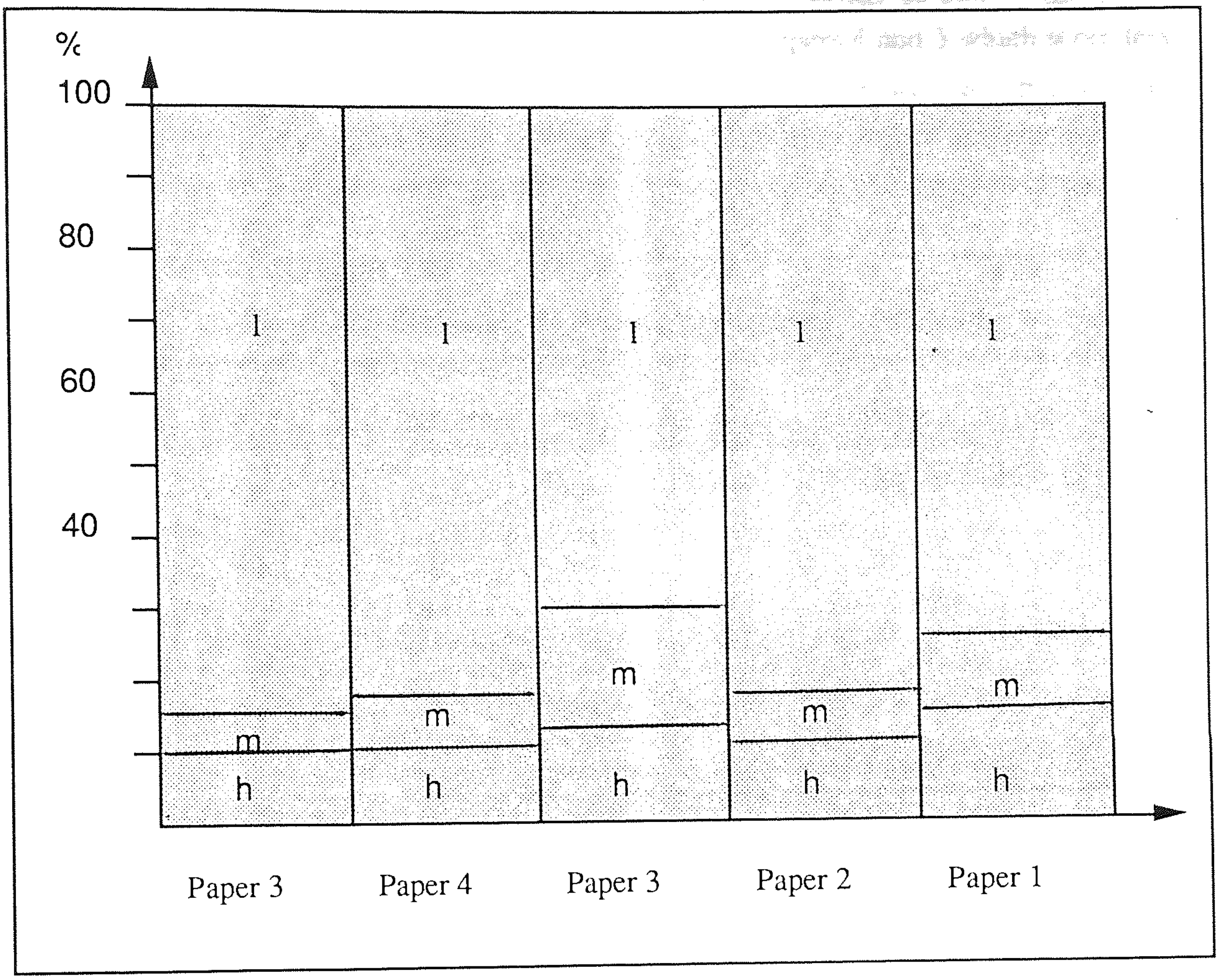


Figure 6.3 Graphical representation of INFORMEX results

Similarly, the length of the introduction and conclusions has not influenced the size of the highly important category as illustrated in table 6.4. It is true that paper 1 with the largest number of words (1292) may have scored the highest percentage, and that paper with the smallest number of words (226) has scored the lowest percentage in that category, the remaining papers 3,4,5 do not follow the pattern. In fact papers 4 and 5 which score less than paper 3, have a much larger introduction and conclusions section than paper 3. In other words, the size of the paper as well as the sections labeled introduction and conclusions has no influence on the selection procedure of the highly important category.

Paper No	Introduction		Conclusion		Total	
	No of words	No of sentences	No of words	No of sentences	No of words	No of sentences
Paper 1	427	18	865	6	1292	24
Paper 2	103	11	123	13	226	24
Paper 3	300	15	249	8	549	23
Paper 4	427	18	343	22	770	40
Paper 5	596	34	277	16	873	50

Table 6.4 Size of the introductions and conclusions

4.2 Consistency of Selection Criteria

A closer investigation of the sentences in the three categories (high, medium, low/zero) reveals that INFORMEX applied throughout the sample similar rules to similar conditions. A breakdown of the criteria of selection for the high and medium categories for each paper is given in the tables below highlighting the consistency property. Results for the low/zero category are given in Appendix I.

In the category of high importance, most of the sentences were selected because they occupied a privileged position within the section and because the heading of that section was identified as an important heading. These sentences account for 70% in paper 1, 73% in paper 2, 70% in paper 3, 73% in paper 4 and 54% in paper 5. Fewer sentences were selected because of their function, these amount to 35% in paper 1, 36% in paper 2, 30% in paper 3, 27% in paper 4 and 54% in paper 5.

Out of the 5 papers, 46% of the sentences of paper 5 were selected because they contained discourse markers, 20% for paper 3, 12% for paper 1 and 9% for paper 2. No discourse markers were detected by INFORMEX for paper 4. It is important to note that some sentences were selected because they contained specific discourse markers, were functionally relevant as well as occupying privileged positions, as in the case of paper 5, in particular.

In the category of medium importance, most sentences were selected because of their privileged positions. No sentences were identified functionally relevant and paper 5 is the only one whose 25% of its sentences were extracted because they contained discourse markers.

Although emphasis in this research is placed on the first two categories, it is important to mention that INFORMEX was also consistent in attributing low/zero importance to the remaining sentences.

4.3 Usability

The usability requirement states that those sentences identified by INFORMEX as highly important must contain adequate information to generate an abstract.To examine the usability of these sentences, one must select appropriate standards against which to compare the performance of INFORMEX. The same standards used in the previous Chapters for evaluating abstractors' performance will be used. These stipulate that the selected sentences should contain information describing the following elements of an abstract:

1) the purpose of the paper;
2) the methodology or technique used;
3) results or conclusions.

The sentences classified by INFORMEX as highly important will be analysed in the light of the above standards.

4.3.1 Paper 1

Analysis of the content of sentences produced by INFORMEX for this paper reveals that 7 out of the 17 sentences contain information describing the above elements. The purpose of the paper as well as the methodology are stated clearly in sentences 1 and 16. These state that this paper describes a technique called the characteristic error method aiming at capturing the reasoning process involved in conflict resolution. The results of this method are given in sentence 123 whereas the conclusions are highlighted in sentence 126 stressing the unique feature of this approach. These sentences as well as the criteria for their selection are listed in figure 6.5 .

The remaining sentences, which were mainly selected because of their position could be easily disregarded. Three of these sentences (3,7,9) provide a background information to the reader. Sentences 20, 24, 55 and 61 contain additional information about the methodology and describe the distinctive features of the characteristic error approach. Finally, sentence 128 refers to the future plans of the authors.

The above analysis suggests that 41% of the sentences selected by INFORMEX can be used to generate an abstract, 29% contain additional information describing the characteristic error method, 18% provide useful background information to the reader and 12% can be considered as redundant information.

4.3.2 Paper 2

Out of the 11 sentences in paper 2 that belong to the highly important category, 9 sentences provided the relevant information for generating an abstract. These sentences are: 1, 2, 9, 10, 110, 111, 112, 114 and 116. Sentences 1, 111 and 114 state clearly that paper 2 describes a language for representing technical knowledge called COMODEL. Sentences 9 and 10 describes the purpose of this paper which is to show that this approach can be applied to diagnose technical systems and to control technical processes and sentence 3 explains that such an approach is different from the first and well known expert systems developed for medical and scientific applications. Sentence 112 contain the conclusions of the paper confirming the capabilities of COMODEL of representing function and geometric properties of components in a uniform way and sentence 116 claims that COMODEL requires, however, additional heuristic knowledge about symptoms of defects as well as strategies in interpreting diagnosis. Sentence 13 explains that applications of this approach are illustrated by means of examples.

In summary 82% of the sentences extracted by INFORMEX for paper 2 are used to generate the abstract, including the two headings and the title of the paper.

4.3.3 Paper 3

Ten sentences were selected from this paper by INFORMEX as a basis for an abstract. The purpose of this paper which is to describe a program, called CHECK, to help the knowledge engineer examine the consistency and completeness of a knowledge base used in an expert system ,is contained in sentences 75 and 79. CHECK's limitation ,that is, CHECK does not handle certainty factors in the rules is captured by sentence 19. In utilising the LES framework, the program detects potential problems associated with the rules and potential gaps which can be found in the knowledge base. These problems and gaps are found in sentence 75. This program also generates a dependency chart which shows the interaction between the rules and the goals as explained in sentence 79. The distinctive approach of this program is given in sentence 10. The conclusions of this paper are summarised in sentence 79, highlighting the usefulness of this program in building a knowledge base.

In other words, 70% of the sentences identified as highly important contained information directly relevant to the elements of an abstract. The remaining 30% provided background information and future improvements (sentences 3, 11 and 82). One could argue that sentence 3 could be added to the abstract to provide some background information to the paper.

4.3.4 Paper 4

In the case of paper 4, 7 sentences out of 11 contained information that could be used to produce an abstract. The title (sentence 1), here again, stated clearly the content of the paper: it indicated that the paper describes an expert adviser for photolithography. Sentences 2, 3 and 6 introduced the problem with integrated circuit fabrication. The aim of the paper was implicitly stated in sentence 16 when the authors explained that the Adviser is designed to help technicians in solving problems instead of depending on the process engineer. The results of this expert system, found in sentences 20, 87 and 88 showed that the Adviser has been a useful experiment in building an "industrial strength expert system" (sentence 87). It helped design a more thorough testing on a manufacturing line (sentence 20) as well as provide valuable insights about the process (sentence 88). Future enhancements are envisaged (sentence 104) mainly a restructuring of the knowledge representation so that a fault diagnosis shell can be achieved (sentence 108).

What is particularly interesting in this paper is that the sentences produced by INFORMEX are mainly extracted because of their privileged positions as the first or last sentence of an important paragraph. The first sentence introduced the thematic idea of the paragraph and the last sentence captured the conclusion. The sentences in between these privileged sentences developed the theme and provided details for a conclusion or a summary to be stated in the last sentence. Selection of these sentences on the basis of their position in this case seem to provide sufficient information to make up the abstract.

4.3.5 Paper 5

In contrast to paper 4, sentences other than those occupying a privileged position were selected by INFORMEX. With the exception of three sentences (sentences 3, 5 and 145) which provided background information to the paper, all remaining ten sentences included equally important elements which could be used as a basis for an abstract. The title, once more, indicated the topic: being fault diagnosis through responsibility. The purpose of this paper is to describe an approach to functional testing (sentence 12) based on the theory of responsibilities (sentence 130). This approach uses a different formal description and four simple rules of diagnosis (sentence 31). In addition, qualitative reasoning is important to justify the author's approach (sentence 36), which proved to be effective in diagnosing faults in an analog circuit (sentence 135). Although this approach is very powerful, there is still some work to be carried out by the authors (sentences 20 and 142).

4.4 Completeness

The above analysis of the sentences classified as highly important by INFORMEX suggests strongly that these extracts contain sufficient information to generate an abstract. Furthermore, they include additional information which help the reader to place the work undertaken in the paper within the field. The extracts, however require some degree of inference to ensure that the abstract is not only informative but coherent and cohesive.

There are, however, still one question to be resolved. It is whether the information contained in these extracts is complete. The analysis of the results in the previous section is based primarily on those sentences classified as highly important. There may be additional information regarding the aspects of abstracts discussed in the previous section, which may not have been captured by the category of high importance. Consequently, analysis of the sentences enclosed in the remaining categories must be carried out to examine whether any item of information of significant importance to the abstract has been omitted.

4.4.1 Paper 1

In the case of paper 1, those sentences of medium importance belong partly to the introduction, partly to the second section and partly to the conclusion. The introduction part describes other researchers' approach to the problem of conflict resolution in sentences 4, 5, 6 and provides further details of the author's approach in sentences 17, 18 and 19. Sentence 19 is particularly relevant as it explains the distinctive feature about the author's approach, although, one can argue that sentence 20 - which is included in the higher category of importance - has encapsulated the uniqueness of that approach. Sentences 23, 56, 57, 58, 59 and 60 which belong to the section titled "the characteristics error approach", supplies further details of the approach. Finally, the last two sentences 124 and 127 which were extracted from the conclusion, exemplify some aspects of the characteristic error method.

With regard to those sentences which were classified as being of low and zero importance, they amounted to 97 sentences (76% of the paper). They contain detailed description of the characteristic error method and show how this method can be applied to interpret cartographic surface attributes using information obtained from remotely sensed imagery.

4.4.2 Paper 2

Figure 6.11 lists those sentences classified as to be of medium importance. Analysis of the content of these sentences reveals that the first five sentences indicates the position of the current approach among other trends. The remaining three sentences which introduce the COMODEL language, can also be traced in figure 6.6.

Those sentences identified to be of very low or zero relevance to an abstract contain a detailed description of the COMODEL language followed by an example showing the applicability of this approach to represent technical systems. These sentences account for 84% of the total number of sentences in this paper.

4.4.3 Paper 3

The extract in figure 6.12 contain two aspects of information of relevance to abstracting. The first five sentences provide some background information explaining that the program, called CHECK, which is the topic of the paper, is a debugging tool and an enhancement to the previously developed system, LES. The remaining nine sentences list the capabilities of CHECK. The content of these sentences, however, was already covered by those listed in figure 6.7. Although sentences 6 and 7 explain the relationship of CHECK to LES much more clearly than those selected in figure 6.7, they, nevertheless, add no new information.

The aim of CHECK is to detect potential problems in developing the rules and the knowledge base of an expert system. A description of how CHECK diagnoses these problems is given in those sentences labeled low or zero importance.

4.4.4 Paper 4

Figure 6.13 lists those sentences identified as of medium importance. The first half of the paper explains the problems related to photolithography, whereas the second half evaluates the achievements of the expert adviser. Although these sentences contain salient information for an abstraction they do not provide any new item worth of inclusion in an abstract.

Sentences which are classified as low or zero importance were mainly derived from the middle sections of this paper. One section is concerned with the implementation of the expert adviser. The second section lists some of the benefits of this system and evaluates its performance. The performance, however, is closely linked to the user interface rather than the usefulness of the system.

4.4.5 Paper 5

Two sentences (17 and 134) out of 7 from the medium category merit careful evaluation. The purpose of this paper is clearly stated in sentence 17 of figure 6.14. This sentence was selected for two reasons: because it began with a theme marker "in this paper", and because it belonged to the section labeled "introduction". Sentence 134 is the second sentence which was selected because of its position. This sentence states that the work discussed in this paper starts from the principles that an electronic engineer would use to describe how each sub-circuit is built. These two sentences should be moved to the high category.

The category which contains sentences of low and zero importance consists of a detailed description of the authors' approach to the use of structure for diagnosing faults as well as to the assignment of responsibilities to the faulted component(s). Such detailed information is not required for abstracting purposes.

Sent. No.	Criteria of selection	Sentence
1	title	The Characteristic Error Approach to Conflict Resolution.
2	heading, introduction	Introduction.
3	position	Conflict resolution is the integration of alternate opinions about the state of the world into a single coherent world view.
7	position	Their approach to conflict resolution therefore consists of evaluating the classification errors which are most likely to occur in the given context.
9	methodology, theme_marker	Most of this work takes a numeric approach to conflict resolution using, for example, Bayesian or Dempster-Schaefer techniques.
16	methodology, objectives, theme_marker	This paper describes a technique for conflict resolution (termed the characteristic error method) whose goal is to capture the reasoning process behind this form of conflict resolution.
20	position	In other words, the knowledge sources themselves are assumed to provide all of the necessary information for the system to successfully resolve the conflict (given the local context).
21	heading, methodology	The characteristic error approach.
22	position	The term characteristic error was chosen with the realization that certain error conditions are 'characteristic' of the environment in which the knowledge source expects to operate.
24	position	This is possible because information about characteristic error classes enables the resolution of a specific conflict to be dependent upon the immediate context of the problem.
55	position	The items listed above serve as basic information for knowledge (rule) based management of the conflict.
61	position	Strategy setting is implemented by moving error terms from characteristic to uncharacteristic or neutral, thereby altering the source's a priori opinion toward this error type.
122	heading, conclusion	Conclusions/Future work.
123	position	The characteristic error paradigm allows for other types of contextural and programmatic information to be easily added to the system.
125	position	These actions alter the relative 'push' given to a particular classification and can substantially alter the final characteristics of the classified image.
126	position, theme_marker	This paper presents a unique means of combining the opinions of divergent experts in a uniform manner using only their own a priori knowledge about their own characteristics.
128	position	It is planned that the upgraded MSIAS will serve as the testbed for this technique, after it is given the ability to evaluate structural information.

Figure 6.5 Paper1: sentences of high importance.

Sentence No.	Criteria of selection	Sentence
1	title	COMODEL: a language for the representation of technical knowledge.
2	heading, introduction	Introduction.
3	position	The first and most well-known expert systems were developed for medical and scientific applications.
9	position	As possible applications are considered diagnosis and prognosis in technical systems and control of technical processes.
10	position, methodology, theme marker	In our approach we stress attention to the system theoretical principles of knowledge modelling and on the representation of geometric knowledge.
13	position	Its application will be illustrated by some examples (section 4).
110	heading, conclusion	Conclusion.
111	position	COMODEL is a component oriented approach to the representation of knowledge about technical systems.
112	position	In our description of COMODEL we have attached importance to its ability to represent function and geometric properties of components in a uniform way.
114	position	For this purpose a COMODEL description
116	position	At this point additional heuristic knowledge about symptoms of defects as well as strategies for the search for interpretation and diagnosis is needed.

Figure 6.6 Paper2: sentences of high importance

Sentence No.	Criteria of selection	Sentence
1	title	Checking an expert systems knowledge base for consistency and completeness.
2	heading, introduction	Introduction.
3	position	The Lockheed Expert System (LES) is a generic rule-based expert system tool [1] (similar to EMYCIN [2]) that has been used as a framework to construct expert systems in many areas such as electronic equipment diagnosis, design checking, photo interpretation, and hazard analysis.
10	position	CHECK statically analyzes the knowledge base (i.e., after the rules and facts are loaded into the knowledge base), unlike TEIRESIAS, which performs an assessment of rules in the setting of a problem-solving session [3].
11	position	Suwa, Scott, and Shortliffe [4] have written a program for verifying that a set of rules comprehensively spans the knowledge base.
17	position	CHECK does not take into account certainty factors when checking the rule base.
74	heading, conclusion	Summary.
75	position, theme-marker	In this paper we described a program called CHECK whose function is to detect four potential problems (redundant rules, conflicting rules, subsumed rules, and circular rules) and three potential gaps (missing rules, unreachable clauses, and deadend clauses) in a knowledge base utilizing the LES framework.
79	position	Finally, as a by-product of the rule checking processing, CHECK generates a dependency chart which shows how the rules couple and interact with each other and with the goals; this chart should help the knowledge engineer to identify immediately the effects of deleting, adding, or modifying rules.
82	position, theme-marker	The major area of improvement for CHECK is the handling of certainty factors in the rules since LES allows the rules to have certainty factors associated with them; this may require the definitions for the seven conditions covered in this paper to be revised.

Figure 6.7 Paper3: sentences of high importance.

Sentence No.	Criteria of selection	Sentence
1	title	An Expert Advisor For Photolithography.
2	heading, introduction	Introduction.
3	position	An integrated circuit fabrication line, like many manufacturing lines, is organized around a sequence of stations mamned by technicians.
6	position	Wafers continue to be processed at other stations, causing a stockpile to build while the process engineer is called in to fire fight (diagnose and correct the problem).
16	position	We designed the Advisor to mimic the problem-solving behavior of human experts so line technicians would feel confident that the program could help them and they would want to use it instead of depending on the process engineer.
20	position	The results of that test helped us design a more thorough testing, in progress, on a manufacturing line.
86	heading, conclusion	Conclusions.
87	position	Developing the Photolithography Advisor has been a useful experiment in building an industrial strength expert system.
88	position	As with other such systems, the experiment may never be completely finished, but some valuable insights about the process have already been gained.
104	position	We are enhancing the Advisor in several ways.
108	position	We are restructuring the knowledge representation to better separate the problem-solving knowledge (and control) from the domain-specific knowledge so that a fault-diagnosis shell can be extracted.

Figure 6.8 Paper4: sentences of high importance.

Sentence No.	Criteria of selection	Sentence
1	title	Fault diagnosis through responsibility.
2	heading, introduction	Introduction.
3	position	Testing and fault diagnosis of printed circuit cards is a very important task which is done many times each day.
5	position	As a result, the testing tends to be inefficient, missing many faulted components and can often isolate faults to only a large group of components.
12	methodology, theme marker	In this paper, our approach to functional testing is described.
20	methodology, theme marker	Although it is agreed that this is the most desirable approach, the author feels that there is too much work still to be done in order to use this approach in testing today.
31	methodology, theme marker	In our approach, we use a different formal description and four simple rules of diagnosis, rather than compiling the system into production rules.
36	position	Qualitative reasoning (Forbus 1981) is important to justify some of our rules and approaches, but since we are working from second principles, many of these results are compiled into the descriptions.
129	heading, conclusion	Conclusion.
130	position, theme marker	In this paper, the 'theory of responsibilities' has been outlined.
135	theme_marker, methodology	This approach has proved to be very effective in diagnosing faults in analog circuits.
142	position	It should be possible to derive our second principles from first principles.
145	position, theme marker	A more detailed description of this work is in (Milne 1985).

Figure 6.9 Paper5: sentences of high importance.

Sentence No.	Criteria of selection	Sentence
4	position	This operation is often performed by a separate entity (knowledge source) specifically designed for this purpose.
5	position	For example, a system which performs surface material classification in a remote sensing application may encounter conflicts when knowledge sources have differing opinions about the most appropriate classification of a given region.
6	position	Remote sensing experts attempting to resolve these conflicts may employ such knowledge as 'the spectral signature of a road may have been shifted due to the presence of vehicles or an oil slick on the surface' or 'a portion of the road may have been obscured by trees'.
17	position	Characteristic error conflict resolution integrates two closely related concepts: reasoning about uncertainty and constraint propagation.
18	position	Reasoning about uncertainty determines the relative validity of beliefs, while constraint propagation uses the presence of contradictions to restrict the range of applicable choices.
19	position	The characteristic error approach is unique in treating each individual knowledge source as a separate entity whose overall reasoning validity may be entirely selfdetermined.
23	position	Conflict resolution based upon the characteristic error paradigm uses knowledge based semantic information, that is, information which is normally tacitly assumed by the expert during the knowledge acquisition process both to develop the context for a given situation and to resolve differences of opinion.
56	position	However, given the existence of a central (top level) conflict resolving system, additional rule based manipulation is possible.
57	position	The rules used for this manipulation/management of the conflict will be specific to a given system, and perhaps even a specific instance of its use.
58	position	It should be noted that these manipulations require no overall ranking of the knowledge sources in the system.
59	position	Their general thrust is to drive the system results in a specific direction thereby establishing a strategy for the system.
60	position	This strategy embodies more general criteria i.e. always err on the side of caution.
124	position	For example, if it were very important to catch the existence of highways in a particular application, we would move more classes into the highway grower's 'characteristic' list.
127	position	The characteristic error method is a planned enhancement to the MSIAS system.

Figure 6.10 Paper1: sentences of medium importance.

Sentence No.	Criteria of selection	Sentence
4	position	Only since a few years technical applications came in sight.
5	position	This type of application requires in first place new types of knowledge representation.
6	position	Thus, a number of papers appeared on this topic following the ideas of the basic work of Hayes.
7	position	They deal mainly with the modelling of physical processes going on in technical systems.
8	position	All of them reason about physical processes making use of a qualitative description of physical entities.
11	position	On the basis of some principles of the modelling of technical systems (section 2), we give an introduction to the knowledge representation language COMODEL (section 3).
12	position	Its main feature is the component oriented description of technical systems and processes covering geometrical and process knowledge by the same means of description.
115	position	This model together with process signals should be processed by an inference engine.

Figure 6.11 Paper2: sentences of medium importance

Sentence No.	Criteria of selection	Sentence
4	position	LES employs a combination of goal-driven and data-driven rules with the latter being attached to the factual database (demons).
5	position	One objective in the design of LES was to make it easy to use.
6	position,	Thus,many debugging tools and aids were added to the LES program.
7	position	One of these aids is the knowledge base completess and consistency verification program called CHECK.
8	position	Its purpose is to help a knowledge engineer check the knowledge base which he constructed for logically redundant rules, conflicting rules, subsumed rules, missing rules, unreachable clauses, and deadend clauses.
9	position	CHECK does not perform any syntax checking on the rules, since this is done automatically when the rule files are loaded into the knowledge base.
12	position	This program was devised and tested within the context of the ONCOCIN system (an EMYCIN-like system).
13	position	Our work differs from theirs in that CHECK includes unreachable clauses and deadend clauses as two additional rule checking criteria.
14	position	Furthermore, CHECK produces a dependency chart and detects any circular rule chains.
15	position	Also, CHECK was devised and tested on a generic expert system with case-grammar rules and a 'frame' database.
16	position	It has been used to analyze a wide variety of knowledge bases.
76	position	We applied the consistency and completeness verification method of Suwa, Scott, and Shortliffe [4] to the generic expert system LES with good results.
77	position	Furthermore, we have extended the checking to include circular rules, unreachable clauses, and deadend clauses.
78	position	We also showed a general algorithm which performs the checking function efficiently.

Figure 6.12 Paper3: sentences of medium importance.

Sentence No.	Criteria of selection	Sentence
4	position	Most stations process a wafer once or twice before it is finished, but for the photolithography step (placing a pattern on the wafer similar to photography) each wafer is processed up to 15 times.
5	position	When a photolithography problem arises, this step becomes the bottleneck for the whole line.
17	position	This involved not only capturing in rules the knowledge of several photolithography engineers and the field test experience of technicians, but also a lot of effort in implementimg an interface that presents a familiar model of work; e.g., by using electronic versions of the paper forms previously used for process recordkeeping.
18	position	How well we have succeeded is still being evaluated.
19	position	We have done a preliminary evaluation by field testing on a research line.
105	position	The knowledge-base is now being extended to include positive resist processes, which are gaining popularity for fabricating smaller patterns on wafers.
106	position	The user interface is being redesigned to better reflect our model of the program as a team of cooperating specialists using graphics, animation, and video display media.
107	position	The problem-solving strategy used by the program is really quite generic to a wide variety of fault-diagnosis problems important to us.

Figure 6.13 Paper4: sentences of medium importance.

Sentence No.	Criteria of selection	Sentence
4	position	Typically this is performed using pre-determined, static and rigidly structured tests.
17	theme marker	In this paper, we will show how structural information can be used to further propose faults.
131	position	This states that we can perform fault diagnosis by assigning responsibilities for parts of the output to various parts of the circuit.
132	position	These responsibilities are derived from a casual simulation of the function of the circuit.
133	position	It has also demonstrated how this works on a theme marker typical building block circuit.
134	position	In our work we have started from 'second principles', that is, the principles which an electronic engineer would use to describes each subcircuit.
143	position	It should be possible to derive our second principles from first principles.
144	position	We have found that we can generally prove the second principles in this way, but we can't do it automatically, yet.

Figure 6.14 Paper5: sentences of medium importance.

5. Linguistic Issues

The previous section analysed the extracts produced by INFORMEX in the light of their consistency, usability and completeness. The purpose was to examine the performance of INFORMEX and the viability of the rule base system approach to abstracting.

With regard to the corpus used to test the performance of INFORMEX, the above sections showed that the classification of sentences was consistently and successfully achieved, and that, on average, 67% of those sentences identified as highly important contained appropriate information for inclusion in an abstract. This suggests that the rules acquired from experts have provided a valuable departure point to the selection of important information for generating an abstract.

The abstracts listing sentences of high importance to be used as a basis for abstracting must also be among the objects of analysis. These extracts are communicating knowledge to readers and consequently interact with their world knowledge. For these extracts to be considered as texts in their own right, they must meet the standards of textuality which according to de Beaugrande consist of seven standards namely cohesion, coherence, intentionality, acceptability, informativity, situationality and intertextuality [de Beaugrande et al, 1981]. While cohesion and coherence are notions which focus on the text material being analysed, the other standards are notions centred on readers and producers of the text material. One can argue that the standards of acceptability and informativity can be linked to the standards of usability and completeness discussed in the earlier sections. The underlying notion here is applied to the content of the extract. Informativity is understood to "designate the extent to which a presentation is new or unexpected for the receivers" using de Beaugrande terminology [de Beaugrande et al 1981:139]. In the context of an abstract of a scientific text, the new information is captured by the elements of an abstract. The standard of intentionality subsumes the intentions of the author(s) of the paper, whereas the standard of situationality is used to designate the factors which make a text relevant to a communicative situation. Intertextuality, on the other hand, is concerned with the factors which make the understanding of one text dependent upon knowledge of one or more previously encountered texts [de Beaugrande et al, 1981: 10]. These three standards, intentionality, situationality and intertextuality, are linked to the attitudes of text producers and readers. As the focus of this section is on the extracts themselves, the cohesion and coherence standards are the most appropriate measures of textuality as they indicate how the elements of the texts fit together and make sense. Cohesion concerns the ways in which elements of the surface text are "mutually connected within a sequence" [de Beaugrande et al, 1981: 3] whereas coherence demands that the elements are "mutually accessible and relevant" [de Beaugrande et al, 1981: 4].

In the following sections, those sentences which were identified directly relevant to abstracting will be analysed to examine the cohesion and coherence of their conceptual constituents. The initial step towards investigating these standards is to use de Beaugrande's text world model to represent these constituents and to show the way these are connected to each other. The nodes capture the conceptual constituents whereas the links have labels specifying the directions of connections.

5.1. Extract 1

The extract for paper 1 is given in figure 6.15, and its corresponding representation in figure 6.16. There are three sentences with no explicit cohesive devices to the rest of the abstract, two of which consist of headings. To bind these sentences, it requires some inference on the behalf of the reader. The first inference is to link the two constituents, "this paper describes" and "this paper presents" together and then connect the two headings to these constituents using the label "of". To link sentence 123 to the diagram one must set the verb "allow" as a link and then apply a semantic transformation to map the two constituents, i.e. the characteristic error approach and the characteristic error paradigm together. Since sentence 1 is identified as being the title, one can connect the constituent "this paper" to the constituent " the characteristic error approach" using the link "is about". Similarly the sentences following a heading can be attributed to that heading until another one is encountered. Such inferences must be carried out to produce a cohesive extract. This representation indicates that the central constituent of the paper is about "the characteristic error approach" which describes a technique for conflict resolution and presents a unique means of combining the opinions of divergent experts. Features and capabilities of this approach are also shown in this model. In other words, figure 6.16 indicates that the outcome does make sense.

The Characteristic Error Approach to Conflict Resolution.

Introduction.

This paper describes a technique for conflict resolution (termed the characteristic error method) whose goal is to capture the reasoning process behind this form of conflict resolution.
The characteristic error approach.
Conclusions/Future work.
The characteristic error paradigm allows for other types of contextual and programmatic information to be easily added to the system. This paper presents a unique means of combining the opinions of divergent experts in a uniform manner using only their own a priori knowledge about their own characteristics.

Figure 6.15 Extract 1

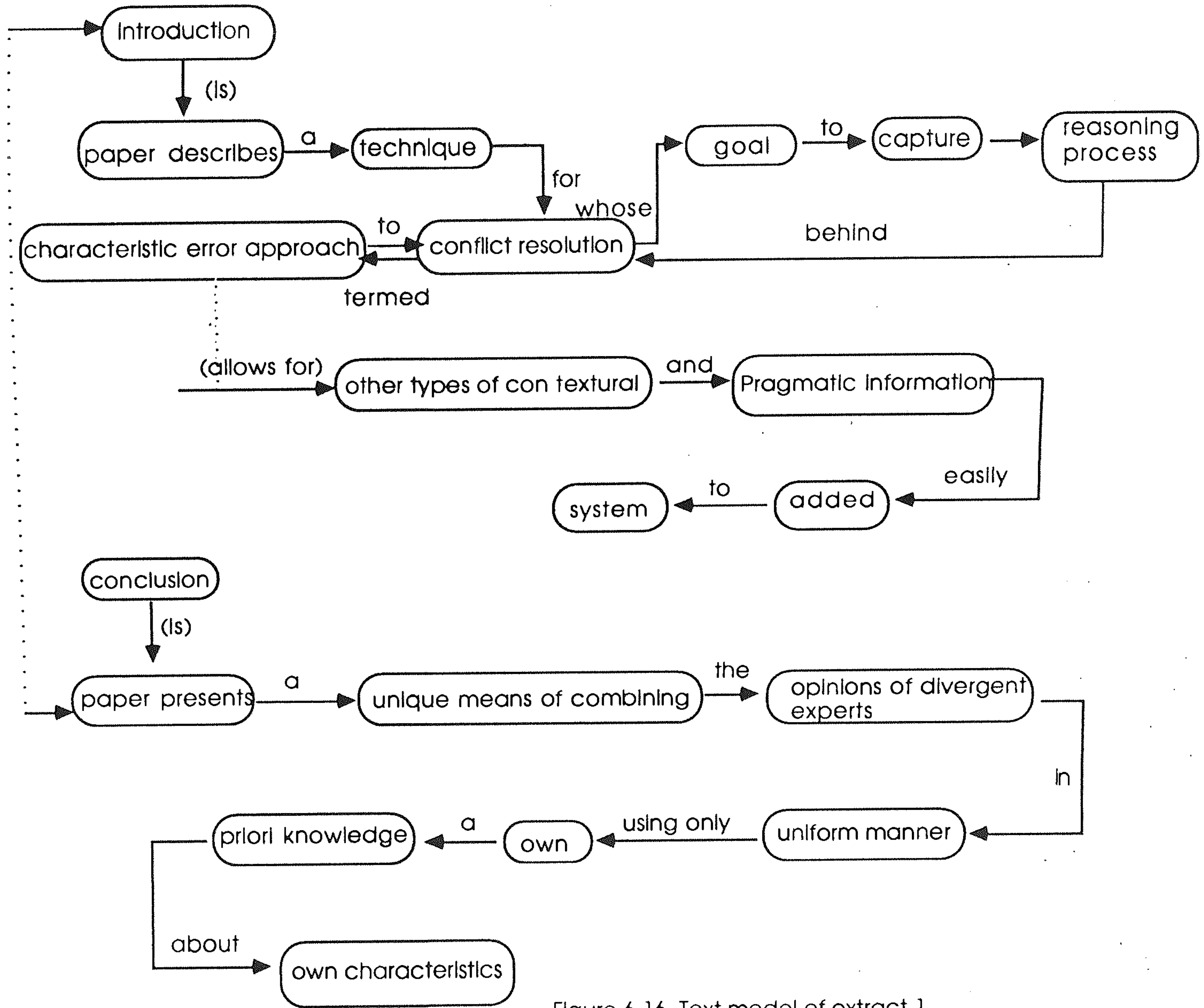


Figure 6.16 Text model of extract 1

5.2 Extract 2

Figure 6.18 illustrates the text world model for the extract of paper 2. This extract represented in figure 6.17 shows no clear cohesive devices for signalling the relationships between its sentences. To connect these sentences together, one has to rely on the conceptual constituent, COMODEL, which occurs in most sentences in different forms. Thus sentence 111 can be linked to sentences 112 and 114 as they all refer to the description of COMODEL. These three sentences can be in turn connected to sentences 1 and 111 using the constituent COMODEL as the link. One has also to infer that the constituent, "as possible applications" refers to the COMODEL approach, thus linking sentence 9 to the model. The two headings captured by the two sentences 2 and 110 can again be linked to the subsequent sentences using the same inference as in the case of extract 1. This link enables sentence 116 to be considered as another statement of conclusion to COMODEL.

COMODEL: a language for the representation of technical knowledge.

Introduction.

As possible applications are considered diagnosis and prognosis in technical systems and control of technical processes.
In our approach we stress attention to the system theoretical principles of knowledge modelling and on the representation of geometric knowledge.
Conclusion.
COMODEL is a component oriented approach to the representation of knowledge about technical systems.
In our description of COMODEL we have attached importance to its ability to represent function and geometric properties of components in a uniform way.
For this purpose a COMODEL description represents a model of a technical system.
At this point additional heuristic knowledge about symptoms of defects as well as strategies for the search for interpretation and diagnosis is needed.

Figure 6.17 Extract 2

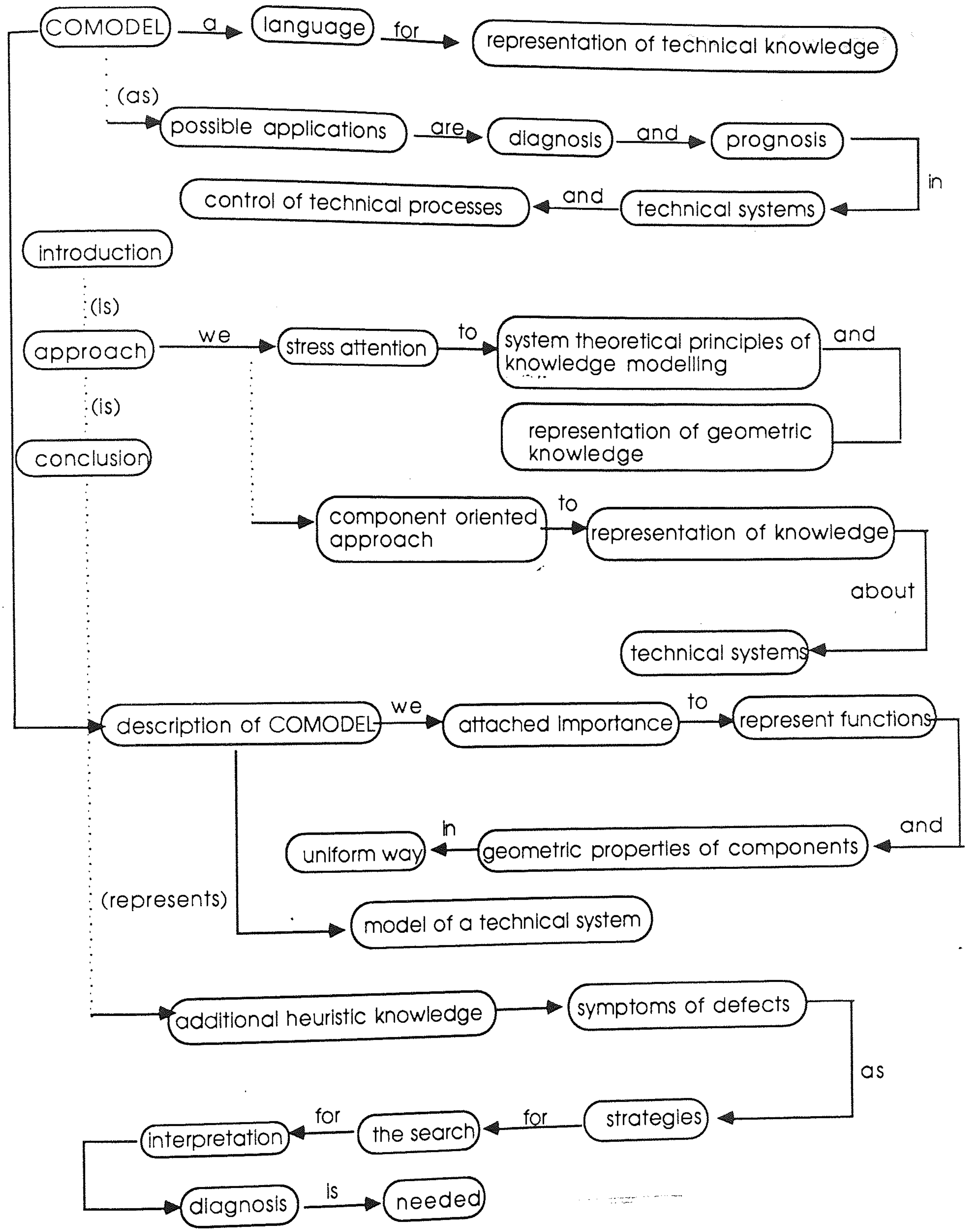


Figure 6.18 Text model of extract 2

The above set of inferences are required to achieve coherence of extract 2. The central conceptual constituent is COMODEL, to which a set of links are assigned, showing namely the possible applications, the main stress of this approach and its features.

5.3 Extract 3

The model for paper 3 which is illustrated in figure 6.20, shows that the central conceptual constituent is the constituent "CHECK". "CHECK" acts a cohesive device linking sentences 10, 17, 75 and 79. Here again, sentences 2 and 74 can be linked to the constituent "this paper". Connecting the title to the diagram can only be resolved if one infers that the title is a generalised statement of sentence 75. This inference, however, requires an understanding of the structure of an expert system to infer the relevance of issues such as completeness and consistency of rules in knowledge base and hence the function of CHECK. To reconstitute the coherence of this extract, the reader is more likely to try to fit the sentences of figure 6.19 together rather than work with the cohesive relationships present in these sentences.

Checking an expert system knowledge base for consistency and completeness.

Introduction.

CHECK statically analyzes the knowledge base (i.e., after the rules and facts are loaded into the knowledge base), unlike TEIRESIAS, which performs an assessment of rules in the setting of a problem-solving session [3]. CHECK does not take into account certainty factors when checking the rule base.

Summary.

In this paper we described a program called CHECK whose function is to detect four potential problems (redundant rules, conflicting rules, subsumed rules, and circular rules) and three potential gaps (missing rules, unreachable clauses, and deadend clauses) in a knowledge base utilizing the LES framework. Finally, as a by-product of the rule checking processing, CHECK generates a dependency chart which shows how the rules couple and interact with each other and with the goals; this chart should help the knowledge engineer to identify immediately the effects of deleting, adding, or modifying rules.

Figure 6.19 Extract 3

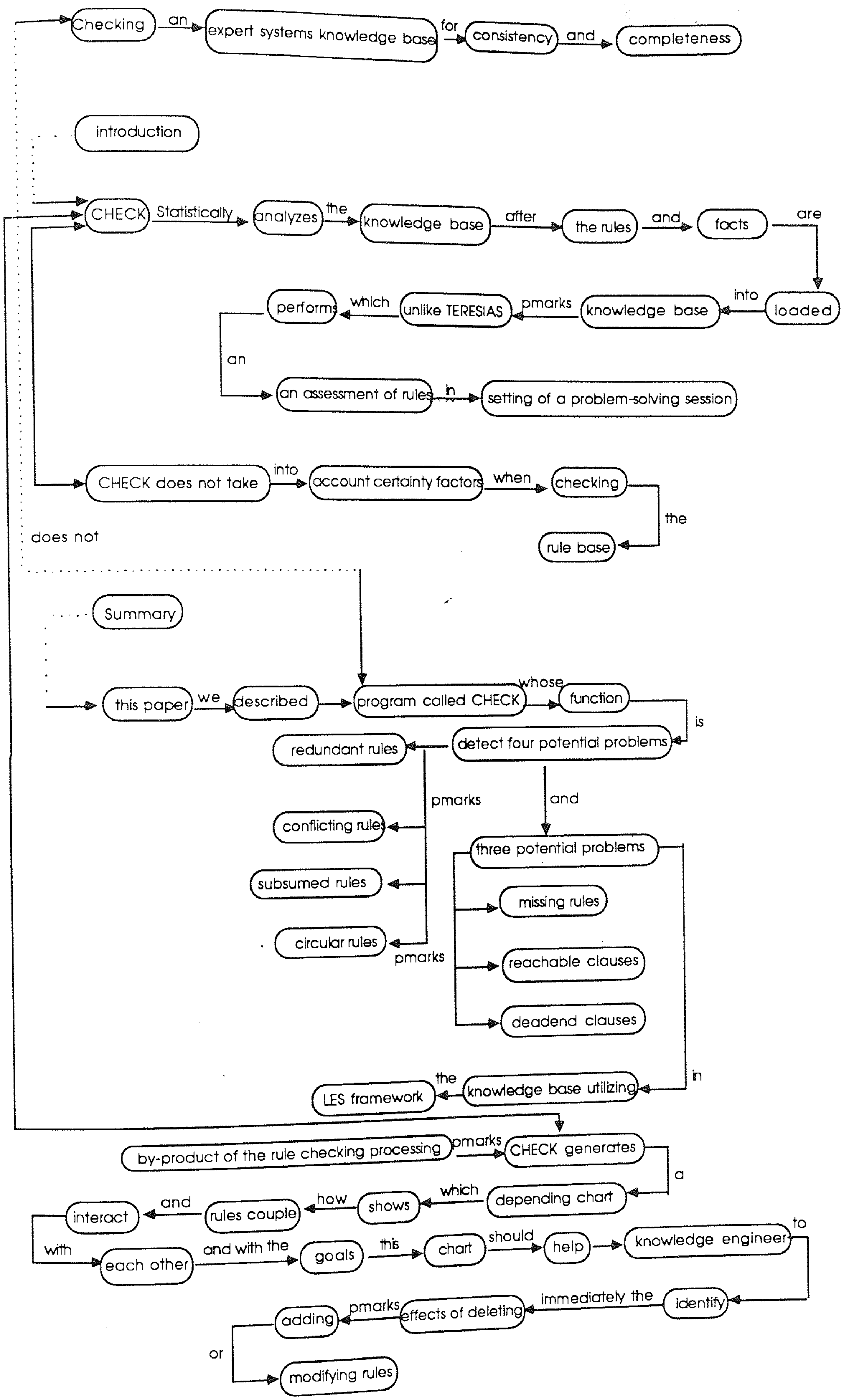


Figure 6.20 Text model of extract 3

5.4 Extract 4

On reading through extract 4 of paper 4 the reader may succeed in making connections across all sentences to the exception of sentence 20 (figure 6.21) . In the original paper this sentence refers to results of a test to evaluate the success of the expert adviser system. Although this sentence does not have clear cohesive ties with the extract the reader can infer that it relates to sentence 16 because of the constituents "design" common to both sentences, and the relationship between the constituent "technicians" and "manufacturing line". Similarly, the constituent "experiment" bridges the gap between sentence 87 and 88. The cohesion of these last two sentences stand out more clearly because they follow each other. The constituent "conclusion" in this extract could provide a more cohesive structure if it occurred before sentence 20.

An Expert Adviser For Photolithography.

Introduction.

We designed the Adviser to mimic the problem-solving behaviour of human experts so line technicians would feel confident that the program could help them and they would want to use it instead of depending on the process engineer.
The results of that test helped us design a more thorough testing, in progress, on a manufacturing line.
Conclusions.
Developing the Photolithography Adviser has been a useful experiment in building an industrial strength expert system.
As with other such systems, the experiment may never be completely finished, but some valuable insights about the process have already been gained.

Figure 6.21 Extract 4

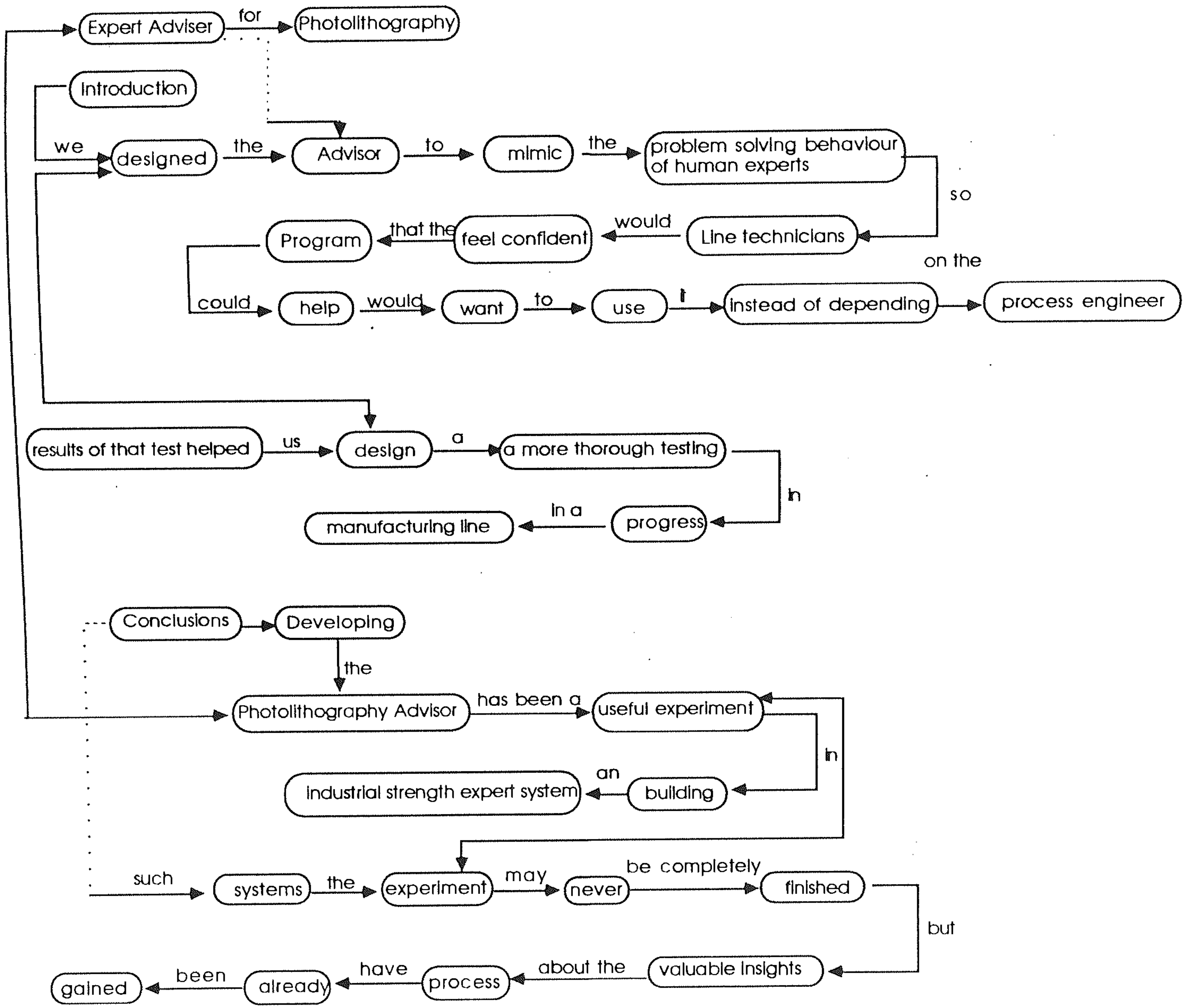


Figure 6.22 Text model of extract 4

5.5 Extract 5

The cohesion and coherence of this extract (figure 6.23), is maintained through the frequent reference to the two constituents, "this paper" and "the approach", thus binding sentences 12, 20, 31, 36, 130 and 135 together. Furthermore, sentence 130 bridges the gap between sentence 1 and 135. Finally, the constituent "second principles" binds sentence 142 to sentence 36 of the extract. The cohesion ties are illustrated in figure 6.24. The headings given in sentences 2 and 129 can be linked easily to the constituent "this paper" which follows on.

> Fault diagnosis through responsibility.
>
> Introduction.
>
> In this paper our approach to functional testing is described.
> Although it is agreed that this is the most desirable approach, the author feels that there is too much work still to be done in order to use this approach in testing today. In our approach, we use a different formal description and four simple rules of diagnosis, rather than compiling the system into a production rules.
> Qualitative reasoning (Forbus 1981) is important to justify some of our rules and approaches, but since we are working from second principles, many of these results are compiled into the descriptions.
> Conclusion.
> In this paper, the 'theory of responsibilities' has been outlined.
> This approach has proved to be very effective in diagnosing faults in analog theme circuits.
> It should be possible to derive our second principles from first principles.

Figure 6.23 Extract 5

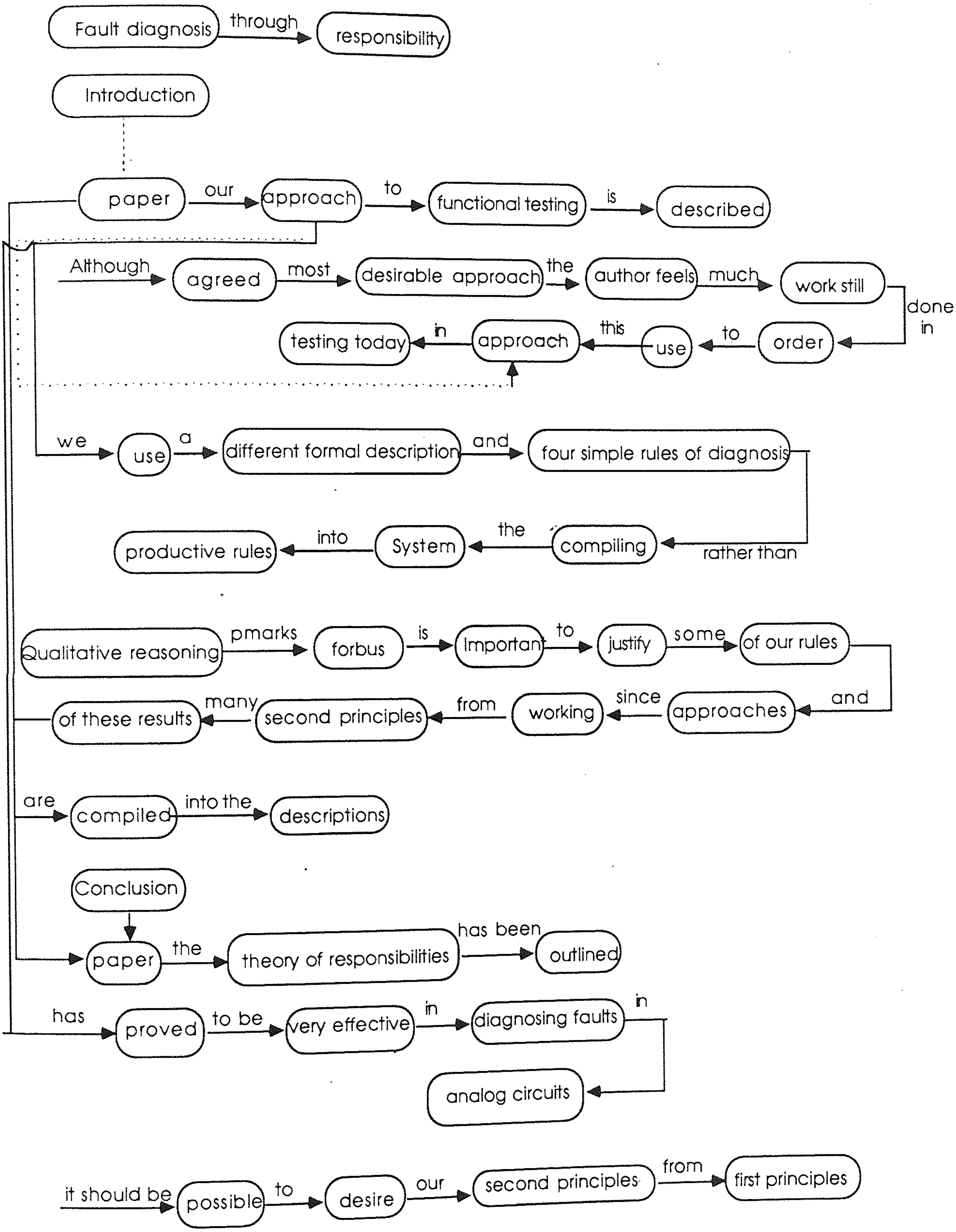


Figure 6.24 Text model of extract 5

5.6 Summary

Each of the above extracts consists of sentences which were selected from the original paper according to criteria of relevance to abstracting. The process of selecting sentences from a paper does not automatically imply that the sentences can be combined into a cohesive and coherent passage. The analysis above, however, suggests that, on the whole, the cohesion within each extract was achieved in two aspects. In some cases, it was derived from constituent collocability, that is, by relating the Adviser to the Expert Adviser, and rules to the knowledge base). In other cases, cohesion is achieved either by means of lexical repetition in constituents (e.g this paper describes, this paper presents; CHECK, a program called CHECK; our description of COMODEL, a COMODEL description; the characteristic error approach, the characteristic error paradigm), or by reference to the same constituent (e.g this paper, this approach) or by stylistic choice (e.g in our approach vs this approach). Sometimes, cohesion was provided by semantic relations associated within the extracts but not explicitly signaled in the constituent. For example, the issues of completeness and consistency in the extract 3 were associated with the rules and clauses that make up a knowledge base . The verb "allow" in sentence 123 of extract 1 is interpreted as an enhancement and a conclusion of the characteristic error approach, thus providing cohesive ties with the other sentences. In the same way, the headings within the five extracts are associated with the two constituents, the paper and the approach.

It is interesting to note that out of the five extracts, extract 4 is the only extract which contained a pronominal expression (the results of that test) whose co-reference could not be identified without going back to the original paper or searching for clues in the constituents within that sentence.

So far, five extracts have revealed some cohesive structuring either of a syntactic or semantic kind. When reading these extracts, the reader is capable of identifying the central constituent around which the main themes seem to emerge. In the first extract, the sentences are linked to two constituents, 'this paper' and 'the characteristic error approach' (see figure 6.16). These two constituents are also connected to each other in sentence 16. The conceptual constituent, COMODEL, appears clearly as the central theme in figure 6.18. Similarly, the constituent, CHECK, appears as the main constituent for extract 3 (see figure 6.20). The Expert Adviser for photolithography are two constituents that appear in most sentences providing extract 4 with a coherent structure (figure 6.22). Finally, the pronominal expression in the constituent "this approach", in sentence 135, established the connection with the constituent, the theory of responsibilities, thus making clear that the approach discussed is diagnosing faults through this theory. The identification of the central theme in each extract as well as the ability to find cohesive devices across the sentences allow the reader to obtain a coherent interpretation of the whole extract.

It is also important to note that the title and the headings provide a useful point of departure for interpreting these abstracts. The title is usually assumed to carry thematic information in a text. This is particularly true in the five extracts: the title captured the main theme and served as an anchor point in the reader's memory around which the incoming sentences were interpreted. These findings are consistent with Kozminsky studies which showed that titles act as advance organisers and serve as pointers which make contact with previous knowledge in memory, thus facilitating comprehension [Kozminski, 1977: 482]. The headings, and in particular the conclusion provided a base which connected the subsequent sentences to the rest of the abstract, and achieved coherence and cohesion of the extract.

The production of these extracts may have been achieved without the use of world knowledge, the understanding of what is being conveyed by these extracts, however, requires the reader to make use of a wide range of possible inferences. This is a reasonable idea since, when authors write a paper, they assume a certain amount of background knowledge and domain knowledge on behalf of their readers.

6. Discussion

The previous sections analysed the results of INFORMEX from two perspectives. The first evaluated the extracts from an artificial intelligence stand-point, focussing on issues related to consistency of the rules used by INFORMEX, the completeness of the knowledge base and the usability of the output. The second perspective focussed on the informativity, cohesion and coherence of these extracts, which are important issues if these extracts are to be acknowledged as linguistic entities. This section discusses the performance of INFORMEX in a wider context by considering the following questions:

1. Can the performance of INFORMEX be improved, if so which components of INFORMEX should be modified?

2. How specific or universal are the rules employed by INFORMEX?

3. What benefits and disadvantages are derived from adopting a rule base system approach to INFORMEX?

6.1 Overall Performance of INFORMEX

The analysis of the extracts revealed that 67% of those sentences belonging to the category of high importance contained information describing either the purpose of the paper, the technique/approach suggested by the author(s) or its results/conclusions. The remaining 33% provided some background information to the methodology discussed in the paper. The main question that follows this analysis is whether INFORMEX can extract only those sentences that contain specific abstract elements, thereby optimising performance. INFORMEX conists of two major modules. The textual analyser module interprets the structure and the content of the paper so that the expert classifyer module can apply its rules which are partly related to reading strategies, partly derived from the short-cut strategy and partly associated with the degree of relevance of a sentence to abstracting. Consequently, to improve the performance of INFORMEX, one must extend the capabilities of these two modules.

In terms of the textual analyser, there are a number of enhancements that can be introduced. Currently, the functional analyser scans each sentence for a limited set of discourse markers, called theme-markers, identified as of significance to abstracting. This set could be extended to include markers of conceptual relevance to scientific papers (see table 4.3). Also it uses a fairly rudimentary approach in classifying sentences into the seven categories (introduction, objectives, methodology, discussion, conclusion, results, investigation). This approach can be improved by scanning for more complex patterns such as those given below when analysing the function of a sentence.

1. "We also showed a general algorithm which performs the checking function efficiently" [Nguyen et al, 1985: 378].

2. "In our description of COMODEL we have attached importance to its ability to represent function and geometric properties of components in a uniform way" [Dilger et al, 1985: 358].

3. "We are enhancing the Adviser in several ways." [Cline et al, 1985: 413].

The functional analyser will need to scan the sentence for relevant cue phrases such as "we showed" and "algorithm" in sentence 1, to infer the association between these phrases and the intended meaning of the author. Similarly, the phrase "we have attached importance" suggests that what follows is important according to the authors. The third sentence contains two cue phrases: "we are enhancing" and "in several ways", indicating that the enhancements are given but not enclosed in this sentence. The functional analyser will need to scan the oncoming sentences looking for connective markers (e.g the first, the second; on one hand, on the other) or for aspects that can be determined only if the program has knowledge of the domain of the topic being discussed, as in the case of this paper. The program will require to know that the Adviser is an expert system consisting of, for instance, a knowledge base, a user interface and a problem solving. So as the program reads the oncoming sentences, it scans for features describing the components of an expert system.

A further measure can be introduced to eliminate detailed descriptions by scanning for discourse markers pointing to exemplification. Such markers include the following phrases:

[for example, for instance, an example of, details of this method are given in the next section....].

The last marker, in particular, indicates that an entire section can be omitted. This can be achieved through the structural organisers which keep a record of the layout of a paper. Similarly, subheadings and subdivisions can be deleted, thus reducing computer processing time.

INFORMEX, at present, does not the exploit typographical features of a paper. The typographical features and other general markup of a paper have a special role but were ignored in this research because of the limitations of the current scanner technology on one hand and the lack of standardisation on the other. These features, however, are becoming important as more and more efforts are being channeled towards the production of a standard generalised markup of a document to facilitate electronic publishing and computer processing. Also the scanner technology is evolving rapidly.

Further performance can also be achieved by adding semantic capabilities to INFORMEX, thus optimising the output of the conceptual analyser and linking it to a knowledge base that is domain specific. It will provide a richer semantic environment to INFORMEX. The semantic dimension will, however, take INFORMEX beyond the objectives of the current research.

The above enhancements will equip INFORMEX to select only those sentences that contain relevant requirements for abstracting, thus achieving a better performance. This does not necessarily generate a better cohesion and coherence of these extracted sentences. A better performance in the extraction process may result in a poorer performance of cohesion and coherence. The additional information produced by the current implementation of INFORMEX aid the reader to situate the relevance of the paper with the field. Such information is particularly valuable for those readers who are new to the field or looking for a general overview of the field. By eliminating additional information, the extracts become more specialised and thus restrict the size of the readership.

6.2 Evaluation of the Rules of INFORMEX

The major contribution of INFORMEX is in demonstrating that reasonable abstracts can be produced by using rules devoid of domain knowledge. INFORMEX has no knowledge of the domain of the papers it has processed. It relies heavily on a set of rules which utilises the structure of the paper and the linguistic manifestations of the content of a paper that are significant to abstracting. In addition to these rules, INFORMEX applies reading strategies captured from expert abstractors used in scanning papers for key information. Like the participants in experiment 1, INFORMEX has a very limited understanding of the material being analysed, and yet capable of extracting the main sentences of a paper and judge their relevance with this shallow knowledge.

Another important contribution of INFORMEX is that the rules which make up its knowledge base are closely associated with the scientific papers of expository type. Although the experimental data belong to the field of artificial intelligence, INFORMEX rules are not domain specific. There is strong feeling that INFORMEX could produce an adequate extract for any scientific paper of expository prose. Authors in the scientific community share a common purpose when they report their work, and therefore apply common practices and conventions in organizing the content of their papers. Skilled writers intuitively know how to organise a progression so that the content is processed in such a way that it captures the readers' eye movement, makes use of the look-ahead principle as described by Kieras [Kieras,1978] suggesting the preferential position for the topic sentence at the start of a paragraph and capitalises on the short-cut strategy. Writers and readers have also at their disposal a wide literature which recommend such structuring of scientific papers. In other words, as long as a paper is written according to the above norms, there is a high probability that INFORMEX will be successful in producing a reasonable abstract. This adds a universal dimension to INFORMEX.

6.3 Evaluation of a Rule Base System Approach

The strategies used by abstractors and the linguistic knowledge required to process a paper were captured by INFORMEX in the form of rules, hence the rule base system approach. Such a formalism was chosen because of the modularity feature, in that rules are independent of each other and of the rest of the system, because of the facility for representing heuristic knowledge, and because rules can be added and modified much more easily than any other available artificial techniques, and finally because of its ability to represent declarative type of knowledge as well as procedural knowledge.

In the process of building INFORMEX, these features revealed to be important assets. First, the rules captured the reasoning of the abstractor in a natural way. Abstractors often described their strategies using a terminology close to the format of rules. A common strategy shared by most abstractors was expressed in the following way:

If you can't find the conclusions of the paper in the introduction, read the last section of the paper.

Rules have a readily understandable form which facilitated not only the building up of the system but also the debugging as well as backtracking aspects. This latter aspect is of critical significance when one is dealing with pragmatic knowledge which contain a certain amount of fuzziness and lack formal structure. If at any point the system failed to apply a given rule, it could backtrack to somewhere a previous choice was made and took a different alternative and tried again. It was also possible to monitor step by step the efficiency of the metarules, described in Chapter 5, in optimising the trigger of appropriate rules.

Finally, the choice of PROLOG as a language to implement the rules proved to be successful. Factual knowledge were adequately described using the declarative assertions while procedural knowledge were implemented in rule form. Equally, the language SPITBOL, used a rule base approach to detect theme markers, identify conceptual constituents of a paper and analyse the function of sentences. Both of these languages operate on the basis of pattern matching, a feature of high importance to this study.

The difficulties encountered by this approach are mainly related to processing time. While the textual analyser produced the knowledge base of a paper in matters of minutes, the expert classifier took an average of 45 minutes to produce one extract. This is partly attributed to the size of the text knowledge base required to consult each time a rule was triggered. The other reason is because INFORMEX did not process the text knowledge base in a linear fashion, but applied abstractors' technique in concentrating on salient sections of the paper first then the rest of the paper. Although this is not a disadvantage of the rule base system approach, it is a limitation associated with the way PROLOG searches its knowledge base.

Finally, because of the way the rules were structured in INFORMEX and the way that they are fired in PROLOG, it is impossible to determine which strategy was more efficient in producing an extract. Also, because it is unrealistic to test exhaustively each aspect of the rules mainly due to the size of the rules, they cannot be used to evaluate the impact of one rule on the other.

In conclusion, the rule base system approach has a long and respectable history as a technique for representing and manipulating knowledge. It has been used to model successfully human cognitive processes and as problem solving strategies [Ringland et al, 1988: 115]. It has also proved to be a valuable technique to model abstractors' strategies and to represent the structure and the content of the experimental papers used in this study. Thus, its greatest asset is that it can be used as a realistic testbed for human performance.

7. Summary

Evaluating the performance of an expert system is a difficult task. On one hand, because of this approach it is still in its infancy, there are no established measures one can use. On the other hand, because it is used as a tool to test theories and models, and because it is based on heuristics, it becomes difficult to asses whether the results produced by the system or those obtained from human experts are incorrect. What is being evaluated? Who is being the object of analysis and investigation? These can be easily confused. For this reason, the purpose of the expert system must be clearly defined so that evaluation of the results can be usefully assessed.

In this study, the expert system was used as a valuable tool to gain insights into the process of abstracting and to investigate if a simplified approach relying on the short-cut strategy, the reading strategy or abstractors combined with the linguistic properties of a given paper could provide a useful basis to extract key information or relevance to abstracting. The results of INFORMEX suggest strongly that such an approach can produce a reasonable extract which could be used as a basis for generating an abstract. It also suggests that certain facets of abstracting lend themselves successfully to computerisation, and ties up closely with findings from cognitive science that it is possible to select information from a text by using very shallow knowledge.

CHAPTER 7

CONCLUSIONS AND FURTHER WORK

1. Summary of Findings.

The primary objective of this research is to understand what kinds of knowledge and skills people use in "extracting " relevant information from text. The approach adopted here is to concentrate on examining how professional abstractors and experienced researchers extract information from scientific papers to produce an abstract. As the process of abstracting is also the focus of study of many disciplines, namely cognitive science, information science, psycholinguistics and textlinguistics, although from different perspectives and for different purposes, findings from these disciplines were acknowledged and integrated in various ways into this research. An investigation into what language resources writers use in communicating their main ideas in a scientific paper was carried out from the standpoint of linguistics. Studies in cognitive science, as well as guidelines produced by the information science discipline, provided a valuable analysis of reading strategies, in general, and of eye-movement, in particular, to infer important information in texts. Also, the pragmatic approach undertaken by this study to investigate how human experts perform the process of abstracting made it possible to decompose this complex process into five main tasks. The first task is concerned with defining the function of the abstract in terms of its role and its readership. If the abstract, for instance, is to help users assess the relevance of the original paper in the light of their research interests, then an informative abstract is recommended. The content of such an abstract should specify the aim of this paper, the technique or approach suggested by the author(s) and summarise the findings. Determining the relevant content of an abstract is the object of the second task. The task of identifying information in the original paper to make up an abstract draw on techniques related to reading strategies. Skimming is used to get an overview of this paper while scanning aids locating specific information. In scanning, human experts know what they are looking for. The reader's eyes move up and down the page in a zig-zag manner searching for textual and contextual relevance clues in the paper. Some of these may take the form of discourse markers, others may be related to the structure of textual space, for example, the distinctive role of the introduction and conclusions in scientific papers of expository type, the emphasis placed on first paragraphs and topic sentences. These reading strategies are important as they assume an interactive relationship between the authors and readers of a text: readers have a set of expectations when reading a scientific paper, and writers assume a specific audience when they organise their ideas and convey their message in their paper. Furthermore, readers and writers take a reversible role. Readers of these scientific papers are writers and writers are readers. It is in the advantage of both readers and writers to ensure that the communication about their work is successfully achieved and this often leads to the use of signalling devices and column structure of textual space on the behalf of writers to highlight the main ideas and the decoding of linguistic features by readers to extract the main issues of a paper. Thus, identifying and developing an effective reading strategy is the concern of task 3 and applying this strategy to a given paper to extract relevant information is the focus of task 4. According to some abstractors, this task entails the readers' knowledge of the topic in which the paper is written as well their linguistic knowledge. The last task is to transform these extracted chunks of information into a coherent, cohesive and informative abstract. These tasks, although described as individual tasks, must not be regarded as an organised sequence of activities. The skills of an abstractor are reflected in the way he/she integrates these tasks in an efficient, accurate and appropriate way. The abstractor combines language skills which involve the manipulation of the written information in a text, with cognitive skills which relate to the ability of reading, interpreting, reducing and communicating information.

In the process of investigating the abstracting tasks, the interviews with professional abstractors revealed two differing views with regard to the kind of specialised knowledge an abstractor should have. One view claimed that domain knowledge is important for abstracting while the other stresses the importance of skills required to identify important information in a paper. Two major questions were addressed in this research: the first one related to the issue of how important is domain knowledge for abstracting purposes, the second investigated whether an adequate abstract can be produced based on skills rather than domain knowledge to extract important sentences from a paper. These sentences were then analysed to assess whether they could be used as a basis for generating an abstract.

These questions were addressed with the view to find out if a simplified approach to the process of abstracting can be achieved. To this end, three experimental studies were carried out. The first experiment which investigated the relevance of the background knowledge in abstracting, showed that the three participants with no background knowledge as well as the one with little knowledge of computing produced an abstract for a paper on expert systems as adequate as the fifth participant who was an active researcher in this field. The two other experiments were primarily concerned in finding out whether introduction and conclusions of scientific papers contain sufficient information for abstracting and consequently examining the validity of the short-cut strategy suggested by some expert abstracters. The above experiments supported strongly that an adequate abstract can be generated without a deep knowledge of the field of the paper. By relying on a set of empirical strategies, a skilled abstractor, without specific domain knowledge, can read, interpret the content of a paper and extract important information to describe the purpose, the methodology and the findings of the work described by the author(s).

If a reasonable abstract could be based on rather limited domain knowledge, could one model this simplified approach to the process of abstracting using computational formalisms? and more specifically could the current artificial intelligence techniques capture the knowledge and skills used by abstractors. Although these knowledge and skills have been simplified in this thesis, they remain, however, complex and intricate. The final objective of this study is to test this simplified approach to abstracting by designing a computer system to contain knowledge and strategies that are necessary for the behaviour of an abstractor and then executing the system to see if such a system can abstract salient information from a paper. These sentences could then be organised by a human generator to compose the abstract. To this end, an information extraction system, called INFORMEX, was developed using a rule base system formalism to capture the knowledge and skills of expert abstractors. This system, which was implemented partly in SPITBOL, and partly in PROLOG, used a set of heuristic rules to analyse five scientific papers of expository type, to interpret the content in relation to the abstract elements and to extract a set of sentences recognised as relevant for abstracting purposes. The analysis of these extracts revealed that the three key elements of an abstract were contained in these sentences, and hence an adequate abstract could be generated. The cohesion and coherence of these abstracts, however, required a certain amount of inferences on the behalf of the human generator. Some of these inferences could be locally achieved, others necessitated the domain knowledge. Furthermore, INFORMEX showed that a rule base system was a suitable computational model to represent experts knowledge and strategies. This computational technique, and more generally, rule based systems provide the basis for a new approach to the modelling of cognition. It showed how experts tackle the task of abstracting by integrating formal knowledge as well as experiential learning; they take short cuts when they can by making use of whatever information happens to be available (e.g. textual and contextual relevance cues). It also highlighted a number of regularities about scientific expository papers. These regularities help the abstractors assess the status of information in a paper, validate their strategies, and allow the modelling of their behaviour computationally. By representing their strategies in the form of rules, one is able to gain insights into how experts organised their cognitive skills and adapt their performance to the requirements of the particular task.

2. Implications for Research

This section will conclude the thesis with a brief discussion of the implications of the approach adopted in this study. The thesis took a primarily pragmatic approach to the study of abstracting, by examining how abstractors use their knowledge and skills to interpret the content of a paper and extract its main ideas. The thesis was not interested in describing which mental processes were necessary for the performance of abstracting, this being the task of the cognitive psychologist, but it focussed rather on the strategy that experts use to apply a subset of their knowledge and skills to respond to a particular task, abstracting in this case. Strategy, in this study, implies the purposeful use of resources. By examining human performance in abstracting, and eliciting the strategy to be later computerised, one is suggesting that there is one general strategy to the process of abstracting. If one follows this strategy, one is able to generate a reasonable abstract. This strategy can be expressed in the form of guidelines and rules to be used for computer generation of abstracts as well as a teaching tool. This tool could serve again as a basis to learn about the effectiveness of these guidelines and could help refining the rules providing thus a new insight into the process of abstracting. Defining strategies as procedures could lead to a powerful way of measuring their strength.

Although the above strategy was applied to generate abstracts of expository type of papers in the field of computer artificial intelligence, the rules that described and were tested by INFORMEX are not domain specific. The strategy which was elicited from abstractors whose expertise covered a wide range of disciplines namely information science, polymer engineering, education, linguistics, psychology, and computer science consists mainly of linguistic knowledge and heuristics. This propounds that such strategy would be applicable to other scientific papers of expository type, giving this strategy a universal dimension. A further extension to this research is to apply INFORMEX to a number of expository papers from various disciplines and investigate the reasons for its success or failure if case may be. Further enhancements are to extend INFORMEX to contain strategies for other types of papers.

A further consideration concerns the testing of the rules describing experts' strategy. The current system consists of a set of independent rules which are triggered by a set of another rules, called metarules. The metarules guide the use of knowledge and decide what rules to be applied. In addition to these rules, one could add another set of rules to monitor which rules are fired and in what circumstances they apply, which rules are redundant and which rules could be superseded. This would provide an insight into the function and value of each individual rule. The application of these monitor rules could explain the impact of each rule on a given type of paper and in a given discipline.

It was mentioned earlier that the structure of INFORMEX is independent of the domain knowledge of the paper. One of the programs in INFORMEX breaks the content of a paper into a set of conceptual constituents, using a set of linguistic and heuristic rules. These constituents have not been fully exploited by the current implementation. They could, however, be used to indicate the material content, the style and type of the paper, on one hand, and aid the reader in assessing whether the content covers general or specific concepts of a given area or research. The additional feature could thus help the production of an abstract to be tai.lored to the needs of a general as well as professional readership.

Furthermore, the extracts produced by INFORMEX consist at present of a set of sentences. It was shown that very little editing is required to generate a cohesive and coherent abstract, a further extension of INFORMEX is to investigate an appropriate formalism which allows INFORMEX to organise these sentences into an acceptable text.

The modelling of experts behaviour in generating an abstract has a profound implication. It suggests that some of the complex intellectual tasks involved in manipulating information are carried out in a mechanical procedure. A careful consideration into the way experts and INFORMEX perform abstracting shows that some facets of abstracting are amenable to computational modelling while others do not lend themselves to automation, at least at present time. For a start, if faced with an unstructured paper, experts can use their specialised knowledge and their common sense to predict the kinds of inference needed such as filling gaps or discontinuities in interpreting the text. Second, they can adjust their strategies to deal with new situations they have never before encountered. Thirdly, they learn from new and unpredictable situations, which in turn strengthen their knowledge and strategies. Their performance is not limited. These dynamic facets are difficult to capture in computational model but there is a great deal of research on machine learning and thinking. Probablistic reasoning and fuzzy logic could extend the capabilities of computational models by allowing them to deal with uncertain and badly defined situations but do not explain why humans perform the way they do.

Finally, the study of information processing, using a heuristic orientated approach is a powerful tool for understanding the tasks involved in such processing and devising plans for achieving an expert performance. Such an approach does not only illuminate some difficult and hidden aspects of information processing but also yields an advantage over the cognitive operations approach. As Nickerson, Perkins and Smith explain, "in the heuristic approach, one tries to improve performance on a task by training the subset of cognitive operations though to be relevant to it, and, of course, providing practice on the task itself. A cognitive operations approach does not normally breakdown a task into an organised series of subtasks, nor develop heuristic specific to the task...[For] many tasks, a plan of attack and heuristics specific to the tasks are helpful. This is particularly so for manifestly complex tasks that have a lot of structure inviting a heuristic analysis". [Nickerson, et al, 1985: 226].

In this thesis, only those issues that are directly relevant to abstracting were discussed. Many aspects related to the cognitive processes, to the cognitive modelling and discourse structure were ignored. Several areas which are closely associated to the study such as the interaction of syntax and semantics in automatic text modelling, language competence and performance, acquisition of skills, have been untouched. The study of abstracting in particular and information processing in general is a complex and multilayered phenomenom, which in general makes it difficult to concentrate on the totality of these issues that are implicated. The results of this research indicate that the heuristic oriented approach is a valuable starting point to the study of abstracting and information processing.

LIST OF REFERENCES

[Aitkenhead et al, 1985]
Aitkenhead, A. M. and Slack, J. M. (1985) Issues in Cognitive Modelling. London: Erlbaum

[Allen, 1977]
Allen, J. (1977) Natural Language Understanding. California: Benjamin & Cumings

[American Chemical Society, 1975]
American Chemical Society (1975) Directions for Abstractors, Ohio: Chemical Abstracts Services

[Anderson, 1980]
Anderson, R. C. (1980) "Schema- Directed Processes in Language Comprehension". In Hartley (ed), The Psychology of Written Communication. London: Kogen Page

[Ansubel, 1968]
Ansubel, D. P. (1968) Education Psychology: A Cognitive View. New York: Holt, Rinehart & Winston

[Arzikulov et al, 1978]
Arzikulov, K.A., Piotrovskii, P. G., Popesku, A. N. and Khazhinskaya, M. S. (1987) "Thesaurus-aided Automatic Abstracting of Scientific and Technical Document". Nauchno Tekhnicheskaya Informatsiya Series, 2 (12), 12-20

[Bachenko et al, 1983]
Bachenko, J., Hindle, D. and Fitzpatrick, E. (1983) Constraining a Deterministic Parser". Proceedings of the National Conference on Artificial Intelligence, Aug 22-26, 1983, 8-11

[Bara et al,1984]
Bara, B. G. and Guida, G. (eds) (1984) Computational Models of Natural Language Processing. Amsterdam: North-Holand

[Baxendale,1958]
Baxendale, P. B. (1958) "Machine-Made Index for Technical Literature - An Experiment". IBM Journal of Research and Development, 2, 354-361

[de Beaugrande, 1980]
Beaugrande, R. de (1980) Text, Discourse and Process. London: Longman

[de Beaugrande, R. 1984]
Beaugrande, R. de (1984) Text Production: Toward a Science of Composition. Advances in Discourse Processing, vol 11. New Jersey: Ablex

[de Beaugrande, 1985]
Beaugrande, R. de (1985) "Text Linguistics in Discourse Studies". In van Dijk (ed), Handbook of Discourse Analysis. London: Academic Press

[de Beaugrande et al, 1981]
Beaugrande, R. de, and Dressler, W. U. (1981) Introduction to Text Linguistics. London: Longman

[Berzon, 1971]
Berzon, V. E. (1971) "An Approach to Problems of Automatic Abstracting and Automatic Text". Automatic Documentation and Mathematical Linguistics, 5 (4), 11-20

[Berzon et al, 1979]
Berzon, V. E. and Brailovskii, A. B. (1979) "Classification of Connectors and Dialogue Systems of Automatic Abstracting". Nauchno Tekhnicheskaya Informatsiya Series, 2 (11), 19-23

[Blymenau et al, 1981]
Blymenau, D. I., Gendina, N.I., Donbronravov, I. S., Lakhuti, D. G., Leonov, V. P., and Fodorov, E. B. (1981) Formalised Abstracting with the Use of Word Stereotypes(Markers)", Nauchno Tekhnicheskaya Informatsiya Series, 2 (2), 16-19

[Bolc,1980]
Bolc, L. (ed) (1980) Representation and Processing of Natural Language. Munich: Carl Hanser Verlag

[Bondarenko, 1976]
Bondarenko, G. V. (1976) "The Use of Formal Text Structure Analysis Methods for Automatic Indexing and Abstracting". Nauchno Tekhnicheskaya Informatsiya Series, 2 (10), 5-26

[Borko, 1967]
Borko, H. (1967) Automated Language Processing. London: Wiley

[Borko et al, 1975]
Borko, H. and Bernier, C. L. (1975) Abstracting Concepts and Methods. London: Academic Press

[Bower et al, 1985]
Bower, G. H. and Cirilo, R. K. (1985) "Cognitive Psychology and Text Processing". In van Dijk (ed), Handbook of Discourse Analysis. London: Academic Press

[Britton et al, 1985]
Britton, B. K. and Black, J. B. (eds) (1985) Understanding Expository Text. London: Erlbaum

[Brown et al,1977]
Brown, P. P. , Faibisoff, S., G., Fairclough, G., T., Gaylor, R., H., Grundmann, W., J. and Wilder, J., C. (1977) Abstracts and Abstracting Services: A Manual for Students, Graduate Library School Monograph #6. Arizona: University of Arizona

[Brown et al, 1983]
Brown, G. and Yule, G. (1983) Discourse Analysis. Cambridge: Cambridge University Press

[Butler, 1985]
Butler, C. (1985) Computers in Linguistics. London: Basil Blackwell

[Carney, 1972]
Carney, T. F. (1972) Content Analysis: A Technique for Systematic Inference from Communications. London: Batsford

[Chafe, 1972]
Chafe, W. L. (1972) "Discourse Structure and Human Knowledge". In Freedle et al (eds), Language Comprehension and the Acquisition of Knowledge.Washington: Winston

[Charniak et al, 1985]
Charniak, E. and McDermott, D. (1985) Introduction to Artificial Intelligence. Canada: Addison-Wesley

[Chaudron et al, 1986]
Chaudron, C. and Richards, J. C. (1986) "The Effect of Discourse Markers on the Comprehension of Lectures". Applied Linguistics, 7 (2), 113-125

[Clark et al, 1977]
Clark, H. H. and Clark, E. V. Psychology and Language: An Introduction to Psycholinguistics. New York: Harcourt, Brace & Jovanovitch

[Cleveland et al,1983]
Cleveland, D. B. and Cleveland, A. D. (1983) Introduction to Indexing and Abstracting. Colorado: Libraries Unlimited

[Climenson et al, 1961]
Climenson, W. D., Hardwick, N. H. and Jacobson, S. N. (1961) "Automatic Syntax Analysis in Machine Indexing and Abstracting". American Documentation, 12, 178-183

[Cline et al, 1985]
Cline, T., Fong, W. and Rosenberg, S. (1985) "An Expert Advisor for Photolithography". Proceedings of the 9th International Joint Conference on Artificial Intelligence, 411-413

[Clocksin et al, 1981]
Clocksin, W. F. and Mellish, C. S. (1981) Programming in PROLOG. Berlin: Springer-Verlag

[Cofer, 1976]
Cofer, C. N. (ed) (1976) The Structure of Human Memory. San Franciso: Freeman

[Cole et al, 1977]
Cole, P. and Sadock, J. (eds) (1977) Syntax and Semantics VIII: Grammatical Relations. New York: Academic Press

[Cook, 1975]
Cook, J. R. S. (1975) "A Communicative Approach to the Analysis of Extended Monologue Discourse and its Relevance to the Development of Teaching Materials in ESP". M. Litt. Thesis, University of Edingburgh, Scotland

[Cooper, 1973]
Cooper, D. E. (1973) Philosophy and the Nature of Language. London: Longman

[Cooper, 1982]
Cooper, M. M. (1982) "Content as Vehicle: Implicatures in Writing". In Nystrand (ed), What Writers Know: The Language, Process and Structure of Written Discourse. New York: Academic Press

[Cremmins, 1982]
Cremmins, E. T. (1982) The Art of Abstracting. Philadelphia: ISI Press

[Danes, 1974]
Danes, f. (ed) (1974) Papers on Functional Sentence Perspective. Prague: Academia

[Davis, 1968]
Davis, F. B. (1968) "Research in comprehension in reading". Research Quarterly, 3, 499-545

[Day, 1979]
Day, R. A. (1979) How to Write and Publish a Scientific Paper. Philadelphia: ISI Press

[Derr, 1985]
Derr, R. L. (1985) "The Concept of Information in Ordinary Discourse". Information Processing and Management, 21 (6), 489-499

[Dietschmann, 1984]
Dietschmann, H. (ed) (1984) Representation and Exchange of Knowledge as a Basis of Information Processes. Proceedings of the 5th International Research Forum in Information Science, Heidelberg, Sep 5-7, 1983. Amsterdam: North-Holland

[van Dijk, 1972]
Dijk, T. van (1972) Some aspects of text grammars. The Hague: Mouton

[van Dijk, 1977]
Dijk, T. van (1977) Text and Context. London: Longman

[van Dijk, 1979]
Dijk, T. van (1979) "Relevance Assignment in Discourse Comprehension". Discourse Processes, 2,113-126

[van Dijk, 1985]
Dijk, T. van (1985) "Introduction: Discourse Analysis as a New Cross-Discipline". In van Dijk (ed), Handbook of Discourse Analysis, vol. 1. London: Academic Press

[van Dijk, 1985a]
Dijk, T. van (1985) Handbook of Discourse Analysis, vol. 1. London: Academic Press

[Dilger et al, 1985]
Dilger, W. and Kippe, J. (1985) "COMODEL: A Language for the Representation of Technical Knowledge". Proceedings of the 9th International Joint Conference on Artificial Intelligence, 353-358

[Dressler, 1970]
Dressler, W. U. (1970) "Textsyntax und Ubersetzung". In Hartmann et al (eds), Sprachwissenschaft und Ubersetzen. Munich: Hueber

[Dressler, 1978]
Dressler, W. U. (ed) (1978) Current Trends in Textlinguistics. Berlin: Water de Gruyter

[Earl,1970]
Earl, L.L. (1970) "Experiments in Automatic Extracting and Indexing". Information Storage and Retrieval, 6 (4), 313-334

[Edmunson et al, 1959]
Edmunson, H. P., Oswald, V. A. and Wyllys, R. E. (1959) "Automatic Indexing and Abstracting of the Contents of Documents". Los Angeles: Planning Research Corporation.

[Edmunson et al, 1961]
Edmunson, H. P., Oswald, V. A. and Wyllys, R. E. (1961) "Automatic Abstracting and Indexing- Survey and Recommendations". Communications of the Association for Computing Machinery, 4, 226-234

[Enkvist, 1978]
Enkvist, N. E. (1978) "Stylistics and Text Linguistics". In Dressler (ed), Current Trends in Textlinguistics. Berlin: Walter de Gruyter

[Eisenstadt et al, 1975]
Eisenstadt, M. and Kareev, Y. (1975) "Aspects of human problem solving: The use of internal representations". In Norman et al (eds), Explorations in Cognition. San Francisco: Freeman

[Feigenbaum, 1982]
Feigenbaum, E. A. (1982) "Knowledge Engineering for the 1980's". Computer Science Department. Stanford: Stanford University

[Ferrante, 1985]
Ferrante, R. D. (1985) "The Characteristic Error Approach to Conflict Resolution". Proceedings of the 9th International Joint Conference on Artificial Intelligence, 331-334

[Fidel, 1986]
Fidel, R. (1986) "Writing Abstracts for Free-Text Searching". Journal of Documentation, 42(1), 11-21

[Fillmore, 1977]
Fillmore, C. (1977) "The Case for Case Reopened". In Cole et al (eds), Syntax and Semantics VIII: Grammatical Relations. New York: Academic Press

[Finsbury Data Services]
Finsbury Data Services. Abstracting Rules - In-House Guidelines. Internal Report. Finsbury Services

[Fitts et al, 1979]
Fitts, P. M. and Posner, M. I. (1979) Human Performance. Westport Connecticut: Greenwood Press

[Forsyth, 1989]
Forsyth, R. (ed) (1989) Expert Systems: Principles and Case Studies. New York: Chapman & Hall

[Foster, 1979]
Foster, J. J. (1979) "The use of visual cues in text". In Kolers et al(eds), Processing Visible Language 1. New York: Plenum

[Freedle et al, 1972]
Freedle, R. O. and Caroll, J. B. (1972) Language Comprehension and the Acquisition of Knowledge. Washington: Winston

[Fum et al, 1984]
Fum, D., Guida, G. and Tasso, C. (1984) "A Propositional Language for Text Representation". In Bara et al (eds), Computational Models of Natural Language Processing. Amsterdam: North-Holand

[Garvin, 1965]
Garvin, P. L. (1965) "The Role of Linguistics in Information Processing". Philologica Pragensia, 8, 177-188

[Gashing et al, 1983]
Gashing, J., Klahr, P., Pople, H., Shortliffe, E. and Terry, A. (1983) "Evaluation of Expert Systems: Issues and Case Studies". In Hayes-Roth et al (eds), Building Expert Systems. Reading MA: Addison-Wesley

[Gevarter, 1984]
Gevarter, W. B. (1984) Artificial Intelligence, Expert Systems, Computer Vision and Natural Language Processing. New Jersey: Noyes

[Giora, 1983]
Giora, R. (1983) "Segmentation and segment cohesion: On the thematic organisation of the text". Text, 3 (2), 155-181

[Graesser et al, 1985]
Graesser, A. C. and Goodman, S. M. (1985) "Implicvit Knowledge, Question Answering, and the Representation of Expository Text". In Britton et al (eds) (1985) Understanding Expository Text. London: Erlbaum

[de Greef et al, 1985]
Greef, P. de and Breuker, J. (1985) "A Case Study in Structured Knowledge Acquisition". Proceedings of the 9th International Joint Conference on Artificial Intelligence, 390-392

[Gulich, 1970]
Gulich, E. (1970) Makrosyntax der Gliederundssignale im gesprochenen Franzosich. Munich: Fink

[Halliday, 1967]
Halliday, M. A. K. (1967) "Notes on Transivity and Theme in English: Part 2". Journal of Linguistics, 3, 199-24

[Halliday, 1978]
Halliday, M. A. K. (1978) Language as Social Semiotic. London: E. Arnold

[Hartley,1980]
Hartley, K. (ed) (1980) The Psychology of Written Communication. London: Kogen Page

[Hartley et al, 1977]
Hartley, J. and Burnhill, P. (1977) "Understanding Instructional Text: Topography, Layout and Design". In Howe (ed), Adult Learning. London: Wiley

[Hartmann et al, 1970]
Hartmann. P. and Vernay, H. (eds) (1970) Sprachwissenschaft und Ubersetzen. Munich: Hueber

[Harris, 1751]
Harris, J. (1751) Hermes: or a Philosophical Inquiry Concerning Language and Universal Grammar, Reproduced Facsimile Edition. Menston (1968): Scolar Press

[Harris,1969]
Harris, Z. S. (1969) Structural Linguistics. Chicago: University of Chicago Press

[Hayes Roth et al, 1983]
Hayes Roth, F., Waterman, D. A. and Lenat, D. B. (1983) Building Expert Systems. Reading MA: Addison-Wesley

[Hoey, 1983]
Hoey, M. P. (1983) On the Surface of Discourse. London: Allen & Unwin

[Hoey, 1985]
Hoey, M. P. (1985) "The Paragraph Boundary as a Marker of Relations between the Parts of a Discourse". Midlands Association for Linguistics Studies, 10, 96-107

[Howe, 1977]
Howe, M. J. A. (1977) Adult Learning. London: Wiley

[Jackson, 1986]
Jackson, P. (1986) Introduction to Expert Systems. Wokingham: Addison-Wesley

[Janos, 1978]
Janos, J. (1978) "Results of an Experiment in Automatic Extracting and Problems in the Use Of Condensed Text in Automated Information Systems". International Forum for Information and Documentation, 3 (1), 13-17

[Janos, 1979]
Janos, J. (1979) "Theory of Functional Sentence Perspective and its Application for the Purpose of Automatic Extracting". Information Processing and Management, 15 (1), 19-25

[Kalverkamper, 1981]
Kalverkamper, H. (1981) Orientierung zur Textlinguistik. Tubingen: Niemeyer

[Karasev, 1978a]
Karasev, S. A. (1978) "Building an Auto-Abstracting System with Text Analysis and Synthesis". Nauchno Tekhnicheskaya Informatsiya Series, 2 (5), 22-28

[Karasev, 1978b]
Karasev, S. A. (1978) "A System of Automatic Abstracting: Computer Experiment Results". Nauchno Tekhnicheskaya Informatsiya Series, 2 (8), 25-27

[Kieras, 1978]
Kieras, D. E. (1978) "Good and bad structure in simple paragraphs: Effects on apparent theme, reading time, and recall". Journal of Verbal Learning and Verbal Behaviour, 17,13-28

[Kieras, 1981]
Kieras, D. E. (1981) "The role of major referents and sentence topics in the construction of passage macrostructure". Discourse Processes, 4, 1-15

[Kieras, 1982]
Kieras, D. E. (1982) "A model of reader strategy for abstracting main ideas from simple technical prose". Text, 2, 47-81

[Kieras, 1985]
Kieras, D. E. (1985) "Thematic Processes in the Comprehension of Technical Prose". In Britton et al (eds), Understanding Expository Text, London: Erlbaum

[Kintsch, 1974]
Kintsch, W. (1974) The Representation of Meaning in Memory. New York: Wiley

[Kintsch et al, 1978]
Kintsch, W. and Dijk, T. van (1978) "Towards a Model of Text Comprehension and Production". Psychology Review, 85 (5), 363-394

[King, 1980]
King, J. (1980) "Modelling Concepts for Reasoning about Access to Knowledge". Newsletter of the Association for Computing Machinery Special Interest Group on Artificial Intelligence, June, 138-140

[Kolers et al, 1979]
Kolers, P., Wroslstad, M. E. and Bouma, H. (eds) (1979) Processing Visible Language 1. New York: Plenum

[Kozminski, 1977]
Kozminski, E. (1977) "Altering Comprehension: The Effect of Biasing Titles on Text Comprehension". Memory and Cognition, 5 (4), 482-490

[Kress,1981]
Kress, G. (1981) Halliday: System and Function in Language. Oxford: Oxford Press

[Krippendorf, 1980]
Krippendorf, K. (1980) Content Analysis: An Introduction to its Methodology. London: Sage

[Kuhlen, 1984]
Kuhlen R. (1984) "Some Similarities and Differences between Intellectual and Machine Text Understanding for the Purpose of Abstracting". In Dietschmann (ed), Representation and Exchange of Knowledge as a Basis of Information Processes. Amsterdam: North- Holland

[Kuno, 1978]
Kuno, S. (1978) "Generative Discourse Analysis in America". In Dressler (ed), Current Trends in Texlinguistics. Berlin: Walter de Gruyter, 275-294

[Lenat, 1982]
Lenat, D. B. (1982) "The nature of heuristics". Artificial Intelligence, 19, 189-249

[Lenat et al, 1979]
Lenat, D. B., Hayes-Roth, F. and Klahr, P. (1979) Cognitive Economy. Report HPP-79-15, Heuristic Programming project, Computer Science Department. Stanford: Stanford University

[Lehnert et al, 1982]
Lehnert, W. G. and Ringle, M. H. (1982) Strategies for Natural Language Processing. New Jersey: Erlbaum

[LISA, 1987]
LISA (1987) LISA: In-House Guidelines. Internal Report. London.

[Loftus et al, 1976]
Loftus, G. and Loftus, E. (1876) Human Memory: The Processing of Information. Hillsdale: Erlbaum

[Luhn,1952]
Luhn, H. P. (1952) "The IBM Electronic Information Searching System". New York: IBM Research centre, Yorktown Heights.

[Luhn, 1958]
Luhn, H.P. (1958] "A Business Intelligence System". IBM Journal of Research and Development, 2 (4), 314-319

[Luhn, 1958a]
Luhn, H. P. (1958) "The Automatic Creation of Literature Abstracts". IBM Journal of Research and Development, 2 (4),159-165

[Lyons, 1977]
Lyons, J. (1977) Semantics. Cambridge: Cambridge University Press

[Meyer, 1975]
Meyer, B. J. F. (1975) The Organisation of Prose and its Effect on Recall. Amsterdam: North-Holland

[Meyer, 1985]
Meyer, B. J. F. (1985) "Prose Analysis: Purposes, Procedures and Problems". In Britton et al (eds) Understanding Expository Text. London: Erlbaum

[Minsky,1975]
Minsky, M. (1975) "A Framework for Representing Knowledge". In Winston (ed), The Psychology of Computer Vision, New York: McGraw Hill

[Montgomery, 1982]
Montgomery, C. (1982) "Automated Abstracting: A Re-evaluation". SIGIR Forum, 6 (3), 7-8

[Moray, 1978]
Moray, N. (1978) "The Strategic Control of Information Processing in Strategies of Information Processing". In Underwood (ed) Strategies of Information Processing. New York: Academic Press, 301-328

[Morgan, 1975]
Morgan, J. L. (1975) "Some remarks on the nature of sentences". Papers from the Parasession on Functionalism. Chicago: Linguistic Society

[Naughton, 1974]
Naughton, J. (1974) Systems Management: Management Information Systems, Unit 10. Bletchley: OU Press

[Nguyen et al, 1985)
Nguyen, T. A., Perkins, W. A., Laffey, T. J. and Pecora, D. (1985) "Checking an Expert Systems Knowledge Base for Consistency and Completeness". Proceedings of the 9th International Joint Conference on Artificial Intelligence, 375-378

[Nickerson et al, 1985]
Nickerson, R. S., Perkins, D. N. and Smith, E. E. (1985) The Teaching of Thinking. New Jersey: Erlbaum

[Nilsson, 1982]
Nilsson, N. J. (1982) Principles of Artificial Intelligence. Heidelberg: Springer-Verlag

[Nistor et al, 1978]
Nistor, E. and Roman, E. (1978) "Certain Detail Aspects in Automated Abstracting". Problems of Information and Documentation , 12 (3), 136-140

[Nistor et al, 1980]
Nistor, E. and Roman, E. (1980) "Some Attempts of Automated Abstracting ". Problems of Information and Documentation, 14 (1), 12-18

[Norman et al, 1976]
Norman, D. A. and Bobrow, D. J. (1976) "On the role of active memory processed in perception and cognition". In Cofer (ed), The Structure of Human Memory. San Francisco: Freeman

[Norvig, 1987]
Norvig, P. (1987) "Inference in Text Understanding", Proceedings of the 6th National Conference of the American Association for Artificial Intelligence, July 13-17, 561-581

[Nystrand, 1982]
Nystrand, M. (ed) (1982) What Writers Know: The Language, Process and Structure of Written Discourse. New York: Academic Press

[Oddy, 1981]
Oddy, R. N., Robertson, S. E., Rijsbergen, C. J. van and Williams, P. W. (eds) (1981) Information Retrieval Research. London: Butterwoths

[Paice, 1977]
Paice, C. D. (1977) Information Retrieval and the Computer. London: MacDonald and Jane's

[Paice, 1981]
Paice, C. D. (1981) "The Automatic Generation of Literature Abstracts: an Approach Based on the Identification of Self-indicating Phrases". In Oddy et al (eds), Information Retrieval Research. London: Butterwoths

[Palek,1968]
Palek, B. (1968) Cross-Reference: A Study from Hyper-Syntax. Pragues: Charles UP

[Partridge, 1986]
Partridge, D. (1986) "Engineering Artificial Intelligence Software". Artificial Intelligence Review, 1 (1), 27-41

[Petofi, 1971]
Petofi, J. (1971) Transformationsgrammatiken und eine ko-textuelle Texttheorie. Frankfurt: Athenaum

[Perfetti et al, 1974]
Perfetti, C. A. and Goldman, S. R. (1974) "Thematisation and sentence retrieval". Journal of Verbal Learning and Verbal Behaviour, 13, 97-116

[Platts, 1972]
Platts, M. (1972) Ways of Meaning: An Introduction to a Philosophy of Language. London: Routledge & kegan

[Pollock et al,1975]
Pollock, J. J. and Zamora, A. (1975) "Automatic Abstracting Research at Chemical Abstracts Service". Journal of the Chemical Information for Computer Science, 15 (4), 226-232

[Prikhod'ko et al, 1982]
Prikhod'ko, S. M. and Skhorokhod'ko, Z. F. (1982) "Automatic Abstracting based on Inter-Phrase Links". Nauchno Tekhnicheskaya Informatsiya Series, 2(1), 27-31

[Ringland et al, 1988]
Ringland, G. A. and Duce, D. A. (eds) (1988) Approaches to Knowledge Representation: an Introduction". London: Wiley

[Rollinger et al,1984]
Rollinger, C. R. and Schneider, H. J. (1984) "Text Understanding as a Knowledge-Based Process". In Dietschmann (ed), Representation and Exchange of Knowledge as a Basis of Information Processes. Amsterdam: North- Holland, 129-141

[Rosenberg et al, 1971]
Rosenberg, J. F. and Travis, C. (1971) Readings in the Philosophy of Language. New Jersey: Prentice-Hall

[Rosenberg, 1977]
Rosenberg, S. (ed) (1977) Sentence Production. Hillsdale: Erlbaum

[Rowley et al, 1978]
Rowley, J. E. and Turner, C. M. D. (1978) The Dissemination of Information. London: Deutsch

[Rumelhart, 1977]
Rumelhart, D. (1977) Introduction to Human Information Processing. New York, Wiley

[Rush et al, 1971]
Rush, J. E., Salvador, R. and Zamora, A. (1971) "Automatic Abstracting and Indexing II. Production of Indicative Abstracts by Application of Contextual Inference and Syntactic Coherence Criteria". Journal of the American Society for Information Science, 22 (4), 260-274

[Schank et al,1977]
Schank, R. C. and Abelson, R. F. (1977) Scripts, Plans, Goals and Understanding. New York: Erlbaum

[Scriven, 1972]
Scriven, M. (1972) "The Concept of Comprehension: From Semantics to Software". In Freedle et al (eds), Language Comprehension and the Acquisition of Knowledge. Washington: Winston

[Sevbo, 1969]
Sevbo, I. P. (1969) Text Structure and Automatic Abstracting. Moscow: Nanka Press

[Shannon et al,1949)
Shannon, C. and Weaver, W. (1949) The Mathematical Theory of Communication. Urbana: University of Illinois Press

[Sherrard, 1985]
Sherrard, C. A. (1985) "The Psychology of Summary-Writing". Journal of Technical Writing and Communication, 15 (3), 247-253

[Shortliffe, 1976]
Shortliffe, E. H. (1976) Computer-Based Medical Consultations: MYCIN. New York: Elsevier

[Sidner, 1978]
Sidner, C. L. (1978) A Progress report on the Discourse and Reference component of PAL. CSCSI-2, 206-213

[Skorokhod'ko, 1972]
Skorokhod'ko, E. F. (1972) "Abstracting Methods of Automatic Abstracting and Indexing" in IFIP Congress 71, Ljubliana, Jugoslavia. Amsterdam: North Holland Publishing, 1179-1182

[Sparck Jones et al,1973]
Sparck Jones, K. and Kay, M. (1973) Linguistics and Information Science. New York: Academic Press

[Sparck Jones et al,1985]
Sparck Jones, K. and Wilks, Y. (eds) (1985) Automatic Natural Language Parsing. Chichester: Ellis Horwood

[Sperber et al, 1986]
Sperber, D. and Wilson, D. (1986) Relevance: Communication and Cognition. Oxford: Basil Blackwell

[Stillings et al, 1987]
Stillings, N. A., Feinstein, M. H., Garfield, J. L., Rissland, L. E., Rosenbaum, D. A., Weisler, S. E. and Baker-Ward, L. (1987) Cognitive Science: An Introduction. Cambridge MA: MIT Press

[Stone et al, 1966]
Stone, P. J., Dunphy, D. C., Smith, M. S. and Olgivie, D. M. (1966) The General Inquirer: A Computer Approach to Content Analysis. Cambridge MA: MIT Press

[Stonier, 1989]
Stonier, T. (1989) "Expert Systems and the Knowledge Revolution". In Forsyth (ed), Expert Systems: Principles and Case Studies. New York: Chapman & Hall

[Stubbs,1985]
Stubbs, M. (1985) Discourse Analysis. Oxford: Basil Blackwell

[Swales, 1981]
Swales, J. (1981) Aspects of Article Introductions. Aston ESP Research Reports No. 1. Birmingham: Aston University

[Taylor et al,1977]
Taylor, S. L. and Krulee, G. K. (1977) "Experiments with an Automatic Abstracting System". Proceedings of the American Society for Information Science Annual Meeting, Information Management in the 1980's, 26 Sept - 1 Oct. Chicago: Knowledge Industry

[Thorndyke, 1977]
Thorndyke, P. W. (1977) "Cognitive structures in comprehension and memory of narrative discourse". Cognitive Psychology, 9, 77-110

[Underwood, 1978]
Underwood, G. (ed) (1978) Strategies of Information Processing. London: Academic Press

[Vervalin, 1981]
Vervalin, C. H. (ed) (1981) Communication and the Technical Professional. Houston: Gulf

[Vickery, 1970]
Vickery, B. C. (1970) Techniques of Information Retrieval. Hamden Connecticut: Archon Books

[Young, 1971]
Young, J. F. (1971) Information Theory. London: Butterworth

[Walker, 1981]
Walker, D. W. (1981) "The Organisation and Use of Information: Contributions of Information Science, Computational Linguistics and Artificial Intelligence". Journal of the American Society for Information Science, 32 (5), 349-358

[Wason, 1980]
Wason, P. (1980) "On Writing Scientific Papers in the Psychology of Written Communication". In Hartley (ed), The Psychology of Written Communication. London: Kogan Page

[Weil,1970]
Weil, B. H. (1970) "Standards for writing abstracts". Journal of the American Society for Information Science, 21 (5), 351-357

[Weisman,1972]
Weisman, H.M. (1972) Information Systems, Services, and Centers. New York: Beckers and Hayes.

[Wellish,1980]
Wellish, H. H. (1980) Indexing and Abstracting: An International Bibliography. Oxford: ABC-Clio

[West, 1981]
West, C. K. (1981) The Social and Psychological Distortion of Information. Chicago: Nelson-Hall

[Wilensky,1978]
Wilensky, R. (1978) "Why John Married Mary: Understanding Stories Involving Recurring Goals". Cognitive Science, 2 (3), 235-266

[Winograd, 1972]
Winograd, T. (1972) Understanding Natural Language. New York: Academic Press

[Winograd, 1982]
Winograd, T. (1982) Language as a Cognitive Process. Volume1: Syntax. Reading MA: Addison-Wesley

[Winston, 1975]
Winston, P. H. (ed) (1975) The Psychology of Computer Vision. New York: McGraw Hill

[Winston, 1975]
Winston, P. H. (1972) Artificial Intelligence. Reading MA: Addison-Wesley

[Witty, 1973]
Witty, F. J. (1973) "The Beginning of Indexing and Abstracting: Some Notes towards a History of Indexing and Abstracting in Antiquity and the Middle Ages". The Indexer, 8 (4), 193-198

[Woods et al,1972]
Woods, W. A., Kaplan, R. M. and Nash-Webber, B. (1972) "The Lunar Sciences Natural Language Information System: Final Report". BBN Report 2378, Cambridge, MA: Bolt Beranek and Newman Inc.

APPENDIX A

A CASE STUDY IN STRUCTURED KNOWLEDGE ACQUISITION: ABSTRACTS

Participant 1.

It has been claimed that structuring knowledge acquisition can be approached either by mixing knowledge acquisition with implementation or by a 'structured knowledge acquisition', called KADS, which involves the separation between knowledge acquisition and implementation. This paper outlines a case study of the latter type which is implemented in a knowledge acquisition support system, written in Prolog, using a Prolog KLONE implementation as a structuring device. The aim of the study was to provide a sufficient basis to enable two students unfamiliar with the statistical domain to implement a prototype in a short time. The outcome of the implementation of this approach showed that after six man months of knowledge acquisition, it took four weeks each of the two computer science students to implement a shell for the prototype. The results showed that a separation between analysis and implementation is possible, a methodology which can help a knowledge engineer to make architectural decisions and build a prototype quickly and efficiently.

Participant 2.

The paper identifies the method of knowledge acquisition as a major issue in the construction of expert systems. It discusses in particular KADS (Knowledge Acquisition Documentation and Structuring) methodology which allows a classification of models used as an interpretative framework and whose crucial point is the use of thinking aloud data. The knowledge engineering tasks prescribed by KADS methodology are classified into three types and then are applied to a specific domain of statistical analysis.. An overview of the first stage of the interpretation model (ORIENTATION) is presented and its usefulness for different categories of users is evaluated.

Participant 3.

A major problem in the construction of expert systems is the method for knowledge acquisition. There are two approaches, one is rapid prototyping and the other is called KADS (Knowledge Acquisition Documentation and Structuring). This methodology is written in Prolog using a Prolog KLONE implementation as a structuring device. The case study, carried out by the authors, shows that KADS is a viable methodology.

Participant 4.

Tests Knowledge Acquisition Documentation and Structuring (KADS) methodology for solving the problem of knowledge acquisition in the construction of expert systems. The KADS methodology is outlined and contrasted against the alternative methodology of rapid prototyping, especially in its separation between knowledge acquisition and implementation. A prototype solving statistical correlational and experimental problem is implemented using Prolog. The paper concludes that the KADS methodology is viable and that a separation between analysis and implementation is possible.

Participant 5.

Two distinct methods have been applied for the acquisition of knowledge for use in expert systems. One approach relies on a knowledge engineer interviewing a domain expert in order that a knowledge base can be constructed. A second approach attempts to separate knowledge acquisition and implementation. This methodology, called the structured knowledge acquisition (KADS) is described in this paper with its application to a case study. The methodology is shown to be viable.

APPENDIX B

QUESTIONNAIRES

RESEARCH INTO AUTOMATIC ABSTRACING: EXPERIMENT 2

I. Participant 1 (Research Interests: Deterministic, Chart, Bottom-up Parsing; PROLOG and Parsing; Machine Translation)

Stage 1: title only

The following is a list of papers that may be relevant to your research interests.

Papers	Ranking Order

1. Frame Selection in Parsing.
2. Deterministic and Bottom-up Parsing in PROLOG.
3. Constraining a Deterministic Parser.
4. New Approaches to Parsing Conjunctions using PROLOG.
5. A Semantic process for Syntactic Disambiguation.
6. A Phrase-Structured Grammatical Framework for Transportable Natural Language Processing.
7. Recovering Strategies for Parsing Extragrammatical Language.
8. A Survey of Machine Translation: its History, Current Status and Future Prospects.
9. Using Semantics in Non-context-Free Parsing of Montague Grammar.
10. EUROTRA and its Objectives.

Please specify the degree of importance of these papers using the following score:

* Score 1 if the paper is extremely relevant (i.e the paper discusses issues of specific interest to your research area).
* Score 2 if the paper is very relevant in the sense that it addresses related issues to your paper.
* Score 3 if the paper is of little relevance (i.e the paper discusses general issues).

* Score 4 if you cannot assess the relevance of the paper from the given information.
* Score 5 if the paper is not relevant to your research interests.

Stage 2: title and author

You are already familiar with the papers given below. I have enclosed the author of these papers and the journal of publication.

Papers	Ranking Order

1. Frame Selection in Parsing.
   S. L. Lutinen, 1984, AAAI, 222-243.
2. Deterministic and Bottom-up Parsing in PROLOG.
   E. P. Stabler, 1983, AAAI, 383-407.
3. Constraining a Deterministic Parser.
   J. Bachenko, D. Hindle and E. Fitzpatrick, 1983, AAAI, 8-11.
4. New Approaches to Parsing Conjunctions using PROLOG.
   S. Fong and R. C. Berwick, 1985, IJCAI, 870-876.
5. A Semantic process for Syntactic Disambiguation.
   G. Hirst, 1984, AAAI, 148-168.
6. A Phrase-Structured Grammatical Framework for Transportable Natural Language Processing.
   Ballard, Bruce, Tinkham, 1985, CL, 10, 81-96.
7. Recovering Strategies for Parsing Extragrammatical Language.
   J. G. Carbonell and P. Hayes, 1985, CL, 9, 123-146.
8. A Survey of Machine Translation: its History, Current Status and Future Prospects.
   J. Slocum, 1985, CL, 11(1), 1-17.
9. Using Semantics in Non-context-Free Parsing of Montague Grammar.
   D. S. Warren and J. Friedman, 1985, CL, 8, 123-138.
10. EUROTRA and its Objectives.
    M. King and S. Perschke, 1982, Multilingua 1(1), 27-32.

i) Could you please select the 5 most relevant papers to your research interest and specify the degree of importance using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year ...).

Stage 3: title, author, introduction and conclusions

Enclosed are the introduction and conclusions of the papers you identified as most important in the previous stage.

i) Could you please read these sections and indicate the degree of importance of these papers using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year, introduction, conclusion, references...).

iii) Could you produce an abstract for each paper.

iv) Could you guess the content of the paper and its structure (that is the number of headings and their titles).

Papers	Ranking Order
1. Frame Selection in Parsing. S. L. Lutinen, 1984, AAAI, 222-243.	
2. Deterministic and Bottom-up Parsing in PROLOG. E. P. Stabler, 1983, AAAI, 383-407.	
3. Constraining a Deterministic Parser. J. Bachenko, D. Hindle and E. Fitzpatrick, 1983, AAAI, 8-11.	

Stage 4: The whole paper

I have enclosed a complete copy of the papers you have identified as most important.

i) Could you please read these papers and indicate their degree of importance using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year, introduction, conclusion, references...).

iii) Could you produce an abstract for each paper.

iv) Are the content and the structure of the paper what you expected?

v) Could you mark the sections relevant to your research and return the papers to me at your earliest convenience.

RESEARCH INTO AUTOMATIC ABSTRACING: EXPERIMENT 2

II. Participant 2 (Research Interests: Knowledge Representation and Classification, Inference Engine, Software Engineering and Artificial Intelligence)

Stage 1: title only

The following is a list of papers that may be relevant to your research interests.

Papers Ranking Order

1. A Case Study in Structured Knowledge Acquisition.
2. Representing Procedural Knowledge in Expert Systems: An Application to Process Control.
3. Competence in Knowledge Representation.
4. SEEK2: A Generalised Approach to Automatic Knowledge Base Refinement.
5. Very-High-Level programming of Knowledge Representation Schemes.
6. A Set-Theoretic Framework for the Processing of Uncertain Knowledge.
7. Control and Integration of Diverse Knowledge in a Diagnostic Expert System.
8. A Guide to the Modal Logics of Knowledge and Belief: Preliminary Draft.
9. A representation for Complex Physical Domains.
10. A Model-Theoretic Analysis of Monotonic Knowledge.

Please specify the degree of importance of these papers using the following score:

* Score 1 if the paper is extremely relevant (i.e the paper discusses issues of specific interest to your research area).
* Score 2 if the paper is very relevant in the sense that it addresses related issues to your paper.
* Score 3 if the paper is of little relevance (i.e the paper discusses general issues).
* Score 4 if you cannot assess the relevance of the paper from the given information.
* Score 5 if the paper is not relevant to your research interests.

Stage 2: title and author

You are already familiar with the papers given below. I have enclosed the author of these papers and the journal of publication.

Papers Ranking Order

1. A Case Study in Structured Knowledge Acquisition.
   P. de Greef and J. Breuker, IJCAI, 1, 1985, 390-392
2. Representing Procedural Knowledge in Expert Systems:
   An Application to Process Control.
   M. Gallanti, G. Guida, L. Spampinato and A. Stefanini,
   IJCAI, 1,1985, 345-352
3. Competence in Knowledge Representation.
   R. J. Brachman and H. J. Levesques, AAAI,1982,
   189-192
4. SEEK2: A Generalised Approach to Automatic Knowledge
   Base Refinement.
   A. Ginsberg, S. Weiss and P. Politakis, IJCAI, 1, 1985,
   367-374
5. Very-High-Level programming of Knowledge
   Representation Schemes.
   S. Westfold, AAAI, 1984, 344-350
6. A Set-Theoretic Framework for the Processing of
   Uncertain Knowledge.

S. Y. Lu, and H. E. Stephanou, AAAI, 1984, 216-232

7. Control and Integration of Diverse Knowledge in a Diagnostic Expert System.
   P. K. Fink, IJCAI, 1, 1985, 426-431
8. A Guide to the Modal Logics of Knowledge and Belief: Preliminary Draft.
   J. Y. Halpern and Y. O. Moses, IJCAI, 1,1985, 480-490
9. A representation for Complex Physical Domains.
   S. Addanki and E. Davis, IJCAI, 1, 1985, 443-446
10. A Model-Theoretic Analysis of Monotonic Knowledge.
   M. Y. Vardi, IJCAI, 1, 1985, 509-512

i) Could you please select the 5 most relevant papers to your research interest and specify the degree of importance using the enclosed scoring system.
ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year ...).

<u>Stage 3: title, author, introduction and conclusions</u>

Enclosed are the introduction and conclusions of the papers you identified as most important in the previous stage.
i) Could you please read these sections and indicate the degree of importance of these papers using the enclosed scoring system.
ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year, introduction, conclusion, references...).
iii) Could you produce an abstract for each paper.
iv) Could you guess the content of the paper and its structure (that is the number of headings and their titles).

Papers	Ranking Order

3. Competence in Knowledge Representation.
   R. J. Brachman and H. J. Levesques, AAAI,1982, 189-192
4. SEEK2: A Generalised Approach to Automatic Knowledge

Base Refinement.
A. Ginsberg, S. Weiss and P. Politakis, IJCAI, 1, 1985, 367-374

5. Very-High-Level programming of Knowledge Representation Schemes.
S. Westfold, AAAI, 1984, 344-350

Stage4: The whole paper

I have enclosed a complete copy of the papers you have identified as most important.

i) Could you please read these papers and indicate their degree of importance using the enclosed scoring system.
ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year, introduction, conclusion, references...).
iii) Could you produce an abstract for each paper.
iv) Are the content and the structure of the paper what you expected?
v) Could you mark the sections relevant to your research and return the papers to me at your earliest convenience.

RESEARCH INTO AUTOMATIC ABSTRACTING: EXPERIMENT 2

III. Participant 3 (Research Interests: Knowledge Representation and Elicitation, Production Systems, Expert Systems)

Stage 1: title only

The following is a list of papers that may be relevant to your research interests.

Papers — Ranking Order

1. A Case Study in Structured Knowledge Acquisition.
2. Representing Procedural Knowledge in Expert

Systems: An Application to Process Control.
3. Competence in Knowledge Representation.
4. SEEK2: A Generalised Approach to Automatic Knowledge Base Refinement.
5. Very-High-Level programming of Knowledge Representation Schemes.
6. A Set-Theoretic Framework for the Processing of Uncertain Knowledge.
7. Control and Integration of Diverse Knowledge in a Diagnostic Expert System.
8. A Guide to the Modal Logics of Knowledge and Belief: Preliminary Draft.
9. A representation for Complex Physical Domains.
10. A Model-Theoretic Analysis of Monotonic Knowledge.

Please specify the degree of importance of these papers using the following score:

* Score 1 if the paper is extremely relevant (i.e the paper discusses issues of specific interest to your research area).
* Score 2 if the paper is very relevant in the sense that it addresses related issues to your paper.
* Score 3 if the paper is of little relevance (i.e the paper discusses general issues).
* Score 4 if you cannot assess the relevance of the paper from the given information.
* Score 5 if the paper is not relevant to your research interests.

Stage 2: title and author

You are already familiar with the papers given below. I have enclosed the author of these papers and the journal of publication.

Papers	Ranking Order

1. A Case Study in Structured Knowledge Acquisition.

P. de Greef and J. Breuker, IJCAI, 1, 1985, 390-392

2. Representing Procedural Knowledge in Expert Systems: An Application to Process Control.
   M. Gallanti, G. Guida, L. Spampinato and A. Stefanini, IJCAI, 1,1985 , 345-352
3. Competence in Knowledge Representation.
   R. J. Brachman and H. J. Levesques, AAAI,1982, 189-192
4. SEEK2: A Generalised Approach to Automatic Knowledge Base Refinement.
   A. Ginsberg, S. Weiss and P. Politakis, IJCAI, 1, 1985, 367-374
5. Very-High-Level programming of Knowledge Representation Schemes.
   S. Westfold, AAAI, 1984, 344-350
6. A Set-Theoretic Framework for the Processing of Uncertain Knowledge.
   S. Y. Lu, and H. E. Stephanou, AAAI, 1984, 216-232
7. Control and Integration of Diverse Knowledge in a Diagnostic Expert System.
   P. K. Fink, IJCAI, 1, 1985, 426-431
8. A Guide to the Modal Logics of Knowledge and Belief: Preliminary Draft.
   J. Y. Halpern and Y. O. Moses, IJCAI, 1,1985, 480-490
9. A representation for Complex Physical Domains.
   S. Addanki and E. Davis, IJCAI, 1, 1985, 443-446
10. A Model-Theoretic Analysis of Monotonic Knowledge.
    M. Y. Vardi, IJCAI, 1, 1985, 509-512

i) Could you please select the 5 most relevant papers to your research interest and specify the degree of importance using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year ...).

Stage 3: title, author, introduction and conclusions

Enclosed are the introduction and conclusions of the papers you identified as most important in the previous stage.

i) Could you please read these sections and indicate the degree of importance of these papers using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year, introduction, conclusion, references...).

iii) Could you produce an abstract for each paper.

iv) Could you guess the content of the paper and its structure (that is the number of headings and their titles).

Papers	Ranking Order
1. A Case Study in Structured Knowledge Acquisition. P. de Greef and J. Breuker, IJCAI, 1, 1985, 390-392	
4. SEEK2: A Generalised Approach to Automatic Knowledge Base Refinement. A. Ginsberg, S. Weiss and P. Politakis, IJCAI, 1, 1985, 367-374	
7. Control and Integration of Diverse Knowledge in a Diagnostic Expert System. P. K. Fink, IJCAI, 1, 1985, 426-431	

Stage 4: The whole paper

I have enclosed a complete copy of the papers you have identified as most important.

i) Could you please read these papers and indicate their degree of importance using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year, introduction, conclusion, references...).

iii) Could you produce an abstract for each paper.

iv) Are the content and the structure of the paper what you expected?

v) Could you mark the sections relevant to your research and return the papers to me at your earliest convenience.

RESEARCH INTO AUTOMATIC ABSTRACING: EXPERIMENT 2

IV. Participant 4 (Research Interests: Semantic Modelling, Database Design, Intelligent Database Systems)

Stage 1: title only

The following is a list of papers that may be relevant to your research interests.

Papers	Ranking Order
1. Schematic Pseudocode for Program Constructs and its Computer Automation by Schemacode.	
2. On Conceptual Database Modelling.	
3. A Perspective for Research on Conceptual Modelling.	
4. Databases: A Logical Perspective.	
5. Issues in the Design of Databases Programming Languages.	
6. Modelling Concepts for Reasoning about Access to Knowledge.	
7. The Use of a Semantic Network in a Deductive Question-Answering System.	

Please specify the degree of importance of these papers using the following score:

* Score 1 if the paper is extremely relevant (i.e the paper discusses issues of specific interest to your research area).
* Score 2 if the paper is very relevant in the sense that it addresses related issues to your paper.

* Score 3 if the paper is of little relevance (i.e the paper discusses general issues).
* Score 4 if you cannot assess the relevance of the paper from the given information.
* Score 5 if the paper is not relevant to your research interests.

Stage 2: title and author

You are already familiar with the papers given below. I have enclosed the author of these papers and the journal of publication.

Papers Ranking Order

1. Schematic Pseudocode for Program Constructs and its Computer Automation by Schemacode.
   P. N. Robillard, CACM, 29 (11), 1986, 1072-1089
2. On Conceptual Database Modelling.
   D. McLeod, SIGART, June, 1980, 161-163
3. A Perspective for Research on Conceptual Modelling.
   J. Mylopoulos, SIGART, June, 1980, 167-170
4. Databases: A Logical Perspective.
   R. Reiter, SIGART, June,1980, 174-176
5. Issues in the Design of Databases Programming Languages.
   L. A. Rowe, SIGART, June, 1980, 180-182
6. Modelling Concepts for Reasoning about Access to Knowledge.
   J. King, SIGART, June, 1980, 138-140
7. The Use of a Semantic Network in a Deductive Question-Answering System.
   J. R. McSkimin and J. Minker, IJCAI-77,1977, 50-58

i) Could you please select the 5 most relevant papers to your research interest and specify the degree of importance using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year ...).

Stage 3: title, author, introduction and conclusions

Enclosed are the introduction and conclusions of the papers you identified as most important in the previous stage.

i) Could you please read these sections and indicate the degree of importance of these papers using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year, introduction, conclusion, references...).

iii) Could you produce an abstract for each paper.

iv) Could you guess the content of the paper and its structure (that is the number of headings and their titles).

Papers	Ranking Order
2. On Conceptual Database Modelling. D. McLeod, SIGART, June, 1980, 161-163	
3. A Perspective for Research on Conceptual Modelling. J. Mylopoulos, SIGART, June, 1980, 167-170	
5. Issues in the Design of Databases Programming Languages. L. A. Rowe, SIGART, June, 1980, 180-182	
6. Modelling Concepts for Reasoning about Access to Knowledge. J. King, SIGART, June, 1980, 138-140	

Stage 4: The whole paper

I have enclosed a complete copy of the papers you have identified as most important.

i) Could you please read these papers and indicate their degree of importance using the enclosed scoring system.

ii) Could you specify what criteria you used when assessing the ranking order for each title (keywords in the title, author, journal, year, introduction, conclusion, references...).

iii) Could you produce an abstract for each paper.

iv) Are the content and the structure of the paper what you expected?

v) Could you mark the sections relevant to your research and return the papers to me at your earliest convenience.

APPENDIX C

ABSTRACTS FOR EXPERIMENT 2

P1.A and P1.B :

The selection of case frames in the presence of large-scale ambiguity has previously been studied using lexically-based and frame-based methods. These are discussed and contrasted with a method using general inference rules. The method is compared with FRUMP and the Incremental Description Refinement process in RUS.

P1.C :

The problem of frame selection in parsing is discussed, with the focus on the selection of a frame for texts which contain highly ambiguous or vague words. An approach to frame selection is presented which involves the use of a small number of general inference rules in conjunction with a hierarchically-organised conceptual memory. This is in contrast to various other methods, which rely either on disambiguation rules stored in the dictionary definitions of ambiguous words, or on previously-activated frames to guide the selection of new frames. The selection of frames for vague or ambiguous words using these previous methods is shown to be problematic. The method presented here does not suffer from these same problems because of the hierarchical organization of memory and the general inference rules used.

P2.A and P2.B :

Context-free parsing and extensions (eg ATNs) are described, Methods of introducing determinism and bottom-up search are discussed.

P2.C :

It is well known that top-down backtracking context free parsers are easy to write in Prolog, and that these parsers can be extended to give them the power of ATNs. This report shows that a number of other familiar parser designs can be naturally implemented in Prolog. The top-down parsers can easily be constrained to do deterministic parsing of LL(K) languages. Bottom-up backtrack parsers can also be elegantly implemented and similarly constrained to do deterministic parsing. Very natural extensions of these LR(K) parser designs suffice for deterministic parsing of natural languages of the sort carried out by the Marcus (1980) parser.

P3.A and P3.B :

Deterministic parsing and left-corner techniques are used to improve the efficientcy of processing technical texts. The influence of these techniques on the structure of the grammar of verb complementation is discussed.

P3.C :

At the Naval Research Laboratory, we are building a deterministic parser, based on principles proposed by Marcus, that can be used in interpreting military message narrative. A central goal of our project is to make the parser useful for real-time applications be constraining the parser's actions and so enhancing its efficiency. In this paper, we propose that a parser can determine the correct structures of English without looking past the "left corner" of a constituent, i.e., the leftmost element of the constituent along with its lexical category (e.g. N, V, Adj). We show that this Left Corner Constraint, which has been built into our parser, leads quite naturally to a description of verb complements in English that is consistent with the findings of recent linguistic theory, in particular, Chomsky's government and binding (GB) framework.

G3.A :

The domain of knowledge, used by expert systems, is becoming increasinly complex and incomplete. This paper examines the impact of these factors on the task of knowledge representation and suggests a way to overcome the problems by pulling ideas from current research. The solution proposed is to distinguish between terminological and assertional competence of a knowledge representation so that the former can be understood in terms of its impact on the latter. The segmentation of a knowledge base into these two components preserves the features of each without sacrificing their powers.

G3.B :

In modern rule-based expert tasks, the domain of knowledge may suffer from two difficulties: complexity and incompleteness. Therefore realistic knowledge representation schemes must be able to deal with the terminological and assertional adequacies of the desired system. This paper examines current research into these fields and proposes a solution which draws the best out of the two areas. Next, components of this new representation methodology are described and their uses justified.

G3.C :

The range of domains and tasks for "knowledge-based systems" has been expanding at a furious pace. As we move away from trivial domains, such as the "blocks world", the demands on knowledge representation systems used by expert programs are becoming more extreme. For one thing, the domains themselves are getting so complex that specialised technical vocabularies are unavoidable; consequently, the issue of a system talking with an expert in his own language cannot be ignored. For another, tasks such as medical diagnosis, scene analysis, speech understanding, and game playing all have as a cenyral feature an incrementally evolving model representing probably incomplete knowledge of part of the task domain. In this paper, we explore some of the impact of these two critical issues - complexity and incompleteness - on knowledge representation systems. We review some aspects of current representation research that offer a foundation for coping with these problems, and finally suggest a way of integrating these ideas into a powerful practical knowledge representation paradigm.

G4.A :

The construction of a knowledge base is one of the most difficult tasks in the development of expert systems. The process of knowedge acquisition can be divided into two phases where the second phase entails refining the rough knowledge acquired from the expert during phase one. This paper describes a method for refining the initial general knowledge to produce a more optimized and formal representation of it. The approach to knowledge base refinement, adopted in SEEK2, is an attempt to construct a more accurate and consistent scheme when designing and validating expert systems.

G4B :

Knowledge acquisition is one of the key problems in the construction of rule-based expert systems. It is a process which entails extracting the initial rough knowledge from the expert and then refining these rules to improve performance. SEEK2 is a system that can present plausible solutions to this refinement problem without expert interaction. The attempt to find a sequence of refinements that optimizes performance is a search problem which can be achieved using a hill-climbing algorithm. This paper describes SEEK2's automatic pilot capabilities and the metalanguage used during the knowledge base refinement.

G5.A :

This paper describes a very high-level programming language which allows both the program and the knowledge base to be specified using the same mathematical techniques which include logic, sets and relations. Thus the language used to express the knowledge is more closely related to the program which manipulates it than the actual implemented representation of the knowledge base. The result is an efficient and flexible language which is more comprehensible and easy to maintain.

G5.B :

A knowledge based system typically consists of a program and a knowledge base that the program uses. In order to increase compiler efficiency and facilitate program maintainability and comprehension, a very high-level mathematical language is needed to express the rules as well as the solution to this problem. This paper provides an overview of such a language, known as BC, and then proceeds to describe how rules can be represented in a knowledge base system.

G5.C :

This paper proposes building knowledge-based systems using a programming system based on a very high-level language. It gives an overview of such a programming system, BC, and shows how BC can be used to implement knowledge representation features, providing as examples, automatic maintenance of inverse links, and property inheritance. The specification language of BC can be extended to include a knowledge representation language by describing its knowledge representation features. This permits a knowledge-based program and its knowledge base to be written in the same very high-level language which allows the knowledge to be more efficiently incorporated into the program as well as making the system as a whole easier to understand and extend.

H1.A :

Knowledge Acquisition has proved to be one of the major bottlenecks in the development of Expert Systems. Two distinct methods have been used to acquire knowledge. In the first, the knowledge engineer interviews the domain expert and from the data obtained attempts to build a prototype expert system. The second method, called KADS, (Knowledge Acquisition Documentation and Structuring), attempts to separate out knowledge acquisition and implementation. This paper describes a case study application of KADS which shows that the methodology is viable and that it is possible to separate knowledge acquisition from implementation.

H1.B :

Two distinct methods have been applied for the acquisition of knowledge for use in expert systems. One approach relies on a knowledge engineer interviewing a domain expert in order that a knowledge base can be constructed. A second approach attempts to separate knowledge acquisition and implementation. This methodology, called the structured knowledge acquisition, is described in this paper together with its application to a case study. The methodology is shown to be viable.

H1.C :

Building an expert system usually comprises an entangled mixture of knowledge acquisition and implementation efforts. An emerging methodology based on cognitive psychology and software development guides and supports knowledge acquisition while implementation is deferred. This allows for a more deliberate choice of architecture. This paper presents a case study to test the methodology.

H4.A :

Knowledge acquisition, which is a major problem in AI, can be divided into two phases. In the first phase, the knowledge engineer in collaboration with the domain expert constructs an initial knowledge base which roughly approximates to the knowledge of the expert. In the second phase, the knowledge is progressively refined through the modification of the knowledge base. SEEK2 is a system which attempts to incorporate a generalised approach to automatic knowledge base refinement. In this paper, we give an outline description of SEEK2 and its successful application to a rhematology knowledge base of approximately 140 rules. We conclude that SEEK2 offers a solid foundation for designing and validating expert system knowledge bases.

H4.B :

This paper describes a generalised approach to the automatic refinement of knowledge bases for use in expert systems. The basic approach as used in the program SEEK is first of all given. This approach relies on the assumption that a finite set of final diagnostic conclusions or endpoints can be identified. Particular attention can then be paid to the refinement of rules involved in concluding a particular endpoint. Refinement usually falls into one of two possible classes: generalisation and specialisation. The different refinement techniques required for those two classes are described. The original SEEK program was developed for interactive use by the knowledge engineer and it could not by itself refine a knowledge base. This restriction has been removed in SEEK2 which is described in this paper together with a simplified example to illustrate its mode of operation.

H4.C (and G4.C) :

This paper describes an approach to knowledge base refinement, an important aspect of knowledge acquisition. Knowledge base refinement is characterised by the addition, deletion and alteration of rule-components in an existing knowledge base in an attempt to improve an expert system's performance. SEEK2 extends the capabilities of its predecessor rule refinement system, SEEK. In this paper, we describe the progress we have made since developing the original SEEK program: (a) SEEK2 works with a more general class of knowledge bases than SEEK, (b) SEEK2 has an "automatic pilot" capability, i.e., it can if desired, perform all of the basic tasks in knowledge base refinement without human interaction, (c) a metalanguage for knowledge base refinement has been specified which describes both domain-independent and domain-specific metaknowledge about the refinement process.

H7.A :

Human experts are able to reason with many different kinds of knowledge. Two types of knowledge are "surface" knowledge and "deep" knowledge. In surface knowledge, rules of thumb correct characteristics of a problem with possible solutions. Deep knowledge allows an expert to reason from first principles. The Integrated Diagnostic Model (IDM) is an expert system which incorporates both forms of knowledge. An outline description of IDM is given together with its range of suitable applications. Future work with IDM is briefly described.

H7.B :

Different kinds of knowledge are often brought to bear by domain experts in the solution of problems. In this paper an expert system is described, called the Integrated diagnostic Model (IDM) which attempts to use both shallow and deep knowledge base. These two types of expert systems are integrated by the control mechanism of the executor, the operation of which is described with regard to a particular example. The paper concludes by giving the limitations of the IDM.

H7.C :

Though current expert system technology has become a major success, existing systems fall often short of human expertise in many ways. One important area is in the use of more basic, deep knowledge as an enhancement to the shalow, surface knowledge commonly employed. The Integrated Diagnostic Model attempts to exploit the use of both types of knowledge by fitting the appropriate knowledge representation and utilization techniques to each. The result is two separate and independent expert systems which are then integrated and controlled by a higher-level module called the executor.

R3.A :

The paper describes work carried out in the area of conceptual modelling, following the principles of object-orientated language, procedural attachment and abstraction (aggregation, generalisation and classification). PSN, a knowledge representation language and TAXIS, a semantic data model are described. PSN treats traditional semantic network concepts within a procedural framework. TAXIS is a programming language for the design of interactive information systems. The paper outlines the similarities and differences between the two and examines their influence on one another. The major difference noted concerns efficiency vs flexibility. TAXIS limits the kind of run-time operation available but is more efficient. Otherwise the features of each have counterparts in the other.

R3.B :

The paper describes tools for conceptual modelling by outlining two models PSN, a knowledge representation language and TAXIS, a semantic data model. Interesting features adopted are the object-oriented approach, idea of procedural attachment for defining the meaning of objects and the notion of organisation in terms of aggregation, generalisation and classification. Other interesting ideas mentioned are mechanisms for handling both static and dynamic exceptions, mapping between classes by similarity and context for modelling multiple views.

R6.A :

The paper is concerned with the specification and use of semantic constraints on a database. It describes various work that has been carried out in the area. A goal of specifying constraints is seen as being the ability to create more powerful systems to mediate between a user and a collection of data, rules and models, thus fulfilling to some extent the task of the data analyst.

R6.B :

The paper discusses semantic constraints and related semantic query optimization. Semantic query optimization exploits semantic constraints on a database to perform one aspect of intelligent database mediation, increasing retrieval efficiency. Several kinds of semantic knowledge can be used, including value ranges for attributes, mutual bounds on value ranges among different attributes, cardinality restrictions on relationships and definitions of derived attributes. Semantically motivated differential access to data can be exploited depending on the underlying file structure indexes and links, etc. Work forms part of a KBMS project.

R6.C :

There is growing agreement about the usefulness of putting semantic database constraints into explicit form that can be manipulated by various database programs. Indeed, this is a prerequisite for building intelligent database mediators. These are programs that perform the task of a good database analyst: to pose the most effective and easily processed queries to help solve a problem. Semantic query optimization is a technique to exploit semantic constraints for one aspect of intelligent database mediation: increasing retrieval efficiency. The technique is to use semantic constraints to transform a query into an equivalent one that can be processed more efficiently. Various kinds of semantic knowledge that can be used for this purpose are described.

APPENDIX D

REFERENCES OF THE PAPERS USED IN THE SHORT-CUT STRATEGY.

CL1:

Hobbs, J. R. and Shieber, S. M. (1987) "An Algorithm for Generating Quantifier Scopings". Computational Linguistics, 13, 47-63

CL2:

Tomita, M. (1987) "An Efficient Augmented-Context-free Parsing Algorithm. Computational Linguistics, 13, 31-46

PE1:

Titomanlio, G. (1982) "Estimation of Injection Pressure During Mold Filling". Polymer Engineering and Science, 22 (5), 324-328

PE2:

Kamal, M. R. and Ryan, M. E. (1980) "The Behavior of Thermosetting Compounds in Injection Molding Cavities". Polymer Engineering and Science, 20 (13), 859-867

PE3:

Tonogai, S. and Seto, S. (1981) "Monohole Flow Test with High-Frequency Preheating for Evaluating Flowability of Thermosetting Compounds". Polymer Engineering and Science, 21 (5), 301-306

CS1:

Ferrante, R. D. (1985) "The Characteristic Error Approach to Conflict Resolution". Proceedings of the International Joint Conference on Artificial Intelligence, 1, 331-334

CS2:

Dilger, W. and Kippe, J. (1985) "COMODEL: A Language for the Representation of Technical Knowledge". Proceedings of the International Joint Conference on Artificial Intelligence, 1, 353-358

CS3:

Middleton, A. G. (1987) "Some Transformations of Relevance to Report Generation". The Computer Journal, 30 91), 37-41

IS1:

Buckles, B. P. and Petry, F. E. (1985) "Uncertainty Models in Information and Database Systems". Journal of Information Science, 11, 77-87

IS2:

Michelet, B. and Turner, W. A. (1985) "Co-Word Search: A System for Information Retrieval". Journal of Information Science, 11, 173-181

APPENDIX E

CODE LISTINGS FOR COMPUTER PROGRAMS

THE TEXTUAL ANALYSER SUBMODULES.

```
*****************************************************************************
* This program  contains three submodules: 1) the conceptual analyser module
*                                          2) the syntactic analyser module
*                                          3) the structural analyser module.
* It identifies headings, paragraghs, sentences, words.
* It records the structure of the paper and stores it in the structural text
* knowledge base.
* It identifies conceptual constituents for each
* sentence and generates a conceptual model.
* In other words it produces a textual world model for a given paper
* expressed
* in the form of PROLOG facts to be used by the expert classifier program.
*
*****************************************************************************
*
*initialisation
         &anchor = 0
         &trim = 1
         &dump = 1
*
*define arrays of delimiters and assume a maximum of 10 headings
         delimiter = table(200)
         newdelim = array(200)
         delchar = table(30)
         delstyl = table(50)
         heading = array('0:9')
         nheading = array('0:9')
         d = table(300)
         verb = table(60)
         conno = array(300)
*
*initialise parameters
         sentcount = 0
         nulline =
         tccount = 1
         flag = 0
         ofcount = 0
         par = 0
         tpar = 0
         nodelim = 0
         nodelstyl = 0
         cptno = 1
*
* files specifications
         input(.input, 'textl.resl')
* detailed analysis of text under study
         output(.output, 'textl.res2')
         output(.output0, 'textl.trace')
         output(.output1, 'delimiters.textl')
         output(.output4, 'textl.parag')
* prolog database
         output(.output2, 'textltxtcpt.pl')
         output(.output3, 'textlcpt.pl')
*
*set up patterns to detect sentence
         sentend = '.!?'
         sent = (break(sentend)  any(sentend)  break(' ')) . sentence
         punct = any(',;:"')
*
*set up word patterns
         ucs = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
         lcs = 'abcdefghijklmnopqrstuvwxyz'
         extras = "'-"
         letters = ucs lcs extras
```

```
        wpat = break(letters) .  gap  span(letters)  . word
        hyphen = '-  '
        cedille = "~"
        spaces = "  "
        space =  " "
        paragmark = "%%"
*
*set up delimiters patterns
        relatlist = 'THAT|WHEN|WHO|WHOM|WHOSE|WHICH|WHAT|WHY|BECAUSE|WHILE|'
+                   'HOW|'
        pronlist = 'WE|I|YOU|HE|SHE|THEY|IT|US|THEM|'
        functlist = 'AMONG|AS|LIKE|SO|THUS|TOWARD|USING|BETWEEN|TOWARDS|'
+                   'ALTHOUGH|WITHIN|WHEREVER|WHENEVER|WHATEVER|THERE|'
+                   'THEREBY|THROUGHOUT|BEFORE|AFTER|OFTEN|THEREFORE|'
+                   'WHEREAS|THEN|TILL|UNTIL|WHEREBY|THOUGH|WHETHER|WHILE|'
+                   'WHILST|HOWEVER|MOREOVER|HERE|'
        preplist = 'AROUND|AT|INTO|BY|FOR|WITH|FROM|AND|TO|WITHIN|WITHOUT|'
+                  'IN|ON|UNDER|UNTIL|OR|AWAY|OFF|OUT|OVER|ABOVE|BEHIND|'
+                  'BELOW|ACROSS|UNDERNEATH|THROUGH|UP|DOWN|ALONG|BEYOND|'
+                  'THROUGHOUT|AGAINST|DESPITE|NOTWITHSTANDING|ABOUT|SINCE|'
+                  'EXCEPT|EXCEPTING|UPON'
        subordlist  = 'IF|BUT|LIKE|UNLESS|OR|RATHER|THAN|'
        conjlist = 'FURTHERMORE|FIRST|FINALLY|SECONDLY|THUS|'
        detlist = 'THE|AN|A|THIS|THESE|THOSE|'
        poslist = 'HIS|ITS|HER|MY|OUR|THEIR|YOUR|OURS|YOURS|THEIRS|'
        quantlist = 'FEW|LOTS|LOT|MANY|PLENTY|MOST|MUCH|SOME|SUCH|'
        modalist = 'CAN|COULD|MAY|MUST|WILL|WOULD|SHALL|SHOULD|MIGHT|'
        verblist = 'BEEN|BEING|BE|HAVE|HAS|SEE|IS|WAS|WERE|DID|DOING|'
+                  'DO|DONE|ARE|'
        verblst = 'CONSISTS|CONSIST|PROVIDES|PROVIDE|CLASSIFIES|CLASSIFY|'
+                 'REVEALS|REVEAL|MAKES|MAKE|UTILISES|UTILIZES|UTILIZE|'
+                 'UTILISE|IDENTIFIES|IDENTIFY|CONTAINS|CONTAIN|'
+                 'SHOWN|SHOWS|SHOW|CAME|COMES|INVOLVE|INVOLVES|DEALS|'
+                 'DESCRIBE|DESCIBES|ATTACHES|DEFINE|DEFINES|INDUCE|'
+                 'INDUCES|REFER|REFERS|INTEGRATE|INTEGRATES|ASSUMES|'
+                 'ASSUME|PLAYS|COMBINE|COMBINES|'
        additlist = 'ALSO|EITHER|EVEN|NEITHER|NOR|TOO|'
        limitlist = 'MAINLY|MOSTLY|ESPECIALLY|CHIEFLY|ALONE|ONLY|PURELY|'
+                    'SIMPLY|MERELY|ALSO|INCLUDING|THERE|NAMELY|ALL|'
        intenslist = 'CLEARLY|CERTAINLY|INDEED|RARELY|ALMOST|JUST|FULLY|QUITE|'
+                    'THOROUGHLY|MOST|EVER'
        pkt = 'PMARKS'
        symbols = '$]){['
        numbers = '1234567890'
        charlist = ' - |-| -|- |'
        adverb = rtab(2) 'LY'
*
*check for stylistic features that can be deleted
        style = 'ACCORDING|WELL|RESPECT|KNOWN|'
+               'ONCE|NOWADAYS|NOW|ETC|EG|ONLY|'
+               'IE|BEST|FIG|SIMILAR|'
*
*give special attention to these
        special = 'OF'
        equations = any('=+/<>')
        parspace = '       '
*
*processing
*set up delimiters arrays
loop1   preplist break('|') . string '|' =        :F(loop2)
        delimiter<string> = string                :(loop1)
loop2   relatlist break('|') . string '|' =       :F(loop3)
        delimiter<string> = string                :(loop2)
loop3   conjlist break('|') . string '|' =        :F(loop4)
        delimiter<string> = string                :(loop3)
```

```
loop5   [illegible] break('|') . string '|' =     :F(loop6)
        delimiter<string> = string                :(loop5)
loop6   pronlist break('|') . string '|' =        :F(loop7)
        delimiter<string> = string                :(loop6)
loop7   functlist break('|') . string '|' =       :F(loop8)
        delimiter<string> = string                :(loop7)
loop8   detlist break('|') . string  '|' =        :F(loop9)
        delimiter<string> = string                :(loop8)
loop9   poslist break('|') . string '|' =         :F(lop10)
        delimiter<string> = string                :(loop9)
lop10   quantlist break('|') . string '|' =       :F(lop11)
        delimiter<string> = string                :(lop10)
lop11   modalist break('|') . string '|' =        :F(lop12)
        delimiter<string> = string                :(lop11)
lop12   verblist break('|') . string '|' =        :F(lop13)
        delimiter<string> = string                :(lop12)
lop13   intenslist break('|') . string '|' =      :F(lop14)
        delimiter<string> = string                :(lop13)
lop14   limitlist break('|') . string  '|' =      :F(lop15)
        delimiter<string> = string                :(lop14)
lop15   style break('|') . string '|' =           :F(lop16)
        delstyl<string> = string
        output0 = delstyl<string>                 :(lop15)
lop16   charlist break('|') . string '|' =        :F(lop17)
        delchar<string> = string
        output0 = delchar<string>                 :(lop16)
lop17   verblst break('|') . string '|' =         :F(delp)
        verb<string> = string
        output0 = verb<string>                    :(lop17)
delp    newdelim = sort(convert(delimiter,'array'))
+                                                 :F(empty)
        i = 1
dout    output0 = i '.' newdelim<i,1> '=' newdelim<i,2>
+                                                 :F(prol)
        i = i + 1                                 :(dout)
empty   terminal = 'empty array'                  :(end)
*
*prepare prolog database
*
prol    output2 = "/* prolog database produced by the program called txtcpt.spt"
+        "*/"
        output2 = "/* the format of the database is as follows: */"
        output2 = "/* (note: title set to heading no = 0) */"
        output2 = "/* textconcept(heading no, paragragh no,"
+       "paragraphsentence no, sentence no, concept no, concept identity ) */"
        output3 = "/* prolog database produced by txtcpt.spt */"
        output3 = "/* concept(concept identity, concept) */"
        output1 = "sentence no     delimiter"

*set title-heading count to 0
        hno = 0
        terminal = "********** processing text"
*read a line and concatenate a blank before it and at end of line.
read    line = line ' '  input ' '   :F(print)
*
*test  for end of sentence and initialise local parameters
getsent ccount = 1
        out =
        word =
        gap =
        line sent  =                 :F(read)
        sentcount = sentcount + 1
*
*test for hyphenated words and concatenate them
hyp     sentence  hyphen =           :S(hyp)
```

```
         output = "Sentence no. " sentcount
*
*check sentence length, if >80 char. divide sentence into parts
         slen = size(sentence)
         le(slen,80)                         :F(dup)
*         output = sentence                   :(equa) removed 14.1.88
         output = sentence                   :(head)
dup      dupsent = sentence
loop     sentlen = size(dupsent)
         gt(sentlen,80)                      :F(out0)
         dupsent len(80) . sentl rem . sent2
         output =  sentl
         dupsent = sent2                     :(loop)
out0     output = dupsent
*
*check if sentence contains equations symbols, if so get next sentence
*equa     sentence equations =                :S(read) removed 14.1.88
*
*check if sentence is a heading
head     newsentence = sentence
         newsentence  len(1) cedille =    :F(paraph)
         newsentence   digits  . char2  rem . sentence  :F(errorl)
*         hno = char2
         terminal = "heading no ::::::::" hno " " sentence

*
*heading found so initialise local paragraph parameters and label headings
         par = 0
         psent = 0
         pflag = 1
         heading<hno> = sentence
         hno = hno + 1                       :(getwd)
*
*increase psent parameter only if sentence not a heading
incr     psent = psent + 1                   :(getwd)
*
*test for sentence patterns and identify conceptual units
getwd     sentence wpat =                    :F(null)
*
*do not count headings as paragraph sentences
cap       capword = replace(word,lcs,ucs)
          ident(pkt,capword)                 :S(nul)
          wcount = wcount + 1
*
*check for ending of words to identify adverbs
          capword adverb                     :S(nod)
*
*check for the use of verbs listed under verblst
          ident(verb<capword>,capword) :S(vcase)
*
*check for stylistic words and remove them
stl       ident(delstyl<capword>,capword)
+                                            :F(char)
          output0 = "s" sentcount ":         " capword "(style)"
          d<capword> = d<capword> + 1
*count the no of delimiters in this text
          nodelstyl = nodelstyl + 1     :(getwd)
*
*check for funny characters
char      ident(delchar<capword>,capword)
+                                            :S(getwd)
* check for of/the type concepts
oftyp     capword special . holdel      :F(del)
          ident(capword,holdel)         :F(del)
          flag = 1
```

```
                                        :(pwd)
*
*case of specified verb
vcase      vout = capword
           output0 = "vout= " vout
           ident(nulline,out)               :F(flagv)
           out = replace(vout,ucs,lcs)      :(outp)
flagv      vflag = 1                        :(outp)

*
*check for delimiters
del        ident(delimiter<capword>,capword)     :F(pwd)
*
*if delimiter preceded by 'of' then this is  not a new concept
nod        nodelim = nodelim + 1
           output1 =  "s" sentcount ":                 " capword
           d<capword> = d<capword> + 1
           eq(flag,1)   eq(ofcount,wcount - 1)  :S(flg)
*
*do not output nonsensical concepts
*do not output specified expressions
*
*do not output empty conceptlist
nul        ident(nulline,out)                    :S(getwd)F(outp)
*
* test for paragraph existence, if yes set local paraph paramaters
paraph     newsentence len(6) . chars
           ident(chars,parspace)            :F(incr)
           par = par + 1
           tpar = tpar + 1
           output4 = "parag.no. " par " in sent no. " sentcount
           output4 = newsentence
           pflag = 0
           psent = 0
           newsentence len(6)  rem . sentence :(incr)
*
*output concepts in prolog database form and output concepts in readable form
outp       newout = out
           output0 = "out : " out
new        newout wpat =                    :F(outp1)
           concept = concept gap word ","   :(new)
outp1      concept = replace(concept,ucs,lcs)
           output2 = "textconcept( " hno "," par "," psent "," sentcount ","
+                    ccount  ",  " cptno " )."
           concept rtab(1) . nconcept
           output3 = "concept( " cptno ", [" nconcept "] )."
           output = ccount ":" cptno "." out
           ccount = ccount + 1
           cptno = cptno + 1
           concept =
           nconcept =
           out =
           eq(vflag,1)                  :F(getwd)
           vflag = 0
           out = replace(vout,ucs,lcs)  :(outp)
flg        flag = 0
           ofcount = 0
pwd        out = out gap word           :(getwd)
null       ident(nulline,out)           :F(outp)
           ccount = ccount - 1          :(sumry)
*
*print summary of findings for each sentence
sumry      output = "(concepts:"   ccount " & words: "  wcount  ")"
           tccount = tccount + ccount
           conno<sentcount> = ccount
           twcount = twcount + wcount
```

```
        output = "(total concepts so far: " tccount - 1  ")"
        output = "(total words so far: " twcount ")"
        wcount = 0                         :(read)
*
*print procedure
print
        output =
        output = "the no. of sentences is "  sentcount
        output = "the no. of words is " twcount
        output = "the no. of concepts found is " tccount - 1
        output = "the no of paragraph is " tpar
        output =
        output = "the no of delimiters found is " nodelim
        output =
        output = "the headings are as follows:"
        output3 = "/* heading(no,heading) */ "
        i = 0
prynt   output =  i "." heading<i>
        heading<i> rtab(1) . nheading<i>
        output3 = "heading( " i ",'" nheading<i> " '  )."
        i = i + 1
        ge(i,hno)                          :F(prynt)
        dtable = sort(convert(d,'array'))
        x = nodelim + nodelstyl
        i = 0
write   output1 = dtable<i,1> " = " dtable<i,2>
        i = i + 1
        le(i,x)                 :S(write)
        output1 =
        output1 = "no. of delimiters : " nodelim
        output1 = "no. of style word type : " nodelstyl
        i = 1
        output3 = "/* summary ( sentence no., no. of concepts identified )"
+                 " */"
swrit   output3 = "summary( " i "," conno<i> ")."
        i = i + 1
        le(i,sentcount)                 :S(swrit)
        output3 = "/* textstats (no.of sentences,no. of headings,"
+                 "no.of concepts,no.of words,no. of paragraphs ) */"
        output3 = "textstats( " sentcount ", " hno ", " tccount - 1
+                 ", " twcount ", " tpar " )."
*
proend  terminal = '************ end *****************' :(end)
*
*signal error if heading not nominalised
error1  terminal = 'heading does not conform to the requirement:'
        terminal = 'i.e * followed by digit(s) in the following  sentence '
        terminal = sentence
end
```

```
*************************************************************************
*
* This program is the functional analyser submodule.
* It identifies the function of each sentence and records it in the
* form of PROLOG facts.
* It also looks for discourse markers in each sentence and records it in
* the form of PROLOG facts.
*
*************************************************************************
*
*initialisation
         &anchor = 0
         &trim = 1
         &trace = 200
         trace(.i,'value')
         sentcount = 0
         moves = array(40)

*
* files specifications
         input(.input, 'text1.res1')
         output(.output, 'text1func.pl')
         output(.output1, 'text1func.trace')
         output(.output2, 'text1status.pl')
*
*set up patterns to detect sentence
         sentend = '.!?'
         sent = (break(sentend)  any(sentend)  break(' ')) . sentence
         equations = '='
*
*set up word patterns
         ucs = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
         lcs = 'abcdefghijklmnopqrstuvwxyz'
         extras = "'-"
         cedille = "~"
         letters = ucs lcs extras
         number = any('123456789')
         wpat = break(letters) .  gap  span(letters)   . word
         introduction = 'INTRODUCE|INTRODUCTION|INTRODUCTORY|OVERVIEW|'
         methodology = 'TECHNIQUE|TECHNIQUES|METHODS|METHOD|METHODOLOGY|'
+                      'APPROACH|APPRAOCHES|FORMALISM|FORMALISMS|'
         discussion = 'DISCUSS|DISCUSSES|DISCUSSION|ARGUE|ARGUES|ARGUMENT|'
         conclusion = 'CONCLUSION|CONCLUSIONS|CONCLUDES|CONCLUDE|SUMMARY|'
         objectives = 'AIM|AIMED|OBJECTIVES|GOAL|CONCERNS|CONCERN|INTENDED|'
+                     'PURPOSES|PURPOSE|GOALS|AIMS|FOCUS|'
         results = 'RESULTS|RESULT|FINDINGS|FINDING|'
         investigation = 'INVESTIGATE|INVESTIGATION|'
*
         theme_markers = 'This paper' | 'This research' | 'This casestudy' |
+                   'The authors' | 'our approach' | 'Our approach' |
+                   'the author' | 'this research' | 'the authors' |
+                   'this paper' | 'we are developing' | 'This approach' |
+                   'this approach' | 'this methodology' | 'This casestudy' |
+                   'this case study' | 'This methodology' |
+                   'this casestudy' | 'This work' | 'this work'

*set up delimiters patterns
         l = 0
inv      investigation break('|') . w '|' =      :F(res)
         l = l + 1
         moves<l> = w                             :(inv)
res      v = 1
rl       results break('|') . w '|' =            :F(obj)
         l = l + 1
         moves<l> = v                             :(rl)
```

```
obl      objectives break('|') . w '|' =        :F(conc)
         l = l + 1
         moves<l> = w                          :(obl)
conc     o = l
col      conclusion break('|') . w '|' =        :F(disc)
         l = l + 1
         moves<l> = w                          :(col)
disc     c = l
disl     discussion break('|') . w '|' =        :F(meth)
         l = l + 1
         moves<l> = w                          :(disl)
meth     d = l
mel      methodology break('|') . w '|' =       :F(intro)
         l = l + 1
         moves<l> = w                          :(mel)
intro    m = l
inl      introduction break('|') . w '|' =      :F(score)
         l = l + 1
         moves<l> = w                          :(inl)
score    t = l
*
*processing
*
         output1 = "sentences identified as markers: "
         linecount = 0
*read a line and concatenate a blank before it and at end of line.
read     line = line ' '  input ' '             :F(proend)
         linecount = linecount + 1
         line sent =                            :F(read)
         sentcount = sentcount + 1
         terminal = "processing sentence " sentcount ":"
*
*is the sentence a title?
         ident(sentcount,1)                     :F(head)
         output = "function(" sentcount ",title)."
         output2 = "status(" sentcount ",title )."
+                                               :(read)
*is the sentence a heading?
head     sentence len(1) cedille number         :F(mark)
         output2 = "status(" sentcount ",heading )."
+                                               :(wid)
*test for discourse markers
mark     sentence theme_markers                 :F(bacg)
         output1 = 's.no ' sentcount ':' sentence
         output2 = "status( " sentcount ", theme_marker)."
*
*test for moves in the  sentence
*test for presence of = and assign sentence to equation category
equa     sentence equations                     :F(wid)
         output = "function(" sentcount  ", equation )."
+                                               :(read)
*find a match with the following functions
*set s=0 if no match else s=1
wid      s = 0
getwd    sentence wpat  =                       :F(bakg)
         word = replace(word,lcs,ucs)
         i = 1
move     mpat = moves<i>
         ident(word,mpat)                       :S(type1)
incr     i = i + 1
         gt(i,l)                                :S(getwd)F(move)
parm     s = 1                                  :(incr)
*check type of  sentence
type1    le(i,v)                                :F(type2)
         output = "function(" sentcount  ", investigation )."
```

```
type2      le(i,r)                                          :F(type3)
           output = "function("  sentcount  ", results )."
           i = r                                            :(parm)
type3      le(i,o)                                          :F(type4)
           output = "function("  sentcount  ", objectives )."
           i = o                                            :(parm)
type4      le(i,c)                                          :F(type5)
           output = "function("  sentcount  ", conclusion )."
           i = c                                            :(parm)
type5      le(i,d)                                          :F(type6)
           output = "function("  sentcount  ", discussion )."
           i = d                                            :(parm)
type6      le(i,m)                                          :F(type7)
           output = "function("  sentcount  ", methodology )."
           i = m                                            :(parm)
type7      le(i,t)                                          :F(bakg)
           output = "function("  sentcount  ", introduction )."
           i = t
           s = 1                                            :(getwd)
*
*background
bakg       eq(s,1)                                          :S(read)
bacg       output = "function("  sentcount  ", background )."
+                                                           :(read)
*end
proend     terminal = "no. of sentences processed is " sentcount
           output1 = " no. of lines processed is " linecount
+                    " no. of sentences processed is " sentcount
           terminal = "******** end *************"
*
end
```

THE EXPERT CLASSIFIER PROGRAM.

```
/*                                                      */
/* This program is the expert classifier program        */
/* which interprets the textual world model of a        */
/* given paper and assess the importance of each        */
/* sentence according to the rules set below            */
/*                                                      */


/* MAIN SECTION */
run:-
                    consult(text2txtcpt),
                    consult(text2func),
                    consult(text2status),
                    tell('text2imp_a.pl'),
                    classify_sentences.


/* CLASSIFICATION SECTION */
classify_sentences:-
                    title_imp,
                    heading_imp(1).



/* SET TITLE IMPORTANCE TO HIGH */

/* RULE 1: IF SENTENCE IDENTIFIED AS TITLE
           ASSIGN HIGH IMPORTANCE                              */

title_imp:-
                    assert_high(1).



/* DEFINE MEMBERSHIP */
member(Any,[Any|_]).
member(Any,[_|Tail]):-
                member(Any,Tail).



/* EXAMINING THE IMPORTANCE OF HEADINGS */

/* RULE 2:   IF SENTENCE IDENTIFIED AS HEADING
             EXAMINE ITS IMPORTANCE
   RULE 2.1: IF SENTENCE CONTAINS SALIENT INFORMATION FOR ABSTRACTING
             PURPOSES ASSIGN HIGH IMPORTANCE
             AND EVALUATE IMPORTANCE OF SENTENCES OF THIS HEADING
   RULE 2.2: IF HEADING IS IDENTIFIED AS IRRELEVANT
             ASSIGN ZERO IMPORTANCE                                      */

heading_imp(N):-
                textconcept(N,0,0,S_no,_,_),
                function(S_no,Ftype),
                (( member(Ftype,[introduction,conclusion,methodology,
                                 objectives]) )
                        -> ( assert_high(S_no),
                             process_p_r_h(N)
                           )
                        |  ( assert_zero(S_no),
                             process_s_nr_h(N)
                           )
                 ),
                 Next_Heading_No is N + 1,
                 find_heading(Max),
```

```
                    -> (  heading_imp(Next_Heading_No) )
                    |    output ).


/* PROCESSING PARAGRAPHS OF IMPORTANT HEADINGS */
/* note beginning of paragraph 1 and last paragraph and examine importance*/

/* RULE 2.1.1: IF SENTENCE IDENTIFIED AS 1ST SENTENCE OF PARAGRAPH 1
               ASSIGN HIGH IMPORTANCE
   RULE 2.1.2: IF SENTENCE IDENTIFIED AS LAST SENTENCE OF PARAGRAPH 1
               ASSIGN HIGH IMPORTANCE
   RULE 2.1.3: IF SENTENCE IDENTIFIED AS 1ST SENTENCE OF LAST PARAGRAPH
               ASSIGN HIGH IMPORTANCE
   RULE 2.1.4: IF SENTENCE IDENTIFIED AS LAST SENTENCE OF LAST PARAG.
               ASSIGN HIGH IMPORTANCE                                   */

process_p_r_h(N):-
        check_pl_sl(N,Startl),
        check_pl_ls(N,Plls),
        Startl_Medium is Startl + 1,
        Endl_Medium is Plls - 1,
        process_medium(Startl_Medium,Endl_Medium),
        check_lp_sl(N,Lastp,Lpsl),
        check_lp_ls(N,Lastp,End2),
        Start2_Medium is Lpsl + 1,
        End2_Medium is End2 - 1,
        process_medium(Start2_Medium,End2_Medium),
        Beg is Plls + 1,
        End is Lpsl - 1,
        check_in_between_para(Beg,End).


/* PROCESS SENTENCES  BETWEEN 1ST AND LAST PARAGRAPHS */

/* RULE 2.1.5: IF SENTENCE IDENTIFIED AS FUNCTIONAL
               ASSIGN HIGH IMPORTANCE

   RULE 2.1.6: IF SENTENCE BELONGS TO IMPORTANT HEADING BUT NOT FUNCTIONAL
               ASSIGN MEDIUM IMPORTANCE                                 */


process_medium(Beg,End):-
        ( Beg =< End )
                -> ( function(Beg,Ftype),
                     (( member(Ftype,[introduction,conclusion,methodology,
                                      objectives]
                        ) )
                           -> assert_high(Beg)
                           |  assert_medium(Beg)
                     ),
                     New_Beg is Beg + 1,
                     process_medium(New_Beg,End)
                   )
                |  true .


not(Goal):-call(Goal),!,fail.
not(Goal).


/* FINDING  PARAGRAPH AND SENTENCES BOUNDARIES */
find_last_sentence(Max):-
```

```
                    not((textconcept(_,_,_,Any,_,_), Any>Max )).


find_heading(Max):-
                    textconcept(Max,_,_,_,_,_),
                    not((textconcept(Any,_,_,_,_,_), Any>Max )).


gen_last_sp(N,Large):-
                    textconcept(N,1,Large,_,_,_),
                    not((textconcept(N,1,Any,_,_,_), Any>Large )).


gen_last_p(N,Large):-
                    textconcept(N,Large,_,_,_,_),
                    not((textconcept(N,Any,_,_,_,_), Any>Large )).


gen_last_s(N,Lastp,Lasts):-
                    textconcept(N,Lastp,Lasts,_,_,_),
                    not((textconcept(N,Lastp,Any,_,_,_), Any>Lasts )).


check_pl_sl(N,S_no):-
                    textconcept(N,1,1,S_no,_,_),
                    assert_high(S_no).


check_pl_ls(N,S_no):-
                    gen_last_sp(N,Lastsp),
                    textconcept(N,1,Lastsp,S_no,_,_),
                    assert_high(S_no).


check_lp_sl(N,Lastp,S_no):-
                    gen_last_p(N,Lastp),
                    textconcept(N,Lastp,1,S_no,_,_),
                    assert_high(S_no).


check_lp_ls(N,Lastp,S_no):-
                    gen_last_s(N,Lastp,Lasts),
                    textconcept(N,Lastp,Lasts,S_no,_,_),
                    assert_high(S_no).



/* PROCESS SENTENCES BETWEEN PARAG. 1 AND LAST PARAG. OF IMPORTANT HEADING */

/* RULE 2.1.7: IF SENTENCE BELONGS TO IMPORTANT HEADING
               AND IDENTIFIED AS FUNCTIONAL
               ASSIGN HIGH IMPORTANCE
   RULE 2.1.8: IF SENTENCE BELONGS TO IMPORTANT HEADING
               BUT NOT IDENTIFIED AS FUNCTIONAL OR POSITIONALLY IMPORTANT
               ASSIGN LOW IMPORTANCE
   RULE 2.1.9: IF SENTENCE BELONGS TO IMPORTANT HEADING
               AND CONTAINS AUTHOR MARKER
               ASSIGN MEDIUM IMPORTANCE.                                   */

check_in_between_para(Beg,End):-
               Beg =< End,
               check_function(Beg),
               New_Beg is Beg + 1,
               check_in_between_para(New_Beg,End).
```

```
check_in_between_para(Beg,End):- true.


check_function(Beg):-
                    function(Beg,Ftype),
                    member(Ftype,[introduction,conclusion,methodology,
                                         objectives]),
                    assert_high(Beg).

 check_function(Beg):-
                    status(Beg,theme_marker),
                    assert_medium(Beg).

check_function(Beg):-
                    assert_low(Beg).


/* PROCESS SENTENCES OF NOT RELEVANT HEADINGS */

/* RULE 2.3.1: IF SENTENCE BELONGS TO NOT IMPORTANT HEADING
               ASSIGN ZERO IMPORTANCE                                  */

process_s_nr_h(N):-
        textconcept(N,0,0,Beg,_,_),
        Next_heading is N + 1,
        find_heading(Max),
        (( Next_heading =< Max )
                -> ( textconcept(Next_heading,0,0,End,_,_),
                     New_Beg is Beg+1,
                     New_End is End-1,
                     allocate_zero_imp(New_Beg,New_End)
                   )
                |  ( find_last_sentence(Last),
                     New_Beg is Beg+1,
                     allocate_zero_imp(New_Beg,Last)
                   )
         ).


allocate_zero_imp(Beg,End):-
        (Beg =< End)
                -> ( assert_zero(Beg),
                     New_Beg is Beg + 1,
                     allocate_zero_imp(New_Beg,End)
                   )
                |  true.


output:-
        told,
        reconsult('text2imp_a.pl'),
        tell('text2cotext_a.res'),
        output_high,
        output_medium,
        output_low,
        output_zero,
        told.


output_high:-
        write('List of sentences which are highly important:'),nl,
        importance(S_no,high),
```

```
        write(','),
        fail.


output_high:-
        true.



output_medium:-
        nl,write('List of sentences which are of medium importance:'),nl,
        importance(S_no,medium),
        write(S_no),
        write(','),
        fail.


output_medium:-
        true.

output_low:-
        nl,write('List of sentences which have a low importance:'),nl,
        importance(S_no,low),
        write(S_no),
        write(','),
        fail.


output_low:-
        true.


output_zero:-
        nl,write('List of sentences which have zero importance:'),nl,
        importance(S_no,zero),
        write(S_no),
        write(','),
        fail.

output_zero:-
        true.



assert_high(S_no):-
        write('importance('),
        write(S_no),
        write(',high).'),nl.

assert_medium(S_no):-
        nl,write('importance('),
        write(S_no),
        write(',medium).'),nl.

assert_low(S_no):-
        nl,write('importance('),
        write(S_no),
        write(',low).'),nl.

assert_zero(S_no):-
        nl,write('importance('),
        write(S_no),
        write(',zero).'),nl.
```

Sentence No.	Criteria of selection	Sentence
1	title	The Characteristic Error Approach to Conflict Resolution.
2	heading, introduction	Introduction.
3	position	Conflict resolution is the integration of alternate opinions about the state of the world into a single coherent world view.
7	position	Their approach to conflict resolution therefore consists of evaluating the classification errors which are most likely to occur in the given context.
9	methodology, theme_marker	Most of this work takes a numeric approach to conflict resolution using, for example, Bayesian or Dempster-Schaefer techniques.
16	methodology, objectives, theme_marker	This paper describes a technique for conflict resolution (termed the characteristic error method) whose goal is to capture the reasoning process behind this form of conflict resolution.
20	position	In other words, the knowledge sources themselves are assumed to provide all of the necessary information for the system to successfully resolve the conflict (given the local context).
21	heading, methodology	The characteristic error approach.
22	position	The term characteristic error was chosen with the realization that certain error conditions are 'characteristic' of the environment in which the knowledge source expects to operate.
24	position	This is possible because information about characteristic error classes enables the resolution of a specific conflict to be dependent upon the immediate context of the problem.
55	position	The items listed above serve as basic information for knowledge (rule) based management of the conflict.
61	position	Strategy setting is implemented by moving error terms from characteristic to uncharacteristic or neutral, thereby altering the source's a priori opinion toward this error type.
122	heading, conclusion	Conclusions/Future work.
123	position	The characteristic error paradigm allows for other types of contextural and programmatic information to be easily added to the system.
125	position	These actions alter the relative 'push' given to a particular classification and can substantially alter the final characteristics of the classified image.
126	position, theme_marker	This paper presents a unique means of combining the opinions of divergent experts in a uniform manner using only their own a priori knowledge about their own characteristics.
128	position	It is planned that the upgraded MSIAS will serve as the testbed for this technique, after it is given the ability to evaluate structural information.

Figure 6.5 Paper1: sentences of high importance.

APPENDIX F

REFERENCES OF THE PAPERS USED IN INFORMEX.

Paper1:

Ferrante, R. D. (1985) "The Characteristic Error Approach to Conflict Resolution". Proceedings of the 9th International Joint Conference on Artificial Intelligence, vol 1, 331-334

Paper 2:

Dilger, W. and Kippe, J. (1985) "COMODEL: A Language for the Representation of Technical Knowledge". Proceedings of the 9th International Joint Conference on Artificial Intelligence, vol 1, 353-358

Paper 3:

Nguyen, T. A., Perkins, W. A., Laffey, T. J. and Pecora, D. (1985) "Checking an Expert Systems Knowledge Base for Consistency and Completeness", Proceedings of the 9th International Joint Conference on Artificial Intelligence, vol 1, 375-378

Paper 4:

Cline, T., Fong, W. and Rosenberg, S. (1985) "An Expert Advisor for Photolithography", Proceedings of the 9th International Joint Conference on Artificial Intelligence, vol 1, 411-413

Paper 5:

Milne, R. (1985) "Fault Diagnosis through Responsibility", Proceedings of the 9th International Joint Conference on Artificial Intelligence, vol 1, 423-425

APPENDIX G

CODE LISTINGS AND OUTPUT OF THE EDITOR PROGRAM

```
**********************************************************************
* this program is the textual editor of INFORMEX.
* It replaces all punctuation marks ',;:' found in the paper by pmarks.
* It also includes two functions (procl & repl) which remove
* the dot from a set of specified words.
**********************************************************************
*
*initialisation
         &anchor = 1
         &trim = 1
         &dump = 1
         &trace = 100
         trace(.line)
         &profile = 2
         slist = array('0:20')
         define('repl(s1,s2,s3)c,t,findc')
         define('procl(line)a,b,c,d,line1,flag')
*
*files specification
         input(.input, 'text5.orig')
         output(.output, 'text5.resl')
*
*set up patterns
         speech = any(';,:"()')
         punct = ';,:"()[]*'
         letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ&'
+                   'abcdefghijklmnopqrstuvwxyz'
         number = '1234567890'
         nb = any('1234567890')
         dot = "."
*
main program
*        terminal = "program started"
*
*specified words
         ss2 = 'etc.|i.e.|i.e|ie.|fig.|Fig.|e.g.|e.g|eg.|cf.|al.|&|'
         i = 0
spat     i = i + 1
         ss2 break('|') . s  '|' =      :F(part1)
         slist<i> = s                   :(spat)
*
*process each line and replace dots in digits by '
part1    count = 0
read     oline = input                  :F(proend)
         count = count + 1
proc     line = oline
         newline = procl(line)
*
*remove dots from etc. i.e. i.e fig. Fig. and others
part2    l = 1
         line = newline
try      newline = repl(line,slist<l>,s3)
         l = l + 1
         line = newline
         le(l,i - 1)                  :S(try)
*
*replace punctuation marks by pmarks
part3    terminal = "line " count
more     newline  break(punct) . keep span(punct) rem . rest =
+           keep ' pmarks ' rest        :S(more)
op       output =  newline              :(read)
proend   terminal = 'no. of lines processed is ' count
         terminal = 'end'               :(end)
*
*function repl
```

```
*execute string by string replacement
*scan s1 for s2 and replace s2 by s3
*replace i.e and i.e. into ie and fig. and Fig. into fig
repl     s2 = slist<l>
         s2 len(1) . c =                  :F(freturn)
         findc = break(c) . t len(1)
         s2 = pos(0) s2
repl1    line findc =                     :F(repl2)
         line s2 =                        :F(repl3)
         eq(l,1)                          :F(ie)
         s3 = "etc"                       :(carry)
ie       le(l,4)                          :F(fig)
         s3 = "ie"                        :(carry)
fig      le(l,6)                          :F(eg)
         s3 = "fig "                      :(carry)
eg       le(l,9)                          :F(cf)
         s3 = "eg"                        :(carry)
cf       le(l,10)                         :F(al)
         s3 = "cf"                        :(carry)
al       le(l,11)                         :F(and)
         s3 = "al"                        :(carry)
and      le(l,12)                         :F(error2)
         s3 = "and"
carry    repl = repl t s3                 :(repl1)
repl3    repl = repl t c                  :(repl1)
repl2    repl = repl line                 :(return)
error2   terminal = "l equal to " l " and no corresponding replacement found"
                                          :(return)
repl.end
*
*function procl
procl     linel =
          flag = 0
again     line break(number) . a  span(number) . b rem . c  :F(printl)
          flag = 1
          c dot rem . d =                         :F(print)
          line = d
          linel = linel a b "'"                   :(again)
print     line = linel line
          procl =  line                           :(return)
printl    eq(flag,0)                              :S(print2)
          linel = linel d
          procl = linel                           :(return)
print2    procl =  line                           :(return)
procl.end
*
*end of program
end
```

PAPER 1.

~0 The Characteristic Error Approach to Conflict Resolution.

~1 INTRODUCTION.

Conflict resolution is the integration of alternate opinions about the state of the world into a single coherent world view. This operation is often performed by a separate entity pmarks knowledge source pmarks specifically designed for this purpose. For example pmarks a system which performs surface material classification in a remote sensing application may encounter conflicts when knowledge sources have differing opinions about the most appropriate classification of a given region. Remote sensing experts attempting to resolve these conflicts may employ such knowledge as 'the spectral signature of a road may have been shifted due to the presence of vehicles or an oil slick on the surface' or 'a portion of the road may have been obscured by trees'. Their approach to conflict resolution therefore consists of evaluating the classification errors which are most likely to occur in the given context.

Conflict resolution has been treated by a number of different authors who use various techniques and terminology such as reasoning about uncertainty or 'the integration of knowledge from disparate sources ' pmarks 4 pmarks . Most of this work takes a numeric approach to conflict resolution using pmarks for example pmarks Bayesian or Dempster-Schaefer techniques pmarks 5 pmarks . However pmarks Bayesian techniques assume the ability to generate a priori a table of conditional probabilities pmarks ie the probability of A given B pmarks and only allow the source to posit a single probability at a time. Dempster-Schafer techniques pmarks on the other hand pmarks while allowing multiple beliefs pmarks assume that all possibilities are independent pmarks an assumption not often satisfied in reality. Other approaches to conflict resolution represent a more semantic knowledge based approach and do not rely exclusively upon a numeric algorithm to resolve conflicts. Examples of these approaches include Cohen's theory of endorsements pmarks 2 pmarks and Tenenbaum and Barrow's Interpretation Guided Segmentation pmarks 6 pmarks . Even these approaches however pmarks exhibit some limitations. They lack extensibility pmarks function in limited domains and require either extensive preprogramming or interactive assistance for operation.

This paper describes a technique for conflict resolution pmarks termed the characteristic error method pmarks whose goal is to capture the reasoning process behind this form of conflict resolution. Characteristic error conflict resolution integrates two closely related concepts pmarks reasoning about uncertainty and constraint propagation. Reasoning about uncertainty determines the relative validity of beliefs pmarks while constraint propagation uses the presence of contradictions to restrict the range of applicable choices. The characteristic error approach is unique in treating each individual knowledge source as a separate entity whose overall reasoning validity may be entirely self-determined. In other words pmarks the knowledge sources themselves are assumed to provide all of the necessary information for the system to successfully resolve the conflict pmarks given the local context pmarks .

~2 THE CHARACTERISTIC ERROR APPROACH.

The term characteristic error was chosen with the realization that certain error conditions are 'characteristic' of the environment in which the knowledge source expects to operate. Conflict resolution based upon the mally tacitly assumed by the expert during the knowledge acquisition process pmarks both to develop the context for a given situation and to resolve differences of opinion. This is possible because information about characteristic error classes enables the resolution of a specific conflict to be dependent upon the immediate context of the problem.

Characteristic error is based upon the idea that pmarks within a multi-source system pmarks each individual knowledge source is pmarks to some extent pmarks the best judge of its own performance. The motivation for this becomes apparent if one considers the factors which are normally embedded in the process of developing any given expert system.
The knowledge of the context in which it expects to operate pmarks ie pmarks an expectation of the form which the environment will take and the types of interrelationships that may occur.
The type of boundary or limiting conditions it expects to encounter pmarks ie pmarks what information can occur in the environment which is close to what is expected but is actually different.
The type of misclassifications it postulates can occur pmarks ie pmarks what information may be present in the environment that is often confused with the desired information.

Implementing the concept of characteristic error requires that each knowledge source posit an a priori estimate of the error classes which it might encounter. This estimate is an explicit ranking of its expectations for the immediately local context pmarks and is independent of the strength of the knowledge source's opinion in any particular instance of its application. It is based only upon the expert's knowledge of the underlying structure of its environment and decisions. For example pmarks a region growing knowledge source pmarks designed for road finding within an image understanding system pmarks will know that local classification error may occur because a portion of the road may be partially obscured by trees. By the same token pmarks it will also be aware of the fact that it is very uncharacteristic for any region designated as deep water to actually be a road. The same source will pmarks however pmarks have no real opinion as to the likely misclassification of shallow water. These considerations are independent of its specific belief of the existence of roads in any particular instance.

This error class information is dependent only upon the context assumed by the knowledge source. It is independent of the actual context of any particular instance in which the knowledge source is utilized. In other words pmarks the knowledge source's characteristic error list explicitly states the context which was assumed to be part of the underlying structure of the environment during knowledge acquisition. Therefore pmarks no a priori controls need be imposed upon the overall system using the knowledge source pmarks since all context information is explicitly stated by the knowledge source itself.

To illustrate these points pmarks Table 1 shows the structure of two road finders. The road finder which anticipates being called upon to process satellite information might expect partial road covering by treetops. The ground level road finder on the other hand pmarks has no expectation of such a conflict.

Such a priori identification of errors also promotes the graceful performance degradation of a group of knowledge sources pmarks if and when any knowledge source is operating outside of its area of expertise. A source operating in an unfamiliar area should not have the conflicting class appearing anywhere in its characteristic error list. This absence serves as an immediate flag for the conflict resolver to investigate the situation further.

The minimum implementation of the characteristic error technique requires the following information pmarks
The characteristic a priori errors expected to be encountered by the knowledge source in the environment
The knowledge source's strength of opinion in a particular instance
A methodology for evaluating the error list and determining the context.
The system works by comparing the characteristic error lists of the conflicting knowledge sources. This determines the conflict's local context. This context is pmarks in essence pmarks an indicator of the similarity and structure of these error lists. The implementation is such that similar error lists create a context controlling the relative strength of opinion necessary from the knowledge sources to alter a decision. For example pmarks if all sources agree that a given error is characteristic of the situation pmarks it will not take a large pmarks absolute pmarks opinion strength to shift the decision in another direction. By the same token pmarks if the context is relatively incoherent pmarks a knowledge source's opinion must be stronger for the opinion to change. The interaction of these two factors is the basis for the implementation of the most likely error approach.

The items listed above serve as basic information for knowledge pmarks rule pmarks based management of the conflict. However pmarks given the existence of a central pmarks top level pmarks conflict resolving system pmarks additional rule based manipulation is possible. The rules used for this manipulation/management of the conflict will be specific to a given system pmarks and perhaps even a specific instance of its use. It should be noted that these manipulations require no overall ranking of the knowledge sources in the system. Their general thrust is to drive the system results in a specific direction thereby establishing a strategy for the system. This strategy embodies more general criteria ie always err on the side of caution. Strategy setting is implemented by moving error terms from characteristic to uncharacteristic or neutral pmarks thereby altering the source's a priori opinion toward this error type.

~3 AN APPLICATION OF CHARACTERISTIC ERROR.

This section presents an example application of the characteristic error concept. The focus of our application is TASC's Multi-Spectral Image Analysis System pmarks MSIAS pmarks pmarks 3 pmarks . MSIAS is a knowledge based system that interprets cartographic surface attributes using information obtained from remotely sensed imagery. The MSIAS classifier utilizes the relative spectral signature of an individual pixel location pmarks ie pmarks its relative intensity in various spectral bands pmarks to determine its surface material class eg concrete pmarks vegetation pmarks water. Conflicts will only occur when new knowledge sources pmarks using other criteria pmarks are introduced into the system. These knowledge sources may start with the latter spectral identification pmarks but may base their own classification opinion upon some alternate criteria pmarks eg structural measures pmarks The important feature of these knowledge sources their individual results. These results generate alternate possible classifications for any given pixel/region. In the example below pmarks alternate classifications represent the conflicts. The process of obtaining a final classification is conflict resolution. Using the nomenclature of Clancy pmarks 1 pmarks this is a problem of heuristic classification.

It is planned that the conflict resolver will be activated late in the image classification process. The late activation occurs because a number of different knowledge sources must be polled for a conflict to occur. Conflict resolution is only applied to those objects which are the subject of conflicting opinions. The frequency of these conflicts will increase as the semantic level of the program increases. That is pmarks the use of the resolver will be lower pmarks on a percentage basis for pixels than for regions. Upon activation pmarks the resolver first looks at the pixels/regions immediately adjacent to the one currentiy under investigation and searches for other opinions as to this pixel's classification. Every adjacent pixel is marked with both a specific classification and characteristic error list. The list summarizes the characteristic error classes posited by the knowledge source which identified it. This information serves as the basis for conflict resolution pmarks attempting to address such issues as pmarks How do the other conflicts and their characteristic error classes relate to the identification of this pixel? Do they reinforce the current designation or contest it and in what way.
The error classes may also be completely mute about this possible pixel classification.

As shown in Figure 1 pmarks a pixel can have up to eight nearest neighbors in a digitized image.

There are two extreme cases for the possible opinions of these neighbors pmarks First the original pixel classification pmarks at x pmarks may be a member of the 'uncharacteristic' error class of each and every neighboring pixel pmarks second pmarks the pixel's class appears in the 'characteristic' error class of all neighboring pixels.

In the first case pmarks the conflict resolver would not pmarks under any circumstances pmarks alter the pixel's classification pmarks as as all of the knowledge sources agree that it is unlikely that this pixel is misclassified. Therefore pmarks even if all knowledge sources felt that their regions were worthy of extension pmarks no regions would be extended as they all have preagreed to abide by this pixel designation. That is pmarks they have all agreed that an 'x' class pixel is both something which they expect to see in the environment and something which could be a boundary of their regions. This corresponds to the general case in which an expert would say 'I use x as an indicator of this condition pmarks y is an indicator of an other condition pmarks x and y are rarely confused'.

In the second case however pmarks the conflict resolver will change the classification of this pixel to the competing class with the highest relative likelihood pmarks opinion strength pmarks . Such a change will occur even if the relative likelihood is fairly low in an absolute sense. This is because there is complete agreement that initial misclassification of this pixel is often characteristic of the situation. This corresponds to the general case in which an expert would say ' I use x as an indicator of this condition pmarks however y is often confused with x pmarks so if you have doubts about y you may also use it as an indicator'.

Fundamentally pmarks the characteristic error paradigm example pmarks for seven neighbors to agree is less strong a case than all eight pmarks or if one has the class on its uncharacteristic error list it can possibly be excluded.

There are three basic ways of performing this integration. The simplest method merely tabulates frequency of appearance for the pixel class in question pmarks combines the two frequencies pmarks characteristic and uncharacteristic pmarks and decides to go with the highest combination of relative frequency and opinion strength. This is a relatively simple approach with weights being heuristically determined.

The second approach judges the relative coherence of these opinions. For example pmarks if three divergent region types had found that the class in question was on their characteristic error list pmarks it is a stronger case for misclassification than if a single type of region had reached the same conclusion three tunes. This approach attempts to compensate for systematic bias on the part of the experts.

The third method which might be considered involves merely discounting the opinion of any region having this class on its uncharacteristic error list. This is however not a valid method as in such a situation a region is actually giving a strong vote for retaining the current classification.

The characteristic error paradigm allows the extraction of information about the characteristics of the knowledge source itself. First pmarks it is important to recall that although we have been discussing three classes there actually are four specific class lists in existence. The fourth class being the class pmarks es pmarks for which the structural knowledge source pmarks region grower pmarks was actually designed. For example pmarks a road grower would have concrete/asphalt in this class pmarks some other examples appear in Table 2 .

The characteristic error class is a group of pixel identifications in which one would expect an error to occur in the environment. As shown in the table pmarks if one was growing highways pmarks it would be expected that some road areas may have been misclassified due to overhanging trees pmarks oil slicks etc . The 'uncharacteristic' error classes are the ones which are expected to form a hard boundary between the given class and an alternate. These would be such classes as deep water and rock for roads. Neutral classes are classes which may exist in the expected environment but about which no intelligent statement may be made ie very shallow water pmarks patchy snow for roads etc . As was discussed in section 2 pmarks manipulation of this list can be a significant action in and of itself and may also serve as an ideal place for changing context. For example pmarks if we know that an area had recently been flooded pmarks we would move shallow water to the characteristic error set for all low lying classes.

The explicit designation of the classes expected to occur in the environment has a very useful side effect pmarks It is now possible to determine that a knowledge source is operating outside of its area of expertise. For example pmarks an airport hangar area grower would expect fuel storage tanks appearing next to it as a hard boundary. The highway expert pmarks on the other hand pmarks would not expect such a structure in any immediately adjacent pixel. Fuel storage tanks would therefore not appear anywhere on a highway grower's list of error classes pmarks not in 'characteristic' 'uncharacteristic' nor in 'neutral'. Therefore if the conflict resolution process retained the storage tank would then try to find alternate explanations of the region/pixel which it had designated as 'highway' pmarks by eliminating the highway designation and asking for alternate opinions.

~4 CONCLUSIONS/FUTURE WORK.

The characteristic error paradigm allows for other types of contextural and programmatic information to be easily added to the system. For example pmarks if it were very important to catch the existence of highways in a particular application pmarks we would move more classes into the highway grower's 'characteristic' list. These actions alter the relative 'push' given to a particular classification and can substantially alter the final characteristics of the classified image.

This paper presents a unique means of combining the opinions of divergent experts in a uniform manner using only their own a priori knowledge about their own characteristics. The characteristic error method is a planned enhancement to the MSIAS system. It is planned that the upgraded MSIAS will serve as the testbed for this technique pmarks after it is given the ability to evaluate structural information.

PAPER 2.

~1 INTRODUCTION.

The first and most well-known expert systems were developed for medical and scientific applications. Only since a few years technical applications came in sight. This type of application requires in first place new types of knowledge representation. Thus pmarks a number of papers appeared on this topic following the ideas of the basic work of Hayes pmarks 1 pmarks . They deal mainly with the modelling of physical processes going on in technical systems pmarks cf pmarks 2 pmarks 3 pmarks 4 pmarks 5 pmarks 6 pmarks 7 pmarks 8 pmarks 9 pmarks 10 pmarks . All of them reason about physical processes making use of a qualitative description of physical entities. As possible applications are considered diagnosis and prognosis in technical systems and control of technical processes.

In our approach we stress attention to the system theoretical principles of knowledge modelling and on the representation of geometric knowledge. On the basis of some principles of the modelling of technical systems pmarks section 2 pmarks we give an introduction to the knowledge representation language COMODEL pmarks section 3 pmarks . Its main feature is the component oriented description of technical systems and processes covering geometrical and process knowledge by the same means of description. Its application will be illustrated by some examples pmarks section 4 pmarks .

~2 MODELLING OF TECHNICAL SYSTEMS.

The different works on modelling of technical systems mentioned above have in common that they do not describe a technical system globally pmarks rather they start from context independent descriptions of single parts and their functions and try to infer descriptions of aggregates from those of the parts. The descriptions are based on variables pmarks eg temperature pmarks pressure pmarks cross-section etc pmarks values of variables and constraints. Forbus pmarks 2 pmarks 3 pmarks 4 pmarks takes the process pmarks causing changes of physical situations pmarks as the basic concept of his description pmarks in de Kleer's and Brown's approach pmarks 5 pmarks 6 pmarks the physical component pmarks whose behavior is characterized by confluences pmarks plays this role pmarks and Kuipers' and Kassirer's work pmarks 7 pmarks 8 pmarks 9 pmarks is based on 'universal' constraints pmarks related to different types of variables. Raulefs pmarks 10 pmarks attaches importance to different degrees of granulation in the qualitative description and to reasoning about time.

Our approach has the following basic features.
1' Component oriented descripton. A technical system is regarded as composed of a finite number of components that correspond to the parts of a real system and can be accumulated to aggregates.
2' Qualitative and quantitative representation of the properties of physical objects.
3' System theoretical taxonomy for classifying the variety of components and their connections pmarks component types pmarks in-

4' Uniform description of the function and
the geometric properties of components.

With respect to 1' and 2' pmarks our approach is related to that of de Kleer and Brown and of Kuipers and Kassirer. Component orientation allows modular descriptions of large aggregates pmarks therefore such a description can be easily modified and adapted to other configurations. The behaviour of components and aggregates is defined by means of constraints on physical 'properties pmarks or variables. With respect to 3' and 4' pmarks we differ from the works mentioned above. As a consequence of a component oriented approach pmarks the components can and should be classified in a systematic way according to the principles of system engineering. Geometric properties of components are of interest in so far as they are essential parts of the behaviour of the components pmarks and for this reason they are described in the same way as the functional properties.

~3 ELEMENTS OF COMODEL.

In the COMponent Oriented DEscription Language pmarks COMODEL pmarks a technical system is viewed as a set of phenomenons pmarks that may be observable by automatic measurement or by human senses. Phenomenons can be connected by relations. This results in a network consisting of phenomenons as edges and two types of nodes pmarks called components and interactions pmarks cf figure 1 pmarks . Components pmarks interactions and phenomenons are the basic objects of our description language pmarks they are related to the parts of a technical system pmarks the connections of these parts and the observable physical properties in the real world.

A COMODEL description of a real technical system consists of a number of object-definitions pmarks defining the three basic objects and a compound object called aggregate.

A phenomenon is a collection of a finite number of physical properties pmarks combined from a particular point of view.

A physical property is a part of a phenomenon. It may be observable or not.

A component is an object with a finite number of gates and relations. To each gate a phenomenon is assigned. The gates serve as links between components and interactions. The relations describe connections between values of different physical properties in the gates' phenomenons or define values of physical properties.

An interaction is an object with a finite number of gates and relations pmarks like a component. It differs from a component by the fact that its relations cannot define values pmarks ie interactions do not have characteristic or intrinsic values.

Gates are parts of components and interactions and gates induces identity on the corresponding phenomenons of the gates.

Relationships between physical properties are described by relations. Simple relations are equal pmarks unequal pmarks proportional etc.

Connection of components and interactions results in larger entities called aggregates. In the same way aggregates are combined to form bigger aggregates up to the whole technical system pmarks that can be viewed as one aggregate consisting of a hierarchy of pmarks sub- pmarks aggregates.

In the definition of phenomenons pmarks components pmarks interactions and aggregates one may refer either to build-in types pmarks PHENOMENON pmarks COMPONENT pmarks INTERACTION pmarks AGGREGATE pmarks or to user-defined objects pmarks symbolized by the preterminal symbol identifier pmarks . These objects form a hierarchy representing the taxonomy of system theoretical terms. Each object describes some aspects of a part of a real system at a particular level of generality.

~4 SAMPLE APPLICATION OF COMODEL.

As a paradigm for the representation of technical systems we take the nitric acid cooler of Lapp/Powers pmarks 11 pmarks shown in figure 2 . The function of this process is to cool a hot nitric acid stream before reacting it with benzene to form nitrobenzene. One top event for the system is a high temperature in the nitric acid reactor feed pmarks since this could cause a reactor runaway.

In order to show the applicability of COMODEL to describe a broad variety of technical mechanisms we choose two parts of the nitric acid cooler - the heat exchanger as energy flow system and a piston and cylinder pmarks taken from the cooling water pump pmarks which is assumed to be a piston pump pmarks as an example for geometric interactions.

~4 1 Heat Exchanger.

The heat exchanger is an aggregate with four gates pmarks two fluid streams passing through and interchanging heat energy. It can be viewed as constructed of two equal components pmarks each a fluid stream with heat energy flowing off or flowing to pmarks cf figure 3 pmarks . Therefore pmarks such a component is an object with three gates pmarks two fluid streams pmarks and one energy stream pmarks cf figure 4 pmarks . The relevant physical properties pmarks flow direction pmarks temperature pmarks heat stream pmarks are combined to phenomenons.

The heat exchanger component HE is GIVEN BELOW. The heat exchanger component has three gates pmarks two for the fluid stream pmarks M1 pmarks M2 pmarks and one for the heat stream pmarks E3 pmarks . The fluid stream passes through the gates M1 and M2 in opposite directions with respect to the centre of the component pmarks ie at one gate it enters the component and at the other it leaves the component. This is stated by the first of the fluid stream. The heat stream is proportional to the difference of temperatures pmarks ie to the loss or gain of temperature of the fluid stream. If the entrance temperature is greater pmarks less pmarks than the exit-temperature then the heat stream leaves pmarks enters pmarks the component. The average temperature of the fluid stream is equal to the temperature of the heat stream.

Fluid stream and heat stream are described by interactions. The interaction FLUIDSTREAM has two gates pmarks M1 pmarks M2 pmarks . The relation states that the values of corresponding physical properties occurring at gate M1 and gate M2 respectively are equal pmarks except for the values of the flow directions.

The interaction HEATENERGY has the two gates E1 and E2. The relations define some relationships between the physical properties included in the phenomenon ENERGYFLOW occurring at the gates of HEATENERGY. The difference of the temperatures at both gates determines the heat stream. No heat is lost between both gates pmarks therefore E1 Q = E2 Q. The heat flows from higher to lower temperature pmarks ie it enters pmarks leaves pmarks the interaction at E1 if E1 T > pmarks < pmarks E2 T. The same holds for gate E2 because of the opposite directions.

The aggregate HEATEXCHANGER pmarks cf figure 5 pmarks consists of two components of type HE and one interaction of type HEATENERGY. The list of gate-pairs define how the constituents of the aggregate are composed pmarks because a gate-pair represents an identification of two gates. For example HE1 is composed with HF by identification of the gates HE1-E3 and HF-E1. A pair like pmarks HE2-M1 M3 pmarks says that gate M1 of HE2 is 'free' pmarks ie not used for composition pmarks and becomes therefore a gate of the aggregate. The description of the aggregate gates is adapted from the description of the corresponding component gates.

~4 2 Piston and cylinder.

Piston and cylinder are the basic elements of the piston pump. They are viewed as two components and modelled by the two objects full cylinder and hollow cylinder pmarks cf figure 6 . The phenomenons join together geometric and kinematic properties.A hollow body pmarks eg a cylinder pmarks can have no bottoms pmarks pipe pmarks one bottom pmarks opened can pmarks or two bottoms pmarks closed can pmarks . The two components are given below.

In the definition of piston and cylinder radius pmarks length pmarks degrees of freedom and number of bottoms are bound to particular values pmarks XR pmarks XL pmarks TRANS-FR pmarks _ pmarks 1 pmarks . Every piston and cylinder has six degrees of freedom pmarks translation along the R-axis pmarks along the L-axis pmarks and along the R-axis after a 90o-rotation pmarks rotation around the R-axis pmarks around the L-axis pmarks and around the R-axis after a 90o rotation. These definitions describe aspects of individual objects from the class of all fullcylinders and all hollowcylinders respectively pmarks only the values for

The FULLCYLINDER-IN-HOLLOWCYLINDER-interaction describes the geometric relationship pmarks if the full cylinder is put into the hollow cylinder. It creates a new hollow cylinder pmarks called pump chamber pmarks with the length depending on the position of the piston. The piston retains two degrees of freedom in the resulting aggregate pmarks cf figure 7 .

The FULLCYLINDER-IN-HOLLOWCYLINDER-interaction has three gates pmarks F pmarks H pmarks R pmarks . It combines the phenomenons FULLCYLINDERPROPERTIES and HOLLOWCYLINDERPROPERTIES to form a new phenomenon HOLLOWCYLINDERPROPERTIES. Gates R and H have the same radius pmarks ie the radius of the interaction as a whole depends on the hollowcylinder which it is composed with. A similar relation holds for the length. The number of bottoms is 2 . Only two degrees of freedom are left pmarks translation along and rotation around the L-axis. The aggregate PUMPCHAMBER consists of two components of type CYLINDER and PISTON respectively and one interaction of type FULLCYLINDER-IN-HOLLOWCYLINDER. Here pmarks the interaction plays a central role. Gate FIT_R is the gate of the aggregate. The composition of the three constituents is obvious.

The definition of the aggregate reflects the fact that the behaviour of a pumpchamber can be described mainly by the interaction of a piston and a cylinder.

~5 CONCLUSION.

COMODEL is a component oriented approach to the representation of knowledge about technical systems. In our description of COMODEL we have attached importance to its ability to represent function and geometric properties of components in a uniform way.

It is our intention to use COMMODEL as a technical description language for expert systems for the area of technical systems pmarks in particular for interpretation of the states of those systems and diagnosis of failures.

For this purpose a COMODEL description represents a model of a technical system. This model together with process signals should be processed by an inference engine. At this point additional heuristic knowledge about symptoms of defects as well as strategies for the search for interpretation and diagnosis is needed.

PAPER 3.

~0 CHECKING AN EXPERT SYSTEMS KNOWLEDGE BASE FOR CONSISTENCY AND COMPLETENESS.

~1 INTRODUCTION.

The Lockheed Expert System pmarks LES pmarks is a generio rule-based expert system tool pmarks 1 pmarks pmarks similar to EMYCIN pmarks 2 pmarks that has been used as a framework to construct expert systems in many areas such as eleotronic equipment diagnosis pmarks design checking pmarks photo interpretation pmarks and hazard analysis. LES employs a combination of goal-driven and data-driven rules with the latter being attached to the factual database pmarks demons pmarks . One objective in the design of LES was to make it easy to use. Thus pmarks many debugging tools and aids were added to the LES program. One of these aids is the knowledge base completeness and consistency verification program called CHECK. Its purpose is to help a knowledge engineer check the knowledge base which he constructed for logically redundant rules pmarks conflicting rules pmarks subsumed rules pmarks missing rules pmarks unreachable clauses pmarks and deadend clauses. CHECK does not perform any syntax checking on the rules pmarks since this is done automatically when the rule files are loaded into the knowledge base. CHECK statically analyzes the knowledge base pmarks ie pmarks after the rules and facts are loaded into the knowledge base pmarks unlike TEIRESIAS pmarks which performs an assessment of rules in the setting of a problem-solving session pmarks 3 pmarks .

Suwa pmarks Scott pmarks and Shortliffe pmarks 4 pmarks have written a program for verifying that a set of rules comprehensively spans the knowledge base. This program was devised and tested within the context of the ONCOCIN system pmarks an EMYCIN-like system pmarks . Our work differs from theirs in that CHECK includes unreachable clauses and deadend clauses as two additional rule checking criteria. Furthermore pmarks CHECK produces a dependency chart and detects any circular rule chains. Also pmarks CHECK was devised and tested on a generic expert system with case-grammar rules and a 'frame' database. It has been used to analyze a wide variety of knowledge bases. CHECK does not take into account certainty faotors when checking the rule base.

~2 CHECKING FOR CONSISTENCY AND COMPLETENESS.

A static analysis of the rules can detect many potential problems and gaps that exist in a rule base. We now identify and give definitions for seven criteria that are used by CHECK to perform static analysis of any rule base constructed for use with LES. The first four criteria are concerned with potential problems pmarks whereas the last three criteria are concerned with gaps in the knowledge base.

~2 1 POTENTIAL PROBLEMS IN A KNOWLEDGE BASE.

By statically analyzing the logical semantics of the rules represented in LES's case grammar format pmarks CHECK can detect redundant rules pmarks conflicting rules pmarks rules that are subsumed by other rules pmarks and circular-rule chains. The following definitions for these four potential problems are used in CHECK.

Redundant rules pmarks two rules succeed in the same situation and have the same results. In LES pmarks this means that the IF parts of the two rules are equivalent pmarks and one or more THEN clauses are also equivalent. Because LES allows variables in rules pmarks equivalent means that the same specific object names can match their corresponding variables. For example the rule 'p pmarks x pmarks --> q pmarks x pmarks ' is equivalent to the rule 'p pmarks y pmarks --> q pmarks y pmarks ' pmarks where x and y are variables.

Conflicting rules pmarks two rules succeed in the same situation but with conflicting results. In LES pmarks this means that the IF parts of the two rules are squivalent pmarks but one or more THEN clauses are contradictory pmarks or one pair of IF clauses is contradictory while they have equivalent THEN clauses. For example pmarks the rule 'p pmarks x pmarks --> not pmarks q pmarks x pmarks ' is contradictory to the rule 'p pmarks x pmarks --> q pmarks x pmarks '.

Subsumed rules pmarks two rules have the same results pmarks but one contains additional constraints on the situations in which it will succeed. In LES pmarks this means one or more THEN clauses are equivalent pmarks but the IF part of cne rule contains fewer constraints and/or clauses than the IF part of the other rule. For example pmarks the rule ' pmarks p pmarks x pmarks and q pmarks y pmarks --> r pmarks z pmarks ' is subsumed by the rule 'p pmarks x pmarks --> r pmarks z pmarks '.

Circular rules pmarks a set of rules is a circular-rule set if the chaining of those rules in the set forms a cycle. For example pmarks if we had a set of rules as follows pmarks pmarks 1 pmarks 'p pmarks x pmarks --> q pmarks x pmarks ' pmarks pmark 'q pmarks x pmarks --> r pmarks x pmarks ' pmarks pmarks 3 pmarks 'r pmarks x pmarks --> p and the goal is r pmarks A pmarks where A is a constant pmarks then the system will enter an infinite loop at run time pmarks unless the system has a special way of handling circular rules.

~2 2 POTENTIAL GAPS IN A KNOWLEDGE BASE.

The development of a knowledge-based system is an iterative process in which knowledge is encoded pmarks tested pmarks added pmarks changed pmarks and both the knowledge engineer and the expert may have overlooked during the knowledge acquisition process. In LES pmarks we have found three situations indicative of gaps in the knowledge base. These three situations pmarks called pmarks 1 pmarks missing rules pmarks pmarks 2 pmarks unreachable clauses pmarks and pmarks 3 pmarks deadend clauses are described below.

Missing rules pmarks a situation in which some values in the set of possible values pmarks called legal values pmarks of an object's attribute are not covered by any rule's IF clauses pmarks ie pmarks the legal values in the set are covered only partially or not at all pmarks . A partially covered attribute can prohibit the system from attaining a conclusion or cause it to make a wrong conclusion when an uncovered attribute value is encountered during run time.

Unreachable clauses pmarks in a goal-driven production system pmarks a THEN clause of a rule should either match a goal clause or match an IF clause of another rule pmarks in the same rule set pmarks . Otherwise pmarks the THEN clause is unreachable.

Deadend clauses pmarks to achieve a goal pmarks or subgoal pmarks in LES pmarks it is required that either pmarks pmarks 1 pmarks the attributes of the goal clause are askable pmarks user provides needed information pmarks or pmarks 2 pmarks that the goal clause is matched by a THEN clause of one of the rules in the rule sets applying to that goal . If neither of these conditions is satisfied then the goal clause can not be achieved pmarks ie pmarks it is a 'deadend clause'. Similarly pmarks the IF clauses of a rule also must meet one of these two conditions pmarks or they are 'deadend clauses'.

~2 3 DEPENDENCY CHART AND CIRCULAR-RULE CHAINS DETECTION.

As a by-product of the rule checking pmarks CHECK generates a dependency chart which shows the interactions among the rules and between the rules and the goal clauses. An example of a dependency chart for a small problem is shown in Figure 1 . A ' pmarks ' indicates that one or more clauses in the IF part of a rule or a goal clause pmarks GC pmarks matches one or more clauses in the THEN part of a rule. The dependency chart is very useful when the knowledge engineer deletes pmarks modifies pmarks or adds rules to the rule base.

Note that in Figure 1 pmarks the ' pmarks ''s indicate the dependencies for the original rule set. By adding a clause to Rule 2 pmarks the ' pmarks 2' dependencies appeared. Note pmarks Rule 2 now references itself - a self-circular rule. By the addition of one clause to Rule 1 pmarks the ' pmarks 1' dependencies appeared. This also causes the rule set to be circular pmarks since an IF clause of Rule 1 is matched by THEN clauses of Rule 7 and Rule 8 which in turn match an IF clause of Rule 1 . Circular rules should be avoided since they can lead to an infinite loop at run time. Some expert systems pmarks such as EMYCIN pmarks handle circular rules in a special way. Nevertheless pmarks the knowledge engineer will want to know which rules are circular. So pmarks CHECK uses the dependency chart to generate graphs representing the interactions between rules pmarks and uses a cyclic graph detection algorithm to detect circular rule chains.

~3 IMPLEMENTATION OF RULE CHECKER.

In solving a problem pmarks the knowledge engineer may write several sets of goal-driven rules with each set having a unique subject category. pmarks In LES it is convenient to put rules in different subject categories so that the system can solve different goals using only those rule sets which apply to that goal pmarks . To solve a particular goal pmarks often he will select several goal-driven rule sets and WHEN rules ·pmarks demons pmarks . Since these rule sets are generated over a period of time pmarks it is quite possible that their interaction will cause some problems. Thus pmarks for each goal it is necessary to compare the rules pmarks in the rule sets specified by that goal pmarks against each other and against the clauses of that goal . We now show an algorithm pmarks in an Algol-like notation pmarks which CHECK uses. The algorithm does the checking for a set of subject categories with N rules and a goal with G clauses.

Because an IF part or a THEN part can have more than one clause pmarks the comparison between one part and another is handled by comparing a clause of one part to every clause in the other part. These algorithms also work for forward-chaining rules pmarks which are called WHEN rules in LES. However pmarks the criterion unreachable clauses is not applicable to forward-chaining rules.

~4 SUMMARY.

In this paper we described a program called CHECK whose function is to detect four potential problems pmarks redundant rules pmarks conflicting rules pmarks subsumed rules pmarks and circular rules pmarks and three potential gaps pmarks missing rules pmarks unreachable clauses pmarks and deadend clauses pmarks in a knowledge base utilizing the LES framework. We applied the consistency and completeness verification method of Suwa pmarks Scott pmarks and Shortliffe pmarks 4 pmarks to the generic expert system

LES with good results. Furthermore pmarks we have extended the checking to include circular rules pmarks unreachable clauses pmarks and deadend clauses. We also showed a general algorithm which performs the checking function efficiently. Finally pmarks as a by-product of the rule checking processing pmarks CHECK generates a dependenoy chart which shows how the rules couple and interact with each other and with the goals pmarks this chart should help the knowledge engineer to identify immediately the effects of deleting pmarks adding pmarks or modifying rules.

From our experiences with constructing different knowledge bases pmarks we find that many changes and additions to the rule sets occur during the development of a knowledge base. Thus pmarks a tool such as CHECK that can detect many potential problems and gaps in the knowledge base should be very useful to the knowledge engineer in helping him to develop a knowledge base rapidly and accurately.

The major area of improvement for CHECK is the handling of certainty factors in the rules since LES allows the rules to have certainty factors associated with them pmarks this may require the definitions for the seven conditions covered in this paper to be revised.

PAPER 4.

~0 An Expert Advisor For Photolithography.

~1 Introduction.

An integrated circuit fabrication line pmarks like many manufacturing lines pmarks is organized around a sequence of stations mamned by technicians. Most stations process a wafer once or twice before it is finished pmarks but for the photolithography step pmarks placing a pattern on the wafer similar to photography pmarks each wafer is processed up to 15 times. When a photolithography problem arises pmarks this step becomes the bottleneck for the whole line. Wafers continue to be processed at other stations pmarks causing a stockpile to build while the process engineer is called in to fire fight pmarks diagnose and correct the problem pmarks .

When fire fighting pmarks the process engineer's first priority is to get the line up and running again as quickly as possible. Fire fighting consumes about half his day pmarks conflicting with his main responsiblities of coordinating line personnel and monitoring and controlling the process to meet specifications on uniformity of pattern thickness pmarks develop pmarks exposure pmarks baking pmarks and many other variables. Under these circumstances pmarks temporary solutions usually result pmarks with no time left for designing long-term solutions to recurrent problems.

An expert system pmarks like the Photolithography Advisor pmarks can help alleviate fire fighting pmarks monitor and control the process pmarks record process data that can help determine what the recurrent problems are pmarks and provide information needed to coordinate line personnel on different work shifts.

The complex procedures needed to produce correct patterns by photolithography are many and varied. For an overview see pmarks Thompson pmarks et al pmarks 1983 pmarks . In addition pmarks the number of ways a defective pattern can be produced is very large. When armed with this knowledge pmarks the Advisor can help the technician focus on the data pertinent to isolating the defect and determining its cause and can recommend corrective action. Without the Advisor or the process engineer pmarks the technician may search blindly for a very long time and become overwhelmed and frustrated.

We designed the Advisor to mimic the problem-solving behavior of human experts so line technicians would feel confident that the program could help them and they would want to use it instead of depending on the process engineer. This involved not only capturing in rules the knowledge of several photolithography engineers and the field test experience of technicians pmarks but also a lot of effort in implementimg an interface that presents a familiar model of work pmarks eg pmarks by using electronic versions of the paper forms previously used for process recordkeeping. How well we have succeeded is still being evaluated. We have done a preliminary evaluation by field testing on a research line. The results of that test helped us design a more thorough testing pmarks in progress pmarks on a manufacturing line.

~2 The Photolithography Advisor.

The Photolithography Advisor is written in HP-RL pmarks Rosenberg pmarks 1983 pmarks a knowledge representation language that supports structured declarative objects pmarks frames pmarks message-passing pmarks active values pmarks daemons pmarks procedural values pmarks a rule system for logic programming with agenda-based inference engines for forward and backward chaining pmarks and a smooth interface with the host LISP environment. HP-RL runs on HP model 9000 series 200 workstations under both PSL pmarks Griss pmarks 1981 pmarks and Common-Lisp pmarks Steele pmarks 1984 pmarks . The A sor is a small to medium sized expert system. It builds on a number of well-known medical diagnosis programs pmarks borrowing in particular some of their architectural ideas pmarks pmarks Shortliffe pmarks 1976 pmarks pmarks Aikins pmarks 1984 pmarks pmarks Kunz pmarks pmarks Smith and Clayton pmarks 1980 pmarks . The program currently contains 30 class-frames pmarks with up to 100 slots each pmarks and 300 rules pmarks 30 forward-chaining and the rest backward-chaining.

The Advisor is structured as a team of three cooperating specialists that work together to solve problems posed by the user. An inspector works with the user to collect symptoms pmarks passing this information on to a diagnostician. The diagnostician uses knowledge of chemistry and photolithography equipment to diagnose wafer problems and their causes and tells the repairman about them. The repairman then advises the user on how to correct the diagnosed problems. The specialists communicate through the frame database. In this sense pmarks the Advisor uses a blackboard architecture pmarks Erman et al pmarks 1980 pmarks .

Any specialist can ask for more information when needed. This allows the consultation to be focused pmarks eliminates asking for extraneous or irrelevant information that may be expensive in time and effort to gather pmarks and results in a more human-like interaction.

The basic diagnostic strategy is to work with visual symptoms first. A visual inspection revealing no symptoms does not mean the wafer is free of defects. It only means that no gross defects can be detected by the naked eye. In such cases pmarks the Advisor requests the user to examine the wafer under the microscope at low magnification. If symptoms are found by this inspection pmarks they must be verified using high magnification.

The user interface has turned out to be one of the most important components. Initially pmarks we were more interested in problem-solving performance than ease of use pmarks so a simple gloss tty interface was used. This was unsatisfactory. In response pmarks we implemented an interface which required very little typing pmarks all questions being asked through pop-up menus pmarks and recorded all interaction in a transcript window. The user could then answer most questions by using either a single-key command pmarks pressing a softkey pmarks or selecting with the mouse. The consultation could be reviewed by scrolling back in the transcript window. We are now building another interface that implements a virtual advisor's notebook. It uses graphical representations of the wafer and processing equipment pmarks sophisticated browsers for examining the knowledge-base pmarks and a laser-disc database of symptom and defect images. While being as easy to use as the current interface pmarks this interface will use more cognitively stimulating media to provide considerably more explanatory power.

~3 Performance and Evaluation.

The first version of the Photolithography Advisor was installed in a low-volume research lab and was used for about a year. At the time of this writing pmarks we have just installed a second version of the program in a manufacturing line for additional tasting.

The test in the research lab helped us debug the knowledge base and obtain useful ideas on what was needed in the user interface. Overall pmarks the program was considered quite useful and generally diagnosed problems as well as the available process engineers pmarks the same ones who helped us build the program pmarks .

Some of the benefits of the program were.
Substantial time was saved in processing wafers. If a

less than 10 minutes.
The Advisor was always available pmarks even during the night shift. Process engineers like to sleep at night.
The hardware emitted less particles than an engineer and could be installed in the clean room where it was convenient and available.
The hardware cost was less than $35 pmarks 000 pmarks cheaper than a full-time engineer.

Some of the needs uncovered during this informal test were.
Most users were naive computer users. The interface must eliminate most typing and make correcting errors easy when typing is needed. We dealt with this in the next version by extensively using popup menus and customized softkeys pmarks eliminating most typing. Where typing was still needed pmarks customized editor modes were designed that limited the number of ways a user could get into trouble.
Questions asked by the Advisor used jargon to describe wafer defects. Menu items could easily be misinterpreted. We dealt with this terminology problem by adding alternate definitions for defects and coilected photographs to illustrate defects. By showing a photograph of the defect we were trying to describe pmarks we avoided misinterpretation of the text.
Experienced users were frustrated having to answer questions about visual symptoms when they knew a high-power inspection was needed. We accomodated them by structuring the menus to have default answers that provided an express-lane to more detailed inspection.
Despite our attempt to make the problem-solving knowledge generic pmarks most fabrication lines will want to customize for local practice. Domain experts pmarks with training from us pmarks can now edit the knowledge base using a sophisticated set of browsing tools that do not require knowing the syntax of either HP-RL or LISP.
The program's problem-solving ability was inadequately tested. Measuring this ability was difficult since the research processes were continually modified.

Primarily because of the last point pmarks we have started an extensive field test on a production line to more fairly and completely test it in its target environment than could be done in a research facility. This evaluation consists of an in vivo use of the program on a negative resist fabrication line pmarks a user survey pmarks and a comparision of key process statistics using before and after data.

The user survey has two parts. The first is administered with each consultation. Here we are trying to ascertain the program's accuracy and obtain detailed comments about the wording on menus pmarks of advice pmarks etc . The second survey is administered every month to collect more general comments about the friendliness and usefulness of the system.

Statistics measuring process performance include yield histories pmarks rework rates pmarks process hold times pmarks and equipment downtime. Many of these are collected automatically by existing process management software. Thus pmarks a large database of data on past performance of the process line exists. In cases where the necessary data is not logged by existing software pmarks the Advisor automatically collects it so that it can be compared with manually reconstructed past data.

~4 Conclusions.

Developing the Photolithography Advisor has been a useful experiment in building an industrial strength expert system. As with other such systems pmarks the experiment may never be completely finished pmarks but some valuable insights about the process have already been gained.

We have found that peopie not trained in AI can quickly learn to modify an expert system in their domain of expertise if given the right tools. But we have also found that they are limited if they are not an experienced AI programmer.

Domain experts tend to introspect about their problem-solving strategies in terms of plans. Even simple planning constructs are useful in helping them articulate their expertise.

Building an expert system has much in common with other types of software engineering. The programming paradigm is different pmarks but experimentation pmarks iteration pmarks and testing are as important here as in other types of programming.

Meta-level rules can be useful initially to capture an expert's problem-solving heuristics. Once the problem is better understood pmarks however pmarks they can usually be replaced by specific object-level rules that solve the problem much better pmarks and more efficiently pmarks .

Experts disagree. It helps to sample a lot of them. In our application pmarks many problems can be solved several ways. Which solution is used often depends on personal preference or on established local procedures. Vocabulary can also be a problem. Different people will call the same thing by different names. This problem becomes acute when trying to design the user interface.

We are enhancing the Advisor in several ways. The knowledge-base is now being extended to include positive resist processes pmarks which are gaining popularity for fabricating smaller patterns on wafers. The user interface is being redesigned to better reflect our model of the program as a team of cooperating specialists using graphics pmarks animation pmarks and video display media. The problem-solving strategy used by the program is really quite generic to a wide variety of fault-diagnosis problems important to us. We are restructuring the knowledge representation to better separate the problem-solving knowledge pmarks and control pmarks from the domain-specific knowledge so that a fault-diagnosis shell can be extracted.

PAPER 5.

~0 FAULT DIAGNOSIS THROUGH RESPONSIBILITY.

~1 INTRODUCTION.

Testing and fault diagnosis of printed circuit cards is a very important task which is done many times each day. Typically this is performed using pre-determined pmarks static and rigidly structured tests. As a result pmarks the testing tends to be inefficient pmarks missing many faulted components and can often isolate faults to only a large group of components.

We introduce the 'theory of responsibilities' as an approach to automated troubleshooting. Using this approach pmarks the understanding of how a circuit works is recorded by assigning responsibilities for parts of the output waveform to subsections of the circuit. These can be assigned manually ro derived from casual simulation.

Our current efforts are directed at automated troubleshooting of analog circuit cards. Our overall system pmarks described in pmarks Milne 1984 pmarks and pmarks Ramsey 1984 pmarks is designed to automatically test and identify faults in an analog card through the use of automatic test equipment. Our past implementation and papers have conducted testing based on the structural description of the circuit. In this paper pmarks our approach to functional testing is described.

This work differs from others in several significant ways. The work of pmarks Cantone 1985 pmarks has been entirely within the structural area. His algorithm for deciding which test to perform based on the most information gained and possible cost is very helpful to structural reasoning pmarks but can't help us once we can no longer probe within an sub-circuit. Cantone only uses structural information to isolate the fault to a single functional area. In this paper pmarks we will show how structural information can be used to further propose faults.

pmarks de Kleer 1983 pmarks is working to diagnose faults in analog circuit cards from 'first principles'. That is pmarks given the low level electronic description of how a capacitor works pmarks it should be possible to deduce how a filter would work pmarks and consequently pmarks diagnose faults in it. Although it is agreed that this is the most desirable approach pmarks the author feels that there is too much work still to be done in order to use this approach in testing today. In our work pmarks we start from 'second principles' pmarks that is pmarks the type of description that an electronics engineer uses to describes various building blocks of circuits. Several examples are contained below.

pmarks Davis 1983 pmarks has done much work in the area of digital troubleshooting based on the function of the components. He relies on computing the function and inverse of each sub-component. In general pmarks it is not possible to compute the inverse of analog functions. Also the digital domain has a very simple output form pmarks 1 or 0 pmarks while the analog domain may have a very rich signal Hence different techniques are and digital domains.

The work of pmarks Chandrasekaran 1985 pmarks is most similar to our own. He describes the object to be diagnosed in a formal language and then compiles this description into a set of production rules to perform the diagnosis. In our approach pmarks we use a different formal description and four simple rules of diagnosis pmarks rather than compiling the system into production rules. Chandrasekaran's work has not defined a clear role for the interaction of functional and structural reasoning. In our work pmarks structural reasoning plays a dual role. It is first used to isolate the possible fault to a single functional area. Secondly pmarks whenever the output is zero pmarks pmarks giving no information pmarks structural rules are used to propose possible faults.

Qualitative reasoning pmarks Forbus 1981 pmarks is important to justify some of our rules and approaches pmarks but since we are working from second principles pmarks many of these results are compiled into the descriptions.

~2 STRUCTURE.

The traditional role of structure is to isolate the possible fault to one functional module. The system first checks if the output is correct. If it is not pmarks the path of the signal is traced back through the structure of the circuit and a test is chosen which will split the possible fault path in half. This is done until only one functional module is left.

In our work pmarks structure is also used when the output is zero. In this situation pmarks we have no information on which to base the functional diagnosis pmarks so structural reasoning must be used. In an analog circuit pmarks we are interested in the output current and voltage drop. Their relationship is controlled by Ohm's Law pmarks E=IR. Qualitatively pmarks we can see that if E is zero pmarks then I will be zero pmarks and that if I is zero pmarks then E will be zero. We can also know that if R is zero pmarks then E and I will be zero. Because of the product of IR pmarks if E is zero we will need two rules pmarks since I or R could be zero.

Let us look at voltage. We know that we need an R in order to get a voltage drop. If an output is shorted to ground pmarks then R is zero pmarks and hence no output voltage. We can translate this into a diagnosis rule.

Voltage Short Rule pmarks
If one component connects the output and ground pmarks
and the output is zero pmarks
then that component may be shorted.

We could derive this rule from E=IR pmarks but it can be 'compiled' to the above form. This compilation is similar to pmarks Chandra 1985 pmarks and only needs to be done once for the fault diagnosis system. This rule is an example of the high level knowledge that an electronics engineer may use. We call this an example of a 'second principle'.

From Krichoff's Current Law pmarks we know that the current flowing into and out of a node is zero. If there is no current flowing into a node pmarks then there can be no current flowing out of the node. As before pmarks we can compile these facts into a 'second principle'.

Current Open Rule pmarks
If one component connects the input and output pmarks
and the output is zero pmarks
then that component could be open.

We also know that with no current pmarks we will not get a voltage drop. If the output is zero pmarks it could be from no R or no I pmarks based upon Ohm's Law. Because of the dependency between the current and voltage pmarks we cannot be sure pmarks based upon a single zero output pmarks whether we have a short or open pmarks so several hypotheses may be produced.

We will use a simple voltage divider as an example of these structure rules. When the circuit is working properly pmarks the outputs are 01 and 02. Their respective values are determined by the ratio of R1 and R2. If the value of 02 is wrong when 01 is correct pmarks then the ratio R1/R2 is wrong. Qualitatively pmarks we cannot tell which value is wrong.

If R1 is a short pmarks then 02 will equal 01 and not be zero. This can be predicted by the application of the Voltage Short rule. pmarks Note that ground being zero is only a special case pmarks . If R1 is open pmarks then 02 will be zero by the Current Open rule. 01 may or may not be zero depending upon the circuit. If R2 is shorted pmarks then 02 will be zero by the Voltage Short rule. If R2 is open pmarks then 02 will be zero by the Current Open rule.

In this section pmarks we have presented our approach to the use of structure for fault diagnosis. In the traditional way pmarks it is used to isolate a fault to a single functional module. We also use it to propose faults based upon shorts and opens. These two rules are very powerful and alone will propose the correct fault for most of the examples we have encountered.

~3 FUNCTION.

The Rectifier Circuit pmarks When the structural rules have isolated a fault to a single functional module pmarks we turn to functional reasoning. To illustrate the theory of responsibilities presented in this paper pmarks a simple rectifier circuit will be used. In second principles pmarks the rectifier circuit can be described as follows. To build a rectifier circuit pmarks use a diode to convert each peak in the input waveform to a positive output peak. For a typical sine wave pmarks this means two diodes pmarks one for the positive pmarks and one for the negative peak. The output is then filtered. To do this a capacitor is used to store energy and a resistor to drain that energy.

From this description responsibilities can be assigned pmarks each diode produces a peak in the output pmarks one at the positive peak and one at the negative peak. The capacitor charges pmarks giving us the rising

The diagnosis rules pmarks
If X is Y by Z and not pmarks Y pmarks then Z is bad pmarks
If X is Y by Z and not pmarks X pmarks then use structure rules.

In this brief paper pmarks the implementation details have been left out pmarks although the PROLOG user will recognise the role for unification.

These are the only troubleshooting rules we need. To diagnose a fault pmarks the output waveform is compared with the input waveform. The first rule will then identify which component is faulted.

The above description was tested on an actual circuit. When the resistor was faulted to the open position pmarks the output was a constant level. Comparing the desired output from the real output pmarks we are missing the declining ramp pmarks so the resistor is bad. The DC voltage levels are important in this diagnosis pmarks but have been omitted for this paper.

When one of the diodes was opened pmarks the output was missing one of its peaks. By comparing the desired output with the real output and the input pmarks the rule can identify correctly which diode has been opened. When the capacitor is opened pmarks the rising output is not a ramp pmarks so the capacitor is predicted bad. If the resistor is shorted pmarks the output is zero pmarks and the structure rules will correctly propose the fault. It should be note that we are lucky in this case that each component manifests it to a single part of the output waveform. In this brief paper pmarks the voltage levels have been omitted. These are critical to diagnose some faults.

~4 DEEP FUNCTIONAL REASONING.

In the above sections pmarks we have assumed that we understand how to assign responsibilities between the parts of the output waveform and the components of the circuit. In this section pmarks we will see how this can be derived automatically.

The basic strategy is to simulate the working of the circuit through our 'second principles'. As each component makes a contribution to the overall output pmarks a responsibility is assigned. We then use the above troubleshooting technique to trace the responsibilities to the faulted component.

We will use the rectifier circuit as an illustration. We start with the following second principles pmarks
Diode pmarks If the input is positive pmarks the output equals the input pmarks
Capacitor pmarks A capacitor with a rising input will charge pmarks
Capacitor pmarks A charged capacitor with a load pmarks and not a rising input will discharge pmarks
Wire pmarks When two wires meet pmarks the output is the sum of the two inputs.

We also assume low level rules that provide for the addition of signal waveforms pmarks the testing for voltage across a capacitor. The input to the clipper is in the form of a sine wave from a transformer. When we apply the diode rule pmarks The sine wave is transformed into a postive half-wave followed by zero. The second diode is in reverse polarity pmarks so it produces a zero output followed by a half-wave. Responsibility to each half-wave is then assigned. Next pmarks the two wires add pmarks giving us a half-wave rectified waveform.

Now the signal arrives at the capacitor and the input signal is rising. By the capacitor rule pmarks the capacitor now charges. The rise in the output voltage is assigned the capacitor. When the signal stops rising pmarks the capacitor has a load pmarks so discharges. This decreasing output voltage is assigned to the resistor.

In this very brief explanation pmarks we have seen how the waveform gets built up from second principles and the responsibilities are assigned. Using this technique pmarks it is possible to derive the responsiblities from a circuit and then perform troubleshooting based upon these.

~5 CONCLUSION.

In this paper pmarks the 'theory of responsibilities' has been outlined. This states that we can perform fault diagnosis by assigning responsibilities for parts of the output to various parts of the circuit. These responsibilities are derived from a casual simulation of the function of the circuit. It has also demonstrated how this works on a typical building block circuit. In our work we have started from 'second principles' pmarks that is pmarks the principles which an electronic engineer would use to describes each sub-circuit. This approach has proved to be very effective in diagnosing faults in analog circuits.

If one has only a limited understanding of the circuit pmarks then responsibilities can only be assigned in a limited way pmarks and hence the capability to diagnose faults will be limited. If one has thorough understanding of the circuit pmarks can do a better job of assigning the responsibilities. However pmarks because of limitations within the domain pmarks it will not always be possible to describe the circuit in such a way as to isolate faults to one component. This limitation pmarks however pmarks primarily comes from the limits of being able to diagnose a circuit as a black box. This technique has been applied to filters pmarks rectifiers pmarks voltage doublers and wave generators. Because of the modularity of most circuits pmarks it works even on seemingly complex circuits pmarks by dealing with the sub-circuits.

Although this technique is very powerful pmarks there is much to be done. It should be possible to derive our second principles from first principles. We have found that we can generally prove the second principles in this way pmarks but we can't do it automatically pmarks yet. A more detailed description of this work is in pmarks Milne 1985 pmarks .

APPENDIX H

RESULTS OF THE TEXTUAL ANALYSER AND THE EXPERT CLASSIFIER MODULES

```
/* textstatus.pl */

status(1,title ).
status(2,heading ).
status( 9, theme_marker).
status( 16, theme_marker).
status(21,heading ).
status(62,heading ).
status( 102, theme_marker).
status(122,heading ).
status( 126, theme_marker).
```

```
function(1,title).
function(2, introduction ).
function(3, background ).
function(4, background ).
function(5, background ).
function(6, background ).
function(7, background ).
function(8, background ).
function(9, methodology ).
function(9, methodology ).
function(10, background ).
function(11, background ).
function(12, background ).
function(13, background ).
function(14, background ).
function(15, background ).
function(16, methodology ).
function(16, methodology ).
function(16, objectives ).
function(17, background ).
function(18, background ).
function(19, background ).
function(20, background ).
function(21, methodology ).
function(22, background ).
function(23, background ).
function(24, background ).
function(25, background ).
function(26, background ).
function(27, background ).
function(28, background ).
function(29, background ).
function(30, background ).
function(31, background ).
function(32, background ).
function(33, background ).
function(34, background ).
function(35, background ).
function(36, background ).
function(37, background ).
function(38, background ).
function(39, background ).
function(40, background ).
function(41, background ).
function(42, background ).
function(43, background ).
function(44, background ).
function(45, background ).
function(46, background ).
function(47, background ).
function(48, background ).
function(49, background ).
function(50, background ).
function(51, background ).
function(52, background ).
function(53, background ).
function(54, background ).
function(55, background ).
function(56, background ).
function(57, background ).
function(58, background ).
function(59, background ).
function(60, background ).
function(61, background ).
function(63, background ).
function(64, background ).
function(65, background ).
function(66, background ).
function(67, background ).
function(68, background ).
function(69, background ).
function(70, background ).
function(71, background ).
function(72, background ).
function(73, background ).
function(74, background ).
function(75, background ).
function(76, background ).
function(77, background ).
function(78, background ).
function(79, background ).
function(80, background ).
function(81, background ).
function(82, background ).
function(83, background ).
function(84, background ).
function(85, background ).
function(86, background ).
function(87, background ).
function(88, background ).
function(89, background ).
function(90, background ).
function(91, background ).
function(92, background ).
function(93, background ).
function(94, background ).
function(95, background ).
function(96, background ).
function(97, background ).
function(98, background ).
function(99, background ).
function(100, background ).
function(101, background ).
function(102, methodology ).
function(103, background ).
function(104, background ).
function(105, background ).
function(106, background ).
function(107, background ).
function(108, background ).
function(109, background ).
function(110, background ).
function(111, background ).
function(112, background ).
function(113, background ).
function(114, background ).
function(115, background ).
function(116, background ).
function(117, background ).
function(118, background ).
function(119, background ).
function(120, background ).
function(121, background ).
function(122, conclusion ).
function(123, background ).
function(124, background ).
function(125, background ).
function(126, background ).
function(127, background ).
function(128, background ).
```

```
/*            textlcpt.pl                    */
/*                                           */
/* prolog database produced by txtcpt.spt */
/* concept(concept identity, concept) */
concept( 1, [ characteristic, error, approach] ).
concept( 2, [ conflict, resolution] ).
concept( 3, [ introduction] ).
concept( 4, [  conflict, resolution] ).
concept( 5, [ integration, of, alternate,  opinions] ).
concept( 6, [ state, of, the, world] ).
concept( 7, [ single,  coherent, world, view] ).
concept( 8, [ operation] ).
concept( 9, [ performed] ).
concept( 10, [ separate, entity] ).
concept( 11, [ knowledge, source] ).
concept( 12, [  designed] ).
concept( 13, [ purpose] ).
concept( 14, [ example] ).
concept( 15, [ system] ).
concept( 16, [  performs, surface, material, classification] ).
concept( 17, [ remote, sensing, application] ).
concept( 18, [ encounter, conflicts] ).
concept( 19, [ knowledge,  sources] ).
concept( 20, [ differing, opinions] ).
concept( 21, [  appropriate, classification, of, a, given, region] ).
concept( 22, [  remote,  sensing, experts, attempting] ).
concept( 23, [ resolve] ).
concept( 24, [ conflicts] ).
concept( 25, [  employ] ).
concept( 26, [ knowledge] ).
concept( 27, [ 'the, spectral, signature, of, a,  road] ).
concept( 28, [ shifted, due] ).
concept( 29, [ presence, of, vehicles] ).
concept( 30, [ oil, slick] ).
concept( 31, [ surface'] ).
concept( 32, [ 'a, portion, of, the,  road] ).
concept( 33, [ obscured] ).
concept( 34, [ trees'] ).
concept( 35, [ approach] ).
concept( 36, [ conflict, resolution] ).
concept( 37, [consists] ).
concept( 38, [ of, evaluating] ).
concept( 39, [  classification, errors] ).
concept( 40, [ occur] ).
concept( 41, [  given, context] ).
concept( 42, [  conflict, resolution] ).
concept( 43, [ treated] ).
concept( 44, [ number, of,  different, authors] ).
concept( 45, [ use, various, techniques] ).
concept( 46, [ terminology] ).
concept( 47, [ reasoning] ).
concept( 48, [ uncertainty] ).
concept( 49, [ 'the,  integration, of, knowledge] ).
concept( 50, [ disparate, sources, '] ).
concept( 51, [ of, this, work, takes] ).
concept( 52, [ numeric, approach] ).
concept( 53, [ conflict,  resolution] ).
concept( 54, [ example] ).
concept( 55, [  bayesian] ).
concept( 56, [ dempsterschaefer, techniques] ).
concept( 57, [  bayesian, techniques] ).
concept( 58, [assume] ).
concept( 59, [ ability] ).
concept( 60, [ generate] ).
concept( 61, [ priori] ).
concept( 62, [ table, of, conditional, probabilities] ).
```

```
concept( 64, [ allow] ).
concept( 65, [ source] ).
concept( 66, [ posit] ).
concept( 67, [ single, probability] ).
concept( 68, [ time] ).
concept( 69, [ dempster-schafer, techniques] ).
concept( 70, [ other, hand] ).
concept( 71, [ allowing, multiple, beliefs] ).
concept( 72, [assume] ).
concept( 73, [ possibilities] ).
concept( 74, [ independent] ).
concept( 75, [ assumption, not] ).
concept( 76, [ satisfied] ).
concept( 77, [  reality] ).
concept( 78, [ other, approaches] ).
concept( 79, [ conflict, resolution, represent] ).
concept( 80, [ more, semantic, knowledge, based, approach] ).
concept( 81, [ not] ).
concept( 82, [ upon] ).
concept( 83, [ numeric, algorithm] ).
concept( 84, [ resolve,  conflicts] ).
concept( 85, [ examples, of, these, approaches, include, cohen's,  theory, of,
concept( 86, [ tenenbaum] ).
concept( 87, [  barrow's, interpretation, guided, segmentation] ).
concept( 88, [ approaches] ).
concept( 89, [  exhibit] ).
concept( 90, [ limitations] ).
concept( 91, [ lack. extensibility] ).
concept( 92, [  function] ).
concept( 93, [ limited, domains] ).
concept( 94, [  require] ).
concept( 95, [ extensive, preprogramming] ).
concept( 96, [ interactive,  assistance] ).
concept( 97, [ operation] ).
concept( 98, [ paper, describes] ).
concept( 99, [ technique] ).
concept( 100, [ conflict, resolution] ).
concept( 101, [ termed] ).
concept( 102, [ characteristic, error, method] ).
concept( 103, [ goal] ).
concept( 104, [ capture] ).
concept( 105, [ reasoning, process] ).
concept( 106, [ form, of,  conflict, resolution] ).
concept( 107, [  characteristic, error, conflict, resolution] ).
concept( 108, [integrates] ).
concept( 109, [ two] ).
concept( 110, [ related, concepts] ).
concept( 111, [  reasoning] ).
concept( 112, [ uncertainty] ).
concept( 113, [ constraint, propagation] ).
concept( 114, [  reasoning] ).
concept( 115, [ uncertainty, determines] ).
concept( 116, [ relative, validity, of,  beliefs] ).
concept( 117, [ constraint, propagation, uses] ).
concept( 118, [ presence, of,  contradictions] ).
concept( 119, [ restrict] ).
concept( 120, [ range, of, applicable, choices] ).
concept( 121, [ characteristic, error, approach] ).
concept( 122, [ unique] ).
concept( 123, [ treating,  each, individual, knowledge, source] ).
concept( 124, [ separate, entity] ).
concept( 125, [ overall, reasoning, validity] ).
concept( 126, [ entirely, selfdetermined] ).
concept( 127, [ other, words] ).
concept( 128, [ knowledge, sources,  themselves] ).
```

```
concept( 129, [ assumed] ).
concept( 130, [provide] ).
concept( 131, [ of, the, necessary,  information] ).
concept( 132, [ system] ).
concept( 133, [ resolve] ).
concept( 134, [  conflict] ).
concept( 135, [ given] ).
concept( 136, [ local, context] ).
concept( 137, [ characteristic, error, approach] ).
concept( 138, [ term, characteristic, error] ).
concept( 139, [ chosen] ).
concept( 140, [  realization] ).
concept( 141, [ certain, error, conditions] ).
concept( 142, [ 'characteristic', of, the, environment] ).
concept( 143, [ knowledge, source,  expects] ).
concept( 144, [ operate] ).
concept( 145, [  conflict, resolution, based, upon] ).
concept( 146, [  characteristic, error, paradigm, uses, knowledge, based,  sema
concept( 147, [  information] ).
concept( 148, [ assumed] ).
concept( 149, [ expert, during] ).
concept( 150, [  knowledge, acquisition, process] ).
concept( 151, [  both] ).
concept( 152, [ develop] ).
concept( 153, [ context] ).
concept( 154, [ given, situation] ).
concept( 155, [ resolve, differences, of,  opinion] ).
concept( 156, [ possible] ).
concept( 157, [ information] ).
concept( 158, [  characteristic, error, classes, enables] ).
concept( 159, [ resolution, of, a,  specific, conflict] ).
concept( 160, [ dependent, upon] ).
concept( 161, [ immediate,  context, of, the, problem] ).
concept( 162, [  characteristic, error] ).
concept( 163, [ based, upon] ).
concept( 164, [ idea] ).
concept( 165, [ multi-source, system] ).
concept( 166, [  each, individual, knowledge,  source] ).
concept( 167, [ extent] ).
concept( 168, [ judge, of, its, own, performance] ).
concept( 169, [ motivation] ).
concept( 170, [ becomes, apparent] ).
concept( 171, [  one, considers] ).
concept( 172, [ factors] ).
concept( 173, [ embedded] ).
concept( 174, [ process, of, developing, any, given, expert, system] ).
concept( 175, [ knowledge, of, the, context] ).
concept( 176, [ expects] ).
concept( 177, [ operate] ).
concept( 178, [ expectation, of, the, form] ).
concept( 179, [ environment] ).
concept( 180, [ take] ).
concept( 181, [ types, of, interrelationships] ).
concept( 182, [ occur] ).
concept( 183, [ type, of, boundary] ).
concept( 184, [ limiting, conditions] ).
concept( 185, [  expects] ).
concept( 186, [ encounter] ).
concept( 187, [ information] ).
concept( 188, [  occur] ).
concept( 189, [ environment] ).
concept( 190, [ close] ).
concept( 191, [  expected] ).
concept( 192, [ different] ).
concept( 193, [ type, of, misclassifications] ).
concept( 194, [ postulates] ).
```

```
concept( 195, [  occur] ).
concept( 196, [ information] ).
concept( 197, [ present] ).
concept( 198, [  environment] ).
concept( 199, [ confused] ).
concept( 200, [ desired,  information] ).
concept( 201, [  implementing] ).
concept( 202, [ concept, of, characteristic, error,  requires] ).
concept( 203, [ each, knowledge, source, posit] ).
concept( 204, [ priori,  estimate, of, the, error, classes] ).
concept( 205, [ encounter] ).
concept( 206, [ estimate] ).
concept( 207, [ explicit, ranking, of, its, expectations] ).
concept( 208, [ local, context] ).
concept( 209, [ independent, of,  the, strength, of, the, knowledge, source's,
concept( 210, [ any,  particular, instance, of, its, application] ).
concept( 211, [ based] ).
concept( 212, [  upon] ).
concept( 213, [ expert's, knowledge, of, the, underlying, structure,  of, its,
concept( 214, [ decisions] ).
concept( 215, [ example] ).
concept( 216, [ region,  growing, knowledge, source] ).
concept( 217, [  designed] ).
concept( 218, [ road, finding] ).
concept( 219, [ image, understanding, system] ).
concept( 220, [ know] ).
concept( 221, [  local, classification, error] ).
concept( 222, [ occur] ).
concept( 223, [ portion, of,  the, road] ).
concept( 224, [ obscured] ).
concept( 225, [ trees] ).
concept( 226, [  same, token] ).
concept( 227, [ aware, of, the, fact] ).
concept( 228, [  very, uncharacteristic] ).
concept( 229, [ any, region, designated] ).
concept( 230, [ deep,  water] ).
concept( 231, [ road] ).
concept( 232, [ same, source] ).
concept( 233, [ no, real, opinion] ).
concept( 234, [  misclassification, of, shallow, water] ).
concept( 235, [ considerations] ).
concept( 236, [ independent, of, its, specific, belief, of, the, existence, of,
concept( 237, [ any, particular, instance] ).
concept( 238, [ error, class, information] ).
concept( 239, [ dependent] ).
concept( 240, [ upon] ).
concept( 241, [ context, assumed] ).
concept( 242, [ the, knowledge, source] ).
concept( 243, [  independent, of, the, actual, context, of, any, particular,  i
concept( 244, [ knowledge, source] ).
concept( 245, [ utilized] ).
concept( 246, [  other, words] ).
concept( 247, [ knowledge, source's, characteristic, error,  list] ).
concept( 248, [ states] ).
concept( 249, [ context] ).
concept( 250, [ assumed] ).
concept( 251, [  part, of, the, underlying, structure, of, the, environment, du
concept( 252, [  no] ).
concept( 253, [ priori, controls, need] ).
concept( 254, [ imposed, upon] ).
concept( 255, [ overall, system] ).
concept( 256, [  knowledge, source] ).
concept( 257, [ context, information] ).
concept( 258, [ stated] ).
concept( 259, [ knowledge, source, itself] ).
concept( 260, [ illustrate] ).
```

```
concept( 263, [shows] ).
concept( 264, [ structure, of, two, road, finders] ).
concept( 265, [ road, finder] ).
concept( 266, [ anticipates] ).
concept( 267, [ called, upon] ).
concept( 268, [ process, satellite, information] ).
concept( 269, [ expect, partial, road, covering] ).
concept( 270, [ treetops] ).
concept( 271, [  ground, level, road, finder] ).
concept( 272, [ other, hand] ).
concept( 273, [  has, no,  expectation, of, such] ).
concept( 274, [ conflict] ).
concept( 275, [ priori, identification, of, errors] ).
concept( 276, [ promotes] ).
concept( 277, [ graceful, performance, degradation, of, a, group, of,  knowledg
concept( 278, [ any, knowledge, source] ).
concept( 279, [ operating, outside, of, its, area, of, expertise] ).
concept( 280, [ source,  operating] ).
concept( 281, [ unfamiliar, area] ).
concept( 282, [ not] ).
concept( 283, [  conflicting, class, appearing, anywhere] ).
concept( 284, [ characteristic,  error, list] ).
concept( 285, [ absence, serves] ).
concept( 286, [ immediate, flag] ).
concept( 287, [ conflict, resolver] ).
concept( 288, [ investigate] ).
concept( 289, [ situation, further] ).
concept( 290, [ minimum, implementation, of, the, characteristic,  error, techn
concept( 291, [ following, information] ).
concept( 292, [ characteristic] ).
concept( 293, [ priori, errors, expected] ).
concept( 294, [  encountered] ).
concept( 295, [ knowledge, source] ).
concept( 296, [  environment] ).
concept( 297, [ knowledge, source's, strength, of, opinion] ).
concept( 298, [  particular, instance] ).
concept( 299, [ methodology] ).
concept( 300, [ evaluating] ).
concept( 301, [ error, list] ).
concept( 302, [  determining] ).
concept( 303, [ context] ).
concept( 304, [ system, works] ).
concept( 305, [ comparing] ).
concept( 306, [ characteristic, error,  lists, of, the, conflicting, knowledge,
concept( 307, [ determines] ).
concept( 308, [ conflict's, local, context] ).
concept( 309, [ context] ).
concept( 310, [  essence] ).
concept( 311, [ indicator, of, the, similarity] ).
concept( 312, [ structure, of,  these, error, lists] ).
concept( 313, [ implementation] ).
concept( 314, [  error, lists, create] ).
concept( 315, [ context, controlling] ).
concept( 316, [ relative,  strength, of, opinion, necessary] ).
concept( 317, [ knowledge,  sources] ).
concept( 318, [ alter] ).
concept( 319, [ decision] ).
concept( 320, [ example] ).
concept( 321, [ sources,  agree] ).
concept( 322, [ given, error] ).
concept( 323, [ characteristic, of, the, situation] ).
concept( 324, [ not, take] ).
concept( 325, [ large] ).
concept( 326, [ absolute] ).
```

```
concept( 328, [ shift] ).
concept( 329, [ decision] ).
concept( 330, [ another, direction] ).
concept( 331, [ same, token] ).
concept( 332, [ context] ).
concept( 333, [ incoherent] ).
concept( 334, [ knowledge, source's, opinion] ).
concept( 335, [ stronger] ).
concept( 336, [ opinion] ).
concept( 337, [ change] ).
concept( 338, [ interaction, of, these, two, factors] ).
concept( 339, [ basis] ).
concept( 340, [ implementation, of, the] ).
concept( 341, [ error, approach] ).
concept( 342, [ items, listed] ).
concept( 343, [ serve] ).
concept( 344, [ basic, information] ).
concept( 345, [ knowledge] ).
concept( 346, [ rule] ).
concept( 347, [ based, management, of, the, conflict] ).
concept( 348, [ given] ).
concept( 349, [ existence, of, a, central] ).
concept( 350, [ top, level] ).
concept( 351, [ conflict, resolving, system] ).
concept( 352, [ additional, rule, based, manipulation] ).
concept( 353, [ possible] ).
concept( 354, [ rules, used] ).
concept( 355, [ manipulation,/management, of, the, conflict] ).
concept( 356, [ specific] ).
concept( 357, [ given, system] ).
concept( 358, [ perhaps] ).
concept( 359, [ specific, instance, of, its, use] ).
concept( 360, [ noted] ).
concept( 361, [ manipulations, require, no, overall, ranking, of, the, knowled
concept( 362, [ system] ).
concept( 363, [ general, thrust] ).
concept( 364, [ drive] ).
concept( 365, [ system, results] ).
concept( 366, [ specific, direction] ).
concept( 367, [ establishing] ).
concept( 368, [ strategy] ).
concept( 369, [ system] ).
concept( 370, [ strategy, embodies, more, general, criteria, always, err] ).
concept( 371, [ side, of, caution] ).
concept( 372, [ strategy, setting] ).
concept( 373, [ implemented] ).
concept( 374, [ moving, error, terms] ).
concept( 375, [ characteristic] ).
concept( 376, [ uncharacteristic] ).
concept( 377, [ neutral] ).
concept( 378, [ altering] ).
concept( 379, [ source's] ).
concept( 380, [ priori, opinion] ).
concept( 381, [ error, type] ).
concept( 382, [ application, of, characteristic, error] ).
concept( 383, [ section, presents, an, example, application, of, the, characte
concept( 384, [ focus, of, our, application] ).
concept( 385, [ tasc's, multi-spectral, image, analysis, system] ).
concept( 386, [ msias] ).
concept( 387, [ msias] ).
concept( 388, [ knowledge, based, system] ).
concept( 389, [ interprets, cartographic, surface, attributes] ).
concept( 390, [ information, obtained] ).
concept( 391, [ sensed, imagery] ).
concept( 392, [ msias, classifier] ).
```

```
concept( 394, [ relative, spectral, signature, of, an, individual, pixel, locat
concept( 395, [ relative, intensity] ).
concept( 396, [ various, spectral, bands] ).
concept( 397, [ determine] ).
concept( 398, [ surface,  material, class, concrete] ).
concept( 399, [  vegetation] ).
concept( 400, [  water] ).
concept( 401, [  conflicts] ).
concept( 402, [ occur] ).
concept( 403, [ new, knowledge, sources] ).
concept( 404, [  other, criteria] ).
concept( 405, [ introduced] ).
concept( 406, [ system] ).
concept( 407, [  knowledge, sources] ).
concept( 408, [ start] ).
concept( 409, [ latter, spectral,  identification] ).
concept( 410, [ base] ).
concept( 411, [ own, classification, opinion, upon] ).
concept( 412, [ alternate, criteria] ).
concept( 413, [  structural, measures] ).
concept( 414, [ important, feature, of, these, knowledge, sources] ).
concept( 415, [ use] ).
concept( 416, [ same, initial, information] ).
concept( 417, [ spectral,  classification] ).
concept( 418, [ independent, criteria] ).
concept( 419, [ obtain] ).
concept( 420, [ individual, results] ).
concept( 421, [ results, generate, alternate,  possible, classifications] ).
concept( 422, [ any, given, pixel,/region] ).
concept( 423, [  example] ).
concept( 424, [  alternate, classifications, represent] ).
concept( 425, [  conflicts] ).
concept( 426, [ process, of, obtaining] ).
concept( 427, [ final, classification] ).
concept( 428, [  conflict, resolution] ).
concept( 429, [ nomenclature, of, clancy] ).
concept( 430, [ problem, of, heuristic, classification] ).
concept( 431, [ planned] ).
concept( 432, [ conflict, resolver] ).
concept( 433, [  activated, late] ).
concept( 434, [ image, classification, process] ).
concept( 435, [  late, activation, occurs] ).
concept( 436, [ number, of, different,  knowledge, sources] ).
concept( 437, [ polled] ).
concept( 438, [ conflict] ).
concept( 439, [ occur] ).
concept( 440, [  conflict, resolution] ).
concept( 441, [ applied] ).
concept( 442, [ objects, which] ).
concept( 443, [ subject, of, conflicting, opinions] ).
concept( 444, [ frequency, of,  these, conflicts] ).
concept( 445, [ increase] ).
concept( 446, [ semantic, level, of, the,  program, increases] ).
concept( 447, [ use, of, the, resolver] ).
concept( 448, [ lower] ).
concept( 449, [ percentage, basis] ).
concept( 450, [ pixels] ).
concept( 451, [  regions] ).
concept( 452, [  upon, activation] ).
concept( 453, [ resolver] ).
concept( 454, [ looks] ).
concept( 455, [  pixels,/regions] ).
concept( 456, [ adjacent] ).
concept( 457, [ one, currentiy] ).
concept( 458, [ investigation] ).
```

```
concept( 460, [ other, opinions] ).
concept( 461, [ pixel's, classification] ).
concept( 462, [ every, adjacent, pixel] ).
concept( 463, [ marked] ).
concept( 464, [ both] ).
concept( 465, [ specific, classification] ).
concept( 466, [ characteristic, error,  list] ).
concept( 467, [ list, summarizes] ).
concept( 468, [ characteristic, error, classes,  posited] ).
concept( 469, [ knowledge, source] ).
concept( 470, [ identified] ).
concept( 471, [ information, serves] ).
concept( 472, [ basis] ).
concept( 473, [ conflict, resolution] ).
concept( 474, [  attempting] ).
concept( 475, [ address] ).
concept( 476, [ issues] ).
concept( 477, [ other, conflicts] ).
concept( 478, [ characteristic,  error, classes, relate] ).
concept( 479, [ identification, of, this, pixel] ).
concept( 480, [ reinforce] ).
concept( 481, [ current, designation] ).
concept( 482, [ contest] ).
concept( 483, [ way] ).
concept( 484, [ error, classes] ).
concept( 485, [ mute] ).
concept( 486, [ possible, pixel, classification] ).
concept( 487, [shown] ).
concept( 488, [ figure] ).
concept( 489, [ pixel] ).
concept( 490, [ eight,  nearest, neighbors] ).
concept( 491, [ digitized, image] ).
concept( 492, [ two, extreme, cases] ).
concept( 493, [ possible, opinions, of, these, neighbors] ).
concept( 494, [ original, pixel,  classification] ).
concept( 495, [ x] ).
concept( 496, [ member, of, the, 'uncharacteristic', error, class, of, each] ).
concept( 497, [ every, neighboring, pixel] ).
concept( 498, [  second] ).
concept( 499, [ pixel's, class, appears] ).
concept( 500, [ 'characteristic',  error, class, of, all, neighboring, pixels]
concept( 501, [ case] ).
concept( 502, [ conflict, resolver] ).
concept( 503, [ not] ).
concept( 504, [ any, circumstances] ).
concept( 505, [  alter] ).
concept( 506, [ pixel's, classification] ).
concept( 507, [ of, the, knowledge, sources, agree] ).
concept( 508, [ pixel] ).
concept( 509, [ misclassified] ).
concept( 510, [ knowledge, sources, felt] ).
concept( 511, [ regions] ).
concept( 512, [ worthy,  of, extension] ).
concept( 513, [  no, regions] ).
concept( 514, [ extended] ).
concept( 515, [ preagreed] ).
concept( 516, [ abide] ).
concept( 517, [ pixel, designation] ).
concept( 518, [ agreed] ).
concept( 519, [ 'x', class, pixel] ).
concept( 520, [ both,  something] ).
concept( 521, [ expect] ).
concept( 522, [ environment] ).
concept( 523, [ something] ).
concept( 524, [ boundary, of, their,  regions] ).
```

```
concept( 526, [ general, case] ).
concept( 527, [ expert] ).
concept( 528, [ say, 'i, use, x] ).
concept( 529, [ indicator, of, this, condition] ).
concept( 530, [  y] ).
concept( 531, [ indicator, of, an, other, condition] ).
concept( 532, [  x] ).
concept( 533, [ y] ).
concept( 534, [ confused'] ).
concept( 535, [ second, case] ).
concept( 536, [ conflict, resolver] ).
concept( 537, [ change] ).
concept( 538, [ classification, of, this, pixel] ).
concept( 539, [ competing, class] ).
concept( 540, [ highest, relative, likelihood] ).
concept( 541, [ opinion,  strength] ).
concept( 542, [ change] ).
concept( 543, [ occur] ).
concept( 544, [ relative,  likelihood] ).
concept( 545, [ low] ).
concept( 546, [ absolute, sense] ).
concept( 547, [ complete, agreement] ).
concept( 548, [ initial,  misclassification, of, this, pixel] ).
concept( 549, [ characteristic, of, the,  situation] ).
concept( 550, [ corresponds] ).
concept( 551, [ general, case] ).
concept( 552, [ expert] ).
concept( 553, [ say, '] ).
concept( 554, [ use, x] ).
concept( 555, [ indicator, of, this, condition] ).
concept( 556, [ y] ).
concept( 557, [ confused] ).
concept( 558, [ x] ).
concept( 559, [ doubts] ).
concept( 560, [ y] ).
concept( 561, [ use] ).
concept( 562, [ indicator'] ).
concept( 563, [ characteristic, error, paradigm,  gives, one] ).
concept( 564, [ methodology] ).
concept( 565, [ juggling, cases, intermediate] ).
concept( 566, [ two, extremes] ).
concept( 567, [ example] ).
concept( 568, [ seven,  neighbors] ).
concept( 569, [ agree] ).
concept( 570, [ less, strong] ).
concept( 571, [ case] ).
concept( 572, [ eight] ).
concept( 573, [ one] ).
concept( 574, [ class] ).
concept( 575, [ uncharacteristic, error, list] ).
concept( 576, [ excluded] ).
concept( 577, [ three, basic, ways, of, performing] ).
concept( 578, [  integration] ).
concept( 579, [ simplest, method] ).
concept( 580, [ tabulates, frequency, of, appearance] ).
concept( 581, [ pixel, class] ).
concept( 582, [ question] ).
concept( 583, [combines] ).
concept( 584, [  two,  frequencies] ).
concept( 585, [ characteristic] ).
concept( 586, [  uncharacteristic] ).
concept( 587, [ decides] ).
concept( 588, [ go] ).
concept( 589, [ highest, combination, of, relative, frequency] ).
concept( 590, [ opinion, strength] ).
```

```
concept( 592, [ weights] ).
concept( 593, [ determined] ).
concept( 594, [ second, approach, judges] ).
concept( 595, [ relative, coherence,  of, these, opinions] ).
concept( 596, [ example] ).
concept( 597, [ three, divergent, region,  types, had, found] ).
concept( 598, [ class] ).
concept( 599, [ question] ).
concept( 600, [  characteristic, error, list] ).
concept( 601, [ stronger, case] ).
concept( 602, [  misclassification] ).
concept( 603, [ single, type, of, region, had,  reached] ).
concept( 604, [ same, conclusion, three, tunes] ).
concept( 605, [ approach,  attempts] ).
concept( 606, [ compensate] ).
concept( 607, [ systematic, bias] ).
concept( 608, [ part,  of, the, experts] ).
concept( 609, [ third, method] ).
concept( 610, [ considered] ).
concept( 611, [involves] ).
concept( 612, [ discounting] ).
concept( 613, [ opinion, of, any, region,  having] ).
concept( 614, [ class] ).
concept( 615, [ uncharacteristic, error, list] ).
concept( 616, [ not] ).
concept( 617, [ valid, method] ).
concept( 618, [ situation] ).
concept( 619, [  region] ).
concept( 620, [ giving] ).
concept( 621, [ strong, vote] ).
concept( 622, [ retaining] ).
concept( 623, [  current, classification] ).
concept( 624, [ characteristic, error, paradigm, allows] ).
concept( 625, [ extraction, of, information] ).
concept( 626, [ characteristics, of, the,  knowledge, source, itself] ).
concept( 627, [ important] ).
concept( 628, [ recall] ).
concept( 629, [ discussing, three, classes] ).
concept( 630, [ four, specific, class, lists] ).
concept( 631, [ existence] ).
concept( 632, [  fourth, class] ).
concept( 633, [ class] ).
concept( 634, [ es] ).
concept( 635, [ structural,  knowledge, source] ).
concept( 636, [ region, grower] ).
concept( 637, [ designed] ).
concept( 638, [ example] ).
concept( 639, [ road, grower] ).
concept( 640, [ concrete,/asphalt] ).
concept( 641, [ class] ).
concept( 642, [ other, examples, appear] ).
concept( 643, [ table] ).
concept( 644, [ characteristic, error, class] ).
concept( 645, [ group, of, pixel,  identifications] ).
concept( 646, [ one] ).
concept( 647, [ expect] ).
concept( 648, [ error] ).
concept( 649, [  occur] ).
concept( 650, [ environment] ).
concept( 651, [shown] ).
concept( 652, [ table] ).
concept( 653, [ one] ).
concept( 654, [ growing, highways] ).
concept( 655, [ expected] ).
concept( 656, [  road, areas] ).
```

```
concept( 658, [ overhanging, trees] ).
concept( 659, [  oil, slicks] ).
concept( 660, [ 'uncharacteristic', error,  classes] ).
concept( 661, [ ones] ).
concept( 662, [ are, expected] ).
concept( 663, [ form] ).
concept( 664, [ hard,  boundary] ).
concept( 665, [ given, class] ).
concept( 666, [ alternate] ).
concept( 667, [ classes] ).
concept( 668, [ deep, water] ).
concept( 669, [ rock] ).
concept( 670, [  roads] ).
concept( 671, [ neutral, classes] ).
concept( 672, [ classes] ).
concept( 673, [ exist] ).
concept( 674, [  expected, environment] ).
concept( 675, [ no, intelligent,  statement] ).
concept( 676, [ made,  very, shallow, water] ).
concept( 677, [  patchy,  snow] ).
concept( 678, [ roads] ).
concept( 679, [ discussed] ).
concept( 680, [ section] ).
concept( 681, [  manipulation, of, this, list] ).
concept( 682, [ significant, action] ).
concept( 683, [  of, itself] ).
concept( 684, [ serve] ).
concept( 685, [ ideal, place] ).
concept( 686, [ changing, context] ).
concept( 687, [ example] ).
concept( 688, [ know] ).
concept( 689, [ area, had] ).
concept( 690, [ flooded] ).
concept( 691, [ move, shallow, water, to] ).
concept( 692, [ characteristic, error, set] ).
concept( 693, [ low, lying, classes] ).
concept( 694, [ explicit, designation, of, the, classes, expected] ).
concept( 695, [  occur] ).
concept( 696, [ environment] ).
concept( 697, [ very, useful, side, effect] ).
concept( 698, [ possible] ).
concept( 699, [ determine] ).
concept( 700, [ knowledge, source] ).
concept( 701, [  operating, outside, of, its, area, of, expertise] ).
concept( 702, [ example] ).
concept( 703, [ airport, hangar, area, grower] ).
concept( 704, [ expect, fuel, storage,  tanks, appearing, next] ).
concept( 705, [ hard, boundary] ).
concept( 706, [ highway, expert] ).
concept( 707, [ other, hand] ).
concept( 708, [ not, expect] ).
concept( 709, [  structure] ).
concept( 710, [ any] ).
concept( 711, [ adjacent, pixel] ).
concept( 712, [  fuel,  storage, tanks] ).
concept( 713, [ not, appear, anywhere] ).
concept( 714, [ highway, grower's, list, of, error, classes] ).
concept( 715, [  not] ).
concept( 716, [ 'characteristic', 'uncharacteristic'] ).
concept( 717, [ 'neutral'] ).
concept( 718, [ conflict, resolution, process, retained] ).
concept( 719, [ storage, tank,  classification] ).
concept( 720, [ conflict, resolver, would, disregard] ).
concept( 721, [  results, of, the, highway, finder] ).
concept( 722, [ abutting, region] ).
```

```
concept( 724, [ find, alternate, explanations, of, the,  region,/pixel] ).
concept( 725, [ had, designated] ).
concept( 726, [ 'highway'] ).
concept( 727, [  eliminating] ).
concept( 728, [ highway, designation] ).
concept( 729, [ asking] ).
concept( 730, [ alternate, opinions] ).
concept( 731, [ conclusions,/future, work] ).
concept( 732, [ characteristic, error, paradigm, allows] ).
concept( 733, [ other,  types, of, contextural] ).
concept( 734, [ programmatic, information] ).
concept( 735, [ added] ).
concept( 736, [ system] ).
concept( 737, [ example] ).
concept( 738, [  very, important] ).
concept( 739, [ catch, the, existence, of, highways] ).
concept( 740, [  particular, application] ).
concept( 741, [ move, more, classes] ).
concept( 742, [ highway, grower's, 'characteristic', list] ).
concept( 743, [ actions,  alter] ).
concept( 744, [ relative, 'push', given] ).
concept( 745, [ particular,  classification] ).
concept( 746, [ can] ).
concept( 747, [ alter] ).
concept( 748, [ final, characteristics, of, the, classified, image] ).
concept( 749, [ paper, presents] ).
concept( 750, [ unique, means, of, combining] ).
concept( 751, [ opinions, of, divergent, experts] ).
concept( 752, [ uniform, manner] ).
concept( 753, [ own] ).
concept( 754, [ priori, knowledge] ).
concept( 755, [ own,  characteristics] ).
concept( 756, [ characteristic, error, method] ).
concept( 757, [  planned, enhancement] ).
concept( 758, [ msias, system] ).
concept( 759, [  planned] ).
concept( 760, [ upgraded, msias] ).
concept( 761, [ serve] ).
concept( 762, [  testbed] ).
concept( 763, [ technique] ).
concept( 764, [ given] ).
concept( 765, [ ability] ).
concept( 766, [  evaluate, structural, information] ).
/* heading(no,heading) */
heading( 0,' The Characteristic Error Approach to Conflict Resolution '  ).
heading( 1,' INTRODUCTION '  ).
heading( 2,' THE CHARACTERISTIC ERROR APPROACH '  ).
heading( 3,' AN  APPLICATION OF CHARACTERISTIC ERROR '  ).
heading( 4,' CONCLUSIONS/FUTURE WORK '  ).
/* summary ( sentence no., no. of concepts identified ) */
summary( 1,2).
summary( 2,1).
summary( 3,4).
summary( 4,6).
summary( 5,8).
summary( 6,13).
summary( 7,7).
summary( 8,9).
summary( 9,6).
summary( 10,12).
summary( 11,9).
summary( 12,7).
summary( 13,3).
summary( 14,3).
summary( 15,7).
```

```
summary( 18,7).
summary( 19,6).
summary( 20,10).
summary( 21,1).
summary( 22,7).
summary( 23,11).
summary( 24,6).
summary( 25,7).
summary( 26,6).
summary( 27,8).
summary( 28,10).
summary( 29,8).
summary( 30,5).
summary( 31,5).
summary( 32,4).
summary( 33,11).
summary( 34,6).
summary( 35,3).
summary( 36,3).
summary( 37,5).
summary( 38,3).
summary( 39,6).
summary( 40,8).
summary( 41,5).
summary( 42,6).
summary( 43,4).
summary( 44,5).
summary( 45,5).
summary( 46,5).
summary( 47,14).
summary( 48,3).
summary( 49,2).
summary( 50,4).
summary( 51,7).
summary( 52,11).
summary( 53,7).
summary( 54,4).
summary( 55,6).
summary( 56,6).
summary( 57,6).
summary( 58,3).
summary( 59,7).
summary( 60,2).
summary( 61,10).
summary( 62,1).
summary( 63,1).
summary( 64,3).
summary( 65,5).
summary( 66,9).
summary( 67,6).
summary( 68,7).
summary( 69,7).
summary( 70,2).
summary( 71,3).
summary( 72,3).
summary( 73,2).
summary( 74,4).
summary( 75,5).
summary( 76,4).
summary( 77,3).
summary( 78,5).
summary( 79,10).
summary( 80,5).
summary( 81,4).
```

```
summary( 83,4).
summary( 84,3).
summary( 85,5).
summary( 86,9).
summary( 87,9).
summary( 88,8).
summary( 89,7).
summary( 90,10).
summary( 91,7).
summary( 92,5).
summary( 93,3).
summary( 94,13).
summary( 95,4).
summary( 96,10).
summary( 97,2).
summary( 98,12).
summary( 99,3).
summary( 100,2).
summary( 101,9).
summary( 102,4).
summary( 103,7).
summary( 104,8).
summary( 105,3).
summary( 106,5).
summary( 107,6).
summary( 108,6).
summary( 109,7).
summary( 110,9).
summary( 111,7).
summary( 112,4).
summary( 113,8).
summary( 114,8).
summary( 115,7).
summary( 116,8).
summary( 117,4).
summary( 118,6).
summary( 119,6).
summary( 120,5).
summary( 121,8).
summary( 122,1).
summary( 123,5).
summary( 124,6).
summary( 125,6).
summary( 126,7).
summary( 127,3).
summary( 128,8).
/* textstats (no.of sentences,no. of headings,no.of concepts,no.of words,no. of
textstats( 128, 5, 766, 2804, 26 ).
```

```
                                                                */
/* prolog database produced by the program called txtcpt.spt*/
/* the format of the database is as follows: */
/* (note: title set to heading no = 0) */
/* textconcept(heading no, paragragh no,paragraphsentence no, sentence no, conc
textconcept( 1,0,0,1,1,  1 ).
textconcept( 1,0,0,1,2,  2 ).
textconcept( 2,0,0,2,1,  3 ).
textconcept( 2,1,1,3,1,  4 ).
textconcept( 2,1,1,3,2,  5 ).
textconcept( 2,1,1,3,3,  6 ).
textconcept( 2,1,1,3,4,  7 ).
textconcept( 2,1,2,4,1,  8 ).
textconcept( 2,1,2,4,2,  9 ).
textconcept( 2,1,2,4,3,  10 ).
textconcept( 2,1,2,4,4,  11 ).
textconcept( 2,1,2,4,5,  12 ).
textconcept( 2,1,2,4,6,  13 ).
textconcept( 2,1,3,5,1,  14 ).
textconcept( 2,1,3,5,2,  15 ).
textconcept( 2,1,3,5,3,  16 ).
textconcept( 2,1,3,5,4,  17 ).
textconcept( 2,1,3,5,5,  18 ).
textconcept( 2,1,3,5,6,  19 ).
textconcept( 2,1,3,5,7,  20 ).
textconcept( 2,1,3,5,8,  21 ).
textconcept( 2,1,4,6,1,  22 ).
textconcept( 2,1,4,6,2,  23 ).
textconcept( 2,1,4,6,3,  24 ).
textconcept( 2,1,4,6,4,  25 ).
textconcept( 2,1,4,6,5,  26 ).
textconcept( 2,1,4,6,6,  27 ).
textconcept( 2,1,4,6,7,  28 ).
textconcept( 2,1,4,6,8,  29 ).
textconcept( 2,1,4,6,9,  30 ).
textconcept( 2,1,4,6,10,  31 ).
textconcept( 2,1,4,6,11,  32 ).
textconcept( 2,1,4,6,12,  33 ).
textconcept( 2,1,4,6,13,  34 ).
textconcept( 2,1,5,7,1,  35 ).
textconcept( 2,1,5,7,2,  36 ).
textconcept( 2,1,5,7,3,  37 ).
textconcept( 2,1,5,7,4,  38 ).
textconcept( 2,1,5,7,5,  39 ).
textconcept( 2,1,5,7,6,  40 ).
textconcept( 2,1,5,7,7,  41 ).
textconcept( 2,2,1,8,1,  42 ).
textconcept( 2,2,1,8,2,  43 ).
textconcept( 2,2,1,8,3,  44 ).
textconcept( 2,2,1,8,4,  45 ).
textconcept( 2,2,1,8,5,  46 ).
textconcept( 2,2,1,8,6,  47 ).
textconcept( 2,2,1,8,7,  48 ).
textconcept( 2,2,1,8,8,  49 ).
textconcept( 2,2,1,8,9,  50 ).
textconcept( 2,2,2,9,1,  51 ).
textconcept( 2,2,2,9,2,  52 ).
textconcept( 2,2,2,9,3,  53 ).
textconcept( 2,2,2,9,4,  54 ).
textconcept( 2,2,2,9,5,  55 ).
textconcept( 2,2,2,9,6,  56 ).
textconcept( 2,2,3,10,1,  57 ).
textconcept( 2,2,3,10,2,  58 ).
textconcept( 2,2,3,10,3,  59 ).
textconcept( 2,2,3,10,4,  60 ).
```

```
textconcept( 2,2,3,10,6,  62 ).
textconcept( 2,2,3,10,7,  63 ).
textconcept( 2,2,3,10,8,  64 ).
textconcept( 2,2,3,10,9,  65 ).
textconcept( 2,2,3,10,10,  66 ).
textconcept( 2,2,3,10,11,  67 ).
textconcept( 2,2,3,10,12,  68 ).
textconcept( 2,2,4,11,1,  69 ).
textconcept( 2,2,4,11,2,  70 ).
textconcept( 2,2,4,11,3,  71 ).
textconcept( 2,2,4,11,4,  72 ).
textconcept( 2,2,4,11,5,  73 ).
textconcept( 2,2,4,11,6,  74 ).
textconcept( 2,2,4,11,7,  75 ).
textconcept( 2,2,4,11,8,  76 ).
textconcept( 2,2,4,11,9,  77 ).
textconcept( 2,2,5,12,1,  78 ).
textconcept( 2,2,5,12,2,  79 ).
textconcept( 2,2,5,12,3,  80 ).
textconcept( 2,2,5,12,4,  81 ).
textconcept( 2,2,5,12,5,  82 ).
textconcept( 2,2,5,12,6,  83 ).
textconcept( 2,2,5,12,7,  84 ).
textconcept( 2,2,6,13,1,  85 ).
textconcept( 2,2,6,13,2,  86 ).
textconcept( 2,2,6,13,3,  87 ).
textconcept( 2,2,7,14,1,  88 ).
textconcept( 2,2,7,14,2,  89 ).
textconcept( 2,2,7,14,3,  90 ).
textconcept( 2,2,8,15,1,  91 ).
textconcept( 2,2,8,15,2,  92 ).
textconcept( 2,2,8,15,3,  93 ).
textconcept( 2,2,8,15,4,  94 ).
textconcept( 2,2,8,15,5,  95 ).
textconcept( 2,2,8,15,6,  96 ).
textconcept( 2,2,8,15,7,  97 ).
textconcept( 2,3,1,16,1,  98 ).
textconcept( 2,3,1,16,2,  99 ).
textconcept( 2,3,1,16,3,  100 ).
textconcept( 2,3,1,16,4,  101 ).
textconcept( 2,3,1,16,5,  102 ).
textconcept( 2,3,1,16,6,  103 ).
textconcept( 2,3,1,16,7,  104 ).
textconcept( 2,3,1,16,8,  105 ).
textconcept( 2,3,1,16,9,  106 ).
textconcept( 2,3,2,17,1,  107 ).
textconcept( 2,3,2,17,2,  108 ).
textconcept( 2,3,2,17,3,  109 ).
textconcept( 2,3,2,17,4,  110 ).
textconcept( 2,3,2,17,5,  111 ).
textconcept( 2,3,2,17,6,  112 ).
textconcept( 2,3,2,17,7,  113 ).
textconcept( 2,3,3,18,1,  114 ).
textconcept( 2,3,3,18,2,  115 ).
textconcept( 2,3,3,18,3,  116 ).
textconcept( 2,3,3,18,4,  117 ).
textconcept( 2,3,3,18,5,  118 ).
textconcept( 2,3,3,18,6,  119 ).
textconcept( 2,3,3,18,7,  120 ).
textconcept( 2,3,4,19,1,  121 ).
textconcept( 2,3,4,19,2,  122 ).
textconcept( 2,3,4,19,3,  123 ).
textconcept( 2,3,4,19,4,  124 ).
textconcept( 2,3,4,19,5,  125 ).
textconcept( 2,3,4,19,6,  126 ).
textconcept( 2,3,5,20,2,  128 ).
textconcept( 2,3,5,20,3,  129 ).
textconcept( 2,3,5,20,4,  130 ).
textconcept( 2,3,5,20,5,  131 ).
textconcept( 2,3,5,20,6,  132 ).
textconcept( 2,3,5,20,7,  133 ).
textconcept( 2,3,5,20,8,  134 ).
textconcept( 2,3,5,20,9,  135 ).
textconcept( 2,3,5,20,10,  136 ).
textconcept( 3,0,0,21,1,  137 ).
textconcept( 3,1,1,22,1,  138 ).
textconcept( 3,1,1,22,2,  139 ).
textconcept( 3,1,1,22,3,  140 ).
textconcept( 3,1,1,22,4,  141 ).
textconcept( 3,1,1,22,5,  142 ).
textconcept( 3,1,1,22,6,  143 ).
textconcept( 3,1,1,22,7,  144 ).
textconcept( 3,1,2,23,1,  145 ).
textconcept( 3,1,2,23,2,  146 ).
textconcept( 3,1,2,23,3,  147 ).
textconcept( 3,1,2,23,4,  148 ).
textconcept( 3,1,2,23,5,  149 ).
textconcept( 3,1,2,23,6,  150 ).
textconcept( 3,1,2,23,7,  151 ).
textconcept( 3,1,2,23,8,  152 ).
textconcept( 3,1,2,23,9,  153 ).
textconcept( 3,1,2,23,10,  154 ).
textconcept( 3,1,2,23,11,  155 ).
textconcept( 3,1,3,24,1,  156 ).
textconcept( 3,1,3,24,2,  157 ).
textconcept( 3,1,3,24,3,  158 ).
textconcept( 3,1,3,24,4,  159 ).
textconcept( 3,1,3,24,5,  160 ).
textconcept( 3,1,3,24,6,  161 ).
textconcept( 3,2,1,25,1,  162 ).
textconcept( 3,2,1,25,2,  163 ).
textconcept( 3,2,1,25,3,  164 ).
textconcept( 3,2,1,25,4,  165 ).
textconcept( 3,2,1,25,5,  166 ).
textconcept( 3,2,1,25,6,  167 ).
textconcept( 3,2,1,25,7,  168 ).
textconcept( 3,2,2,26,1,  169 ).
textconcept( 3,2,2,26,2,  170 ).
textconcept( 3,2,2,26,3,  171 ).
textconcept( 3,2,2,26,4,  172 ).
textconcept( 3,2,2,26,5,  173 ).
textconcept( 3,2,2,26,6,  174 ).
textconcept( 3,2,3,27,1,  175 ).
textconcept( 3,2,3,27,2,  176 ).
textconcept( 3,2,3,27,3,  177 ).
textconcept( 3,2,3,27,4,  178 ).
textconcept( 3,2,3,27,5,  179 ).
textconcept( 3,2,3,27,6,  180 ).
textconcept( 3,2,3,27,7,  181 ).
textconcept( 3,2,3,27,8,  182 ).
textconcept( 3,2,4,28,1,  183 ).
textconcept( 3,2,4,28,2,  184 ).
textconcept( 3,2,4,28,3,  185 ).
textconcept( 3,2,4,28,4,  186 ).
textconcept( 3,2,4,28,5,  187 ).
textconcept( 3,2,4,28,6,  188 ).
textconcept( 3,2,4,28,7,  189 ).
textconcept( 3,2,4,28,8,  190 ).
textconcept( 3,2,4,28,9,  191 ).
textconcept( 3,2,4,28,10,  192 ).
```

```
textconcept( 3,2,5,29,4,  196 ).
textconcept( 3,2,5,29,5,  197 ).
textconcept( 3,2,5,29,6,  198 ).
textconcept( 3,2,5,29,7,  199 ).
textconcept( 3,2,5,29,8,  200 ).
textconcept( 3,3,1,30,1,  201 ).
textconcept( 3,3,1,30,2,  202 ).
textconcept( 3,3,1,30,3,  203 ).
textconcept( 3,3,1,30,4,  204 ).
textconcept( 3,3,1,30,5,  205 ).
textconcept( 3,3,2,31,1,  206 ).
textconcept( 3,3,2,31,2,  207 ).
textconcept( 3,3,2,31,3,  208 ).
textconcept( 3,3,2,31,4,  209 ).
textconcept( 3,3,2,31,5,  210 ).
textconcept( 3,3,3,32,1,  211 ).
textconcept( 3,3,3,32,2,  212 ).
textconcept( 3,3,3,32,3,  213 ).
textconcept( 3,3,3,32,4,  214 ).
textconcept( 3,3,4,33,1,  215 ).
textconcept( 3,3,4,33,2,  216 ).
textconcept( 3,3,4,33,3,  217 ).
textconcept( 3,3,4,33,4,  218 ).
textconcept( 3,3,4,33,5,  219 ).
textconcept( 3,3,4,33,6,  220 ).
textconcept( 3,3,4,33,7,  221 ).
textconcept( 3,3,4,33,8,  222 ).
textconcept( 3,3,4,33,9,  223 ).
textconcept( 3,3,4,33,10,  224 ).
textconcept( 3,3,4,33,11,  225 ).
textconcept( 3,3,5,34,1,  226 ).
textconcept( 3,3,5,34,2,  227 ).
textconcept( 3,3,5,34,3,  228 ).
textconcept( 3,3,5,34,4,  229 ).
textconcept( 3,3,5,34,5,  230 ).
textconcept( 3,3,5,34,6,  231 ).
textconcept( 3,3,6,35,1,  232 ).
textconcept( 3,3,6,35,2,  233 ).
textconcept( 3,3,6,35,3,  234 ).
textconcept( 3,3,7,36,1,  235 ).
textconcept( 3,3,7,36,2,  236 ).
textconcept( 3,3,7,36,3,  237 ).
textconcept( 3,4,1,37,1,  238 ).
textconcept( 3,4,1,37,2,  239 ).
textconcept( 3,4,1,37,3,  240 ).
textconcept( 3,4,1,37,4,  241 ).
textconcept( 3,4,1,37,5,  242 ).
textconcept( 3,4,2,38,1,  243 ).
textconcept( 3,4,2,38,2,  244 ).
textconcept( 3,4,2,38,3,  245 ).
textconcept( 3,4,3,39,1,  246 ).
textconcept( 3,4,3,39,2,  247 ).
textconcept( 3,4,3,39,3,  248 ).
textconcept( 3,4,3,39,4,  249 ).
textconcept( 3,4,3,39,5,  250 ).
textconcept( 3,4,3,39,6,  251 ).
textconcept( 3,4,4,40,1,  252 ).
textconcept( 3,4,4,40,2,  253 ).
textconcept( 3,4,4,40,3,  254 ).
textconcept( 3,4,4,40,4,  255 ).
textconcept( 3,4,4,40,5,  256 ).
textconcept( 3,4,4,40,6,  257 ).
textconcept( 3,4,4,40,7,  258 ).
textconcept( 3,5,1,41,3,  262 ).
textconcept( 3,5,1,41,4,  263 ).
textconcept( 3,5,1,41,5,  264 ).
textconcept( 3,5,2,42,1,  265 ).
textconcept( 3,5,2,42,2,  266 ).
textconcept( 3,5,2,42,3,  267 ).
textconcept( 3,5,2,42,4,  268 ).
textconcept( 3,5,2,42,5,  269 ).
textconcept( 3,5,2,42,6,  270 ).
textconcept( 3,5,3,43,1,  271 ).
textconcept( 3,5,3,43,2,  272 ).
textconcept( 3,5,3,43,3,  273 ).
textconcept( 3,5,3,43,4,  274 ).
textconcept( 3,6,1,44,1,  275 ).
textconcept( 3,6,1,44,2,  276 ).
textconcept( 3,6,1,44,3,  277 ).
textconcept( 3,6,1,44,4,  278 ).
textconcept( 3,6,1,44,5,  279 ).
textconcept( 3,6,2,45,1,  280 ).
textconcept( 3,6,2,45,2,  281 ).
textconcept( 3,6,2,45,3,  282 ).
textconcept( 3,6,2,45,4,  283 ).
textconcept( 3,6,2,45,5,  284 ).
textconcept( 3,6,3,46,1,  285 ).
textconcept( 3,6,3,46,2,  286 ).
textconcept( 3,6,3,46,3,  287 ).
textconcept( 3,6,3,46,4,  288 ).
textconcept( 3,6,3,46,5,  289 ).
textconcept( 3,7,1,47,1,  290 ).
textconcept( 3,7,1,47,2,  291 ).
textconcept( 3,7,1,47,3,  292 ).
textconcept( 3,7,1,47,4,  293 ).
textconcept( 3,7,1,47,5,  294 ).
textconcept( 3,7,1,47,6,  295 ).
textconcept( 3,7,1,47,7,  296 ).
textconcept( 3,7,1,47,8,  297 ).
textconcept( 3,7,1,47,9,  298 ).
textconcept( 3,7,1,47,10,  299 ).
textconcept( 3,7,1,47,11,  300 ).
textconcept( 3,7,1,47,12,  301 ).
textconcept( 3,7,1,47,13,  302 ).
textconcept( 3,7,1,47,14,  303 ).
textconcept( 3,7,2,48,1,  304 ).
textconcept( 3,7,2,48,2,  305 ).
textconcept( 3,7,2,48,3,  306 ).
textconcept( 3,7,3,49,1,  307 ).
textconcept( 3,7,3,49,2,  308 ).
textconcept( 3,7,4,50,1,  309 ).
textconcept( 3,7,4,50,2,  310 ).
textconcept( 3,7,4,50,3,  311 ).
textconcept( 3,7,4,50,4,  312 ).
textconcept( 3,7,5,51,1,  313 ).
textconcept( 3,7,5,51,2,  314 ).
textconcept( 3,7,5,51,3,  315 ).
textconcept( 3,7,5,51,4,  316 ).
textconcept( 3,7,5,51,5,  317 ).
textconcept( 3,7,5,51,6,  318 ).
textconcept( 3,7,5,51,7,  319 ).
textconcept( 3,7,6,52,1,  320 ).
textconcept( 3,7,6,52,2,  321 ).
textconcept( 3,7,6,52,3,  322 ).
textconcept( 3,7,6,52,4,  323 ).
textconcept( 3,7,6,52,5,  324 ).
```

```
textconcept( 3,7,6,52,7,  326 ).
textconcept( 3,7,6,52,8,  327 ).
textconcept( 3,7,6,52,9,  328 ).
textconcept( 3,7,6,52,10,  329 ).
textconcept( 3,7,6,52,11,  330 ).
textconcept( 3,7,7,53,1,  331 ).
textconcept( 3,7,7,53,2,  332 ).
textconcept( 3,7,7,53,3,  333 ).
textconcept( 3,7,7,53,4,  334 ).
textconcept( 3,7,7,53,5,  335 ).
textconcept( 3,7,7,53,6,  336 ).
textconcept( 3,7,7,53,7,  337 ).
textconcept( 3,7,8,54,1,  338 ).
textconcept( 3,7,8,54,2,  339 ).
textconcept( 3,7,8,54,3,  340 ).
textconcept( 3,7,8,54,4,  341 ).
textconcept( 3,8,1,55,1,  342 ).
textconcept( 3,8,1,55,2,  343 ).
textconcept( 3,8,1,55,3,  344 ).
textconcept( 3,8,1,55,4,  345 ).
textconcept( 3,8,1,55,5,  346 ).
textconcept( 3,8,1,55,6,  347 ).
textconcept( 3,8,2,56,1,  348 ).
textconcept( 3,8,2,56,2,  349 ).
textconcept( 3,8,2,56,3,  350 ).
textconcept( 3,8,2,56,4,  351 ).
textconcept( 3,8,2,56,5,  352 ).
textconcept( 3,8,2,56,6,  353 ).
textconcept( 3,8,3,57,1,  354 ).
textconcept( 3,8,3,57,2,  355 ).
textconcept( 3,8,3,57,3,  356 ).
textconcept( 3,8,3,57,4,  357 ).
textconcept( 3,8,3,57,5,  358 ).
textconcept( 3,8,3,57,6,  359 ).
textconcept( 3,8,4,58,1,  360 ).
textconcept( 3,8,4,58,2,  361 ).
textconcept( 3,8,4,58,3,  362 ).
textconcept( 3,8,5,59,1,  363 ).
textconcept( 3,8,5,59,2,  364 ).
textconcept( 3,8,5,59,3,  365 ).
textconcept( 3,8,5,59,4,  366 ).
textconcept( 3,8,5,59,5,  367 ).
textconcept( 3,8,5,59,6,  368 ).
textconcept( 3,8,5,59,7,  369 ).
textconcept( 3,8,6,60,1,  370 ).
textconcept( 3,8,6,60,2,  371 ).
textconcept( 3,8,7,61,1,  372 ).
textconcept( 3,8,7,61,2,  373 ).
textconcept( 3,8,7,61,3,  374 ).
textconcept( 3,8,7,61,4,  375 ).
textconcept( 3,8,7,61,5,  376 ).
textconcept( 3,8,7,61,6,  377 ).
textconcept( 3,8,7,61,7,  378 ).
textconcept( 3,8,7,61,8,  379 ).
textconcept( 3,8,7,61,9,  380 ).
textconcept( 3,8,7,61,10,  381 ).
textconcept( 4,0,0,62,1,  382 ).
textconcept( 4,1,1,63,1,  383 ).
textconcept( 4,1,2,64,1,  384 ).
textconcept( 4,1,2,64,2,  385 ).
textconcept( 4,1,2,64,3,  386 ).
textconcept( 4,1,3,65,1,  387 ).
textconcept( 4,1,3,65,2,  388 ).
textconcept( 4,1,3,65,3,  389 ).
textconcept( 4,1,3,65,4,  390 ).
textconcept( 4,1,4,66,1,  392 ).
textconcept( 4,1,4,66,2,  393 ).
textconcept( 4,1,4,66,3,  394 ).
textconcept( 4,1,4,66,4,  395 ).
textconcept( 4,1,4,66,5,  396 ).
textconcept( 4,1,4,66,6,  397 ).
textconcept( 4,1,4,66,7,  398 ).
textconcept( 4,1,4,66,8,  399 ).
textconcept( 4,1,4,66,9,  400 ).
textconcept( 4,1,5,67,1,  401 ).
textconcept( 4,1,5,67,2,  402 ).
textconcept( 4,1,5,67,3,  403 ).
textconcept( 4,1,5,67,4,  404 ).
textconcept( 4,1,5,67,5,  405 ).
textconcept( 4,1,5,67,6,  406 ).
textconcept( 4,1,6,68,1,  407 ).
textconcept( 4,1,6,68,2,  408 ).
textconcept( 4,1,6,68,3,  409 ).
textconcept( 4,1,6,68,4,  410 ).
textconcept( 4,1,6,68,5,  411 ).
textconcept( 4,1,6,68,6,  412 ).
textconcept( 4,1,6,68,7,  413 ).
textconcept( 4,1,7,69,1,  414 ).
textconcept( 4,1,7,69,2,  415 ).
textconcept( 4,1,7,69,3,  416 ).
textconcept( 4,1,7,69,4,  417 ).
textconcept( 4,1,7,69,5,  418 ).
textconcept( 4,1,7,69,6,  419 ).
textconcept( 4,1,7,69,7,  420 ).
textconcept( 4,1,8,70,1,  421 ).
textconcept( 4,1,8,70,2,  422 ).
textconcept( 4,1,9,71,1,  423 ).
textconcept( 4,1,9,71,2,  424 ).
textconcept( 4,1,9,71,3,  425 ).
textconcept( 4,1,10,72,1,  426 ).
textconcept( 4,1,10,72,2,  427 ).
textconcept( 4,1,10,72,3,  428 ).
textconcept( 4,1,11,73,1,  429 ).
textconcept( 4,1,11,73,2,  430 ).
textconcept( 4,2,1,74,1,  431 ).
textconcept( 4,2,1,74,2,  432 ).
textconcept( 4,2,1,74,3,  433 ).
textconcept( 4,2,1,74,4,  434 ).
textconcept( 4,2,2,75,1,  435 ).
textconcept( 4,2,2,75,2,  436 ).
textconcept( 4,2,2,75,3,  437 ).
textconcept( 4,2,2,75,4,  438 ).
textconcept( 4,2,2,75,5,  439 ).
textconcept( 4,2,3,76,1,  440 ).
textconcept( 4,2,3,76,2,  441 ).
textconcept( 4,2,3,76,3,  442 ).
textconcept( 4,2,3,76,4,  443 ).
textconcept( 4,2,4,77,1,  444 ).
textconcept( 4,2,4,77,2,  445 ).
textconcept( 4,2,4,77,3,  446 ).
textconcept( 4,2,5,78,1,  447 ).
textconcept( 4,2,5,78,2,  448 ).
textconcept( 4,2,5,78,3,  449 ).
textconcept( 4,2,5,78,4,  450 ).
textconcept( 4,2,5,78,5,  451 ).
textconcept( 4,2,6,79,1,  452 ).
textconcept( 4,2,6,79,2,  453 ).
textconcept( 4,2,6,79,3,  454 ).
textconcept( 4,2,6,79,4,  455 ).
textconcept( 4,2,6,79,5,  456 ).
```

```
textconcept( 4,2,6,79,7,  458 ).
textconcept( 4,2,6,79,8,  459 ).
textconcept( 4,2,6,79,9,  460 ).
textconcept( 4,2,6,79,10,  461 ).
textconcept( 4,2,7,80,1,  462 ).
textconcept( 4,2,7,80,2,  463 ).
textconcept( 4,2,7,80,3,  464 ).
textconcept( 4,2,7,80,4,  465 ).
textconcept( 4,2,7,80,5,  466 ).
textconcept( 4,2,8,81,1,  467 ).
textconcept( 4,2,8,81,2,  468 ).
textconcept( 4,2,8,81,3,  469 ).
textconcept( 4,2,8,81,4,  470 ).
textconcept( 4,2,9,82,1,  471 ).
textconcept( 4,2,9,82,2,  472 ).
textconcept( 4,2,9,82,3,  473 ).
textconcept( 4,2,9,82,4,  474 ).
textconcept( 4,2,9,82,5,  475 ).
textconcept( 4,2,9,82,6,  476 ).
textconcept( 4,2,9,82,7,  477 ).
textconcept( 4,2,9,82,8,  478 ).
textconcept( 4,2,9,82,9,  479 ).
textconcept( 4,2,10,83,1,  480 ).
textconcept( 4,2,10,83,2,  481 ).
textconcept( 4,2,10,83,3,  482 ).
textconcept( 4,2,10,83,4,  483 ).
textconcept( 4,2,11,84,1,  484 ).
textconcept( 4,2,11,84,2,  485 ).
textconcept( 4,2,11,84,3,  486 ).
textconcept( 4,3,1,85,1,  487 ).
textconcept( 4,3,1,85,2,  488 ).
textconcept( 4,3,1,85,3,  489 ).
textconcept( 4,3,1,85,4,  490 ).
textconcept( 4,3,1,85,5,  491 ).
textconcept( 4,4,1,86,1,  492 ).
textconcept( 4,4,1,86,2,  493 ).
textconcept( 4,4,1,86,3,  494 ).
textconcept( 4,4,1,86,4,  495 ).
textconcept( 4,4,1,86,5,  496 ).
textconcept( 4,4,1,86,6,  497 ).
textconcept( 4,4,1,86,7,  498 ).
textconcept( 4,4,1,86,8,  499 ).
textconcept( 4,4,1,86,9,  500 ).
textconcept( 4,5,1,87,1,  501 ).
textconcept( 4,5,1,87,2,  502 ).
textconcept( 4,5,1,87,3,  503 ).
textconcept( 4,5,1,87,4,  504 ).
textconcept( 4,5,1,87,5,  505 ).
textconcept( 4,5,1,87,6,  506 ).
textconcept( 4,5,1,87,7,  507 ).
textconcept( 4,5,1,87,8,  508 ).
textconcept( 4,5,1,87,9,  509 ).
textconcept( 4,5,2,88,1,  510 ).
textconcept( 4,5,2,88,2,  511 ).
textconcept( 4,5,2,88,3,  512 ).
textconcept( 4,5,2,88,4,  513 ).
textconcept( 4,5,2,88,5,  514 ).
textconcept( 4,5,2,88,6,  515 ).
textconcept( 4,5,2,88,7,  516 ).
textconcept( 4,5,2,88,8,  517 ).
textconcept( 4,5,3,89,1,  518 ).
textconcept( 4,5,3,89,2,  519 ).
textconcept( 4,5,3,89,3,  520 ).
textconcept( 4,5,3,89,4,  521 ).
textconcept( 4,5,3,89,5,  522 ).
textconcept( 4,5,3,89,7,  524 ).
textconcept( 4,5,4,90,1,  525 ).
textconcept( 4,5,4,90,2,  526 ).
textconcept( 4,5,4,90,3,  527 ).
textconcept( 4,5,4,90,4,  528 ).
textconcept( 4,5,4,90,5,  529 ).
textconcept( 4,5,4,90,6,  530 ).
textconcept( 4,5,4,90,7,  531 ).
textconcept( 4,5,4,90,8,  532 ).
textconcept( 4,5,4,90,9,  533 ).
textconcept( 4,5,4,90,10,  534 ).
textconcept( 4,6,1,91,1,  535 ).
textconcept( 4,6,1,91,2,  536 ).
textconcept( 4,6,1,91,3,  537 ).
textconcept( 4,6,1,91,4,  538 ).
textconcept( 4,6,1,91,5,  539 ).
textconcept( 4,6,1,91,6,  540 ).
textconcept( 4,6,1,91,7,  541 ).
textconcept( 4,6,2,92,1,  542 ).
textconcept( 4,6,2,92,2,  543 ).
textconcept( 4,6,2,92,3,  544 ).
textconcept( 4,6,2,92,4,  545 ).
textconcept( 4,6,2,92,5,  546 ).
textconcept( 4,6,3,93,1,  547 ).
textconcept( 4,6,3,93,2,  548 ).
textconcept( 4,6,3,93,3,  549 ).
textconcept( 4,6,4,94,1,  550 ).
textconcept( 4,6,4,94,2,  551 ).
textconcept( 4,6,4,94,3,  552 ).
textconcept( 4,6,4,94,4,  553 ).
textconcept( 4,6,4,94,5,  554 ).
textconcept( 4,6,4,94,6,  555 ).
textconcept( 4,6,4,94,7,  556 ).
textconcept( 4,6,4,94,8,  557 ).
textconcept( 4,6,4,94,9,  558 ).
textconcept( 4,6,4,94,10,  559 ).
textconcept( 4,6,4,94,11,  560 ).
textconcept( 4,6,4,94,12,  561 ).
textconcept( 4,6,4,94,13,  562 ).
textconcept( 4,7,1,95,1,  563 ).
textconcept( 4,7,1,95,2,  564 ).
textconcept( 4,7,1,95,3,  565 ).
textconcept( 4,7,1,95,4,  566 ).
textconcept( 4,7,2,96,1,  567 ).
textconcept( 4,7,2,96,2,  568 ).
textconcept( 4,7,2,96,3,  569 ).
textconcept( 4,7,2,96,4,  570 ).
textconcept( 4,7,2,96,5,  571 ).
textconcept( 4,7,2,96,6,  572 ).
textconcept( 4,7,2,96,7,  573 ).
textconcept( 4,7,2,96,8,  574 ).
textconcept( 4,7,2,96,9,  575 ).
textconcept( 4,7,2,96,10,  576 ).
textconcept( 4,8,1,97,1,  577 ).
textconcept( 4,8,1,97,2,  578 ).
textconcept( 4,8,2,98,1,  579 ).
textconcept( 4,8,2,98,2,  580 ).
textconcept( 4,8,2,98,3,  581 ).
textconcept( 4,8,2,98,4,  582 ).
textconcept( 4,8,2,98,5,  583 ).
textconcept( 4,8,2,98,6,  584 ).
textconcept( 4,8,2,98,7,  585 ).
textconcept( 4,8,2,98,8,  586 ).
textconcept( 4,8,2,98,9,  587 ).
textconcept( 4,8,2,98,10,  588 ).
```

```
textconcept( 4,8,2,98,12,  590 ).
textconcept( 4,8,3,99,1,  591 ).
textconcept( 4,8,3,99,2,  592 ).
textconcept( 4,8,3,99,3,  593 ).
textconcept( 4,9,1,100,1,  594 ).
textconcept( 4,9,1,100,2,  595 ).
textconcept( 4,9,2,101,1,  596 ).
textconcept( 4,9,2,101,2,  597 ).
textconcept( 4,9,2,101,3,  598 ).
textconcept( 4,9,2,101,4,  599 ).
textconcept( 4,9,2,101,5,  600 ).
textconcept( 4,9,2,101,6,  601 ).
textconcept( 4,9,2,101,7,  602 ).
textconcept( 4,9,2,101,8,  603 ).
textconcept( 4,9,2,101,9,  604 ).
textconcept( 4,9,3,102,1,  605 ).
textconcept( 4,9,3,102,2,  606 ).
textconcept( 4,9,3,102,3,  607 ).
textconcept( 4,9,3,102,4,  608 ).
textconcept( 4,10,1,103,1,  609 ).
textconcept( 4,10,1,103,2,  610 ).
textconcept( 4,10,1,103,3,  611 ).
textconcept( 4,10,1,103,4,  612 ).
textconcept( 4,10,1,103,5,  613 ).
textconcept( 4,10,1,103,6,  614 ).
textconcept( 4,10,1,103,7,  615 ).
textconcept( 4,10,2,104,1,  616 ).
textconcept( 4,10,2,104,2,  617 ).
textconcept( 4,10,2,104,3,  618 ).
textconcept( 4,10,2,104,4,  619 ).
textconcept( 4,10,2,104,5,  620 ).
textconcept( 4,10,2,104,6,  621 ).
textconcept( 4,10,2,104,7,  622 ).
textconcept( 4,10,2,104,8,  623 ).
textconcept( 4,11,1,105,1,  624 ).
textconcept( 4,11,1,105,2,  625 ).
textconcept( 4,11,1,105,3,  626 ).
textconcept( 4,11,2,106,1,  627 ).
textconcept( 4,11,2,106,2,  628 ).
textconcept( 4,11,2,106,3,  629 ).
textconcept( 4,11,2,106,4,  630 ).
textconcept( 4,11,2,106,5,  631 ).
textconcept( 4,11,3,107,1,  632 ).
textconcept( 4,11,3,107,2,  633 ).
textconcept( 4,11,3,107,3,  634 ).
textconcept( 4,11,3,107,4,  635 ).
textconcept( 4,11,3,107,5,  636 ).
textconcept( 4,11,3,107,6,  637 ).
textconcept( 4,11,4,108,1,  638 ).
textconcept( 4,11,4,108,2,  639 ).
textconcept( 4,11,4,108,3,  640 ).
textconcept( 4,11,4,108,4,  641 ).
textconcept( 4,11,4,108,5,  642 ).
textconcept( 4,11,4,108,6,  643 ).
textconcept( 4,12,1,109,1,  644 ).
textconcept( 4,12,1,109,2,  645 ).
textconcept( 4,12,1,109,3,  646 ).
textconcept( 4,12,1,109,4,  647 ).
textconcept( 4,12,1,109,5,  648 ).
textconcept( 4,12,1,109,6,  649 ).
textconcept( 4,12,1,109,7,  650 ).
textconcept( 4,12,2,110,1,  651 ).
textconcept( 4,12,2,110,2,  652 ).
textconcept( 4,12,2,110,3,  653 ).
textconcept( 4,12,2,110,4,  654 ).
textconcept( 4,12,2,110,6,  656 ).
textconcept( 4,12,2,110,7,  657 ).
textconcept( 4,12,2,110,8,  658 ).
textconcept( 4,12,2,110,9,  659 ).
textconcept( 4,12,3,111,1,  660 ).
textconcept( 4,12,3,111,2,  661 ).
textconcept( 4,12,3,111,3,  662 ).
textconcept( 4,12,3,111,4,  663 ).
textconcept( 4,12,3,111,5,  664 ).
textconcept( 4,12,3,111,6,  665 ).
textconcept( 4,12,3,111,7,  666 ).
textconcept( 4,12,4,112,1,  667 ).
textconcept( 4,12,4,112,2,  668 ).
textconcept( 4,12,4,112,3,  669 ).
textconcept( 4,12,4,112,4,  670 ).
textconcept( 4,12,5,113,1,  671 ).
textconcept( 4,12,5,113,2,  672 ).
textconcept( 4,12,5,113,3,  673 ).
textconcept( 4,12,5,113,4,  674 ).
textconcept( 4,12,5,113,5,  675 ).
textconcept( 4,12,5,113,6,  676 ).
textconcept( 4,12,5,113,7,  677 ).
textconcept( 4,12,5,113,8,  678 ).
textconcept( 4,12,6,114,1,  679 ).
textconcept( 4,12,6,114,2,  680 ).
textconcept( 4,12,6,114,3,  681 ).
textconcept( 4,12,6,114,4,  682 ).
textconcept( 4,12,6,114,5,  683 ).
textconcept( 4,12,6,114,6,  684 ).
textconcept( 4,12,6,114,7,  685 ).
textconcept( 4,12,6,114,8,  686 ).
textconcept( 4,12,7,115,1,  687 ).
textconcept( 4,12,7,115,2,  688 ).
textconcept( 4,12,7,115,3,  689 ).
textconcept( 4,12,7,115,4,  690 ).
textconcept( 4,12,7,115,5,  691 ).
textconcept( 4,12,7,115,6,  692 ).
textconcept( 4,12,7,115,7,  693 ).
textconcept( 4,13,1,116,1,  694 ).
textconcept( 4,13,1,116,2,  695 ).
textconcept( 4,13,1,116,3,  696 ).
textconcept( 4,13,1,116,4,  697 ).
textconcept( 4,13,1,116,5,  698 ).
textconcept( 4,13,1,116,6,  699 ).
textconcept( 4,13,1,116,7,  700 ).
textconcept( 4,13,1,116,8,  701 ).
textconcept( 4,13,2,117,1,  702 ).
textconcept( 4,13,2,117,2,  703 ).
textconcept( 4,13,2,117,3,  704 ).
textconcept( 4,13,2,117,4,  705 ).
textconcept( 4,13,3,118,1,  706 ).
textconcept( 4,13,3,118,2,  707 ).
textconcept( 4,13,3,118,3,  708 ).
textconcept( 4,13,3,118,4,  709 ).
textconcept( 4,13,3,118,5,  710 ).
textconcept( 4,13,3,118,6,  711 ).
textconcept( 4,13,4,119,1,  712 ).
textconcept( 4,13,4,119,2,  713 ).
textconcept( 4,13,4,119,3,  714 ).
textconcept( 4,13,4,119,4,  715 ).
textconcept( 4,13,4,119,5,  716 ).
textconcept( 4,13,4,119,6,  717 ).
textconcept( 4,13,5,120,1,  718 ).
textconcept( 4,13,5,120,2,  719 ).
textconcept( 4,13,5,120,3,  720 ).
```

```
textconcept( 4,13,5,120,5,   722 ).
textconcept( 4,13,6,121,1,   723 ).
textconcept( 4,13,6,121,2,   724 ).
textconcept( 4,13,6,121,3,   725 ).
textconcept( 4,13,6,121,4,   726 ).
textconcept( 4,13,6,121,5,   727 ).
textconcept( 4,13,6,121,6,   728 ).
textconcept( 4,13,6,121,7,   729 ).
textconcept( 4,13,6,121,8,   730 ).
textconcept( 5,0,0,122,1,   731 ).
textconcept( 5,1,1,123,1,   732 ).
textconcept( 5,1,1,123,2,   733 ).
textconcept( 5,1,1,123,3,   734 ).
textconcept( 5,1,1,123,4,   735 ).
textconcept( 5,1,1,123,5,   736 ).
textconcept( 5,1,2,124,1,   737 ).
textconcept( 5,1,2,124,2,   738 ).
textconcept( 5,1,2,124,3,   739 ).
textconcept( 5,1,2,124,4,   740 ).
textconcept( 5,1,2,124,5,   741 ).
textconcept( 5,1,2,124,6,   742 ).
textconcept( 5,1,3,125,1,   743 ).
textconcept( 5,1,3,125,2,   744 ).
textconcept( 5,1,3,125,3,   745 ).
textconcept( 5,1,3,125,4,   746 ).
textconcept( 5,1,3,125,5,   747 ).
textconcept( 5,1,3,125,6,   748 ).
textconcept( 5,2,1,126,1,   749 ).
textconcept( 5,2,1,126,2,   750 ).
textconcept( 5,2,1,126,3,   751 ).
textconcept( 5,2,1,126,4,   752 ).
textconcept( 5,2,1,126,5,   753 ).
textconcept( 5,2,1,126,6,   754 ).
textconcept( 5,2,1,126,7,   755 ).
textconcept( 5,2,2,127,1,   756 ).
textconcept( 5,2,2,127,2,   757 ).
textconcept( 5,2,2,127,3,   758 ).
textconcept( 5,2,3,128,1,   759 ).
textconcept( 5,2,3,128,2,   760 ).
textconcept( 5,2,3,128,3,   761 ).
textconcept( 5,2,3,128,4,   762 ).
textconcept( 5,2,3,128,5,   763 ).
textconcept( 5,2,3,128,6,   764 ).
textconcept( 5,2,3,128,7,   765 ).
textconcept( 5,2,3,128,8,   766 ).
```

```
/* text2status.pl */

status(1,title ).
status(2,heading ).
status( 10, theme_marker).
status(14,heading ).
status( 19, theme_marker).
status( 25, theme_marker).
status(31,heading ).
status(55,heading ).
status(60,heading ).
status(87,heading ).
status(110,heading ).
```

```
/* text2func.pl */

function(1,title).
function(2, introduction ).
function(3, background ).
function(4, background ).
function(5, background ).
function(6, background ).
function(7, background ).
function(8, background ).
function(9, background ).
function(10, methodology ).
function(11, background ).
function(12, background ).
function(13, background ).
function(14, background ).
function(15, background ).
function(16, background ).
function(17, background ).
function(18, background ).
function(19, methodology ).
function(20, background ).
function(21, background ).
function(22, background ).
function(23, background ).
function(24, background ).
function(25, methodology ).
function(26, background ).
function(27, background ).
function(28, background ).
function(29, background ).
function(30, background ).
function(31, background ).
function(32, background ).
function(33, background ).
function(34, background ).
function(35, background ).
function(36, background ).
function(37, background ).
function(38, background ).
function(39, background ).
function(40, background ).
function(41, background ).
function(42, background ).
function(43, background ).
function(44, background ).
function(45, background ).
function(46, background ).
function(47, background ).
function(48, background ).
function(49, background ).
function(50, background ).
function(51, background ).
function(52, background ).
function(53, background ).
function(54, background ).
function(55, background ).
function(56, background ).
function(57, background ).
function(58, background ).
function(59, background ).
function(60, background ).
function(61, background ).
function(62, background ).
function(63, background ).
function(64, background ).
function(65, background ).
function(66, background ).
function(67, background ).
function(68, background ).
function(69, background ).
function(70, background ).
function(71, background ).
function(72, background ).
function(73, background ).
function(74, background ).
function(75, background ).
function(76, background ).
function(77, background ).
function(78, background ).
function(79, background ).
function(80, background ).
function(81, background ).
function(82, background ).
function(83, background ).
function(84, background ).
function(85, background ).
function(86, background ).
function(87, background ).
function(88, background ).
function(89, background ).
function(90, background ).
function(91, background ).
function(92, background ).
function(93, background ).
function(94, background ).
function(95, background ).
function(96, background ).
function(97, background ).
function(98, background ).
function(99, background ).
function(100, background ).
function(101, background ).
function(102, background ).
function(103, background ).
function(104, background ).
function(105, background ).
function(106, background ).
function(107, background ).
function(108, background ).
function(109, background ).
function(110, conclusion ).
function(111, background ).
function(112, background ).
function(113, background ).
function(114, background ).
function(115, background ).
function(116, background ).
```

```
/*                txtcpt.pl                */
/*                                         */
/* prolog database produced by txtcpt.spt */
/* concept(concept identity, concept) */
concept( 1, [ comodel] ).
concept( 2, [ language] ).
concept( 3, [ representation, of, technical, knowledge] ).
concept( 4, [ introduction] ).
concept( 5, [ well-known, expert, systems, were,  developed] ).
concept( 6, [ medical] ).
concept( 7, [ scientific, applications] ).
concept( 8, [ years, technical, applications] ).
concept( 9, [came] ).
concept( 10, [ sight] ).
concept( 11, [ type, of, application, requires] ).
concept( 12, [  place, new, types, of, knowledge, representation] ).
concept( 13, [ number, of, papers, appeared] ).
concept( 14, [ topic, following] ).
concept( 15, [ ideas, of, the, basic, work, of, hayes] ).
concept( 16, [ deal] ).
concept( 17, [ modelling, of, physical,  processes, going] ).
concept( 18, [ technical, systems] ).
concept( 19, [  cf] ).
concept( 20, [ of, them, reason] ).
concept( 21, [  physical, processes, making, use, of, a, qualitative,  description, of, physical, entities] ).
concept( 22, [ possible, applications] ).
concept( 23, [ considered, diagnosis] ).
concept( 24, [ prognosis] ).
concept( 25, [  technical, systems] ).
concept( 26, [ control, of, technical, processes] ).
concept( 27, [ approach] ).
concept( 28, [ stress, attention] ).
concept( 29, [ system,  theoretical, principles, of, knowledge, modelling] ).
concept( 30, [ representation, of, geometric, knowledge] ).
concept( 31, [ basis, of, some, principles, of, the, modelling, of,  technical, systems] ).
concept( 32, [ section] ).
concept( 33, [ give] ).
concept( 34, [ introduction] ).
concept( 35, [ knowledge, representation, language,  comodel] ).
concept( 36, [ section] ).
concept( 37, [ main, feature] ).
concept( 38, [  component, oriented, description, of, technical, systems] ).
concept( 39, [ processes, covering, geometrical] ).
concept( 40, [ process, knowledge] ).
concept( 41, [ same, means, of, description] ).
concept( 42, [ application] ).
concept( 43, [ illustrated] ).
concept( 44, [ examples] ).
concept( 45, [ section] ).
concept( 46, [ modelling, of, technical, systems] ).
concept( 47, [ different, works] ).
concept( 48, [ modelling, of, technical,  systems, mentioned] ).
concept( 49, [ common] ).
concept( 50, [ not] ).
concept( 51, [describe] ).
concept( 52, [ technical, system] ).
concept( 53, [ start] ).
concept( 54, [ context, independent, descriptions, of, single, parts] ).
concept( 55, [ functions] ).
concept( 56, [  try] ).
concept( 57, [ infer, descriptions, of, aggregates] ).
concept( 58, [  of, the, parts] ).
concept( 59, [ descriptions] ).
concept( 60, [ based] ).
concept( 61, [ variables] ).
```

```
concept( 63, [  pressure] ).
concept( 64, [  cross-section] ).
concept( 65, [  values, of, variables] ).
concept( 66, [ constraints] ).
concept( 67, [ forbus] ).
concept( 68, [  takes] ).
concept( 69, [ process] ).
concept( 70, [  causing, changes, of, physical,  situations] ).
concept( 71, [ basic, concept, of, his, description] ).
concept( 72, [ de, kleer's] ).
concept( 73, [ brown's, approach] ).
concept( 74, [  physical, component] ).
concept( 75, [ behavior] ).
concept( 76, [ characterized] ).
concept( 77, [ confluences] ).
concept( 78, [plays] ).
concept( 79, [ role] ).
concept( 80, [ kuipers'] ).
concept( 81, [ kassirer's, work] ).
concept( 82, [ based] ).
concept( 83, [ 'universal', constraints] ).
concept( 84, [  related] ).
concept( 85, [ different, types, of, variables] ).
concept( 86, [ raulefs] ).
concept( 87, [attaches] ).
concept( 88, [ importance] ).
concept( 89, [ different, degrees, of, granulation] ).
concept( 90, [ qualitative,  description] ).
concept( 91, [ reasoning] ).
concept( 92, [ time] ).
concept( 93, [ approach] ).
concept( 94, [ following, basic, features] ).
concept( 95, [  1', component, oriented, descripton] ).
concept( 96, [ technical, system] ).
concept( 97, [ regarded] ).
concept( 98, [ composed, of,  a, finite, number, of, components] ).
concept( 99, [ correspond] ).
concept( 100, [ parts, of, a, real, system] ).
concept( 101, [ accumulated] ).
concept( 102, [ aggregates] ).
concept( 103, [  2', qualitative] ).
concept( 104, [ quantitative, representation, of, the, properties, of, physical, objects] ).
concept( 105, [  3', system, theoretical, taxonomy] ).
concept( 106, [ classifying] ).
concept( 107, [ variety, of, components] ).
concept( 108, [ connections] ).
concept( 109, [ component, types] ).
concept( 110, [  interaction, types] ).
concept( 111, [  4', uniform, description, of, the, function] ).
concept( 112, [ geometric, properties, of, components] ).
concept( 113, [ 1'] ).
concept( 114, [ 2'] ).
concept( 115, [ approach] ).
concept( 116, [ related] ).
concept( 117, [ that, of, de, kleer] ).
concept( 118, [ brown] ).
concept( 119, [ of, kuipers] ).
concept( 120, [  kassirer] ).
concept( 121, [ component, orientation, allows, modular,  descriptions, of, large, aggregates] ).
concept( 122, [  description] ).
concept( 123, [ modified] ).
concept( 124, [ adapted] ).
concept( 125, [  other, configurations] ).
concept( 126, [ behaviour, of, components] ).
concept( 127, [ aggregates] ).
```

```
concept( 129, [ means, of, constraints] ).
concept( 130, [ physical, 'properties] ).
concept( 131, [ variables] ).
concept( 132, [ 3'] ).
concept( 133, [ 4'] ).
concept( 134, [ differ] ).
concept( 135, [ works, mentioned] ).
concept( 136, [ consequence, of, a, component, oriented, approach] ).
concept( 137, [ components] ).
concept( 138, [ classified] ).
concept( 139, [ systematic, way] ).
concept( 140, [ principles, of, system, engineering] ).
concept( 141, [ geometric, properties, of, components] ).
concept( 142, [ of, interest] ).
concept( 143, [ far] ).
concept( 144, [ essential, parts, of, the, behaviour, of, the, components] ).
concept( 145, [ reason] ).
concept( 146, [ described] ).
concept( 147, [ same, way] ).
concept( 148, [ functional, properties] ).
concept( 149, [ elements, of, comodel] ).
concept( 150, [ component, oriented, description, language] ).
concept( 151, [ comodel] ).
concept( 152, [ technical, system] ).
concept( 153, [ viewed] ).
concept( 154, [ set, of, phenomenons] ).
concept( 155, [ observable] ).
concept( 156, [ automatic, measurement] ).
concept( 157, [ human, senses] ).
concept( 158, [ phenomenons] ).
concept( 159, [ connected] ).
concept( 160, [ relations] ).
concept( 161, [ results] ).
concept( 162, [ network, consisting, of, phenomenons] ).
concept( 163, [ edges] ).
concept( 164, [ two, types, of, nodes] ).
concept( 165, [ called, components] ).
concept( 166, [ interactions] ).
concept( 167, [ cf, figure] ).
concept( 168, [ components] ).
concept( 169, [ interactions] ).
concept( 170, [ phenomenons] ).
concept( 171, [ basic, objects, of, our, description, language] ).
concept( 172, [ related] ).
concept( 173, [ parts, of, a, technical, system] ).
concept( 174, [ connections, of, these, parts] ).
concept( 175, [ observable, physical, properties] ).
concept( 176, [ real, world] ).
concept( 177, [ comodel, description, of, a, real, technical, system] ).
concept( 178, [consists] ).
concept( 179, [ of, a, number, of, object-definitions] ).
concept( 180, [ defining] ).
concept( 181, [ three, basic, objects] ).
concept( 182, [ compound, object, called, aggregate] ).
concept( 183, [ phenomenon] ).
concept( 184, [ collection, of, a, finite, number, of, physical, properties] ).
concept( 185, [ combined] ).
concept( 186, [ particular, point, of, view] ).
concept( 187, [ physical, property] ).
concept( 188, [ part, of, a, phenomenon] ).
concept( 189, [ observable] ).
concept( 190, [ not] ).
concept( 191, [ component] ).
concept( 192, [ object] ).
concept( 193, [ finite, number, of, gates] ).
```

```
concept( 195, [ each, gate] ).
concept( 196, [ phenomenon] ).
concept( 197, [  assigned] ).
concept( 198, [ gates, serve] ).
concept( 199, [ links] ).
concept( 200, [ components] ).
concept( 201, [ interactions] ).
concept( 202, [ relations] ).
concept( 203, [describe] ).
concept( 204, [ connections] ).
concept( 205, [ values, of, different, physical, properties] ).
concept( 206, [ gates', phenomenons] ).
concept( 207, [define] ).
concept( 208, [ values,  of, physical, properties] ).
concept( 209, [ interaction] ).
concept( 210, [ object] ).
concept( 211, [ finite, number, of,  gates] ).
concept( 212, [ relations] ).
concept( 213, [ component] ).
concept( 214, [ differs] ).
concept( 215, [ component] ).
concept( 216, [ fact] ).
concept( 217, [ relations,  cannot] ).
concept( 218, [define] ).
concept( 219, [ values] ).
concept( 220, [ interactions] ).
concept( 221, [ not] ).
concept( 222, [  characteristic] ).
concept( 223, [ intrinsic, values] ).
concept( 224, [    gates] ).
concept( 225, [ parts, of, components] ).
concept( 226, [ interactions] ).
concept( 227, [ used] ).
concept( 228, [ links] ).
concept( 229, [ both] ).
concept( 230, [ connection, of, a,  component] ).
concept( 231, [ interaction] ).
concept( 232, [ corresponding,  gates] ).
concept( 233, [induces] ).
concept( 234, [ identity] ).
concept( 235, [ corresponding, phenomenons, of, the, gates] ).
concept( 236, [    relationships] ).
concept( 237, [ physical, properties] ).
concept( 238, [ described] ).
concept( 239, [ relations] ).
concept( 240, [ simple, relations] ).
concept( 241, [ equal] ).
concept( 242, [  unequal] ).
concept( 243, [  proportional] ).
concept( 244, [    connection, of, components] ).
concept( 245, [ interactions, results] ).
concept( 246, [ larger, entities, called, aggregates] ).
concept( 247, [ same,  way, aggregates] ).
concept( 248, [ combined] ).
concept( 249, [ form, bigger,  aggregates] ).
concept( 250, [ whole, technical, system] ).
concept( 251, [ viewed] ).
concept( 252, [ one, aggregate, consisting, of, a,  hierarchy, of] ).
concept( 253, [ sub-] ).
concept( 254, [  aggregates] ).
concept( 255, [ definition, of, phenomenons] ).
concept( 256, [  components] ).
concept( 257, [  interactions] ).
concept( 258, [ aggregates, one] ).
concept( 259, [refer] ).
```

```
concept( 261, [ phenomenon] ).
concept( 262, [ component] ).
concept( 263, [ interaction] ).
concept( 264, [ aggregate] ).
concept( 265, [ user-defined, objects] ).
concept( 266, [ symbolized] ).
concept( 267, [ preterminal, symbol, identifier] ).
concept( 268, [ objects, form] ).
concept( 269, [ hierarchy, representing] ).
concept( 270, [ taxonomy, of, system, theoretical, terms] ).
concept( 271, [ each, object, describes] ).
concept( 272, [ aspects, of, a, part, of, a, real, system] ).
concept( 273, [ particu,lar, level, of, generality] ).
concept( 274, [ sample, application, of, comodel] ).
concept( 275, [ paradigm, for] ).
concept( 276, [ representation, of, technical, systems] ).
concept( 277, [ take] ).
concept( 278, [ nitric, acid, cooler, of, lapp,/powers] ).
concept( 279, [shown] ).
concept( 280, [ figure] ).
concept( 281, [ function, of, this, process] ).
concept( 282, [ cool] ).
concept( 283, [ hot, nitric, acid, stream] ).
concept( 284, [ reacting] ).
concept( 285, [ benzene] ).
concept( 286, [ form, nitrobenzene] ).
concept( 287, [ one, top, event] ).
concept( 288, [ system] ).
concept( 289, [ high, temperature] ).
concept( 290, [ nitric, acid, reactor, feed] ).
concept( 291, [ cause] ).
concept( 292, [ reactor, runaway] ).
concept( 293, [ order] ).
concept( 294, [show] ).
concept( 295, [ applicability, of, comodel] ).
concept( 296, [describe] ).
concept( 297, [ broad, variety, of, technical, mechanisms] ).
concept( 298, [ choose, two, parts, of, the, nitric, acid, cooler] ).
concept( 299, [ heat, exchanger] ).
concept( 300, [ energy, flow, system] ).
concept( 301, [ piston] ).
concept( 302, [ cylinder] ).
concept( 303, [ taken] ).
concept( 304, [ cooling, water, pump] ).
concept( 305, [ assumed] ).
concept( 306, [ piston, pump] ).
concept( 307, [ example] ).
concept( 308, [ geometric, interactions] ).
concept( 309, [ 1 heat, exchanger] ).
concept( 310, [ heat, exchanger] ).
concept( 311, [ aggregate] ).
concept( 312, [ four, gates] ).
concept( 313, [ two, fluid, streams, passing] ).
concept( 314, [ interchanging, heat, energy] ).
concept( 315, [ 1t] ).
concept( 316, [ viewed] ).
concept( 317, [ constructed, of, two, equal, components] ).
concept( 318, [ each] ).
concept( 319, [ fluid, stream] ).
concept( 320, [ heat, energy, flowing] ).
concept( 321, [ flowing] ).
concept( 322, [ cf, figure] ).
concept( 323, [ component] ).
concept( 324, [ object, with, three, gates] ).
concept( 325, [ two, fluid, streams] ).
```

```
concept( 327, [ of, figure] ).
concept( 328, [ relevant, physical, properties] ).
concept( 329, [ flow, direction] ).
concept( 330, [ temperature] ).
concept( 331, [ heat, stream] ).
concept( 332, [ combined] ).
concept( 333, [ phenomenons] ).
concept( 334, [ heat, exchanger, component] ).
concept( 335, [ given] ).
concept( 336, [ heat, exchanger, component] ).
concept( 337, [ three, gates] ).
concept( 338, [ two] ).
concept( 339, [ fluid, stream] ).
concept( 340, [ m] ).
concept( 341, [ m] ).
concept( 342, [ one] ).
concept( 343, [ heat, stream] ).
concept( 344, [ e] ).
concept( 345, [ fluid, stream, passes] ).
concept( 346, [ gates, m] ).
concept( 347, [ m] ).
concept( 348, [ opposite, directions] ).
concept( 349, [ centre, of, the, component] ).
concept( 350, [ one, gate] ).
concept( 351, [ enters] ).
concept( 352, [ component] ).
concept( 353, [ other] ).
concept( 354, [ leaves] ).
concept( 355, [ component] ).
concept( 356, [ stated] ).
concept( 357, [ relation] ).
concept( 358, [ other, relations] ).
concept( 359, [define] ).
concept( 360, [ relationships] ).
concept( 361, [ heat, stream] ).
concept( 362, [ temperature, of, the, fluid, stream] ).
concept( 363, [ heat, stream] ).
concept( 364, [ proportional] ).
concept( 365, [ difference, of, temperatures] ).
concept( 366, [ loss] ).
concept( 367, [ gain, of, temperature, of, the, fluid, stream] ).
concept( 368, [ entrance, temperature] ).
concept( 369, [ greater] ).
concept( 370, [ less] ).
concept( 371, [ exit-temperature] ).
concept( 372, [ heat, stream, leaves] ).
concept( 373, [ enters] ).
concept( 374, [ component] ).
concept( 375, [ average, temperature, of, the, fluid, stream] ).
concept( 376, [ equal] ).
concept( 377, [ temperature, of, the, heat, stream] ).
concept( 378, [ fluid, stream] ).
concept( 379, [ heat, stream] ).
concept( 380, [ described] ).
concept( 381, [ interactions] ).
concept( 382, [ interaction, fluidstream] ).
concept( 383, [ two, gates] ).
concept( 384, [ m] ).
concept( 385, [ m] ).
concept( 386, [ relation, states] ).
concept( 387, [ values, of, corresponding, physical, properties, occurring] ).
concept( 388, [ gate, m] ).
concept( 389, [ gate, m,2 respective,ly] ).
concept( 390, [ equal] ).
concept( 391, [ values, of, the, flow, directions] ).
```

```
concept( 393, [ two, gates, e] ).
concept( 394, [ e] ).
concept( 395, [ relations] ).
concept( 396, [define] ).
concept( 397, [ relationships] ).
concept( 398, [ physical, properties, included] ).
concept( 399, [  phenomenon, energyflow, occurring] ).
concept( 400, [ gates,  of, heatenergy] ).
concept( 401, [ difference, of, the, temperatures] ).
concept( 402, [ both, gates, determines] ).
concept( 403, [ heat, stream] ).
concept( 404, [ no,  heat] ).
concept( 405, [ lost] ).
concept( 406, [ both, gates] ).
concept( 407, [ e,1 q, =  e,2 q] ).
concept( 408, [ heat, flows] ).
concept( 409, [ higher] ).
concept( 410, [ lower, temperature] ).
concept( 411, [ enters] ).
concept( 412, [ leaves] ).
concept( 413, [ interaction] ).
concept( 414, [ e] ).
concept( 415, [  e,1 t] ).
concept( 416, [  e,2 t] ).
concept( 417, [ same, holds] ).
concept( 418, [ gate, e] ).
concept( 419, [ of, the, opposite, directions] ).
concept( 420, [ aggregate, heatexchanger] ).
concept( 421, [ cf, figure] ).
concept( 422, [consists] ).
concept( 423, [ of, two, components, of, type] ).
concept( 424, [ one,  interaction, of, type, heatenergy] ).
concept( 425, [ list, of,  gate-pairs] ).
concept( 426, [define] ).
concept( 427, [ constituents, of, the, aggregate] ).
concept( 428, [ composed] ).
concept( 429, [ gate-pair, represents] ).
concept( 430, [ identification, of, two, gates] ).
concept( 431, [ example] ).
concept( 432, [ composed] ).
concept( 433, [ hf] ).
concept( 434, [ identification, of, the,  gates] ).
concept( 435, [1-e] ).
concept( 436, [ hf-e] ).
concept( 437, [ pair] ).
concept( 438, [2-m,1 m] ).
concept( 439, [  says] ).
concept( 440, [ gate, m,1 of, he] ).
concept( 441, [ 'free'] ).
concept( 442, [ not, used] ).
concept( 443, [  composition] ).
concept( 444, [ becomes] ).
concept( 445, [ gate, of, the,  aggregate] ).
concept( 446, [ description, of, the, aggregate, gates] ).
concept( 447, [ adapted] ).
concept( 448, [ description, of, the, corresponding, component, gates] ).
concept( 449, [ 2 piston] ).
concept( 450, [ cylinder] ).
concept( 451, [    piston] ).
concept( 452, [ cylinder] ).
concept( 453, [ basic, elements, of, the,  piston, pump] ).
concept( 454, [ viewed] ).
concept( 455, [ two, components] ).
concept( 456, [ modelled] ).
concept( 457, [ two, objects, full, cylinder] ).
```

```
concept( 459, [ cf, figure] ).
concept( 460, [ phenomenons, join, together, geometric] ).
concept( 461, [ kinematic, properties] ).
concept( 462, [ hollow, body] ).
concept( 463, [ cylinder] ).
concept( 464, [ no, bottoms] ).
concept( 465, [ pipe] ).
concept( 466, [ one, bottom] ).
concept( 467, [ opened] ).
concept( 468, [ two, bottoms] ).
concept( 469, [ closed] ).
concept( 470, [ two, components] ).
concept( 471, [ given] ).
concept( 472, [ definition, of, piston] ).
concept( 473, [ cylinder, radius] ).
concept( 474, [ length] ).
concept( 475, [ degrees, of, freedom] ).
concept( 476, [ number, of, bottoms] ).
concept( 477, [ bound] ).
concept( 478, [ particular, values] ).
concept( 479, [ xr] ).
concept( 480, [ xl] ).
concept( 481, [ trans-fr] ).
concept( 482, [ every, piston] ).
concept( 483, [ cylinder] ).
concept( 484, [ six, degrees, of, freedom] ).
concept( 485, [ translation] ).
concept( 486, [ r-axis] ).
concept( 487, [ l-axis] ).
concept( 488, [ r-axis] ).
concept( 489, [ 90orotation] ).
concept( 490, [ rotation] ).
concept( 491, [ r-axis] ).
concept( 492, [ laxis] ).
concept( 493, [ r-axis] ).
concept( 494, [ 90o, rotation] ).
concept( 495, [ definitions] ).
concept( 496, [describe] ).
concept( 497, [ aspects, of, individual, objects] ).
concept( 498, [ class, of, all, fullcylinders] ).
concept( 499, [ hollowcylinders] ).
concept( 500, [ values] ).
concept( 501, [ position] ).
concept( 502, [ space] ).
concept( 503, [ left, open] ).
concept( 504, [ fullcylinder-in-hollowcylinder-interaction, describes] ).
concept( 505, [ geometric, relationship] ).
concept( 506, [ ful,1 cylinder] ).
concept( 507, [ put] ).
concept( 508, [ hollow, cylinder] ).
concept( 509, [ creates] ).
concept( 510, [ new, hollow, cylinder] ).
concept( 511, [ called, pump, chamber] ).
concept( 512, [ length, depending] ).
concept( 513, [ position, of, the, piston] ).
concept( 514, [ piston, retains, two, degrees, of, freedom] ).
concept( 515, [ resulting, aggregate] ).
concept( 516, [ cf, figure] ).
concept( 517, [ fullcylinder-in-hollowcylinder-interaction] ).
concept( 518, [ three, gates] ).
concept( 519, [ f] ).
concept( 520, [ h] ).
concept( 521, [ r] ).
concept( 522, [combines] ).
concept( 523, [ phenomenons, fullcylinderproperties] ).
```

```
concept( 525, [ form] ).
concept( 526, [ new, phenomenon, hollowcylinderproperties] ).
concept( 527, [ gates, r] ).
concept( 528, [ h] ).
concept( 529, [ same, radius] ).
concept( 530, [ radius, of, the, interaction] ).
concept( 531, [ whole, depends] ).
concept( 532, [ hollowcylinder] ).
concept( 533, [ composed] ).
concept( 534, [ relation, holds] ).
concept( 535, [ length] ).
concept( 536, [ number, of, bottoms] ).
concept( 537, [ two, degrees, of, freedom] ).
concept( 538, [ left] ).
concept( 539, [ translation] ).
concept( 540, [ rotation] ).
concept( 541, [ l-axis] ).
concept( 542, [ aggregate, pumpchamber] ).
concept( 543, [consists] ).
concept( 544, [ of, two, components, of, type, cylinder] ).
concept( 545, [ piston, respective,ly] ).
concept( 546, [ one, interaction, of, type, fullcylinder-in-hollowcylinder] ).
concept( 547, [ interaction] ).
concept( 548, [plays] ).
concept( 549, [ central, role] ).
concept( 550, [ gate, fit,_r] ).
concept( 551, [ gate, of, the, aggregate] ).
concept( 552, [ composition, of, the, three, constituents] ).
concept( 553, [ obvious] ).
concept( 554, [ definition, of, the, aggregate, reflects] ).
concept( 555, [ fact] ).
concept( 556, [ behaviour, of, a, pumpchamber] ).
concept( 557, [ described] ).
concept( 558, [ interaction, of, a, piston] ).
concept( 559, [ cylinder] ).
concept( 560, [ conclusion] ).
concept( 561, [ comodel] ).
concept( 562, [ component, oriented, approach] ).
concept( 563, [ representation, of, knowledge] ).
concept( 564, [ technical, systems] ).
concept( 565, [ description, of, comodel] ).
concept( 566, [ attached, importance] ).
concept( 567, [ ability] ).
concept( 568, [ represent, function] ).
concept( 569, [ geometric, properties, of, components] ).
concept( 570, [ uniform, way] ).
concept( 571, [ intention] ).
concept( 572, [ use, commodel] ).
concept( 573, [ technical, description, language] ).
concept( 574, [ expert, systems] ).
concept( 575, [ area, of, technical, systems] ).
concept( 576, [ particular] ).
concept( 577, [ interpretation, of, the, states, of, those, systems] ).
concept( 578, [ diagnosis, of, failures] ).
concept( 579, [ purpose] ).
concept( 580, [ comodel, description, represents] ).
concept( 581, [ model, of, a, technical, system] ).
concept( 582, [ model, together] ).
concept( 583, [ process, signals] ).
concept( 584, [ processed] ).
concept( 585, [ inference, engine] ).
concept( 586, [ point, additional, heuristic, knowledge] ).
concept( 587, [ symptoms, of, defects] ).
concept( 588, [ strategies] ).
concept( 589, [ search] ).
```

```
concept( 591, [ diagnosis] ).
concept( 592, [ needed] ).
/* heading(no,heading) */
heading( 0,' COMODEL pmarks  A LANGUAGE FOR THE REPRESENTATION OF TECHNICAL K
heading( 1,' INTRODUCTION '  ).
heading( 2,' MODELLING OF TECHNICAL SYSTEMS '  ).
heading( 3,' ELEMENTS OF COMODEL '  ).
heading( 4,' SAMPLE APPLICATION OF COMODEL '  ).
heading( 5,' 1 Heat Exchanger '  ).
heading( 6,' 2 Piston and cylinder '  ).
heading( 7,' CONCLUSION '  ).
/* summary ( sentence no., no. of concepts identified ) */
summary( 1,3).
summary( 2,1).
summary( 3,3).
summary( 4,3).
summary( 5,2).
summary( 6,3).
summary( 7,4).
summary( 8,2).
summary( 9,5).
summary( 10,4).
summary( 11,6).
summary( 12,5).
summary( 13,4).
summary( 14,1).
summary( 15,12).
summary( 16,8).
summary( 17,19).
summary( 18,7).
summary( 19,2).
summary( 20,1).
summary( 21,7).
summary( 22,2).
summary( 23,6).
summary( 24,2).
summary( 25,8).
summary( 26,5).
summary( 27,6).
summary( 28,4).
summary( 29,5).
summary( 30,8).
summary( 31,1).
summary( 32,8).
summary( 33,3).
summary( 34,7).
summary( 35,9).
summary( 36,6).
summary( 37,4).
summary( 38,2).
summary( 39,2).
summary( 40,4).
summary( 41,3).
summary( 42,4).
summary( 43,7).
summary( 44,5).
summary( 45,10).
summary( 46,6).
summary( 47,6).
summary( 48,4).
summary( 49,4).
summary( 50,3).
summary( 51,8).
summary( 52,13).
summary( 53,3).
```

```
summary( 55,1).
summary( 56,6).
summary( 57,6).
summary( 58,6).
summary( 59,16).
summary( 60,1).
summary( 61,5).
summary( 62,8).
summary( 63,5).
summary( 64,6).
summary( 65,2).
summary( 66,9).
summary( 67,11).
summary( 68,2).
summary( 69,5).
summary( 70,5).
summary( 71,7).
summary( 72,3).
summary( 73,4).
summary( 74,4).
summary( 75,6).
summary( 76,3).
summary( 77,6).
summary( 78,3).
summary( 79,4).
summary( 80,9).
summary( 81,3).
summary( 82,5).
summary( 83,6).
summary( 84,6).
summary( 85,9).
summary( 86,3).
summary( 87,2).
summary( 88,3).
summary( 89,6).
summary( 90,2).
summary( 91,8).
summary( 92,2).
summary( 93,10).
summary( 94,13).
summary( 95,9).
summary( 96,5).
summary( 97,5).
summary( 98,3).
summary( 99,5).
summary( 100,5).
summary( 101,7).
summary( 102,2).
summary( 103,1).
summary( 104,5).
summary( 105,5).
summary( 106,3).
summary( 107,2).
summary( 108,2).
summary( 109,6).
summary( 110,1).
summary( 111,4).
summary( 112,6).
summary( 113,8).
summary( 114,3).
summary( 115,4).
summary( 116,7).
/* textstats (no.of sentences,no. of headings,no.of concepts,no.of words,no. of paragraphs ) */
textstats( 116, 8, 592, 1983, 30 ).
```

```
/* prolog database produced by the program called txtcpt.spt*/
/* the format of the database is as follows: */
/* (note: title set to heading no = 0) */
/* textconcept(heading no, paragragh no,paragraphsentence no, sentence no, concept no, concept identit
textconcept( 1,0,0,1,1,  1 ).
textconcept( 1,0,0,1,2,  2 ).
textconcept( 1,0,0,1,3,  3 ).
textconcept( 2,0,0,2,1,  4 ).
textconcept( 2,1,1,3,1,  5 ).
textconcept( 2,1,1,3,2,  6 ).
textconcept( 2,1,1,3,3,  7 ).
textconcept( 2,1,2,4,1,  8 ).
textconcept( 2,1,2,4,2,  9 ).
textconcept( 2,1,2,4,3,  10 ).
textconcept( 2,1,3,5,1,  11 ).
textconcept( 2,1,3,5,2,  12 ).
textconcept( 2,1,4,6,1,  13 ).
textconcept( 2,1,4,6,2,  14 ).
textconcept( 2,1,4,6,3,  15 ).
textconcept( 2,1,5,7,1,  16 ).
textconcept( 2,1,5,7,2,  17 ).
textconcept( 2,1,5,7,3,  18 ).
textconcept( 2,1,5,7,4,  19 ).
textconcept( 2,1,6,8,1,  20 ).
textconcept( 2,1,6,8,2,  21 ).
textconcept( 2,1,7,9,1,  22 ).
textconcept( 2,1,7,9,2,  23 ).
textconcept( 2,1,7,9,3,  24 ).
textconcept( 2,1,7,9,4,  25 ).
textconcept( 2,1,7,9,5,  26 ).
textconcept( 2,2,1,10,1,  27 ).
textconcept( 2,2,1,10,2,  28 ).
textconcept( 2,2,1,10,3,  29 ).
textconcept( 2,2,1,10,4,  30 ).
textconcept( 2,2,2,11,1,  31 ).
textconcept( 2,2,2,11,2,  32 ).
textconcept( 2,2,2,11,3,  33 ).
textconcept( 2,2,2,11,4,  34 ).
textconcept( 2,2,2,11,5,  35 ).
textconcept( 2,2,2,11,6,  36 ).
textconcept( 2,2,3,12,1,  37 ).
textconcept( 2,2,3,12,2,  38 ).
textconcept( 2,2,3,12,3,  39 ).
textconcept( 2,2,3,12,4,  40 ).
textconcept( 2,2,3,12,5,  41 ).
textconcept( 2,2,4,13,1,  42 ).
textconcept( 2,2,4,13,2,  43 ).
textconcept( 2,2,4,13,3,  44 ).
textconcept( 2,2,4,13,4,  45 ).
textconcept( 3,0,0,14,1,  46 ).
textconcept( 3,1,1,15,1,  47 ).
textconcept( 3,1,1,15,2,  48 ).
textconcept( 3,1,1,15,3,  49 ).
textconcept( 3,1,1,15,4,  50 ).
textconcept( 3,1,1,15,5,  51 ).
textconcept( 3,1,1,15,6,  52 ).
textconcept( 3,1,1,15,7,  53 ).
textconcept( 3,1,1,15,8,  54 ).
textconcept( 3,1,1,15,9,  55 ).
textconcept( 3,1,1,15,10,  56 ).
textconcept( 3,1,1,15,11,  57 ).
textconcept( 3,1,1,15,12,  58 ).
textconcept( 3,1,2,16,1,  59 ).
textconcept( 3,1,2,16,2,  60 ).
```

```
textconcept( 3,1,2,16,4,  62 ).
textconcept( 3,1,2,16,5,  63 ).
textconcept( 3,1,2,16,6,  64 ).
textconcept( 3,1,2,16,7,  65 ).
textconcept( 3,1,2,16,8,  66 ).
textconcept( 3,1,3,17,1,  67 ).
textconcept( 3,1,3,17,2,  68 ).
textconcept( 3,1,3,17,3,  69 ).
textconcept( 3,1,3,17,4,  70 ).
textconcept( 3,1,3,17,5,  71 ).
textconcept( 3,1,3,17,6,  72 ).
textconcept( 3,1,3,17,7,  73 ).
textconcept( 3,1,3,17,8,  74 ).
textconcept( 3,1,3,17,9,  75 ).
textconcept( 3,1,3,17,10,  76 ).
textconcept( 3,1,3,17,11,  77 ).
textconcept( 3,1,3,17,12,  78 ).
textconcept( 3,1,3,17,13,  79 ).
textconcept( 3,1,3,17,14,  80 ).
textconcept( 3,1,3,17,15,  81 ).
textconcept( 3,1,3,17,16,  82 ).
textconcept( 3,1,3,17,17,  83 ).
textconcept( 3,1,3,17,18,  84 ).
textconcept( 3,1,3,17,19,  85 ).
textconcept( 3,1,4,18,1,  86 ).
textconcept( 3,1,4,18,2,  87 ).
textconcept( 3,1,4,18,3,  88 ).
textconcept( 3,1,4,18,4,  89 ).
textconcept( 3,1,4,18,5,  90 ).
textconcept( 3,1,4,18,6,  91 ).
textconcept( 3,1,4,18,7,  92 ).
textconcept( 3,2,1,19,1,  93 ).
textconcept( 3,2,1,19,2,  94 ).
textconcept( 3,2,2,20,1,  95 ).
textconcept( 3,2,3,21,1,  96 ).
textconcept( 3,2,3,21,2,  97 ).
textconcept( 3,2,3,21,3,  98 ).
textconcept( 3,2,3,21,4,  99 ).
textconcept( 3,2,3,21,5,  100 ).
textconcept( 3,2,3,21,6,  101 ).
textconcept( 3,2,3,21,7,  102 ).
textconcept( 3,2,4,22,1,  103 ).
textconcept( 3,2,4,22,2,  104 ).
textconcept( 3,2,5,23,1,  105 ).
textconcept( 3,2,5,23,2,  106 ).
textconcept( 3,2,5,23,3,  107 ).
textconcept( 3,2,5,23,4,  108 ).
textconcept( 3,2,5,23,5,  109 ).
textconcept( 3,2,5,23,6,  110 ).
textconcept( 3,2,6,24,1,  111 ).
textconcept( 3,2,6,24,2,  112 ).
textconcept( 3,3,1,25,1,  113 ).
textconcept( 3,3,1,25,2,  114 ).
textconcept( 3,3,1,25,3,  115 ).
textconcept( 3,3,1,25,4,  116 ).
textconcept( 3,3,1,25,5,  117 ).
textconcept( 3,3,1,25,6,  118 ).
textconcept( 3,3,1,25,7,  119 ).
textconcept( 3,3,1,25,8,  120 ).
textconcept( 3,3,2,26,1,  121 ).
textconcept( 3,3,2,26,2,  122 ).
textconcept( 3,3,2,26,3,  123 ).
textconcept( 3,3,2,26,4,  124 ).
textconcept( 3,3,2,26,5,  125 ).
textconcept( 3,3,3,27,1,  126 ).
textconcept( 3,3,3,27,3,  128 ).
textconcept( 3,3,3,27,4,  129 ).
textconcept( 3,3,3,27,5,  130 ).
textconcept( 3,3,3,27,6,  131 ).
textconcept( 3,3,4,28,1,  132 ).
textconcept( 3,3,4,28,2,  133 ).
textconcept( 3,3,4,28,3,  134 ).
textconcept( 3,3,4,28,4,  135 ).
textconcept( 3,3,5,29,1,  136 ).
textconcept( 3,3,5,29,2,  137 ).
textconcept( 3,3,5,29,3,  138 ).
textconcept( 3,3,5,29,4,  139 ).
textconcept( 3,3,5,29,5,  140 ).
textconcept( 3,3,6,30,1,  141 ).
textconcept( 3,3,6,30,2,  142 ).
textconcept( 3,3,6,30,3,  143 ).
textconcept( 3,3,6,30,4,  144 ).
textconcept( 3,3,6,30,5,  145 ).
textconcept( 3,3,6,30,6,  146 ).
textconcept( 3,3,6,30,7,  147 ).
textconcept( 3,3,6,30,8,  148 ).
textconcept( 4,0,0,31,1,  149 ).
textconcept( 4,1,1,32,1,  150 ).
textconcept( 4,1,1,32,2,  151 ).
textconcept( 4,1,1,32,3,  152 ).
textconcept( 4,1,1,32,4,  153 ).
textconcept( 4,1,1,32,5,  154 ).
textconcept( 4,1,1,32,6,  155 ).
textconcept( 4,1,1,32,7,  156 ).
textconcept( 4,1,1,32,8,  157 ).
textconcept( 4,1,2,33,1,  158 ).
textconcept( 4,1,2,33,2,  159 ).
textconcept( 4,1,2,33,3,  160 ).
textconcept( 4,1,3,34,1,  161 ).
textconcept( 4,1,3,34,2,  162 ).
textconcept( 4,1,3,34,3,  163 ).
textconcept( 4,1,3,34,4,  164 ).
textconcept( 4,1,3,34,5,  165 ).
textconcept( 4,1,3,34,6,  166 ).
textconcept( 4,1,3,34,7,  167 ).
textconcept( 4,1,4,35,1,  168 ).
textconcept( 4,1,4,35,2,  169 ).
textconcept( 4,1,4,35,3,  170 ).
textconcept( 4,1,4,35,4,  171 ).
textconcept( 4,1,4,35,5,  172 ).
textconcept( 4,1,4,35,6,  173 ).
textconcept( 4,1,4,35,7,  174 ).
textconcept( 4,1,4,35,8,  175 ).
textconcept( 4,1,4,35,9,  176 ).
textconcept( 4,2,1,36,1,  177 ).
textconcept( 4,2,1,36,2,  178 ).
textconcept( 4,2,1,36,3,  179 ).
textconcept( 4,2,1,36,4,  180 ).
textconcept( 4,2,1,36,5,  181 ).
textconcept( 4,2,1,36,6,  182 ).
textconcept( 4,3,1,37,1,  183 ).
textconcept( 4,3,1,37,2,  184 ).
textconcept( 4,3,1,37,3,  185 ).
textconcept( 4,3,1,37,4,  186 ).
textconcept( 4,4,1,38,1,  187 ).
textconcept( 4,4,1,38,2,  188 ).
textconcept( 4,4,2,39,1,  189 ).
textconcept( 4,4,2,39,2,  190 ).
textconcept( 4,5,1,40,1,  191 ).
textconcept( 4,5,1,40,2,  192 ).
```

```
textconcept( 4,5,2,41,2, 196 ).
textconcept( 4,5,2,41,3, 197 ).
textconcept( 4,5,3,42,1, 198 ).
textconcept( 4,5,3,42,2, 199 ).
textconcept( 4,5,3,42,3, 200 ).
textconcept( 4,5,3,42,4, 201 ).
textconcept( 4,5,4,43,1, 202 ).
textconcept( 4,5,4,43,2, 203 ).
textconcept( 4,5,4,43,3, 204 ).
textconcept( 4,5,4,43,4, 205 ).
textconcept( 4,5,4,43,5, 206 ).
textconcept( 4,5,4,43,6, 207 ).
textconcept( 4,5,4,43,7, 208 ).
textconcept( 4,6,1,44,1, 209 ).
textconcept( 4,6,1,44,2, 210 ).
textconcept( 4,6,1,44,3, 211 ).
textconcept( 4,6,1,44,4, 212 ).
textconcept( 4,6,1,44,5, 213 ).
textconcept( 4,6,2,45,1, 214 ).
textconcept( 4,6,2,45,2, 215 ).
textconcept( 4,6,2,45,3, 216 ).
textconcept( 4,6,2,45,4, 217 ).
textconcept( 4,6,2,45,5, 218 ).
textconcept( 4,6,2,45,6, 219 ).
textconcept( 4,6,2,45,7, 220 ).
textconcept( 4,6,2,45,8, 221 ).
textconcept( 4,6,2,45,9, 222 ).
textconcept( 4,6,2,45,10, 223 ).
textconcept( 4,7,1,46,1, 224 ).
textconcept( 4,7,1,46,2, 225 ).
textconcept( 4,7,1,46,3, 226 ).
textconcept( 4,7,1,46,4, 227 ).
textconcept( 4,7,1,46,5, 228 ).
textconcept( 4,7,1,46,6, 229 ).
textconcept( 4,7,2,47,1, 230 ).
textconcept( 4,7,2,47,2, 231 ).
textconcept( 4,7,2,47,3, 232 ).
textconcept( 4,7,2,47,4, 233 ).
textconcept( 4,7,2,47,5, 234 ).
textconcept( 4,7,2,47,6, 235 ).
textconcept( 4,8,1,48,1, 236 ).
textconcept( 4,8,1,48,2, 237 ).
textconcept( 4,8,1,48,3, 238 ).
textconcept( 4,8,1,48,4, 239 ).
textconcept( 4,8,2,49,1, 240 ).
textconcept( 4,8,2,49,2, 241 ).
textconcept( 4,8,2,49,3, 242 ).
textconcept( 4,8,2,49,4, 243 ).
textconcept( 4,9,1,50,1, 244 ).
textconcept( 4,9,1,50,2, 245 ).
textconcept( 4,9,1,50,3, 246 ).
textconcept( 4,9,2,51,1, 247 ).
textconcept( 4,9,2,51,2, 248 ).
textconcept( 4,9,2,51,3, 249 ).
textconcept( 4,9,2,51,4, 250 ).
textconcept( 4,9,2,51,5, 251 ).
textconcept( 4,9,2,51,6, 252 ).
textconcept( 4,9,2,51,7, 253 ).
textconcept( 4,9,2,51,8, 254 ).
textconcept( 4,10,1,52,1, 255 ).
textconcept( 4,10,1,52,2, 256 ).
textconcept( 4,10,1,52,3, 257 ).
textconcept( 4,10,1,52,4, 258 ).
textconcept( 4,10,1,52,7, 261 ).
textconcept( 4,10,1,52,8, 262 ).
textconcept( 4,10,1,52,9, 263 ).
textconcept( 4,10,1,52,10, 264 ).
textconcept( 4,10,1,52,11, 265 ).
textconcept( 4,10,1,52,12, 266 ).
textconcept( 4,10,1,52,13, 267 ).
textconcept( 4,10,2,53,1, 268 ).
textconcept( 4,10,2,53,2, 269 ).
textconcept( 4,10,2,53,3, 270 ).
textconcept( 4,10,3,54,1, 271 ).
textconcept( 4,10,3,54,2, 272 ).
textconcept( 4,10,3,54,3, 273 ).
textconcept( 5,0,0,55,1, 274 ).
textconcept( 5,1,1,56,1, 275 ).
textconcept( 5,1,1,56,2, 276 ).
textconcept( 5,1,1,56,3, 277 ).
textconcept( 5,1,1,56,4, 278 ).
textconcept( 5,1,1,56,5, 279 ).
textconcept( 5,1,1,56,6, 280 ).
textconcept( 5,1,2,57,1, 281 ).
textconcept( 5,1,2,57,2, 282 ).
textconcept( 5,1,2,57,3, 283 ).
textconcept( 5,1,2,57,4, 284 ).
textconcept( 5,1,2,57,5, 285 ).
textconcept( 5,1,2,57,6, 286 ).
textconcept( 5,1,3,58,1, 287 ).
textconcept( 5,1,3,58,2, 288 ).
textconcept( 5,1,3,58,3, 289 ).
textconcept( 5,1,3,58,4, 290 ).
textconcept( 5,1,3,58,5, 291 ).
textconcept( 5,1,3,58,6, 292 ).
textconcept( 5,2,1,59,1, 293 ).
textconcept( 5,2,1,59,2, 294 ).
textconcept( 5,2,1,59,3, 295 ).
textconcept( 5,2,1,59,4, 296 ).
textconcept( 5,2,1,59,5, 297 ).
textconcept( 5,2,1,59,6, 298 ).
textconcept( 5,2,1,59,7, 299 ).
textconcept( 5,2,1,59,8, 300 ).
textconcept( 5,2,1,59,9, 301 ).
textconcept( 5,2,1,59,10, 302 ).
textconcept( 5,2,1,59,11, 303 ).
textconcept( 5,2,1,59,12, 304 ).
textconcept( 5,2,1,59,13, 305 ).
textconcept( 5,2,1,59,14, 306 ).
textconcept( 5,2,1,59,15, 307 ).
textconcept( 5,2,1,59,16, 308 ).
textconcept( 6,0,0,60,1, 309 ).
textconcept( 6,1,1,61,1, 310 ).
textconcept( 6,1,1,61,2, 311 ).
textconcept( 6,1,1,61,3, 312 ).
textconcept( 6,1,1,61,4, 313 ).
textconcept( 6,1,1,61,5, 314 ).
textconcept( 6,1,2,62,1, 315 ).
textconcept( 6,1,2,62,2, 316 ).
textconcept( 6,1,2,62,3, 317 ).
textconcept( 6,1,2,62,4, 318 ).
textconcept( 6,1,2,62,5, 319 ).
textconcept( 6,1,2,62,6, 320 ).
textconcept( 6,1,2,62,7, 321 ).
textconcept( 6,1,2,62,8, 322 ).
textconcept( 6,1,3,63,1, 323 ).
textconcept( 6,1,3,63,2, 324 ).
```

```
textconcept( 6,1,3,63,4, 326 ).
textconcept( 6,1,3,63,5, 327 ).
textconcept( 6,1,4,64,1, 328 ).
textconcept( 6,1,4,64,2, 329 ).
textconcept( 6,1,4,64,3, 330 ).
textconcept( 6,1,4,64,4, 331 ).
textconcept( 6,1,4,64,5, 332 ).
textconcept( 6,1,4,64,6, 333 ).
textconcept( 6,2,1,65,1, 334 ).
textconcept( 6,2,1,65,2, 335 ).
textconcept( 6,2,2,66,1, 336 ).
textconcept( 6,2,2,66,2, 337 ).
textconcept( 6,2,2,66,3, 338 ).
textconcept( 6,2,2,66,4, 339 ).
textconcept( 6,2,2,66,5, 340 ).
textconcept( 6,2,2,66,6, 341 ).
textconcept( 6,2,2,66,7, 342 ).
textconcept( 6,2,2,66,8, 343 ).
textconcept( 6,2,2,66,9, 344 ).
textconcept( 6,2,3,67,1, 345 ).
textconcept( 6,2,3,67,2, 346 ).
textconcept( 6,2,3,67,3, 347 ).
textconcept( 6,2,3,67,4, 348 ).
textconcept( 6,2,3,67,5, 349 ).
textconcept( 6,2,3,67,6, 350 ).
textconcept( 6,2,3,67,7, 351 ).
textconcept( 6,2,3,67,8, 352 ).
textconcept( 6,2,3,67,9, 353 ).
textconcept( 6,2,3,67,10, 354 ).
textconcept( 6,2,3,67,11, 355 ).
textconcept( 6,2,4,68,1, 356 ).
textconcept( 6,2,4,68,2, 357 ).
textconcept( 6,2,5,69,1, 358 ).
textconcept( 6,2,5,69,2, 359 ).
textconcept( 6,2,5,69,3, 360 ).
textconcept( 6,2,5,69,4, 361 ).
textconcept( 6,2,5,69,5, 362 ).
textconcept( 6,2,6,70,1, 363 ).
textconcept( 6,2,6,70,2, 364 ).
textconcept( 6,2,6,70,3, 365 ).
textconcept( 6,2,6,70,4, 366 ).
textconcept( 6,2,6,70,5, 367 ).
textconcept( 6,2,7,71,1, 368 ).
textconcept( 6,2,7,71,2, 369 ).
textconcept( 6,2,7,71,3, 370 ).
textconcept( 6,2,7,71,4, 371 ).
textconcept( 6,2,7,71,5, 372 ).
textconcept( 6,2,7,71,6, 373 ).
textconcept( 6,2,7,71,7, 374 ).
textconcept( 6,2,8,72,1, 375 ).
textconcept( 6,2,8,72,2, 376 ).
textconcept( 6,2,8,72,3, 377 ).
textconcept( 6,3,1,73,1, 378 ).
textconcept( 6,3,1,73,2, 379 ).
textconcept( 6,3,1,73,3, 380 ).
textconcept( 6,3,1,73,4, 381 ).
textconcept( 6,3,2,74,1, 382 ).
textconcept( 6,3,2,74,2, 383 ).
textconcept( 6,3,2,74,3, 384 ).
textconcept( 6,3,2,74,4, 385 ).
textconcept( 6,3,3,75,1, 386 ).
textconcept( 6,3,3,75,2, 387 ).
textconcept( 6,3,3,75,3, 388 ).
textconcept( 6,3,3,75,4, 389 ).
textconcept( 6,3,3,75,5, 390 ).
textconcept( 6,4,1,76,1, 392 ).
textconcept( 6,4,1,76,2, 393 ).
textconcept( 6,4,1,76,3, 394 ).
textconcept( 6,4,2,77,1, 395 ).
textconcept( 6,4,2,77,2, 396 ).
textconcept( 6,4,2,77,3, 397 ).
textconcept( 6,4,2,77,4, 398 ).
textconcept( 6,4,2,77,5, 399 ).
textconcept( 6,4,2,77,6, 400 ).
textconcept( 6,4,3,78,1, 401 ).
textconcept( 6,4,3,78,2, 402 ).
textconcept( 6,4,3,78,3, 403 ).
textconcept( 6,4,4,79,1, 404 ).
textconcept( 6,4,4,79,2, 405 ).
textconcept( 6,4,4,79,3, 406 ).
textconcept( 6,4,4,79,4, 407 ).
textconcept( 6,4,5,80,1, 408 ).
textconcept( 6,4,5,80,2, 409 ).
textconcept( 6,4,5,80,3, 410 ).
textconcept( 6,4,5,80,4, 411 ).
textconcept( 6,4,5,80,5, 412 ).
textconcept( 6,4,5,80,6, 413 ).
textconcept( 6,4,5,80,7, 414 ).
textconcept( 6,4,5,80,8, 415 ).
textconcept( 6,4,5,80,9, 416 ).
textconcept( 6,4,6,81,1, 417 ).
textconcept( 6,4,6,81,2, 418 ).
textconcept( 6,4,6,81,3, 419 ).
textconcept( 6,5,1,82,1, 420 ).
textconcept( 6,5,1,82,2, 421 ).
textconcept( 6,5,1,82,3, 422 ).
textconcept( 6,5,1,82,4, 423 ).
textconcept( 6,5,1,82,5, 424 ).
textconcept( 6,5,2,83,1, 425 ).
textconcept( 6,5,2,83,2, 426 ).
textconcept( 6,5,2,83,3, 427 ).
textconcept( 6,5,2,83,4, 428 ).
textconcept( 6,5,2,83,5, 429 ).
textconcept( 6,5,2,83,6, 430 ).
textconcept( 6,5,3,84,1, 431 ).
textconcept( 6,5,3,84,2, 432 ).
textconcept( 6,5,3,84,3, 433 ).
textconcept( 6,5,3,84,4, 434 ).
textconcept( 6,5,3,84,5, 435 ).
textconcept( 6,5,3,84,6, 436 ).
textconcept( 6,5,4,85,1, 437 ).
textconcept( 6,5,4,85,2, 438 ).
textconcept( 6,5,4,85,3, 439 ).
textconcept( 6,5,4,85,4, 440 ).
textconcept( 6,5,4,85,5, 441 ).
textconcept( 6,5,4,85,6, 442 ).
textconcept( 6,5,4,85,7, 443 ).
textconcept( 6,5,4,85,8, 444 ).
textconcept( 6,5,4,85,9, 445 ).
textconcept( 6,5,5,86,1, 446 ).
textconcept( 6,5,5,86,2, 447 ).
textconcept( 6,5,5,86,3, 448 ).
textconcept( 7,0,0,87,1, 449 ).
textconcept( 7,0,0,87,2, 450 ).
textconcept( 7,1,1,88,1, 451 ).
textconcept( 7,1,1,88,2, 452 ).
textconcept( 7,1,1,88,3, 453 ).
textconcept( 7,1,2,89,1, 454 ).
textconcept( 7,1,2,89,2, 455 ).
textconcept( 7,1,2,89,3, 456 ).
```

```
textconcept( 7,1,2,89,5, 458 ).
textconcept( 7,1,2,89,6, 459 ).
textconcept( 7,1,3,90,1, 460 ).
textconcept( 7,1,3,90,2, 461 ).
textconcept( 7,1,4,91,1, 462 ).
textconcept( 7,1,4,91,2, 463 ).
textconcept( 7,1,4,91,3, 464 ).
textconcept( 7,1,4,91,4, 465 ).
textconcept( 7,1,4,91,5, 466 ).
textconcept( 7,1,4,91,6, 467 ).
textconcept( 7,1,4,91,7, 468 ).
textconcept( 7,1,4,91,8, 469 ).
textconcept( 7,1,5,92,1, 470 ).
textconcept( 7,1,5,92,2, 471 ).
textconcept( 7,2,1,93,1, 472 ).
textconcept( 7,2,1,93,2, 473 ).
textconcept( 7,2,1,93,3, 474 ).
textconcept( 7,2,1,93,4, 475 ).
textconcept( 7,2,1,93,5, 476 ).
textconcept( 7,2,1,93,6, 477 ).
textconcept( 7,2,1,93,7, 478 ).
textconcept( 7,2,1,93,8, 479 ).
textconcept( 7,2,1,93,9, 480 ).
textconcept( 7,2,1,93,10, 481 ).
textconcept( 7,2,2,94,1, 482 ).
textconcept( 7,2,2,94,2, 483 ).
textconcept( 7,2,2,94,3, 484 ).
textconcept( 7,2,2,94,4, 485 ).
textconcept( 7,2,2,94,5, 486 ).
textconcept( 7,2,2,94,6, 487 ).
textconcept( 7,2,2,94,7, 488 ).
textconcept( 7,2,2,94,8, 489 ).
textconcept( 7,2,2,94,9, 490 ).
textconcept( 7,2,2,94,10, 491 ).
textconcept( 7,2,2,94,11, 492 ).
textconcept( 7,2,2,94,12, 493 ).
textconcept( 7,2,2,94,13, 494 ).
textconcept( 7,2,3,95,1, 495 ).
textconcept( 7,2,3,95,2, 496 ).
textconcept( 7,2,3,95,3, 497 ).
textconcept( 7,2,3,95,4, 498 ).
textconcept( 7,2,3,95,5, 499 ).
textconcept( 7,2,3,95,6, 500 ).
textconcept( 7,2,3,95,7, 501 ).
textconcept( 7,2,3,95,8, 502 ).
textconcept( 7,2,3,95,9, 503 ).
textconcept( 7,3,1,96,1, 504 ).
textconcept( 7,3,1,96,2, 505 ).
textconcept( 7,3,1,96,3, 506 ).
textconcept( 7,3,1,96,4, 507 ).
textconcept( 7,3,1,96,5, 508 ).
textconcept( 7,3,2,97,1, 509 ).
textconcept( 7,3,2,97,2, 510 ).
textconcept( 7,3,2,97,3, 511 ).
textconcept( 7,3,2,97,4, 512 ).
textconcept( 7,3,2,97,5, 513 ).
textconcept( 7,3,3,98,1, 514 ).
textconcept( 7,3,3,98,2, 515 ).
textconcept( 7,3,3,98,3, 516 ).
textconcept( 7,4,1,99,1, 517 ).
textconcept( 7,4,1,99,2, 518 ).
textconcept( 7,4,1,99,3, 519 ).
textconcept( 7,4,1,99,4, 520 ).
textconcept( 7,4,1,99,5, 521 ).
textconcept( 7,4,2,100,1, 522 ).
textconcept( 7,4,2,100,3, 524 ).
textconcept( 7,4,2,100,4, 525 ).
textconcept( 7,4,2,100,5, 526 ).
textconcept( 7,4,3,101,1, 527 ).
textconcept( 7,4,3,101,2, 528 ).
textconcept( 7,4,3,101,3, 529 ).
textconcept( 7,4,3,101,4, 530 ).
textconcept( 7,4,3,101,5, 531 ).
textconcept( 7,4,3,101,6, 532 ).
textconcept( 7,4,3,101,7, 533 ).
textconcept( 7,4,4,102,1, 534 ).
textconcept( 7,4,4,102,2, 535 ).
textconcept( 7,4,5,103,1, 536 ).
textconcept( 7,4,6,104,1, 537 ).
textconcept( 7,4,6,104,2, 538 ).
textconcept( 7,4,6,104,3, 539 ).
textconcept( 7,4,6,104,4, 540 ).
textconcept( 7,4,6,104,5, 541 ).
textconcept( 7,4,7,105,1, 542 ).
textconcept( 7,4,7,105,2, 543 ).
textconcept( 7,4,7,105,3, 544 ).
textconcept( 7,4,7,105,4, 545 ).
textconcept( 7,4,7,105,5, 546 ).
textconcept( 7,4,8,106,1, 547 ).
textconcept( 7,4,8,106,2, 548 ).
textconcept( 7,4,8,106,3, 549 ).
textconcept( 7,4,9,107,1, 550 ).
textconcept( 7,4,9,107,2, 551 ).
textconcept( 7,4,10,108,1, 552 ).
textconcept( 7,4,10,108,2, 553 ).
textconcept( 7,5,1,109,1, 554 ).
textconcept( 7,5,1,109,2, 555 ).
textconcept( 7,5,1,109,3, 556 ).
textconcept( 7,5,1,109,4, 557 ).
textconcept( 7,5,1,109,5, 558 ).
textconcept( 7,5,1,109,6, 559 ).
textconcept( 8,0,0,110,1, 560 ).
textconcept( 8,1,1,111,1, 561 ).
textconcept( 8,1,1,111,2, 562 ).
textconcept( 8,1,1,111,3, 563 ).
textconcept( 8,1,1,111,4, 564 ).
textconcept( 8,1,2,112,1, 565 ).
textconcept( 8,1,2,112,2, 566 ).
textconcept( 8,1,2,112,3, 567 ).
textconcept( 8,1,2,112,4, 568 ).
textconcept( 8,1,2,112,5, 569 ).
textconcept( 8,1,2,112,6, 570 ).
textconcept( 8,2,1,113,1, 571 ).
textconcept( 8,2,1,113,2, 572 ).
textconcept( 8,2,1,113,3, 573 ).
textconcept( 8,2,1,113,4, 574 ).
textconcept( 8,2,1,113,5, 575 ).
textconcept( 8,2,1,113,6, 576 ).
textconcept( 8,2,1,113,7, 577 ).
textconcept( 8,2,1,113,8, 578 ).
textconcept( 8,3,1,114,1, 579 ).
textconcept( 8,3,1,114,2, 580 ).
textconcept( 8,3,1,114,3, 581 ).
textconcept( 8,3,2,115,1, 582 ).
textconcept( 8,3,2,115,2, 583 ).
textconcept( 8,3,2,115,3, 584 ).
textconcept( 8,3,2,115,4, 585 ).
textconcept( 8,3,3,116,1, 586 ).
textconcept( 8,3,3,116,2, 587 ).
textconcept( 8,3,3,116,3, 588 ).
```

```
textconcept( 8,3,3,116,4,  589 ).
textconcept( 8,3,3,116,5,  590 ).
textconcept( 8,3,3,116,6,  591 ).
textconcept( 8,3,3,116,7,  592 ).
```

```
/* text3status.pl */

status(1,title ).
status(2,heading ).
status(18,heading ).
status(22,heading ).
status(37,heading ).
status(49,heading ).
status(63,heading ).
status(74,heading ).
status( 75, theme_marker).
status( 82, theme_marker).
```

```
/*text3func.pl*/

function(1,title).
function(2, introduction ).
function(3, background ).
function(4, background ).
function(5, background ).
function(6, background ).
function(7, background ).
function(8, background ).
function(9, background ).
function(10, background ).
function(11, background ).
function(12, background ).
function(13, background ).
function(14, background ).
function(15, background ).
function(16, background ).
function(17, background ).
function(18, background ).
function(19, background ).
function(20, background ).
function(21, background ).
function(22, background ).
function(23, background ).
function(24, background ).
function(25, background ).
function(26, background ).
function(27, background ).
function(28, background ).
function(29, background ).
function(30, background ).
function(31, background ).
function(32, background ).
function(33, background ).
function(34, background ).
function(35, background ).
function(36, background ).
function(37, background ).
function(38, background ).
function(39, background ).
function(40, background ).
function(41, background ).
function(42, background ).
function(43, background ).
function(44, background ).
function(45, background ).
function(46, background ).
function(47, background ).
function(48, background ).
function(49, background ).
function(50, background ).
function(51, background ).
function(52, background ).
function(53, background ).
function(54, background ).
function(55, background ).
function(56, background ).
function(57, background ).
function(58, background ).
function(59, background ).
function(60, background ).
function(61, background ).
function(62, background ).
function(63, background ).
function(64, background ).
function(65, background ).
function(66, background ).
function(67, background ).
function(68, background ).
function(69, background ).
function(70, background ).
function(71, background ).
function(72, background ).
function(73, background ).
function(74, conclusion ).
function(75, background ).
function(76, background ).
function(77, background ).
function(78, background ).
function(79, background ).
function(80, background ).
function(81, background ).
function(82, background ).
```

```
/* text3cpt.pl */
/* prolog database produced by txtcpt.spt */
/* concept(concept identity, concept) */
concept( 1, [ checking] ).
concept( 2, [ expert, systems, knowledge, base] ).
concept( 3, [ consistency] ).
concept( 4, [ completeness] ).
concept( 5, [ introduction] ).
concept( 6, [ lockheed, expert, system] ).
concept( 7, [ les] ).
concept( 8, [ generio, rule-based, expert, system, tool] ).
concept( 9, [ emycin] ).
concept( 10, [ used] ).
concept( 11, [ framework] ).
concept( 12, [ construct, expert, systems] ).
concept( 13, [ areas] ).
concept( 14, [ eleotronic, equipment, diągnosis] ).
concept( 15, [ design, checking] ).
concept( 16, [ photo, interpretation] ).
concept( 17, [ hazard, analysis] ).
concept( 18, [ les, employs] ).
concept( 19, [ combination, of, goal-driven] ).
concept( 20, [ data-driven, rules] ).
concept( 21, [ latter] ).
concept( 22, [ attached] ).
concept( 23, [ factual, database] ).
concept( 24, [ demons] ).
concept( 25, [ one, objective] ).
concept( 26, [ design, of, les] ).
concept( 27, [make] ).
concept( 28, [ easy] ).
concept( 29, [ use] ).
concept( 30, [ debugging, tools] ).
concept( 31, [ aids, were, added] ).
concept( 32, [ les, program] ).
concept( 33, [ one, of, these, aids] ).
concept( 34, [ knowledge, base, completeness] ).
concept( 35, [ consistency, verification, program, called, check] ).
concept( 36, [ purpose] ).
concept( 37, [ help] ).
concept( 38, [ knowledge, engineer, check] ).
concept( 39, [ knowledge, base] ).
concept( 40, [ constructed] ).
concept( 41, [ redundant, rules] ).
concept( 42, [ conflicting, rules] ).
concept( 43, [ subsumed, rules] ).
concept( 44, [ missing, rules] ).
concept( 45, [ unreachable, clauses] ).
concept( 46, [ deadend, clauses] ).
concept( 47, [ check, does, not, perform, any, syntax, checking] ).
concept( 48, [ rules] ).
concept( 49, [ rule, files] ).
concept( 50, [ loaded] ).
concept( 51, [ knowledge, base] ).
concept( 52, [ check] ).
concept( 53, [ analyzes] ).
concept( 54, [ knowledge, base] ).
concept( 55, [ rules] ).
concept( 56, [ facts] ).
concept( 57, [ loaded] ).
concept( 58, [ knowledge, base] ).
concept( 59, [ unlike, teiresias] ).
concept( 60, [ performs] ).
concept( 61, [ assessment, of, rules] ).
concept( 62, [ setting, of, a, problem-solving, session] ).
```

```
concept( 63, [ suwa] ).
concept( 64, [ scott] ).
concept( 65, [ shortliffe] ).
concept( 66, [ written] ).
concept( 67, [ program] ).
concept( 68, [ verifying] ).
concept( 69, [ set, of, rules] ).
concept( 70, [ spans] ).
concept( 71, [ knowledge, base] ).
concept( 72, [ program] ).
concept( 73, [ devised] ).
concept( 74, [ tested] ).
concept( 75, [ context, of, the, oncocin, system] ).
concept( 76, [ emycin-like, system] ).
concept( 77, [ work, differs] ).
concept( 78, [ check, includes, unreachable, clauses] ).
concept( 79, [ deadend, clauses] ).
concept( 80, [ two, additional, rule, checking, criteria] ).
concept( 81, [ check, produces] ).
concept( 82, [ dependency, chart] ).
concept( 83, [ detects, any, circular, rule, chains] ).
concept( 84, [ check] ).
concept( 85, [ devised] ).
concept( 86, [ tested] ).
concept( 87, [ generic, expert, system] ).
concept( 88, [ case-grammar, rules] ).
concept( 89, [ 'frame', database] ).
concept( 90, [ used] ).
concept( 91, [ analyze] ).
concept( 92, [ wide, variety, of, knowledge, bases] ).
concept( 93, [ check, does, not, take] ).
concept( 94, [ account, certainty, faotors] ).
concept( 95, [ checking, the, rule, base] ).
concept( 96, [ checking] ).
concept( 97, [ consistency] ).
concept( 98, [ completeness] ).
concept( 99, [ static, analysis, of, the, rules] ).
concept( 100, [ detect] ).
concept( 101, [ potential, problems] ).
concept( 102, [ gaps] ).
concept( 103, [ exist] ).
concept( 104, [ rule, base] ).
concept( 105, [identify] ).
concept( 106, [ give, definitions] ).
concept( 107, [ seven, criteria] ).
concept( 108, [ used] ).
concept( 109, [ check] ).
concept( 110, [ perform, static, analysis, of, any, rule, base, constructed] ).
concept( 111, [ use] ).
concept( 112, [ les] ).
concept( 113, [ four, criteria] ).
concept( 114, [ concerned] ).
concept( 115, [ potential, problems] ).
concept( 116, [ last, three, criteria] ).
concept( 117, [ concerned] ).
concept( 118, [ gaps] ).
concept( 119, [ the, knowledge, base] ).
concept( 120, [ 1 potential, problems] ).
concept( 121, [ knowledge, base] ).
concept( 122, [ analyzing] ).
concept( 123, [ logical, semantics, of, the, rules, represented] ).
concept( 124, [ les's, case, grammar, format] ).
concept( 125, [ check] ).
concept( 126, [ detect, redundant, rules] ).
concept( 127, [ conflicting, rules] ).
concept( 128, [ rules] ).
```

```
concept( 129, [ subsumed] ).
concept( 130, [ other, rules] ).
concept( 131, [ circular-rule, chains] ).
concept( 132, [ following, definitions] ).
concept( 133, [ four, potential, problems] ).
concept( 134, [ used] ).
concept( 135, [ check] ).
concept( 136, [ redundant, rules] ).
concept( 137, [ two, rules, succeed] ).
concept( 138, [ same, situation] ).
concept( 139, [ same, results] ).
concept( 140, [ les] ).
concept( 141, [ means] ).
concept( 142, [ parts, of, the, two, rules] ).
concept( 143, [ equivalent] ).
concept( 144, [ one] ).
concept( 145, [ more] ).
concept( 146, [ clauses] ).
concept( 147, [ equivalent] ).
concept( 148, [ les, allows, variables] ).
concept( 149, [ rules] ).
concept( 150, [ equivalent, means] ).
concept( 151, [ same, specific, object, names] ).
concept( 152, [ match] ).
concept( 153, [ corresponding, variables] ).
concept( 154, [ example] ).
concept( 155, [ rule, 'p] ).
concept( 156, [ x] ).
concept( 157, [ --,> q] ).
concept( 158, [ x] ).
concept( 159, [ '] ).
concept( 160, [ equivalent] ).
concept( 161, [ rule, 'p] ).
concept( 162, [ y] ).
concept( 163, [ --,> q] ).
concept( 164, [ y] ).
concept( 165, [ '] ).
concept( 166, [ where, x] ).
concept( 167, [ y] ).
concept( 168, [ variables] ).
concept( 169, [ conflicting, rules] ).
concept( 170, [ two, rules, succeed] ).
concept( 171, [ same, situation] ).
concept( 172, [ conflicting, results] ).
concept( 173, [ les] ).
concept( 174, [ means] ).
concept( 175, [ parts, of, the, two, rules] ).
concept( 176, [ squivalent] ).
concept( 177, [ one] ).
concept( 178, [ more] ).
concept( 179, [ clauses] ).
concept( 180, [ contradictory] ).
concept( 181, [ one, pair, of, if, clauses] ).
concept( 182, [ contradictory] ).
concept( 183, [ equivalent] ).
concept( 184, [ clauses] ).
concept( 185, [ example] ).
concept( 186, [ rule, 'p] ).
concept( 187, [ x] ).
concept( 188, [ --,> not] ).
concept( 189, [ q] ).
concept( 190, [ x] ).
concept( 191, [ '] ).
concept( 192, [ contradictory] ).
concept( 193, [ rule, 'p] ).
concept( 194, [ x] ).
```

```
concept( 195, [ --,> q] ).
concept( 196, [ x] ).
concept( 197, [ '] ).
concept( 198, [ subsumed, rules] ).
concept( 199, [ two, rules] ).
concept( 200, [ same, results] ).
concept( 201, [ one] ).
concept( 202, [contains] ).
concept( 203, [ additional, constraints] ).
concept( 204, [ situations] ).
concept( 205, [ succeed] ).
concept( 206, [ les] ).
concept( 207, [ means, one] ).
concept( 208, [ more] ).
concept( 209, [ clauses] ).
concept( 210, [ equivalent] ).
concept( 211, [ part, of, cne, rule] ).
concept( 212, [contains] ).
concept( 213, [ fewer, constraints] ).
concept( 214, [ clauses] ).
concept( 215, [ part, of, the, other, rule] ).
concept( 216, [ example] ).
concept( 217, [ rule, '] ).
concept( 218, [ p] ).
concept( 219, [ x] ).
concept( 220, [ q] ).
concept( 221, [ y] ).
concept( 222, [ --,> r] ).
concept( 223, [ z] ).
concept( 224, [ '] ).
concept( 225, [ subsumed] ).
concept( 226, [ rule, 'p] ).
concept( 227, [ x] ).
concept( 228, [ --,> r] ).
concept( 229, [ z] ).
concept( 230, [ '] ).
concept( 231, [ circular, rules] ).
concept( 232, [ set, of, rules] ).
concept( 233, [ circular-rule, set] ).
concept( 234, [ chaining, of, those, rules] ).
concept( 235, [ set, forms] ).
concept( 236, [ cycle] ).
concept( 237, [ example] ).
concept( 238, [ had] ).
concept( 239, [ set, of, rules] ).
concept( 240, [ follows] ).
concept( 241, [ 'p] ).
concept( 242, [ x] ).
concept( 243, [ --,> q] ).
concept( 244, [ x] ).
concept( 245, [ '] ).
concept( 246, [ 'q] ).
concept( 247, [ x] ).
concept( 248, [ --,> r] ).
concept( 249, [ x] ).
concept( 250, [ '] ).
concept( 251, [ 'r] ).
concept( 252, [ x] ).
concept( 253, [ --,> p] ).
concept( 254, [ x] ).
concept( 255, [ '] ).
concept( 256, [ goal] ).
concept( 257, [ r] ).
concept( 258, [ where] ).
concept( 259, [ constant] ).
concept( 260, [ system] ).
```

```
concept( 261, [ enter] ).
concept( 262, [ infinite, loop] ).
concept( 263, [ run, time] ).
concept( 264, [ system] ).
concept( 265, [ special, way, of,  handling, circular, rules] ).
concept( 266, [ 2  potential, gaps] ).
concept( 267, [ knowledge, base] ).
concept( 268, [ development, of, a, knowledge-based, system] ).
concept( 269, [ iterative, process] ).
concept( 270, [ knowledge] ).
concept( 271, [ encoded] ).
concept( 272, [  tested] ).
concept( 273, [  added] ).
concept( 274, [  changed] ).
concept( 275, [  refined] ).
concept( 276, [ iterative, process] ).
concept( 277, [  leaves, gaps] ).
concept( 278, [ knowledge, base] ).
concept( 279, [  both] ).
concept( 280, [ knowledge, engineer] ).
concept( 281, [ expert] ).
concept( 282, [ overlooked, during] ).
concept( 283, [ knowledge,  acquisition, process] ).
concept( 284, [ les] ).
concept( 285, [ found,  three, situations, indicative, of, gaps] ).
concept( 286, [  knowledge, base] ).
concept( 287, [ three, situations] ).
concept( 288, [  called] ).
concept( 289, [  missing, rules] ).
concept( 290, [  unreachable,  clauses] ).
concept( 291, [  deadend, clauses] ).
concept( 292, [  described] ).
concept( 293, [  missing, rules] ).
concept( 294, [ situation] ).
concept( 295, [ values] ).
concept( 296, [ the, set, of, possible,  values] ).
concept( 297, [ called, legal, values] ).
concept( 298, [  of, an,  object's, attribute] ).
concept( 299, [ not, covered] ).
concept( 300, [  any, rule's] ).
concept( 301, [ clauses] ).
concept( 302, [ legal,  values] ).
concept( 303, [ set] ).
concept( 304, [ covered] ).
concept( 305, [ not] ).
concept( 306, [  covered, attribute] ).
concept( 307, [ prohibit] ).
concept( 308, [  system] ).
concept( 309, [ attaining] ).
concept( 310, [ conclusion] ).
concept( 311, [  cause] ).
concept( 312, [make] ).
concept( 313, [ wrong, conclusion] ).
concept( 314, [ uncovered, attribute, value] ).
concept( 315, [  encountered, during, run, time] ).
concept( 316, [  unreachable, clauses] ).
concept( 317, [ goal-driven,  production, system] ).
concept( 318, [ clause, of, a,  rule] ).
concept( 319, [ match] ).
concept( 320, [ goal, clause] ).
concept( 321, [ match] ).
concept( 322, [ clause, of, another, rule] ).
concept( 323, [ same,  rule,  set] ).
concept( 324, [ otherwise] ).
concept( 325, [ clause] ).
concept( 326, [ unreachable] ).
```

```
concept( 327, [ deadend, clauses] ).
concept( 328, [ achieve] ).
concept( 329, [ goal] ).
concept( 330, [ subgoal] ).
concept( 331, [ les] ).
concept( 332, [ required] ).
concept( 333, [ attributes, of, the, goal, clause] ).
concept( 334, [ askable] ).
concept( 335, [ user] ).
concept( 336, [provides] ).
concept( 337, [ needed, information] ).
concept( 338, [ goal, clause] ).
concept( 339, [ matched] ).
concept( 340, [ clause, of, one, of, the, rules] ).
concept( 341, [ rule, sets, applying] ).
concept( 342, [ goal] ).
concept( 343, [ of, these, conditions] ).
concept( 344, [ satisfied] ).
concept( 345, [ goal, clause] ).
concept( 346, [ not] ).
concept( 347, [ achieved] ).
concept( 348, [ 'deadend, clause'] ).
concept( 349, [ clauses, of, a, rule] ).
concept( 350, [ meet, one, of, these, two, conditions] ).
concept( 351, [ 'deadend, clauses'] ).
concept( 352, [ 3 dependency, chart] ).
concept( 353, [ circular-rule, chains, detection] ).
concept( 354, [ by-product, of, the, rule, checking] ).
concept( 355, [ check, generates] ).
concept( 356, [ dependency, chart] ).
concept( 357, [shows] ).
concept( 358, [ interactions] ).
concept( 359, [ rules] ).
concept( 360, [ rules] ).
concept( 361, [ goal, clauses] ).
concept( 362, [ example, of, a, dependency, chart] ).
concept( 363, [ small, problem] ).
concept( 364, [shown] ).
concept( 365, [ figure] ).
concept( 366, [ '] ).
concept( 367, [ ', indicates] ).
concept( 368, [ one] ).
concept( 369, [ more, clauses] ).
concept( 370, [ part, of, a, rule] ).
concept( 371, [ goal, clause] ).
concept( 372, [ gc] ).
concept( 373, [ matches, one] ).
concept( 374, [ more, clauses] ).
concept( 375, [ part, of, a, rule] ).
concept( 376, [ dependency, chart] ).
concept( 377, [ very, useful] ).
concept( 378, [ knowledge, engineer, deletes] ).
concept( 379, [ modifies] ).
concept( 380, [ adds, rules] ).
concept( 381, [ rule, base] ).
concept( 382, [ note] ).
concept( 383, [ figure] ).
concept( 384, [ '] ).
concept( 385, [ ''s, indicate] ).
concept( 386, [ dependencies] ).
concept( 387, [ original, rule, set] ).
concept( 388, [ adding] ).
concept( 389, [ clause] ).
concept( 390, [ rule] ).
concept( 391, [ '] ).
concept( 392, [ 2', dependencies, appeared] ).
```

```
concept( 393, [  note] ).
concept( 394, [  rule,  references, itself] ).
concept( 395, [ self-circular, rule] ).
concept( 396, [ addition, of, one, clause] ).
concept( 397, [ rule] ).
concept( 398, [ '] ).
concept( 399, [ 1', dependencies, appeared] ).
concept( 400, [  causes] ).
concept( 401, [ rule, set] ).
concept( 402, [ circular] ).
concept( 403, [ clause, of, rule] ).
concept( 404, [ matched] ).
concept( 405, [  clauses, of, rule] ).
concept( 406, [ rule] ).
concept( 407, [ turn,  match] ).
concept( 408, [ clause, of, rule] ).
concept( 409, [ circular,  rules] ).
concept( 410, [ avoided] ).
concept( 411, [ lead] ).
concept( 412, [ infinite, loop] ).
concept( 413, [ run, time] ).
concept( 414, [  expert, systems] ).
concept( 415, [ emycin] ).
concept( 416, [  handle,  circular, rules] ).
concept( 417, [ special, way] ).
concept( 418, [  nevertheless] ).
concept( 419, [ knowledge, engineer] ).
concept( 420, [  want] ).
concept( 421, [ know] ).
concept( 422, [ rules] ).
concept( 423, [ circular] ).
concept( 424, [  check, uses] ).
concept( 425, [ dependency, chart] ).
concept( 426, [ generate,  graphs, representing] ).
concept( 427, [ interactions] ).
concept( 428, [ rules] ).
concept( 429, [ uses] ).
concept( 430, [ cyclic, graph,  detection, algorithm] ).
concept( 431, [ detect, circular, rule,  chains] ).
concept( 432, [  implementation, of, rule, checker] ).
concept( 433, [ solving, a, problem] ).
concept( 434, [ knowledge,  engineer] ).
concept( 435, [ write, several, sets, of,  goal-driven, rules] ).
concept( 436, [ each, set, having] ).
concept( 437, [  unique, subject, category] ).
concept( 438, [ les] ).
concept( 439, [  convenient] ).
concept( 440, [ put,  rules] ).
concept( 441, [   different,  subject, categories, so] ).
concept( 442, [ system] ).
concept( 443, [  solve, different, goals] ).
concept( 444, [ rule,  sets] ).
concept( 445, [ goal] ).
concept( 446, [ solve] ).
concept( 447, [  particular, goal] ).
concept( 448, [ select,  several, goal-driven, rule, sets] ).
concept( 449, [  rules] ).
concept( 450, [ demons] ).
concept( 451, [ rule, sets] ).
concept( 452, [  generated] ).
concept( 453, [ period, of, time] ).
concept( 454, [ possible] ).
concept( 455, [ interaction] ).
concept( 456, [  cause] ).
concept( 457, [ problems] ).
concept( 458, [ each, goal] ).
```

```
concept( 459, [ necessary] ).
concept( 460, [ compare] ).
concept( 461, [ rules] ).
concept( 462, [ rule, sets, specified] ).
concept( 463, [ goal] ).
concept( 464, [ each, other] ).
concept( 465, [ clauses, of, that, goal] ).
concept( 466, [show] ).
concept( 467, [ algorithm] ).
concept( 468, [ algol-like, notation] ).
concept( 469, [ check, uses] ).
concept( 470, [ algorithm, does] ).
concept( 471, [ checking] ).
concept( 472, [ set, of, subject, categories] ).
concept( 473, [ n, rules] ).
concept( 474, [ goal] ).
concept( 475, [ g, clauses] ).
concept( 476, [ part] ).
concept( 477, [ part] ).
concept( 478, [ have, more] ).
concept( 479, [ one, clause] ).
concept( 480, [ comparison] ).
concept( 481, [ one, part] ).
concept( 482, [ another] ).
concept( 483, [ handled] ).
concept( 484, [ comparing] ).
concept( 485, [ clause, of, one, part] ).
concept( 486, [ every, clause] ).
concept( 487, [ other, part] ).
concept( 488, [ algorithms] ).
concept( 489, [ work] ).
concept( 490, [ forward-chaining, rules] ).
concept( 491, [ called] ).
concept( 492, [ rules] ).
concept( 493, [ les] ).
concept( 494, [ criterion, unreachable, clauses] ).
concept( 495, [ not, applicable] ).
concept( 496, [ forward-chaining, rules] ).
concept( 497, [ summary] ).
concept( 498, [ paper] ).
concept( 499, [ described] ).
concept( 500, [ program, called, check] ).
concept( 501, [ function] ).
concept( 502, [ detect, four, potential, problems] ).
concept( 503, [ redundant, rules] ).
concept( 504, [ conflicting, rules] ).
concept( 505, [ subsumed, rules] ).
concept( 506, [ circular, rules] ).
concept( 507, [ three, potential, gaps] ).
concept( 508, [ missing, rules] ).
concept( 509, [ unreachable, clauses] ).
concept( 510, [ deadend, clauses] ).
concept( 511, [ knowledge, base, utilizing] ).
concept( 512, [ les, framework] ).
concept( 513, [ applied] ).
concept( 514, [ consistency] ).
concept( 515, [ completeness, verification, method, of, suwa] ).
concept( 516, [ scott] ).
concept( 517, [ shortliffe] ).
concept( 518, [ generic, expert, system, les] ).
concept( 519, [ good, results] ).
concept( 520, [ extended] ).
concept( 521, [ checking] ).
concept( 522, [ include, circular, rules] ).
concept( 523, [ unreachable, clauses] ).
concept( 524, [ deadend, clauses] ).
```

```
concept( 525, [ showed] ).
concept( 526, [ general,  algorithm] ).
concept( 527, [ performs] ).
concept( 528, [ checking,  function] ).
concept( 529, [  by-product, of, the, rule, checking, processing] ).
concept( 530, [  check, generates] ).
concept( 531, [ dependenoy, chart] ).
concept( 532, [shows] ).
concept( 533, [ rules, couple] ).
concept( 534, [ interact] ).
concept( 535, [ each, other] ).
concept( 536, [ goals] ).
concept( 537, [  chart] ).
concept( 538, [ help] ).
concept( 539, [ knowledge, engineer] ).
concept( 540, [identify] ).
concept( 541, [ effects, of,  deleting] ).
concept( 542, [  adding] ).
concept( 543, [ modifying, rules] ).
concept( 544, [ experiences] ).
concept( 545, [ constructing,  different, knowledge, bases] ).
concept( 546, [ find] ).
concept( 547, [ changes] ).
concept( 548, [ additions] ).
concept( 549, [ rule, sets,  occur, during] ).
concept( 550, [ development, of, a, knowledge,  base] ).
concept( 551, [ tool] ).
concept( 552, [ check] ).
concept( 553, [  detect] ).
concept( 554, [ potential, problems] ).
concept( 555, [ gaps] ).
concept( 556, [ knowledge, base] ).
concept( 557, [ very, useful] ).
concept( 558, [ knowledge, engineer] ).
concept( 559, [ helping, him] ).
concept( 560, [  develop] ).
concept( 561, [ knowledge, base] ).
concept( 562, [ major, area, of, improvement] ).
concept( 563, [ check] ).
concept( 564, [ handling, of, certainty, factors] ).
concept( 565, [  rules] ).
concept( 566, [ les, allows] ).
concept( 567, [ rules] ).
concept( 568, [  certainty, factors, associated] ).
concept( 569, [ require] ).
concept( 570, [ definitions] ).
concept( 571, [  seven, conditions, covered] ).
concept( 572, [ paper] ).
concept( 573, [ revised] ).
/* heading(no,heading) */
heading( 0,' CHECKING AN EXPERT SYSTEMS KNOWLEDGE BASE  FOR CONSISTENCY AND C
heading( 1,' INTRODUCTION '  ).
heading( 2,'  CHECKING FOR CONSISTENCY AND COMPLETENESS '  ).
heading( 3,' 1 POTENTIAL PROBLEMS IN A KNOWLEDGE BASE '  ).
heading( 4,' 2  POTENTIAL GAPS IN A KNOWLEDGE BASE '  ).
heading( 5,' 3  DEPENDENCY CHART AND CIRCULAR-RULE CHAINS DETECTION '  ).
heading( 6,'  IMPLEMENTATION OF RULE CHECKER '  ).
heading( 7,'  SUMMARY '  ).
/* summary ( sentence no., no. of concepts identified ) */
summary( 1,4).
summary( 2,1).
summary( 3,12).
summary( 4,7).
summary( 5,5).
summary( 6,3).
summary( 7,3).
```

```
summary( 8,11).
summary( 9,5).
summary( 10,11).
summary( 11,9).
summary( 12,5).
summary( 13,4).
summary( 14,3).
summary( 15,6).
summary( 16,3).
summary( 17,3).
summary( 18,3).
summary( 19,6).
summary( 20,8).
summary( 21,7).
summary( 22,2).
summary( 23,10).
summary( 24,4).
summary( 25,4).
summary( 26,8).
summary( 27,6).
summary( 28,15).
summary( 29,4).
summary( 30,12).
summary( 31,13).
summary( 32,8).
summary( 33,10).
summary( 34,15).
summary( 35,6).
summary( 36,29).
summary( 37,2).
summary( 38,8).
summary( 39,8).
summary( 40,3).
summary( 41,6).
summary( 42,13).
summary( 43,10).
summary( 44,8).
summary( 45,3).
summary( 46,16).
summary( 47,6).
summary( 48,3).
summary( 49,2).
summary( 50,8).
summary( 51,4).
summary( 52,10).
summary( 53,6).
summary( 54,6).
summary( 55,5).
summary( 56,3).
summary( 57,4).
summary( 58,9).
summary( 59,5).
summary( 60,4).
summary( 61,6).
summary( 62,8).
summary( 63,1).
summary( 64,5).
summary( 65,8).
summary( 66,5).
summary( 67,7).
summary( 68,8).
summary( 69,4).
summary( 70,6).
summary( 71,12).
summary( 72,6).
summary( 73,3).
```

```
summary( 74,1).
summary( 75,15).
summary( 76,7).
summary( 77,5).
summary( 78,4).
summary( 79,15).
summary( 80,7).
summary( 81,11).
summary( 82,12).
/* textstats (no.of sentences,no. of headings,no.of concepts,no.of words,no. of paragraphs ) */
textstats( 82, 8, 573, 1749, 12 ).
```

```
/* text3txtcpt.pl */
/* prolog database produced by the program called txtcpt.spt*/
/* the format of the database is as follows: */
/* (note: title set to heading no = 0) */
/* textconcept(heading no, paragragh no,paragraphsentence no, sentence no, concept no, concept identit
textconcept( 1,0,0,1,1,  1 ).
textconcept( 1,0,0,1,2,  2 ).
textconcept( 1,0,0,1,3,  3 ).
textconcept( 1,0,0,1,4,  4 ).
textconcept( 2,0,0,2,1,  5 ).
textconcept( 2,1,1,3,1,  6 ).
textconcept( 2,1,1,3,2,  7 ).
textconcept( 2,1,1,3,3,  8 ).
textconcept( 2,1,1,3,4,  9 ).
textconcept( 2,1,1,3,5,  10 ).
textconcept( 2,1,1,3,6,  11 ).
textconcept( 2,1,1,3,7,  12 ).
textconcept( 2,1,1,3,8,  13 ).
textconcept( 2,1,1,3,9,  14 ).
textconcept( 2,1,1,3,10,  15 ).
textconcept( 2,1,1,3,11,  16 ).
textconcept( 2,1,1,3,12,  17 ).
textconcept( 2,1,2,4,1,  18 ).
textconcept( 2,1,2,4,2,  19 ).
textconcept( 2,1,2,4,3,  20 ).
textconcept( 2,1,2,4,4,  21 ).
textconcept( 2,1,2,4,5,  22 ).
textconcept( 2,1,2,4,6,  23 ).
textconcept( 2,1,2,4,7,  24 ).
textconcept( 2,1,3,5,1,  25 ).
textconcept( 2,1,3,5,2,  26 ).
textconcept( 2,1,3,5,3,  27 ).
textconcept( 2,1,3,5,4,  28 ).
textconcept( 2,1,3,5,5,  29 ).
textconcept( 2,1,4,6,1,  30 ).
textconcept( 2,1,4,6,2,  31 ).
textconcept( 2,1,4,6,3,  32 ).
textconcept( 2,1,5,7,1,  33 ).
textconcept( 2,1,5,7,2,  34 ).
textconcept( 2,1,5,7,3,  35 ).
textconcept( 2,1,6,8,1,  36 ).
textconcept( 2,1,6,8,2,  37 ).
textconcept( 2,1,6,8,3,  38 ).
textconcept( 2,1,6,8,4,  39 ).
textconcept( 2,1,6,8,5,  40 ).
textconcept( 2,1,6,8,6,  41 ).
textconcept( 2,1,6,8,7,  42 ).
textconcept( 2,1,6,8,8,  43 ).
textconcept( 2,1,6,8,9,  44 ).
textconcept( 2,1,6,8,10,  45 ).
textconcept( 2,1,6,8,11,  46 ).
textconcept( 2,1,7,9,1,  47 ).
textconcept( 2,1,7,9,2,  48 ).
textconcept( 2,1,7,9,3,  49 ).
textconcept( 2,1,7,9,4,  50 ).
textconcept( 2,1,7,9,5,  51 ).
textconcept( 2,1,8,10,1,  52 ).
textconcept( 2,1,8,10,2,  53 ).
textconcept( 2,1,8,10,3,  54 ).
textconcept( 2,1,8,10,4,  55 ).
textconcept( 2,1,8,10,5,  56 ).
textconcept( 2,1,8,10,6,  57 ).
textconcept( 2,1,8,10,7,  58 ).
textconcept( 2,1,8,10,8,  59 ).
textconcept( 2,1,8,10,9,  60 ).
textconcept( 2,1,8,10,10,  61 ).
```

```
textconcept( 5,1,8,45,3, 326 ).
textconcept( 5,1,9,46,1, 327 ).
textconcept( 5,1,9,46,2, 328 ).
textconcept( 5,1,9,46,3, 329 ).
textconcept( 5,1,9,46,4, 330 ).
textconcept( 5,1,9,46,5, 331 ).
textconcept( 5,1,9,46,6, 332 ).
textconcept( 5,1,9,46,7, 333 ).
textconcept( 5,1,9,46,8, 334 ).
textconcept( 5,1,9,46,9, 335 ).
textconcept( 5,1,9,46,10, 336 ).
textconcept( 5,1,9,46,11, 337 ).
textconcept( 5,1,9,46,12, 338 ).
textconcept( 5,1,9,46,13, 339 ).
textconcept( 5,1,9,46,14, 340 ).
textconcept( 5,1,9,46,15, 341 ).
textconcept( 5,1,9,46,16, 342 ).
textconcept( 5,1,10,47,1, 343 ).
textconcept( 5,1,10,47,2, 344 ).
textconcept( 5,1,10,47,3, 345 ).
textconcept( 5,1,10,47,4, 346 ).
textconcept( 5,1,10,47,5, 347 ).
textconcept( 5,1,10,47,6, 348 ).
textconcept( 5,1,11,48,1, 349 ).
textconcept( 5,1,11,48,2, 350 ).
textconcept( 5,1,11,48,3, 351 ).
textconcept( 6,0,0,49,1, 352 ).
textconcept( 6,0,0,49,2, 353 ).
textconcept( 6,1,1,50,1, 354 ).
textconcept( 6,1,1,50,2, 355 ).
textconcept( 6,1,1,50,3, 356 ).
textconcept( 6,1,1,50,4, 357 ).
textconcept( 6,1,1,50,5, 358 ).
textconcept( 6,1,1,50,6, 359 ).
textconcept( 6,1,1,50,7, 360 ).
textconcept( 6,1,1,50,8, 361 ).
textconcept( 6,1,2,51,1, 362 ).
textconcept( 6,1,2,51,2, 363 ).
textconcept( 6,1,2,51,3, 364 ).
textconcept( 6,1,2,51,4, 365 ).
textconcept( 6,1,3,52,1, 366 ).
textconcept( 6,1,3,52,2, 367 ).
textconcept( 6,1,3,52,3, 368 ).
textconcept( 6,1,3,52,4, 369 ).
textconcept( 6,1,3,52,5, 370 ).
textconcept( 6,1,3,52,6, 371 ).
textconcept( 6,1,3,52,7, 372 ).
textconcept( 6,1,3,52,8, 373 ).
textconcept( 6,1,3,52,9, 374 ).
textconcept( 6,1,3,52,10, 375 ).
textconcept( 6,1,4,53,1, 376 ).
textconcept( 6,1,4,53,2, 377 ).
textconcept( 6,1,4,53,3, 378 ).
textconcept( 6,1,4,53,4, 379 ).
textconcept( 6,1,4,53,5, 380 ).
textconcept( 6,1,4,53,6, 381 ).
textconcept( 6,2,1,54,1, 382 ).
textconcept( 6,2,1,54,2, 383 ).
textconcept( 6,2,1,54,3, 384 ).
textconcept( 6,2,1,54,4, 385 ).
textconcept( 6,2,1,54,5, 386 ).
textconcept( 6,2,1,54,6, 387 ).
textconcept( 6,2,2,55,1, 388 ).
textconcept( 6,2,2,55,2, 389 ).
textconcept( 6,2,2,55,3, 390 ).
textconcept( 6,2,2,55,4, 391 ).
textconcept( 6,2,2,55,5, 392 ).
textconcept( 6,2,3,56,1, 393 ).
textconcept( 6,2,3,56,2, 394 ).
textconcept( 6,2,3,56,3, 395 ).
textconcept( 6,2,4,57,1, 396 ).
textconcept( 6,2,4,57,2, 397 ).
textconcept( 6,2,4,57,3, 398 ).
textconcept( 6,2,4,57,4, 399 ).
textconcept( 6,2,5,58,1, 400 ).
textconcept( 6,2,5,58,2, 401 ).
textconcept( 6,2,5,58,3, 402 ).
textconcept( 6,2,5,58,4, 403 ).
textconcept( 6,2,5,58,5, 404 ).
textconcept( 6,2,5,58,6, 405 ).
textconcept( 6,2,5,58,7, 406 ).
textconcept( 6,2,5,58,8, 407 ).
textconcept( 6,2,5,58,9, 408 ).
textconcept( 6,2,6,59,1, 409 ).
textconcept( 6,2,6,59,2, 410 ).
textconcept( 6,2,6,59,3, 411 ).
textconcept( 6,2,6,59,4, 412 ).
textconcept( 6,2,6,59,5, 413 ).
textconcept( 6,2,7,60,1, 414 ).
textconcept( 6,2,7,60,2, 415 ).
textconcept( 6,2,7,60,3, 416 ).
textconcept( 6,2,7,60,4, 417 ).
textconcept( 6,2,8,61,1, 418 ).
textconcept( 6,2,8,61,2, 419 ).
textconcept( 6,2,8,61,3, 420 ).
textconcept( 6,2,8,61,4, 421 ).
textconcept( 6,2,8,61,5, 422 ).
textconcept( 6,2,8,61,6, 423 ).
textconcept( 6,2,9,62,1, 424 ).
textconcept( 6,2,9,62,2, 425 ).
textconcept( 6,2,9,62,3, 426 ).
textconcept( 6,2,9,62,4, 427 ).
textconcept( 6,2,9,62,5, 428 ).
textconcept( 6,2,9,62,6, 429 ).
textconcept( 6,2,9,62,7, 430 ).
textconcept( 6,2,9,62,8, 431 ).
textconcept( 7,0,0,63,1, 432 ).
textconcept( 7,1,1,64,1, 433 ).
textconcept( 7,1,1,64,2, 434 ).
textconcept( 7,1,1,64,3, 435 ).
textconcept( 7,1,1,64,4, 436 ).
textconcept( 7,1,1,64,5, 437 ).
textconcept( 7,1,2,65,1, 438 ).
textconcept( 7,1,2,65,2, 439 ).
textconcept( 7,1,2,65,3, 440 ).
textconcept( 7,1,2,65,4, 441 ).
textconcept( 7,1,2,65,5, 442 ).
textconcept( 7,1,2,65,6, 443 ).
textconcept( 7,1,2,65,7, 444 ).
textconcept( 7,1,2,65,8, 445 ).
textconcept( 7,1,3,66,1, 446 ).
textconcept( 7,1,3,66,2, 447 ).
textconcept( 7,1,3,66,3, 448 ).
textconcept( 7,1,3,66,4, 449 ).
textconcept( 7,1,3,66,5, 450 ).
textconcept( 7,1,4,67,1, 451 ).
textconcept( 7,1,4,67,2, 452 ).
textconcept( 7,1,4,67,3, 453 ).
textconcept( 7,1,4,67,4, 454 ).
textconcept( 7,1,4,67,5, 455 ).
textconcept( 7,1,4,67,6, 456 ).
textconcept( 7,1,4,67,7, 457 ).
```

```
textconcept( 4,1,9,31,10, 194 ).
textconcept( 4,1,9,31,11, 195 ).
textconcept( 4,1,9,31,12, 196 ).
textconcept( 4,1,9,31,13, 197 ).
textconcept( 4,1,10,32,1, 198 ).
textconcept( 4,1,10,32,2, 199 ).
textconcept( 4,1,10,32,3, 200 ).
textconcept( 4,1,10,32,4, 201 ).
textconcept( 4,1,10,32,5, 202 ).
textconcept( 4,1,10,32,6, 203 ).
textconcept( 4,1,10,32,7, 204 ).
textconcept( 4,1,10,32,8, 205 ).
textconcept( 4,1,11,33,1, 206 ).
textconcept( 4,1,11,33,2, 207 ).
textconcept( 4,1,11,33,3, 208 ).
textconcept( 4,1,11,33,4, 209 ).
textconcept( 4,1,11,33,5, 210 ).
textconcept( 4,1,11,33,6, 211 ).
textconcept( 4,1,11,33,7, 212 ).
textconcept( 4,1,11,33,8, 213 ).
textconcept( 4,1,11,33,9, 214 ).
textconcept( 4,1,11,33,10, 215 ).
textconcept( 4,1,12,34,1, 216 ).
textconcept( 4,1,12,34,2, 217 ).
textconcept( 4,1,12,34,3, 218 ).
textconcept( 4,1,12,34,4, 219 ).
textconcept( 4,1,12,34,5, 220 ).
textconcept( 4,1,12,34,6, 221 ).
textconcept( 4,1,12,34,7, 222 ).
textconcept( 4,1,12,34,8, 223 ).
textconcept( 4,1,12,34,9, 224 ).
textconcept( 4,1,12,34,10, 225 ).
textconcept( 4,1,12,34,11, 226 ).
textconcept( 4,1,12,34,12, 227 ).
textconcept( 4,1,12,34,13, 228 ).
textconcept( 4,1,12,34,14, 229 ).
textconcept( 4,1,12,34,15, 230 ).
textconcept( 4,1,13,35,1, 231 ).
textconcept( 4,1,13,35,2, 232 ).
textconcept( 4,1,13,35,3, 233 ).
textconcept( 4,1,13,35,4, 234 ).
textconcept( 4,1,13,35,5, 235 ).
textconcept( 4,1,13,35,6, 236 ).
textconcept( 4,1,14,36,1, 237 ).
textconcept( 4,1,14,36,2, 238 ).
textconcept( 4,1,14,36,3, 239 ).
textconcept( 4,1,14,36,4, 240 ).
textconcept( 4,1,14,36,5, 241 ).
textconcept( 4,1,14,36,6, 242 ).
textconcept( 4,1,14,36,7, 243 ).
textconcept( 4,1,14,36,8, 244 ).
textconcept( 4,1,14,36,9, 245 ).
textconcept( 4,1,14,36,10, 246 ).
textconcept( 4,1,14,36,11, 247 ).
textconcept( 4,1,14,36,12, 248 ).
textconcept( 4,1,14,36,13, 249 ).
textconcept( 4,1,14,36,14, 250 ).
textconcept( 4,1,14,36,15, 251 ).
textconcept( 4,1,14,36,16, 252 ).
textconcept( 4,1,14,36,17, 253 ).
textconcept( 4,1,14,36,18, 254 ).
textconcept( 4,1,14,36,19, 255 ).
textconcept( 4,1,14,36,20, 256 ).
textconcept( 4,1,14,36,21, 257 ).
textconcept( 4,1,14,36,22, 258 ).
textconcept( 4,1,14,36,23, 259 ).
textconcept( 4,1,14,36,24, 260 ).
textconcept( 4,1,14,36,25, 261 ).
textconcept( 4,1,14,36,26, 262 ).
textconcept( 4,1,14,36,27, 263 ).
textconcept( 4,1,14,36,28, 264 ).
textconcept( 4,1,14,36,29, 265 ).
textconcept( 5,0,0,37,1, 266 ).
textconcept( 5,0,0,37,2, 267 ).
textconcept( 5,1,1,38,1, 268 ).
textconcept( 5,1,1,38,2, 269 ).
textconcept( 5,1,1,38,3, 270 ).
textconcept( 5,1,1,38,4, 271 ).
textconcept( 5,1,1,38,5, 272 ).
textconcept( 5,1,1,38,6, 273 ).
textconcept( 5,1,1,38,7, 274 ).
textconcept( 5,1,1,38,8, 275 ).
textconcept( 5,1,2,39,1, 276 ).
textconcept( 5,1,2,39,2, 277 ).
textconcept( 5,1,2,39,3, 278 ).
textconcept( 5,1,2,39,4, 279 ).
textconcept( 5,1,2,39,5, 280 ).
textconcept( 5,1,2,39,6, 281 ).
textconcept( 5,1,2,39,7, 282 ).
textconcept( 5,1,2,39,8, 283 ).
textconcept( 5,1,3,40,1, 284 ).
textconcept( 5,1,3,40,2, 285 ).
textconcept( 5,1,3,40,3, 286 ).
textconcept( 5,1,4,41,1, 287 ).
textconcept( 5,1,4,41,2, 288 ).
textconcept( 5,1,4,41,3, 289 ).
textconcept( 5,1,4,41,4, 290 ).
textconcept( 5,1,4,41,5, 291 ).
textconcept( 5,1,4,41,6, 292 ).
textconcept( 5,1,5,42,1, 293 ).
textconcept( 5,1,5,42,2, 294 ).
textconcept( 5,1,5,42,3, 295 ).
textconcept( 5,1,5,42,4, 296 ).
textconcept( 5,1,5,42,5, 297 ).
textconcept( 5,1,5,42,6, 298 ).
textconcept( 5,1,5,42,7, 299 ).
textconcept( 5,1,5,42,8, 300 ).
textconcept( 5,1,5,42,9, 301 ).
textconcept( 5,1,5,42,10, 302 ).
textconcept( 5,1,5,42,11, 303 ).
textconcept( 5,1,5,42,12, 304 ).
textconcept( 5,1,5,42,13, 305 ).
textconcept( 5,1,6,43,1, 306 ).
textconcept( 5,1,6,43,2, 307 ).
textconcept( 5,1,6,43,3, 308 ).
textconcept( 5,1,6,43,4, 309 ).
textconcept( 5,1,6,43,5, 310 ).
textconcept( 5,1,6,43,6, 311 ).
textconcept( 5,1,6,43,7, 312 ).
textconcept( 5,1,6,43,8, 313 ).
textconcept( 5,1,6,43,9, 314 ).
textconcept( 5,1,6,43,10, 315 ).
textconcept( 5,1,7,44,1, 316 ).
textconcept( 5,1,7,44,2, 317 ).
textconcept( 5,1,7,44,3, 318 ).
textconcept( 5,1,7,44,4, 319 ).
textconcept( 5,1,7,44,5, 320 ).
textconcept( 5,1,7,44,6, 321 ).
textconcept( 5,1,7,44,7, 322 ).
textconcept( 5,1,7,44,8, 323 ).
textconcept( 5,1,8,45,1, 324 ).
textconcept( 5,1,8,45,2, 325 ).
```

```
textconcept( 2,1,8,10,11, 62 ).
textconcept( 2,2,1,11,1, 63 ).
textconcept( 2,2,1,11,2, 64 ).
textconcept( 2,2,1,11,3, 65 ).
textconcept( 2,2,1,11,4, 66 ).
textconcept( 2,2,1,11,5, 67 ).
textconcept( 2,2,1,11,6, 68 ).
textconcept( 2,2,1,11,7, 69 ).
textconcept( 2,2,1,11,8, 70 ).
textconcept( 2,2,1,11,9, 71 ).
textconcept( 2,2,2,12,1, 72 ).
textconcept( 2,2,2,12,2, 73 ).
textconcept( 2,2,2,12,3, 74 ).
textconcept( 2,2,2,12,4, 75 ).
textconcept( 2,2,2,12,5, 76 ).
textconcept( 2,2,3,13,1, 77 ).
textconcept( 2,2,3,13,2, 78 ).
textconcept( 2,2,3,13,3, 79 ).
textconcept( 2,2,3,13,4, 80 ).
textconcept( 2,2,4,14,1, 81 ).
textconcept( 2,2,4,14,2, 82 ).
textconcept( 2,2,4,14,3, 83 ).
textconcept( 2,2,5,15,1, 84 ).
textconcept( 2,2,5,15,2, 85 ).
textconcept( 2,2,5,15,3, 86 ).
textconcept( 2,2,5,15,4, 87 ).
textconcept( 2,2,5,15,5, 88 ).
textconcept( 2,2,5,15,6, 89 ).
textconcept( 2,2,6,16,1, 90 ).
textconcept( 2,2,6,16,2, 91 ).
textconcept( 2,2,6,16,3, 92 ).
textconcept( 2,2,7,17,1, 93 ).
textconcept( 2,2,7,17,2, 94 ).
textconcept( 2,2,7,17,3, 95 ).
textconcept( 3,0,0,18,1, 96 ).
textconcept( 3,0,0,18,2, 97 ).
textconcept( 3,0,0,18,3, 98 ).
textconcept( 3,1,1,19,1, 99 ).
textconcept( 3,1,1,19,2, 100 ).
textconcept( 3,1,1,19,3, 101 ).
textconcept( 3,1,1,19,4, 102 ).
textconcept( 3,1,1,19,5, 103 ).
textconcept( 3,1,1,19,6, 104 ).
textconcept( 3,1,2,20,1, 105 ).
textconcept( 3,1,2,20,2, 106 ).
textconcept( 3,1,2,20,3, 107 ).
textconcept( 3,1,2,20,4, 108 ).
textconcept( 3,1,2,20,5, 109 ).
textconcept( 3,1,2,20,6, 110 ).
textconcept( 3,1,2,20,7, 111 ).
textconcept( 3,1,2,20,8, 112 ).
textconcept( 3,1,3,21,1, 113 ).
textconcept( 3,1,3,21,2, 114 ).
textconcept( 3,1,3,21,3, 115 ).
textconcept( 3,1,3,21,4, 116 ).
textconcept( 3,1,3,21,5, 117 ).
textconcept( 3,1,3,21,6, 118 ).
textconcept( 3,1,3,21,7, 119 ).
textconcept( 4,0,0,22,1, 120 ).
textconcept( 4,0,0,22,2, 121 ).
textconcept( 4,1,1,23,1, 122 ).
textconcept( 4,1,1,23,2, 123 ).
textconcept( 4,1,1,23,3, 124 ).
textconcept( 4,1,1,23,4, 125 ).
textconcept( 4,1,1,23,5, 126 ).
textconcept( 4,1,1,23,6, 127 ).
textconcept( 4,1,1,23,7, 128 ).
textconcept( 4,1,1,23,8, 129 ).
textconcept( 4,1,1,23,9, 130 ).
textconcept( 4,1,1,23,10, 131 ).
textconcept( 4,1,2,24,1, 132 ).
textconcept( 4,1,2,24,2, 133 ).
textconcept( 4,1,2,24,3, 134 ).
textconcept( 4,1,2,24,4, 135 ).
textconcept( 4,1,3,25,1, 136 ).
textconcept( 4,1,3,25,2, 137 ).
textconcept( 4,1,3,25,3, 138 ).
textconcept( 4,1,3,25,4, 139 ).
textconcept( 4,1,4,26,1, 140 ).
textconcept( 4,1,4,26,2, 141 ).
textconcept( 4,1,4,26,3, 142 ).
textconcept( 4,1,4,26,4, 143 ).
textconcept( 4,1,4,26,5, 144 ).
textconcept( 4,1,4,26,6, 145 ).
textconcept( 4,1,4,26,7, 146 ).
textconcept( 4,1,4,26,8, 147 ).
textconcept( 4,1,5,27,1, 148 ).
textconcept( 4,1,5,27,2, 149 ).
textconcept( 4,1,5,27,3, 150 ).
textconcept( 4,1,5,27,4, 151 ).
textconcept( 4,1,5,27,5, 152 ).
textconcept( 4,1,5,27,6, 153 ).
textconcept( 4,1,6,28,1, 154 ).
textconcept( 4,1,6,28,2, 155 ).
textconcept( 4,1,6,28,3, 156 ).
textconcept( 4,1,6,28,4, 157 ).
textconcept( 4,1,6,28,5, 158 ).
textconcept( 4,1,6,28,6, 159 ).
textconcept( 4,1,6,28,7, 160 ).
textconcept( 4,1,6,28,8, 161 ).
textconcept( 4,1,6,28,9, 162 ).
textconcept( 4,1,6,28,10, 163 ).
textconcept( 4,1,6,28,11, 164 ).
textconcept( 4,1,6,28,12, 165 ).
textconcept( 4,1,6,28,13, 166 ).
textconcept( 4,1,6,28,14, 167 ).
textconcept( 4,1,6,28,15, 168 ).
textconcept( 4,1,7,29,1, 169 ).
textconcept( 4,1,7,29,2, 170 ).
textconcept( 4,1,7,29,3, 171 ).
textconcept( 4,1,7,29,4, 172 ).
textconcept( 4,1,8,30,1, 173 ).
textconcept( 4,1,8,30,2, 174 ).
textconcept( 4,1,8,30,3, 175 ).
textconcept( 4,1,8,30,4, 176 ).
textconcept( 4,1,8,30,5, 177 ).
textconcept( 4,1,8,30,6, 178 ).
textconcept( 4,1,8,30,7, 179 ).
textconcept( 4,1,8,30,8, 180 ).
textconcept( 4,1,8,30,9, 181 ).
textconcept( 4,1,8,30,10, 182 ).
textconcept( 4,1,8,30,11, 183 ).
textconcept( 4,1,8,30,12, 184 ).
textconcept( 4,1,9,31,1, 185 ).
textconcept( 4,1,9,31,2, 186 ).
textconcept( 4,1,9,31,3, 187 ).
textconcept( 4,1,9,31,4, 188 ).
textconcept( 4,1,9,31,5, 189 ).
textconcept( 4,1,9,31,6, 190 ).
textconcept( 4,1,9,31,7, 191 ).
textconcept( 4,1,9,31,8, 192 ).
textconcept( 4,1,9,31,9, 193 ).
```

```
textconcept( 7,1,5,68,1,  458 ).
textconcept( 7,1,5,68,2,  459 ).
textconcept( 7,1,5,68,3,  460 ).
textconcept( 7,1,5,68,4,  461 ).
textconcept( 7,1,5,68,5,  462 ).
textconcept( 7,1,5,68,6,  463 ).
textconcept( 7,1,5,68,7,  464 ).
textconcept( 7,1,5,68,8,  465 ).
textconcept( 7,1,6,69,1,  466 ).
textconcept( 7,1,6,69,2,  467 ).
textconcept( 7,1,6,69,3,  468 ).
textconcept( 7,1,6,69,4,  469 ).
textconcept( 7,1,7,70,1,  470 ).
textconcept( 7,1,7,70,2,  471 ).
textconcept( 7,1,7,70,3,  472 ).
textconcept( 7,1,7,70,4,  473 ).
textconcept( 7,1,7,70,5,  474 ).
textconcept( 7,1,7,70,6,  475 ).
textconcept( 7,2,1,71,1,  476 ).
textconcept( 7,2,1,71,2,  477 ).
textconcept( 7,2,1,71,3,  478 ).
textconcept( 7,2,1,71,4,  479 ).
textconcept( 7,2,1,71,5,  480 ).
textconcept( 7,2,1,71,6,  481 ).
textconcept( 7,2,1,71,7,  482 ).
textconcept( 7,2,1,71,8,  483 ).
textconcept( 7,2,1,71,9,  484 ).
textconcept( 7,2,1,71,10,  485 ).
textconcept( 7,2,1,71,11,  486 ).
textconcept( 7,2,1,71,12,  487 ).
textconcept( 7,2,2,72,1,  488 ).
textconcept( 7,2,2,72,2,  489 ).
textconcept( 7,2,2,72,3,  490 ).
textconcept( 7,2,2,72,4,  491 ).
textconcept( 7,2,2,72,5,  492 ).
textconcept( 7,2,2,72,6,  493 ).
textconcept( 7,2,3,73,1,  494 ).
textconcept( 7,2,3,73,2,  495 ).
textconcept( 7,2,3,73,3,  496 ).
textconcept( 8,0,0,74,1,  497 ).
textconcept( 8,1,1,75,1,  498 ).
textconcept( 8,1,1,75,2,  499 ).
textconcept( 8,1,1,75,3,  500 ).
textconcept( 8,1,1,75,4,  501 ).
textconcept( 8,1,1,75,5,  502 ).
textconcept( 8,1,1,75,6,  503 ).
textconcept( 8,1,1,75,7,  504 ).
textconcept( 8,1,1,75,8,  505 ).
textconcept( 8,1,1,75,9,  506 ).
textconcept( 8,1,1,75,10,  507 ).
textconcept( 8,1,1,75,11,  508 ).
textconcept( 8,1,1,75,12,  509 ).
textconcept( 8,1,1,75,13,  510 ).
textconcept( 8,1,1,75,14,  511 ).
textconcept( 8,1,1,75,15,  512 ).
textconcept( 8,1,2,76,1,  513 ).
textconcept( 8,1,2,76,2,  514 ).
textconcept( 8,1,2,76,3,  515 ).
textconcept( 8,1,2,76,4,  516 ).
textconcept( 8,1,2,76,5,  517 ).
textconcept( 8,1,2,76,6,  518 ).
textconcept( 8,1,2,76,7,  519 ).
textconcept( 8,1,3,77,1,  520 ).
textconcept( 8,1,3,77,2,  521 ).
textconcept( 8,1,3,77,3,  522 ).
textconcept( 8,1,3,77,4,  523 ).
textconcept( 8,1,3,77,5,  524 ).
textconcept( 8,1,4,78,1,  525 ).
textconcept( 8,1,4,78,2,  526 ).
textconcept( 8,1,4,78,3,  527 ).
textconcept( 8,1,4,78,4,  528 ).
textconcept( 8,1,5,79,1,  529 ).
textconcept( 8,1,5,79,2,  530 ).
textconcept( 8,1,5,79,3,  531 ).
textconcept( 8,1,5,79,4,  532 ).
textconcept( 8,1,5,79,5,  533 ).
textconcept( 8,1,5,79,6,  534 ).
textconcept( 8,1,5,79,7,  535 ).
textconcept( 8,1,5,79,8,  536 ).
textconcept( 8,1,5,79,9,  537 ).
textconcept( 8,1,5,79,10,  538 ).
textconcept( 8,1,5,79,11,  539 ).
textconcept( 8,1,5,79,12,  540 ).
textconcept( 8,1,5,79,13,  541 ).
textconcept( 8,1,5,79,14,  542 ).
textconcept( 8,1,5,79,15,  543 ).
textconcept( 8,2,1,80,1,  544 ).
textconcept( 8,2,1,80,2,  545 ).
textconcept( 8,2,1,80,3,  546 ).
textconcept( 8,2,1,80,4,  547 ).
textconcept( 8,2,1,80,5,  548 ).
textconcept( 8,2,1,80,6,  549 ).
textconcept( 8,2,1,80,7,  550 ).
textconcept( 8,2,2,81,1,  551 ).
textconcept( 8,2,2,81,2,  552 ).
textconcept( 8,2,2,81,3,  553 ).
textconcept( 8,2,2,81,4,  554 ).
textconcept( 8,2,2,81,5,  555 ).
textconcept( 8,2,2,81,6,  556 ).
textconcept( 8,2,2,81,7,  557 ).
textconcept( 8,2,2,81,8,  558 ).
textconcept( 8,2,2,81,9,  559 ).
textconcept( 8,2,2,81,10,  560 ).
textconcept( 8,2,2,81,11,  561 ).
textconcept( 8,3,1,82,1,  562 ).
textconcept( 8,3,1,82,2,  563 ).
textconcept( 8,3,1,82,3,  564 ).
textconcept( 8,3,1,82,4,  565 ).
textconcept( 8,3,1,82,5,  566 ).
textconcept( 8,3,1,82,6,  567 ).
textconcept( 8,3,1,82,7,  568 ).
textconcept( 8,3,1,82,8,  569 ).
textconcept( 8,3,1,82,9,  570 ).
textconcept( 8,3,1,82,10,  571 ).
textconcept( 8,3,1,82,11,  572 ).
```

```
/* text4status.pl */

status(1,title ).
status(2,heading ).
status(21,heading ).
status(49,heading ).
status(86,heading ).
```

```
/* text4func.pl */

function(1,title).
function(2, introduction ).
function(3, background ).
function(4, background ).
function(5, background ).
function(6, background ).
function(7, background ).
function(8, background ).
function(9, background ).
function(10, background ).
function(11, background ).
function(12, background ).
function(13, background ).
function(14, background ).
function(15, background ).
function(16, background ).
function(17, background ).
function(18, background ).
function(19, background ).
function(20, background ).
function(21, background ).
function(22, background ).
function(23, background ).
function(24, background ).
function(25, background ).
function(26, background ).
function(27, background ).
function(28, background ).
function(29, background ).
function(30, background ).
function(31, background ).
function(32, background ).
function(33, background ).
function(34, background ).
function(35, background ).
function(36, background ).
function(37, background ).
function(38, background ).
function(39, background ).
function(40, background ).
function(41, background ).
function(42, background ).
function(43, background ).
function(44, background ).
function(45, background ).
function(46, background ).
function(47, background ).
function(48, background ).
function(49, background ).
function(50, background ).
function(51, background ).
function(52, background ).
function(53, background ).
function(54, background ).
function(55, background ).
function(56, background ).
function(57, background ).
function(58, background ).
function(59, background ).
function(60, background ).
function(61, background ).
function(62, background ).
function(63, background ).
function(64, background ).
function(65, background ).
function(66, background ).
function(67, background ).
function(68, background ).
function(69, background ).
function(70, background ).
function(71, background ).
function(72, background ).
function(73, background ).
function(74, background ).
function(75, background ).
function(76, background ).
function(77, background ).
function(78, background ).
function(79, background ).
function(80, background ).
function(81, background ).
function(82, background ).
function(83, background ).
function(84, background ).
function(85, background ).
function(86, conclusion ).
function(87, background ).
function(88, background ).
function(89, background ).
function(90, background ).
function(91, background ).
function(92, background ).
function(93, background ).
function(94, background ).
function(95, background ).
function(96, background ).
function(97, background ).
function(98, background ).
function(99, background ).
function(100, background ).
function(101, background ).
function(102, background ).
function(103, background ).
function(104, background ).
function(105, background ).
function(106, background ).
function(107, background ).
function(108, background ).
```

```
/* text4txtcpt.pl */
/* prolog database produced by the program called txtcpt.spt*/
/* the format of the database is as follows: */
/* (note: title set to heading no = 0) */
/* textconcept(heading no, paragragh no,paragraphsentence no, sentence no, concept no, concept identit
textconcept( 1,0,0,1,1,  1 ).
textconcept( 1,0,0,1,2,  2 ).
textconcept( 2,0,0,2,1,  3 ).
textconcept( 2,1,1,3,1,  4 ).
textconcept( 2,1,1,3,2,  5 ).
textconcept( 2,1,1,3,3,  6 ).
textconcept( 2,1,1,3,4,  7 ).
textconcept( 2,1,1,3,5,  8 ).
textconcept( 2,1,2,4,1,  9 ).
textconcept( 2,1,2,4,2,  10 ).
textconcept( 2,1,2,4,3,  11 ).
textconcept( 2,1,2,4,4,  12 ).
textconcept( 2,1,2,4,5,  13 ).
textconcept( 2,1,2,4,6,  14 ).
textconcept( 2,1,2,4,7,  15 ).
textconcept( 2,1,2,4,8,  16 ).
textconcept( 2,1,2,4,9,  17 ).
textconcept( 2,1,2,4,10,  18 ).
textconcept( 2,1,2,4,11,  19 ).
textconcept( 2,1,2,4,12,  20 ).
textconcept( 2,1,3,5,1,  21 ).
textconcept( 2,1,3,5,2,  22 ).
textconcept( 2,1,3,5,3,  23 ).
textconcept( 2,1,3,5,4,  24 ).
textconcept( 2,1,4,6,1,  25 ).
textconcept( 2,1,4,6,2,  26 ).
textconcept( 2,1,4,6,3,  27 ).
textconcept( 2,1,4,6,4,  28 ).
textconcept( 2,1,4,6,5,  29 ).
textconcept( 2,1,4,6,6,  30 ).
textconcept( 2,1,4,6,7,  31 ).
textconcept( 2,1,4,6,8,  32 ).
textconcept( 2,1,4,6,9,  33 ).
textconcept( 2,1,4,6,10,  34 ).
textconcept( 2,1,4,6,11,  35 ).
textconcept( 2,1,4,6,12,  36 ).
textconcept( 2,2,1,7,1,  37 ).
textconcept( 2,2,1,7,2,  38 ).
textconcept( 2,2,1,7,3,  39 ).
textconcept( 2,2,1,7,4,  40 ).
textconcept( 2,2,1,7,5,  41 ).
textconcept( 2,2,1,7,6,  42 ).
textconcept( 2,2,1,7,7,  43 ).
textconcept( 2,2,2,8,1,  44 ).
textconcept( 2,2,2,8,2,  45 ).
textconcept( 2,2,2,8,3,  46 ).
textconcept( 2,2,2,8,4,  47 ).
textconcept( 2,2,2,8,5,  48 ).
textconcept( 2,2,2,8,6,  49 ).
textconcept( 2,2,2,8,7,  50 ).
textconcept( 2,2,2,8,8,  51 ).
textconcept( 2,2,2,8,9,  52 ).
textconcept( 2,2,2,8,10,  53 ).
textconcept( 2,2,2,8,11,  54 ).
textconcept( 2,2,2,8,12,  55 ).
textconcept( 2,2,2,8,13,  56 ).
textconcept( 2,2,2,8,14,  57 ).
textconcept( 2,2,3,9,1,  58 ).
textconcept( 2,2,3,9,2,  59 ).
textconcept( 2,2,3,9,3,  60 ).
```

```
textconcept( 2,2,3,9,4,  61 ).
textconcept( 2,2,3,9,5,  62 ).
textconcept( 2,2,3,9,6,  63 ).
textconcept( 2,3,1,10,1,  64 ).
textconcept( 2,3,1,10,2,  65 ).
textconcept( 2,3,1,10,3,  66 ).
textconcept( 2,3,1,10,4,  67 ).
textconcept( 2,3,1,10,5,  68 ).
textconcept( 2,3,1,10,6,  69 ).
textconcept( 2,3,1,10,7,  70 ).
textconcept( 2,3,1,10,8,  71 ).
textconcept( 2,3,1,10,9,  72 ).
textconcept( 2,3,1,10,10,  73 ).
textconcept( 2,3,1,10,11,  74 ).
textconcept( 2,3,1,10,12,  75 ).
textconcept( 2,3,1,10,13,  76 ).
textconcept( 2,4,1,11,1,  77 ).
textconcept( 2,4,1,11,2,  78 ).
textconcept( 2,4,1,11,3,  79 ).
textconcept( 2,4,1,11,4,  80 ).
textconcept( 2,4,2,12,1,  81 ).
textconcept( 2,4,2,12,2,  82 ).
textconcept( 2,4,2,12,3,  83 ).
textconcept( 2,4,3,13,1,  84 ).
textconcept( 2,4,3,13,2,  85 ).
textconcept( 2,4,3,13,3,  86 ).
textconcept( 2,4,3,13,4,  87 ).
textconcept( 2,4,3,13,5,  88 ).
textconcept( 2,4,4,14,1,  89 ).
textconcept( 2,4,4,14,2,  90 ).
textconcept( 2,4,4,14,3,  91 ).
textconcept( 2,4,4,14,4,  92 ).
textconcept( 2,4,4,14,5,  93 ).
textconcept( 2,4,4,14,6,  94 ).
textconcept( 2,4,4,14,7,  95 ).
textconcept( 2,4,4,14,8,  96 ).
textconcept( 2,4,4,14,9,  97 ).
textconcept( 2,4,4,14,10,  98 ).
textconcept( 2,4,4,14,11,  99 ).
textconcept( 2,4,5,15,1,  100 ).
textconcept( 2,4,5,15,2,  101 ).
textconcept( 2,4,5,15,3,  102 ).
textconcept( 2,4,5,15,4,  103 ).
textconcept( 2,4,5,15,5,  104 ).
textconcept( 2,4,5,15,6,  105 ).
textconcept( 2,4,5,15,7,  106 ).
textconcept( 2,5,1,16,1,  107 ).
textconcept( 2,5,1,16,2,  108 ).
textconcept( 2,5,1,16,3,  109 ).
textconcept( 2,5,1,16,4,  110 ).
textconcept( 2,5,1,16,5,  111 ).
textconcept( 2,5,1,16,6,  112 ).
textconcept( 2,5,1,16,7,  113 ).
textconcept( 2,5,1,16,8,  114 ).
textconcept( 2,5,1,16,9,  115 ).
textconcept( 2,5,1,16,10,  116 ).
textconcept( 2,5,1,16,11,  117 ).
textconcept( 2,5,1,16,12,  118 ).
textconcept( 2,5,2,17,1,  119 ).
textconcept( 2,5,2,17,2,  120 ).
textconcept( 2,5,2,17,3,  121 ).
textconcept( 2,5,2,17,4,  122 ).
textconcept( 2,5,2,17,5,  123 ).
textconcept( 2,5,2,17,6,  124 ).
textconcept( 2,5,2,17,7,  125 ).
textconcept( 2,5,2,17,8,  126 ).
textconcept( 2,5,2,17,9,  127 ).
textconcept( 2,5,2,17,10,  128 ).
textconcept( 2,5,2,17,11,  129 ).
textconcept( 2,5,2,17,12,  130 ).
textconcept( 2,5,2,17,13,  131 ).
textconcept( 2,5,3,18,1,  132 ).
textconcept( 2,5,3,18,2,  133 ).
textconcept( 2,5,3,18,3,  134 ).
textconcept( 2,5,4,19,1,  135 ).
textconcept( 2,5,4,19,2,  136 ).
textconcept( 2,5,4,19,3,  137 ).
textconcept( 2,5,5,20,1,  138 ).
textconcept( 2,5,5,20,2,  139 ).
textconcept( 2,5,5,20,3,  140 ).
textconcept( 2,5,5,20,4,  141 ).
textconcept( 2,5,5,20,5,  142 ).
textconcept( 3,0,0,21,1,  143 ).
textconcept( 3,1,1,22,1,  144 ).
textconcept( 3,1,1,22,2,  145 ).
textconcept( 3,1,1,22,3,  146 ).
textconcept( 3,1,1,22,4,  147 ).
textconcept( 3,1,1,22,5,  148 ).
textconcept( 3,1,1,22,6,  149 ).
textconcept( 3,1,1,22,7,  150 ).
textconcept( 3,1,1,22,8,  151 ).
textconcept( 3,1,1,22,9,  152 ).
textconcept( 3,1,1,22,10,  153 ).
textconcept( 3,1,1,22,11,  154 ).
textconcept( 3,1,1,22,12,  155 ).
textconcept( 3,1,1,22,13,  156 ).
textconcept( 3,1,1,22,14,  157 ).
textconcept( 3,1,1,22,15,  158 ).
textconcept( 3,1,1,22,16,  159 ).
textconcept( 3,1,1,22,17,  160 ).
textconcept( 3,1,1,22,18,  161 ).
textconcept( 3,1,2,23,1,  162 ).
textconcept( 3,1,2,23,2,  163 ).
textconcept( 3,1,2,23,3,  164 ).
textconcept( 3,1,2,23,4,  165 ).
textconcept( 3,1,2,23,5,  166 ).
textconcept( 3,1,2,23,6,  167 ).
textconcept( 3,1,3,24,1,  168 ).
textconcept( 3,1,3,24,2,  169 ).
textconcept( 3,1,3,24,3,  170 ).
textconcept( 3,1,4,25,1,  171 ).
textconcept( 3,1,4,25,2,  172 ).
textconcept( 3,1,4,25,3,  173 ).
textconcept( 3,1,4,25,4,  174 ).
textconcept( 3,1,4,25,5,  175 ).
textconcept( 3,1,4,25,6,  176 ).
textconcept( 3,1,4,25,7,  177 ).
textconcept( 3,1,4,25,8,  178 ).
textconcept( 3,1,4,25,9,  179 ).
textconcept( 3,1,4,25,10,  180 ).
textconcept( 3,1,4,25,11,  181 ).
textconcept( 3,1,4,25,12,  182 ).
textconcept( 3,1,5,26,1,  183 ).
textconcept( 3,1,5,26,2,  184 ).
textconcept( 3,1,5,26,3,  185 ).
textconcept( 3,1,5,26,4,  186 ).
textconcept( 3,1,5,26,5,  187 ).
textconcept( 3,1,5,26,6,  188 ).
textconcept( 3,1,5,26,7,  189 ).
textconcept( 3,2,1,27,1,  190 ).
textconcept( 3,2,1,27,2,  191 ).
textconcept( 3,2,1,27,3,  192 ).
```

```
textconcept( 3,2,1,27,4,  193 ).
textconcept( 3,2,1,27,5,  194 ).
textconcept( 3,2,1,27,6,  195 ).
textconcept( 3,2,2,28,1,  196 ).
textconcept( 3,2,2,28,2,  197 ).
textconcept( 3,2,2,28,3,  198 ).
textconcept( 3,2,2,28,4,  199 ).
textconcept( 3,2,2,28,5,  200 ).
textconcept( 3,2,2,28,6,  201 ).
textconcept( 3,2,3,29,1,  202 ).
textconcept( 3,2,3,29,2,  203 ).
textconcept( 3,2,3,29,3,  204 ).
textconcept( 3,2,3,29,4,  205 ).
textconcept( 3,2,3,29,5,  206 ).
textconcept( 3,2,3,29,6,  207 ).
textconcept( 3,2,4,30,1,  208 ).
textconcept( 3,2,4,30,2,  209 ).
textconcept( 3,2,4,30,3,  210 ).
textconcept( 3,2,4,30,4,  211 ).
textconcept( 3,2,4,30,5,  212 ).
textconcept( 3,2,5,31,1,  213 ).
textconcept( 3,2,5,31,2,  214 ).
textconcept( 3,2,6,32,1,  215 ).
textconcept( 3,2,6,32,2,  216 ).
textconcept( 3,2,6,32,3,  217 ).
textconcept( 3,2,6,32,4,  218 ).
textconcept( 3,3,1,33,1,  219 ).
textconcept( 3,3,1,33,2,  220 ).
textconcept( 3,3,1,33,3,  221 ).
textconcept( 3,3,1,33,4,  222 ).
textconcept( 3,3,2,34,1,  223 ).
textconcept( 3,3,2,34,2,  224 ).
textconcept( 3,3,2,34,3,  225 ).
textconcept( 3,3,2,34,4,  226 ).
textconcept( 3,3,2,34,5,  227 ).
textconcept( 3,3,2,34,6,  228 ).
textconcept( 3,3,2,34,7,  229 ).
textconcept( 3,3,2,34,8,  230 ).
textconcept( 3,3,2,34,9,  231 ).
textconcept( 3,3,2,34,10,  232 ).
textconcept( 3,3,2,34,11,  233 ).
textconcept( 3,3,2,34,12,  234 ).
textconcept( 3,4,1,35,1,  235 ).
textconcept( 3,4,1,35,2,  236 ).
textconcept( 3,4,1,35,3,  237 ).
textconcept( 3,4,2,36,1,  238 ).
textconcept( 3,4,2,36,2,  239 ).
textconcept( 3,4,2,36,3,  240 ).
textconcept( 3,4,3,37,1,  241 ).
textconcept( 3,4,3,37,2,  242 ).
textconcept( 3,4,3,37,3,  243 ).
textconcept( 3,4,3,37,4,  244 ).
textconcept( 3,4,4,38,1,  245 ).
textconcept( 3,4,4,38,2,  246 ).
textconcept( 3,4,4,38,3,  247 ).
textconcept( 3,4,4,38,4,  248 ).
textconcept( 3,4,4,38,5,  249 ).
textconcept( 3,4,4,38,6,  250 ).
textconcept( 3,4,4,38,7,  251 ).
textconcept( 3,4,5,39,1,  252 ).
textconcept( 3,4,5,39,2,  253 ).
textconcept( 3,4,5,39,3,  254 ).
textconcept( 3,4,5,39,4,  255 ).
textconcept( 3,4,5,39,5,  256 ).
textconcept( 3,5,1,40,1,  257 ).
textconcept( 3,5,1,40,2,  258 ).
textconcept( 3,5,1,40,3,  259 ).
textconcept( 3,5,1,40,4,  260 ).
textconcept( 3,5,2,41,1,  261 ).
textconcept( 3,5,2,41,2,  262 ).
textconcept( 3,5,2,41,3,  263 ).
textconcept( 3,5,2,41,4,  264 ).
textconcept( 3,5,2,41,5,  265 ).
textconcept( 3,5,3,42,1,  266 ).
textconcept( 3,5,4,43,1,  267 ).
textconcept( 3,5,4,43,2,  268 ).
textconcept( 3,5,4,43,3,  269 ).
textconcept( 3,5,4,43,4,  270 ).
textconcept( 3,5,4,43,5,  271 ).
textconcept( 3,5,4,43,6,  272 ).
textconcept( 3,5,4,43,7,  273 ).
textconcept( 3,5,4,43,8,  274 ).
textconcept( 3,5,4,43,9,  275 ).
textconcept( 3,5,4,43,10,  276 ).
textconcept( 3,5,5,44,1,  277 ).
textconcept( 3,5,5,44,2,  278 ).
textconcept( 3,5,5,44,3,  279 ).
textconcept( 3,5,5,44,4,  280 ).
textconcept( 3,5,5,44,5,  281 ).
textconcept( 3,5,5,44,6,  282 ).
textconcept( 3,5,5,44,7,  283 ).
textconcept( 3,5,5,44,8,  284 ).
textconcept( 3,5,6,45,1,  285 ).
textconcept( 3,5,6,45,2,  286 ).
textconcept( 3,5,6,45,3,  287 ).
textconcept( 3,5,6,45,4,  288 ).
textconcept( 3,5,7,46,1,  289 ).
textconcept( 3,5,7,46,2,  290 ).
textconcept( 3,5,7,46,3,  291 ).
textconcept( 3,5,8,47,1,  292 ).
textconcept( 3,5,8,47,2,  293 ).
textconcept( 3,5,8,47,3,  294 ).
textconcept( 3,5,8,47,4,  295 ).
textconcept( 3,5,8,47,5,  296 ).
textconcept( 3,5,8,47,6,  297 ).
textconcept( 3,5,8,47,7,  298 ).
textconcept( 3,5,9,48,1,  299 ).
textconcept( 3,5,9,48,2,  300 ).
textconcept( 3,5,9,48,3,  301 ).
textconcept( 3,5,9,48,4,  302 ).
textconcept( 3,5,9,48,5,  303 ).
textconcept( 3,5,9,48,6,  304 ).
textconcept( 3,5,9,48,7,  305 ).
textconcept( 3,5,9,48,8,  306 ).
textconcept( 4,0,0,49,1,  307 ).
textconcept( 4,0,0,49,2,  308 ).
textconcept( 4,1,1,50,1,  309 ).
textconcept( 4,1,1,50,2,  310 ).
textconcept( 4,1,1,50,3,  311 ).
textconcept( 4,1,1,50,4,  312 ).
textconcept( 4,1,1,50,5,  313 ).
textconcept( 4,1,2,51,1,  314 ).
textconcept( 4,1,2,51,2,  315 ).
textconcept( 4,1,2,51,3,  316 ).
textconcept( 4,1,2,51,4,  317 ).
textconcept( 4,1,2,51,5,  318 ).
textconcept( 4,2,1,52,1,  319 ).
textconcept( 4,2,1,52,2,  320 ).
textconcept( 4,2,1,52,3,  321 ).
textconcept( 4,2,1,52,4,  322 ).
textconcept( 4,2,1,52,5,  323 ).
textconcept( 4,2,1,52,6,  324 ).
```

```
textconcept( 4,2,1,52,7, 325 ).
textconcept( 4,2,2,53,1, 326 ).
textconcept( 4,2,2,53,2, 327 ).
textconcept( 4,2,2,53,3, 328 ).
textconcept( 4,2,2,53,4, 329 ).
textconcept( 4,2,2,53,5, 330 ).
textconcept( 4,2,2,53,6, 331 ).
textconcept( 4,2,2,53,7, 332 ).
textconcept( 4,2,2,53,8, 333 ).
textconcept( 4,2,2,53,9, 334 ).
textconcept( 4,2,2,53,10, 335 ).
textconcept( 4,3,1,54,1, 336 ).
textconcept( 4,3,2,55,1, 337 ).
textconcept( 4,3,2,55,2, 338 ).
textconcept( 4,3,2,55,3, 339 ).
textconcept( 4,3,3,56,1, 340 ).
textconcept( 4,3,3,56,2, 341 ).
textconcept( 4,3,3,56,3, 342 ).
textconcept( 4,3,3,56,4, 343 ).
textconcept( 4,3,3,56,5, 344 ).
textconcept( 4,3,3,56,6, 345 ).
textconcept( 4,3,3,56,7, 346 ).
textconcept( 4,3,3,56,8, 347 ).
textconcept( 4,3,4,57,1, 348 ).
textconcept( 4,3,4,57,2, 349 ).
textconcept( 4,3,4,57,3, 350 ).
textconcept( 4,3,4,57,4, 351 ).
textconcept( 4,3,5,58,1, 352 ).
textconcept( 4,3,5,58,2, 353 ).
textconcept( 4,3,5,58,3, 354 ).
textconcept( 4,3,6,59,1, 355 ).
textconcept( 4,3,6,59,2, 356 ).
textconcept( 4,3,6,59,3, 357 ).
textconcept( 4,3,6,59,4, 358 ).
textconcept( 4,3,6,59,5, 359 ).
textconcept( 4,3,6,59,6, 360 ).
textconcept( 4,3,7,60,1, 361 ).
textconcept( 4,3,7,60,2, 362 ).
textconcept( 4,3,7,60,3, 363 ).
textconcept( 4,3,7,60,4, 364 ).
textconcept( 4,4,1,61,1, 365 ).
textconcept( 4,4,1,61,2, 366 ).
textconcept( 4,4,2,62,1, 367 ).
textconcept( 4,4,2,62,2, 368 ).
textconcept( 4,4,3,63,1, 369 ).
textconcept( 4,4,3,63,2, 370 ).
textconcept( 4,4,3,63,3, 371 ).
textconcept( 4,4,3,63,4, 372 ).
textconcept( 4,4,3,63,5, 373 ).
textconcept( 4,4,3,63,6, 374 ).
textconcept( 4,4,3,63,7, 375 ).
textconcept( 4,4,4,64,1, 376 ).
textconcept( 4,4,4,64,2, 377 ).
textconcept( 4,4,4,64,3, 378 ).
textconcept( 4,4,4,64,4, 379 ).
textconcept( 4,4,4,64,5, 380 ).
textconcept( 4,4,4,64,6, 381 ).
textconcept( 4,4,5,65,1, 382 ).
textconcept( 4,4,5,65,2, 383 ).
textconcept( 4,4,5,65,3, 384 ).
textconcept( 4,4,5,65,4, 385 ).
textconcept( 4,4,5,65,5, 386 ).
textconcept( 4,4,5,65,6, 387 ).
textconcept( 4,4,5,65,7, 388 ).
textconcept( 4,4,5,65,8, 389 ).
textconcept( 4,4,5,65,9, 390 ).
textconcept( 4,4,6,66,1, 391 ).
textconcept( 4,4,6,66,2, 392 ).
textconcept( 4,4,6,66,3, 393 ).
textconcept( 4,4,6,66,4, 394 ).
textconcept( 4,4,7,67,1, 395 ).
textconcept( 4,4,7,67,2, 396 ).
textconcept( 4,4,8,68,1, 397 ).
textconcept( 4,4,8,68,2, 398 ).
textconcept( 4,4,8,68,3, 399 ).
textconcept( 4,4,8,68,4, 400 ).
textconcept( 4,4,8,68,5, 401 ).
textconcept( 4,4,9,69,1, 402 ).
textconcept( 4,4,9,69,2, 403 ).
textconcept( 4,4,9,69,3, 404 ).
textconcept( 4,4,9,69,4, 405 ).
textconcept( 4,4,9,69,5, 406 ).
textconcept( 4,4,10,70,1, 407 ).
textconcept( 4,4,10,70,2, 408 ).
textconcept( 4,4,10,70,3, 409 ).
textconcept( 4,4,10,70,4, 410 ).
textconcept( 4,4,10,70,5, 411 ).
textconcept( 4,4,10,70,6, 412 ).
textconcept( 4,4,10,70,7, 413 ).
textconcept( 4,4,11,71,1, 414 ).
textconcept( 4,4,11,71,2, 415 ).
textconcept( 4,4,11,71,3, 416 ).
textconcept( 4,4,11,71,4, 417 ).
textconcept( 4,4,11,71,5, 418 ).
textconcept( 4,4,11,71,6, 419 ).
textconcept( 4,4,11,71,7, 420 ).
textconcept( 4,4,12,72,1, 421 ).
textconcept( 4,4,12,72,2, 422 ).
textconcept( 4,4,12,72,3, 423 ).
textconcept( 4,4,12,72,4, 424 ).
textconcept( 4,4,12,72,5, 425 ).
textconcept( 4,4,12,72,6, 426 ).
textconcept( 4,4,12,72,7, 427 ).
textconcept( 4,4,13,73,1, 428 ).
textconcept( 4,4,13,73,2, 429 ).
textconcept( 4,4,13,73,3, 430 ).
textconcept( 4,4,13,73,4, 431 ).
textconcept( 4,4,13,73,5, 432 ).
textconcept( 4,4,13,73,6, 433 ).
textconcept( 4,4,13,73,7, 434 ).
textconcept( 4,4,13,73,8, 435 ).
textconcept( 4,4,14,74,1, 436 ).
textconcept( 4,4,14,74,2, 437 ).
textconcept( 4,4,15,75,1, 438 ).
textconcept( 4,4,15,75,2, 439 ).
textconcept( 4,4,15,75,3, 440 ).
textconcept( 4,4,15,75,4, 441 ).
textconcept( 4,4,15,75,5, 442 ).
textconcept( 4,5,1,76,1, 443 ).
textconcept( 4,5,1,76,2, 444 ).
textconcept( 4,5,1,76,3, 445 ).
textconcept( 4,5,1,76,4, 446 ).
textconcept( 4,5,1,76,5, 447 ).
textconcept( 4,5,1,76,6, 448 ).
textconcept( 4,5,1,76,7, 449 ).
textconcept( 4,5,1,76,8, 450 ).
textconcept( 4,5,2,77,1, 451 ).
textconcept( 4,5,2,77,2, 452 ).
textconcept( 4,5,2,77,3, 453 ).
textconcept( 4,5,2,77,4, 454 ).
textconcept( 4,5,2,77,5, 455 ).
textconcept( 4,5,2,77,6, 456 ).
```

textconcept(4,5,2,77,7, 457).
textconcept(4,5,2,77,8, 458).
textconcept(4,6,1,78,1, 459).
textconcept(4,6,1,78,2, 460).
textconcept(4,6,2,79,1, 461).
textconcept(4,6,2,79,2, 462).
textconcept(4,6,3,80,1, 463).
textconcept(4,6,3,80,2, 464).
textconcept(4,6,3,80,3, 465).
textconcept(4,6,3,80,4, 466).
textconcept(4,6,3,80,5, 467).
textconcept(4,6,3,80,6, 468).
textconcept(4,6,3,80,7, 469).
textconcept(4,6,4,81,1, 470).
textconcept(4,6,4,81,2, 471).
textconcept(4,6,4,81,3, 472).
textconcept(4,6,4,81,4, 473).
textconcept(4,6,4,81,5, 474).
textconcept(4,7,1,82,1, 475).
textconcept(4,7,1,82,2, 476).
textconcept(4,7,1,82,3, 477).
textconcept(4,7,1,82,4, 478).
textconcept(4,7,2,83,1, 479).
textconcept(4,7,2,83,2, 480).
textconcept(4,7,2,83,3, 481).
textconcept(4,7,3,84,1, 482).
textconcept(4,7,3,84,2, 483).
textconcept(4,7,4,85,1, 484).
textconcept(4,7,4,85,2, 485).
textconcept(4,7,4,85,3, 486).
textconcept(4,7,4,85,4, 487).
textconcept(4,7,4,85,5, 488).
textconcept(4,7,4,85,6, 489).
textconcept(4,7,4,85,7, 490).
textconcept(4,7,4,85,8, 491).
textconcept(5,0,0,86,1, 492).
textconcept(5,1,1,87,1, 493).
textconcept(5,1,1,87,2, 494).
textconcept(5,1,1,87,3, 495).
textconcept(5,1,1,87,4, 496).
textconcept(5,1,1,87,5, 497).
textconcept(5,1,2,88,1, 498).
textconcept(5,1,2,88,2, 499).
textconcept(5,1,2,88,3, 500).
textconcept(5,1,2,88,4, 501).
textconcept(5,1,2,88,5, 502).
textconcept(5,1,2,88,6, 503).
textconcept(5,1,2,88,7, 504).
textconcept(5,1,2,88,8, 505).
textconcept(5,1,2,88,9, 506).
textconcept(5,2,1,89,1, 507).
textconcept(5,2,1,89,2, 508).
textconcept(5,2,1,89,3, 509).
textconcept(5,2,1,89,4, 510).
textconcept(5,2,1,89,5, 511).
textconcept(5,2,1,89,6, 512).
textconcept(5,2,1,89,7, 513).
textconcept(5,2,1,89,8, 514).
textconcept(5,2,1,89,9, 515).
textconcept(5,2,2,90,1, 516).
textconcept(5,2,2,90,2, 517).
textconcept(5,2,2,90,3, 518).
textconcept(5,2,2,90,4, 519).
textconcept(5,3,1,91,1, 520).
textconcept(5,3,1,91,2, 521).
textconcept(5,3,1,91,4, 523).
textconcept(5,3,2,92,1, 524).
textconcept(5,3,2,92,2, 525).
textconcept(5,3,2,92,3, 526).
textconcept(5,3,2,92,4, 527).
textconcept(5,3,2,92,5, 528).
textconcept(5,4,1,93,1, 529).
textconcept(5,4,1,93,2, 530).
textconcept(5,4,1,93,3, 531).
textconcept(5,4,1,93,4, 532).
textconcept(5,4,2,94,1, 533).
textconcept(5,4,2,94,2, 534).
textconcept(5,4,2,94,3, 535).
textconcept(5,4,2,94,4, 536).
textconcept(5,4,2,94,5, 537).
textconcept(5,4,2,94,6, 538).
textconcept(5,4,2,94,7, 539).
textconcept(5,5,1,95,1, 540).
textconcept(5,5,1,95,2, 541).
textconcept(5,5,1,95,3, 542).
textconcept(5,5,1,95,4, 543).
textconcept(5,5,2,96,1, 544).
textconcept(5,5,2,96,2, 545).
textconcept(5,5,2,96,3, 546).
textconcept(5,5,2,96,4, 547).
textconcept(5,5,2,96,5, 548).
textconcept(5,5,2,96,6, 549).
textconcept(5,5,2,96,7, 550).
textconcept(5,5,2,96,8, 551).
textconcept(5,6,1,97,1, 552).
textconcept(5,6,2,98,1, 553).
textconcept(5,6,2,98,2, 554).
textconcept(5,6,2,98,3, 555).
textconcept(5,6,3,99,1, 556).
textconcept(5,6,3,99,2, 557).
textconcept(5,6,3,99,3, 558).
textconcept(5,6,4,100,1, 559).
textconcept(5,6,4,100,2, 560).
textconcept(5,6,4,100,3, 561).
textconcept(5,6,4,100,4, 562).
textconcept(5,6,4,100,5, 563).
textconcept(5,6,5,101,1, 564).
textconcept(5,6,5,101,2, 565).
textconcept(5,6,6,102,1, 566).
textconcept(5,6,6,102,2, 567).
textconcept(5,6,6,102,3, 568).
textconcept(5,6,6,102,4, 569).
textconcept(5,6,7,103,1, 570).
textconcept(5,6,7,103,2, 571).
textconcept(5,6,7,103,3, 572).
textconcept(5,6,7,103,4, 573).
textconcept(5,7,1,104,1, 574).
textconcept(5,7,1,104,2, 575).
textconcept(5,7,1,104,3, 576).
textconcept(5,7,2,105,1, 577).
textconcept(5,7,2,105,2, 578).
textconcept(5,7,2,105,3, 579).
textconcept(5,7,2,105,4, 580).
textconcept(5,7,2,105,5, 581).
textconcept(5,7,2,105,6, 582).
textconcept(5,7,3,106,1, 583).
textconcept(5,7,3,106,2, 584).
textconcept(5,7,3,106,3, 585).
textconcept(5,7,3,106,4, 586).
textconcept(5,7,3,106,5, 587).

```
textconcept( 5,7,3,106,7,  589 ).
textconcept( 5,7,3,106,8,  590 ).
textconcept( 5,7,4,107,1,  591 ).
textconcept( 5,7,4,107,2,  592 ).
textconcept( 5,7,4,107,3,  593 ).
textconcept( 5,7,4,107,4,  594 ).
textconcept( 5,7,5,108,1,  595 ).
textconcept( 5,7,5,108,2,  596 ).
textconcept( 5,7,5,108,3,  597 ).
textconcept( 5,7,5,108,4,  598 ).
textconcept( 5,7,5,108,5,  599 ).
textconcept( 5,7,5,108,6,  600 ).
textconcept( 5,7,5,108,7,  601 ).
textconcept( 5,7,5,108,8,  602 ).
```

```
/* text4cpt.pl */
/* prolog database produced by txtcpt.spt */
/* concept(concept identity, concept) */
concept( 1, [ expert, advisor] ).
concept( 2, [ photolithography] ).
concept( 3, [ introduction] ).
concept( 4, [ integrated, circuit, fabrication, line] ).
concept( 5, [ manufacturing, lines] ).
concept( 6, [ organized] ).
concept( 7, [ sequence, of, stations, mamned] ).
concept( 8, [ technicians] ).
concept( 9, [ stations, process] ).
concept( 10, [ wafer] ).
concept( 11, [ twice] ).
concept( 12, [ finished] ).
concept( 13, [ photolithography, step] ).
concept( 14, [ placing] ).
concept( 15, [ pattern] ).
concept( 16, [ wafer] ).
concept( 17, [ photography] ).
concept( 18, [  each, wafer] ).
concept( 19, [ processed] ).
concept( 20, [ 15 times] ).
concept( 21, [ photolithography, problem, arises] ).
concept( 22, [ step, becomes] ).
concept( 23, [ bottleneck] ).
concept( 24, [ whole, line] ).
concept( 25, [  wafers, continue] ).
concept( 26, [ processed] ).
concept( 27, [ other,  stations] ).
concept( 28, [  causing] ).
concept( 29, [ stockpile] ).
concept( 30, [ build] ).
concept( 31, [ process, engineer] ).
concept( 32, [ called] ).
concept( 33, [ fire, fight] ).
concept( 34, [ diagnose] ).
concept( 35, [ correct] ).
concept( 36, [  problem] ).
concept( 37, [ fire, fighting] ).
concept( 38, [ process, engineer's] ).
concept( 39, [ priority] ).
concept( 40, [ get] ).
concept( 41, [ line] ).
concept( 42, [ running, again] ).
concept( 43, [ possible] ).
concept( 44, [  fire, fighting, consumes] ).
concept( 45, [ half] ).
concept( 46, [ day] ).
concept( 47, [  conflicting] ).
concept( 48, [ main, responsiblities, of, coordinating, line, personnel] ).
concept( 49, [  monitoring] ).
concept( 50, [ controlling] ).
concept( 51, [ process] ).
concept( 52, [ meet, specifications] ).
concept( 53, [ uniformity, of, pattern, thickness] ).
concept( 54, [  develop] ).
concept( 55, [  exposure] ).
concept( 56, [  baking] ).
concept( 57, [ other, variables] ).
concept( 58, [ circumstances] ).
concept( 59, [  temporary, solutions] ).
concept( 60, [ result] ).
concept( 61, [ no, time,  left] ).
concept( 62, [ designing, long-term, solutions] ).
```

```
concept( 63, [ recurrent, problems] ).
concept( 64, [ expert, system] ).
concept( 65, [ photolithography, advisor] ).
concept( 66, [ help, alleviate, fire, fighting] ).
concept( 67, [ monitor] ).
concept( 68, [ control] ).
concept( 69, [ process] ).
concept( 70, [ record, process, data] ).
concept( 71, [ help, determine] ).
concept( 72, [ recurrent, problems, are] ).
concept( 73, [provide] ).
concept( 74, [ information, needed] ).
concept( 75, [ coordinate, line, personnel] ).
concept( 76, [ different, work, shifts] ).
concept( 77, [ complex, procedures, needed] ).
concept( 78, [ produce, correct, patterns] ).
concept( 79, [ photolithography] ).
concept( 80, [ varied] ).
concept( 81, [ overview] ).
concept( 82, [ thompson] ).
concept( 83, [ et, al] ).
concept( 84, [ addition] ).
concept( 85, [ number, of, ways] ).
concept( 86, [ defective, pattern] ).
concept( 87, [ produced] ).
concept( 88, [ very, large] ).
concept( 89, [ armed] ).
concept( 90, [ knowledge] ).
concept( 91, [ the, advisor] ).
concept( 92, [ help] ).
concept( 93, [ technician, focus] ).
concept( 94, [ data, pertinent] ).
concept( 95, [ isolating] ).
concept( 96, [ defect] ).
concept( 97, [ determining] ).
concept( 98, [ cause] ).
concept( 99, [ recommend, corrective, action] ).
concept( 100, [ advisor] ).
concept( 101, [ process, engineer] ).
concept( 102, [ technician] ).
concept( 103, [ search] ).
concept( 104, [ very, long, time] ).
concept( 105, [ become, overwhelmed] ).
concept( 106, [ frustrated] ).
concept( 107, [ designed] ).
concept( 108, [ advisor] ).
concept( 109, [ mimic] ).
concept( 110, [ problem-solving, behavior, of, human, experts] ).
concept( 111, [ line, technicians] ).
concept( 112, [ feel, confident] ).
concept( 113, [ program] ).
concept( 114, [ help] ).
concept( 115, [ want] ).
concept( 116, [ use] ).
concept( 117, [ instead, of, depending] ).
concept( 118, [ process, engineer] ).
concept( 119, [ involved, not] ).
concept( 120, [ capturing] ).
concept( 121, [ rules] ).
concept( 122, [ knowledge, of, several, photolithography, engineers] ).
concept( 123, [ field, test, experience, of, technicians] ).
concept( 124, [ of, effort] ).
concept( 125, [ implementimg] ).
concept( 126, [ interface] ).
concept( 127, [ presents] ).
concept( 128, [ familiar, model, of, work] ).
```

```
concept( 129, [ electronic, versions, of, the, paper, forms] ).
concept( 130, [ used] ).
concept( 131, [ process, recordkeeping] ).
concept( 132, [ succeeded] ).
concept( 133, [  still] ).
concept( 134, [ evaluated] ).
concept( 135, [ preliminary, evaluation] ).
concept( 136, [ field, testing] ).
concept( 137, [ research, line] ).
concept( 138, [ results, of, that,  test, helped] ).
concept( 139, [ design] ).
concept( 140, [ more, thorough, testing] ).
concept( 141, [ progress] ).
concept( 142, [ manufacturing, line] ).
concept( 143, [ photolithography, advisor] ).
concept( 144, [ photolithography, advisor] ).
concept( 145, [ written] ).
concept( 146, [ hp-rl] ).
concept( 147, [ rosenberg] ).
concept( 148, [ knowledge, representation, language] ).
concept( 149, [  supports, structured, declarative, objects] ).
concept( 150, [ frames] ).
concept( 151, [  messagepassing] ).
concept( 152, [  active, values] ).
concept( 153, [ daemons] ).
concept( 154, [  procedural, values] ).
concept( 155, [ rule,  system] ).
concept( 156, [ logic, programming] ).
concept( 157, [ agenda-based, inference,  engines] ).
concept( 158, [ forward] ).
concept( 159, [ backward, chaining] ).
concept( 160, [ smooth,  interface] ).
concept( 161, [ host, llsp, environment] ).
concept( 162, [ hp-rl, runs] ).
concept( 163, [  hp, model, 9000 series, 200 workstations] ).
concept( 164, [ both, psl] ).
concept( 165, [ griss] ).
concept( 166, [ common-lisp] ).
concept( 167, [ steele] ).
concept( 168, [ advisor] ).
concept( 169, [ small] ).
concept( 170, [ medium, sized, expert, system] ).
concept( 171, [ builds] ).
concept( 172, [ number, of, well-known, medical, diagnosis, programs] ).
concept( 173, [  borrowing] ).
concept( 174, [ particular] ).
concept( 175, [ of, their, architectural, ideas] ).
concept( 176, [ shortliffe] ).
concept( 177, [  1] ).
concept( 178, [ aikins] ).
concept( 179, [ kunz] ).
concept( 180, [  et, al] ).
concept( 181, [ smith] ).
concept( 182, [ clayton] ).
concept( 183, [ program] ).
concept( 184, [contains] ).
concept( 185, [ 30 class-frames] ).
concept( 186, [ 100 slots, each] ).
concept( 187, [ 300  rules] ).
concept( 188, [  30 forward-chaining] ).
concept( 189, [ rest, backward-chaining] ).
concept( 190, [ advisor] ).
concept( 191, [ structured] ).
concept( 192, [ team, of, three, cooperating, specialists] ).
concept( 193, [ work, together] ).
concept( 194, [ solve, problems, posed] ).
```

```
concept( 195, [ user] ).
concept( 196, [ inspector, works] ).
concept( 197, [ user] ).
concept( 198, [ collect,  symptoms] ).
concept( 199, [  passing] ).
concept( 200, [ information] ).
concept( 201, [ diagnostician] ).
concept( 202, [ diagnostician, uses, knowledge, of, chemistry] ).
concept( 203, [ photolithography, equipment] ).
concept( 204, [ diagnose, wafer, problems] ).
concept( 205, [ causes] ).
concept( 206, [ tells] ).
concept( 207, [ repairman] ).
concept( 208, [ repairman] ).
concept( 209, [ advises] ).
concept( 210, [ user] ).
concept( 211, [ correct] ).
concept( 212, [ diagnosed, problems] ).
concept( 213, [ specialists, communicate] ).
concept( 214, [  frame, database] ).
concept( 215, [ sense] ).
concept( 216, [ advisor, uses] ).
concept( 217, [ blackboard,  architecture] ).
concept( 218, [ erman, et] ).
concept( 219, [ al] ).
concept( 220, [  any, specialist] ).
concept( 221, [ ask] ).
concept( 222, [ more, information] ).
concept( 223, [ needed] ).
concept( 224, [ allows] ).
concept( 225, [ consultation] ).
concept( 226, [ be, focused] ).
concept( 227, [  eliminates,  asking] ).
concept( 228, [ extraneous] ).
concept( 229, [ irrelevant, information] ).
concept( 230, [ expensive] ).
concept( 231, [ time] ).
concept( 232, [ effort] ).
concept( 233, [ gather] ).
concept( 234, [ results] ).
concept( 235, [  more, human-like, interaction] ).
concept( 236, [ basic, diagnostic, strategy] ).
concept( 237, [ work] ).
concept( 238, [ visual,  symptoms] ).
concept( 239, [ visual, inspection, revealing, no, symptoms,  does, not, mean]
concept( 240, [ wafer] ).
concept( 241, [ free, of, defects] ).
concept( 242, [ means] ).
concept( 243, [ no, gross, defects] ).
concept( 244, [ detected] ).
concept( 245, [ naked, eye] ).
concept( 246, [ cases] ).
concept( 247, [ advisor, requests] ).
concept( 248, [ user] ).
concept( 249, [ examine] ).
concept( 250, [  wafer] ).
concept( 251, [ microscope] ).
concept( 252, [ low, magnification] ).
concept( 253, [ symptoms] ).
concept( 254, [ found] ).
concept( 255, [ inspection] ).
concept( 256, [ verified] ).
concept( 257, [ high, magnification] ).
concept( 258, [ user, interface] ).
concept( 259, [ turned] ).
concept( 260, [ one, of, the] ).
```

```
concept( 261, [  important, components] ).
concept( 262, [ more, interested] ).
concept( 263, [ problem-solving, performance] ).
concept( 264, [ ease, of, use] ).
concept( 265, [ simple, gloss, tty, interface] ).
concept( 266, [ used] ).
concept( 267, [ unsatisfactory] ).
concept( 268, [ response] ).
concept( 269, [ implemented] ).
concept( 270, [ interface] ).
concept( 271, [ required,  very, little, typing] ).
concept( 272, [ all, questions] ).
concept( 273, [ asked] ).
concept( 274, [ popup, menus] ).
concept( 275, [ recorded] ).
concept( 276, [ interaction] ).
concept( 277, [ transcript, window] ).
concept( 278, [ user] ).
concept( 279, [ answer] ).
concept( 280, [ questions] ).
concept( 281, [ single-key, command] ).
concept( 282, [  pressing] ).
concept( 283, [ softkey] ).
concept( 284, [ selecting] ).
concept( 285, [ mouse] ).
concept( 286, [ consultation] ).
concept( 287, [ reviewed] ).
concept( 288, [  scrolling, back] ).
concept( 289, [ transcript, window] ).
concept( 290, [ building, another, interface] ).
concept( 291, [ implements] ).
concept( 292, [ virtual, advisor's,  notebook] ).
concept( 293, [ uses, graphical, representations, of, the, wafer] ).
concept( 294, [ processing, equipment] ).
concept( 295, [  sophisticated, browsers] ).
concept( 296, [ examining] ).
concept( 297, [ knowledge-base] ).
concept( 298, [ laser-disc, database, of,  symptom] ).
concept( 299, [ defect, images] ).
concept( 300, [ easy] ).
concept( 301, [ use] ).
concept( 302, [ current, interface] ).
concept( 303, [ interface] ).
concept( 304, [ use, more] ).
concept( 305, [ stimulating, media] ).
concept( 306, [provide] ).
concept( 307, [ more, explanatory, power] ).
concept( 308, [  performance] ).
concept( 309, [ evaluation] ).
concept( 310, [ version, of, the, photolithography, advisor] ).
concept( 311, [ installed] ).
concept( 312, [ low-volume, research, lab] ).
concept( 313, [ used] ).
concept( 314, [ year] ).
concept( 315, [ time, of, this, writing] ).
concept( 316, [ installed] ).
concept( 317, [  second, version, of, the, program] ).
concept( 318, [ manufacturing, line] ).
concept( 319, [  additional, tasting] ).
concept( 320, [ test] ).
concept( 321, [ research, lab, helped] ).
concept( 322, [ debug] ).
concept( 323, [ knowledge, base] ).
concept( 324, [ obtain, useful, ideas] ).
concept( 325, [ needed] ).
concept( 326, [ user, interface] ).
```

```
concept( 327, [  overall] ).
concept( 328, [ program] ).
concept( 329, [ considered] ).
concept( 330, [ useful] ).
concept( 331, [ diagnosed, problems] ).
concept( 332, [ available, process, engineers] ).
concept( 333, [ same, ones] ).
concept( 334, [ helped] ).
concept( 335, [ build] ).
concept( 336, [ program] ).
concept( 337, [ of, the, benefits, of, the, program] ).
concept( 338, [  substantial, time] ).
concept( 339, [ saved] ).
concept( 340, [ processing, wafers] ).
concept( 341, [  process, error] ).
concept( 342, [ not, detected] ).
concept( 343, [  reprocessing] ).
concept( 344, [ take] ).
concept( 345, [ 8 hours] ).
concept( 346, [ running] ).
concept( 347, [ program, required,  less] ).
concept( 348, [ 10 minutes] ).
concept( 349, [ advisor] ).
concept( 350, [ always, available] ).
concept( 351, [ during] ).
concept( 352, [  night, shift] ).
concept( 353, [ process, engineers] ).
concept( 354, [ sleep] ).
concept( 355, [ night] ).
concept( 356, [ hardware, emitted, less, particles] ).
concept( 357, [ engineer] ).
concept( 358, [ installed] ).
concept( 359, [ clean, room, where] ).
concept( 360, [  convenient] ).
concept( 361, [ available] ).
concept( 362, [ hardware, cost] ).
concept( 363, [ less] ).
concept( 364, [  cheaper] ).
concept( 365, [ full-time, engineer] ).
concept( 366, [ of, the, needs, uncovered, during] ).
concept( 367, [ informal, test] ).
concept( 368, [ users] ).
concept( 369, [ naive, computer, users] ).
concept( 370, [ interface] ).
concept( 371, [ eliminate] ).
concept( 372, [ typing] ).
concept( 373, [make] ).
concept( 374, [ correcting,  errors, easy] ).
concept( 375, [ typing] ).
concept( 376, [ needed] ).
concept( 377, [ dealt] ).
concept( 378, [ next, version] ).
concept( 379, [ popup,  menus] ).
concept( 380, [ customized, softkeys] ).
concept( 381, [  eliminating] ).
concept( 382, [ typing] ).
concept( 383, [ where, typing] ).
concept( 384, [ still, needed] ).
concept( 385, [  customized, editor,  modes] ).
concept( 386, [ designed] ).
concept( 387, [ limited] ).
concept( 388, [ number, of, ways] ).
concept( 389, [ user] ).
concept( 390, [ get] ).
concept( 391, [ trouble] ).
```

```
concept( 393, [ advisor, used, jargon] ).
concept( 394, [describe] ).
concept( 395, [ wafer, defects] ).
concept( 396, [ menu, items] ).
concept( 397, [ misinterpreted] ).
concept( 398, [ dealt] ).
concept( 399, [ terminology, problem] ).
concept( 400, [ adding, alternate, definitions] ).
concept( 401, [ defects] ).
concept( 402, [ coilected, photographs, to, illustrate, defects] ).
concept( 403, [ showing] ).
concept( 404, [ photograph, of, the, defect] ).
concept( 405, [ trying] ).
concept( 406, [describe] ).
concept( 407, [ avoided, misinterpretation, of, the, text] ).
concept( 408, [  experienced, users] ).
concept( 409, [ frustrated, having] ).
concept( 410, [ answer,  questions] ).
concept( 411, [ visual, symptoms] ).
concept( 412, [ knew] ).
concept( 413, [  high-power, inspection] ).
concept( 414, [ needed] ).
concept( 415, [ accomodated] ).
concept( 416, [ structuring] ).
concept( 417, [ menus] ).
concept( 418, [ default, answers] ).
concept( 419, [ provided] ).
concept( 420, [ express-lane] ).
concept( 421, [ more, detailed,  inspection] ).
concept( 422, [ attempt] ).
concept( 423, [make] ).
concept( 424, [ problem-solving,  knowledge, generic] ).
concept( 425, [ fabrication, lines] ).
concept( 426, [ want] ).
concept( 427, [ customize] ).
concept( 428, [ local, practice] ).
concept( 429, [ domain, experts] ).
concept( 430, [  training] ).
concept( 431, [ edit] ).
concept( 432, [ knowledge, base] ).
concept( 433, [ sophisticated, set, of, browsing, tools] ).
concept( 434, [ not,  require, knowing] ).
concept( 435, [ syntax, of, either, hp-rl] ).
concept( 436, [ lisp] ).
concept( 437, [ program's, problem-solving, ability] ).
concept( 438, [ tested] ).
concept( 439, [  measuring] ).
concept( 440, [ ability] ).
concept( 441, [ difficult] ).
concept( 442, [ research, processes] ).
concept( 443, [ modified] ).
concept( 444, [ of, the, last, point] ).
concept( 445, [ started] ).
concept( 446, [  extensive, field, test] ).
concept( 447, [ production, line] ).
concept( 448, [ more] ).
concept( 449, [ test] ).
concept( 450, [ target, environment] ).
concept( 451, [ research, facility] ).
concept( 452, [ evaluation] ).
concept( 453, [consists] ).
concept( 454, [ of, an] ).
concept( 455, [  vivo, use, of, the, program] ).
concept( 456, [ negative, resist, fabrication, line] ).
concept( 457, [ user, survey] ).
concept( 458, [ comparision, of, key, process, statistics] ).
```

```
concept( 459, [ data] ).
concept( 460, [ user, survey] ).
concept( 461, [ two, parts] ).
concept( 462, [ administered] ).
concept( 463, [ each, consultation] ).
concept( 464, [ trying] ).
concept( 465, [ ascertain] ).
concept( 466, [  program's, accuracy] ).
concept( 467, [ obtain, detailed, comments] ).
concept( 468, [ wording] ).
concept( 469, [ menus] ).
concept( 470, [  of, advice] ).
concept( 471, [ second, survey] ).
concept( 472, [ administered, every, month] ).
concept( 473, [ collect, more, general, comments] ).
concept( 474, [ friendliness] ).
concept( 475, [ usefulness, of, the, system] ).
concept( 476, [  statistics, measuring, process, performance, include, yield,
concept( 477, [  rework, rates] ).
concept( 478, [  process, hold, times] ).
concept( 479, [ equipment, downtime] ).
concept( 480, [ of, these] ).
concept( 481, [ collected] ).
concept( 482, [ existing, process, management, software] ).
concept( 483, [ large,  database, of, data] ).
concept( 484, [ past, performance, of, the, process, line,  exists] ).
concept( 485, [ cases, where] ).
concept( 486, [ necessary, data] ).
concept( 487, [ not, logged] ).
concept( 488, [  existing, software] ).
concept( 489, [ advisor] ).
concept( 490, [ collects] ).
concept( 491, [ compared] ).
concept( 492, [ reconstructed, past,  data] ).
concept( 493, [ conclusions] ).
concept( 494, [    developing] ).
concept( 495, [ photolithography, advisor] ).
concept( 496, [ useful,  experiment] ).
concept( 497, [ building] ).
concept( 498, [ industrial, strength, expert, system] ).
concept( 499, [ other] ).
concept( 500, [ systems] ).
concept( 501, [ experiment] ).
concept( 502, [ never] ).
concept( 503, [ finished] ).
concept( 504, [ valuable, insights] ).
concept( 505, [  process] ).
concept( 506, [ already] ).
concept( 507, [ gained] ).
concept( 508, [ found] ).
concept( 509, [ peopie, not, trained] ).
concept( 510, [ ai] ).
concept( 511, [  learn] ).
concept( 512, [ modify] ).
concept( 513, [ expert, system] ).
concept( 514, [ domain, of, expertise] ).
concept( 515, [ given] ).
concept( 516, [ right, tools] ).
concept( 517, [ found] ).
concept( 518, [ limited] ).
concept( 519, [ not] ).
concept( 520, [ experienced, ai, programmer] ).
concept( 521, [  domain, experts, tend] ).
concept( 522, [ introspect] ).
concept( 523, [ problemsolving, strategies] ).
concept( 524, [ terms, of, plans] ).
```

```
concept( 525, [ simple, planning,  constructs] ).
concept( 526, [ useful] ).
concept( 527, [ helping] ).
concept( 528, [ articulate] ).
concept( 529, [ expertise] ).
concept( 530, [  building] ).
concept( 531, [ expert, system] ).
concept( 532, [ common] ).
concept( 533, [  other, types, of, software, engineering] ).
concept( 534, [ programming,  paradigm] ).
concept( 535, [ different] ).
concept( 536, [ experimentation] ).
concept( 537, [  iteration] ).
concept( 538, [  testing] ).
concept( 539, [ important] ).
concept( 540, [ other, types, of, programming] ).
concept( 541, [  meta-level, rules] ).
concept( 542, [ useful] ).
concept( 543, [ capture] ).
concept( 544, [ expert's, problem-solving, heuristics] ).
concept( 545, [ problem] ).
concept( 546, [ better, understood] ).
concept( 547, [ replaced] ).
concept( 548, [  specific, object-level, rules] ).
concept( 549, [ solve] ).
concept( 550, [ problem] ).
concept( 551, [ better] ).
concept( 552, [ more] ).
concept( 553, [  experts, disagree] ).
concept( 554, [ helps] ).
concept( 555, [ sample] ).
concept( 556, [ of, them] ).
concept( 557, [ application] ).
concept( 558, [ problems] ).
concept( 559, [ solved, several, ways] ).
concept( 560, [ solution] ).
concept( 561, [ used] ).
concept( 562, [ depends] ).
concept( 563, [ personal, preference] ).
concept( 564, [ established, local, procedures] ).
concept( 565, [ vocabulary] ).
concept( 566, [ problem] ).
concept( 567, [  different, people] ).
concept( 568, [ call] ).
concept( 569, [ same, thing] ).
concept( 570, [  different, names] ).
concept( 571, [ problem, becomes, acute] ).
concept( 572, [ trying] ).
concept( 573, [ design] ).
concept( 574, [ user, interface] ).
concept( 575, [ enhancing] ).
concept( 576, [ advisor] ).
concept( 577, [ several, ways] ).
concept( 578, [  knowledge-base] ).
concept( 579, [ extended] ).
concept( 580, [ include, positive,  resist, processes] ).
concept( 581, [ gaining, popularity] ).
concept( 582, [ fabricating, smaller, patterns] ).
concept( 583, [ wafers] ).
concept( 584, [ user, interface] ).
concept( 585, [  redesigned] ).
concept( 586, [ better, reflect] ).
concept( 587, [ model, of, the, program] ).
concept( 588, [ team, of, cooperating, specialists] ).
concept( 589, [ graphics] ).
concept( 590, [  animation] ).
```

```
concept( 591, [ video, display, media] ).
concept( 592, [ problem-solving, strategy, used] ).
concept( 593, [ program] ).
concept( 594, [ generic] ).
concept( 595, [ wide,  variety, of, fault-diagnosis, problems, important] ).
concept( 596, [  restructuring] ).
concept( 597, [ knowledge, representation] ).
concept( 598, [ better, separate] ).
concept( 599, [ problem-solving, knowledge] ).
concept( 600, [ control] ).
concept( 601, [  domain-specific, knowledge] ).
concept( 602, [ fault-diagnosis, shell] ).
concept( 603, [ extracted] ).
/* heading(no,heading) */
heading( 0,' An Expert Advisor For Photolithography '  ).
heading( 1,' Introduction '  ).
heading( 2,' The Photolithography Advisor '  ).
heading( 3,'  Performance and Evaluation '  ).
heading( 4,' Conclusions '  ).
/* summary ( sentence no., no. of concepts identified ) */
summary( 1,2).
summary( 2,1).
summary( 3,5).
summary( 4,12).
summary( 5,4).
summary( 6,12).
summary( 7,7).
summary( 8,14).
summary( 9,6).
summary( 10,13).
summary( 11,4).
summary( 12,3).
summary( 13,5).
summary( 14,11).
summary( 15,7).
summary( 16,12).
summary( 17,13).
summary( 18,3).
summary( 19,3).
summary( 20,5).
summary( 21,1).
summary( 22,18).
summary( 23,6).
summary( 24,3).
summary( 25,12).
summary( 26,7).
summary( 27,6).
summary( 28,6).
summary( 29,6).
summary( 30,5).
summary( 31,2).
summary( 32,4).
summary( 33,1).
summary( 34,4).
summary( 35,12).
summary( 36,3).
summary( 37,3).
summary( 38,4).
summary( 39,7).
summary( 40,5).
summary( 41,4).
summary( 42,5).
summary( 43,1).
summary( 44,10).
summary( 45,8).
summary( 46,4).
```

```
summary( 47,3).
summary( 48,7).
summary( 49,8).
summary( 50,2).
summary( 51,5).
summary( 52,5).
summary( 53,7).
summary( 54,10).
summary( 55,1).
summary( 56,3).
summary( 57,8).
summary( 58,4).
summary( 59,3).
summary( 60,6).
summary( 61,4).
summary( 62,2).
summary( 63,2).
summary( 64,7).
summary( 65,6).
summary( 66,9).
summary( 67,4).
summary( 68,2).
summary( 69,5).
summary( 70,5).
summary( 71,7).
summary( 72,7).
summary( 73,7).
summary( 74,8).
summary( 75,2).
summary( 76,5).
summary( 77,8).
summary( 78,8).
summary( 79,2).
summary( 80,2).
summary( 81,12).
summary( 82,4).
summary( 83,3).
summary( 84,2).
summary( 85,8).
summary( 86,1).
summary( 87,5).
summary( 88,9).
summary( 89,9).
summary( 90,4).
summary( 91,4).
summary( 92,5).
summary( 93,4).
summary( 94,7).
summary( 95,4).
summary( 96,8).
summary( 97,1).
summary( 98,3).
summary( 99,3).
summary( 100,5).
summary( 101,2).
summary( 102,4).
summary( 103,4).
summary( 104,3).
summary( 105,6).
summary( 106,8).
summary( 107,4).
summary( 108,8).
/* textstats (no.of sentences,no. of headings,no.of concepts,no.of words,no. of
textstats( 108, 5, 603, 1863, 24 ).
```

```
/* text5status.pl */

status(1,title ).
status(2,heading ).
status( 12, theme_marker).
status( 17, theme_marker).
status( 20, theme_marker).
status( 31, theme_marker).
status(37,heading ).
status( 77, theme_marker).
status(81,heading ).
status( 83, theme_marker).
status( 100, theme_marker).
status(108,heading ).
status(129,heading ).
status( 130, theme_marker).
status( 133, theme_marker).
status( 135, theme_marker).
status( 145, theme_marker).
```

```
/* text5func.pl */

function(1,title).
function(2, introduction ).
function(3, background ).
function(4, background ).
function(5, background ).
function(6, background ).
function(7, background ).
function(8, background ).
function(9, background ).
function(10, background ).
function(11, background ).
function(12, methodology ).
function(13, background ).
function(14, background ).
function(15, background ).
function(16, background ).
function(17, background ).
function(18, background ).
function(19, background ).
function(20, methodology ).
function(20, methodology ).
function(21, background ).
function(22, background ).
function(23, background ).
function(24, background ).
function(25, background ).
function(26, background ).
function(27, background ).
function(28, background ).
function(29, background ).
function(30, background ).
function(31, methodology ).
function(32, background ).
function(33, background ).
function(34, background ).
function(35, background ).
function(36, background ).
function(37, background ).
function(38, background ).
function(39, background ).
function(40, background ).
function(41, background ).
function(42, background ).
function(43, background ).
function(44, background ).
function(45, background ).
function(46, background ).
function(47, background ).
function(48, background ).
function(49, background ).
function(50, background ).
function(51, background ).
function(52, background ).
function(53, background ).
function(54, background ).
function(55, background ).
function(56, background ).
function(57, background ).
function(58, background ).
function(59, background ).
function(60, background ).
function(61, background ).
function(62, background ).
function(63, background ).
function(64, background ).
function(65, background ).
function(66, background ).
function(67, background ).
function(68, background ).
function(69, background ).
function(70, background ).
function(71, background ).
function(72, background ).
function(73, background ).
function(74, background ).
function(75, background ).
function(76, background ).
function(77, methodology ).
function(78, background ).
function(79, background ).
function(80, background ).
function(81, background ).
function(82, background ).
function(83, background ).
function(84, background ).
function(85, background ).
function(86, background ).
function(87, background ).
function(88, background ).
function(89, background ).
function(90, background ).
function(91, background ).
function(92, background ).
function(93, background ).
function(94, background ).
function(95, background ).
function(96, background ).
function(97, background ).
function(98, background ).
function(99, background ).
function(100, background ).
function(101, background ).
function(102, background ).
function(103, background ).
function(104, background ).
function(105, background ).
function(106, background ).
function(107, background ).
function(108, background ).
function(109, background ).
function(110, background ).
function(111, background ).
function(112, background ).
function(113, background ).
function(114, background ).
function(115, background ).
function(116, background ).
function(117, background ).
function(118, background ).
function(119, background ).
function(120, background ).
function(121, background ).
function(122, background ).
function(123, background ).
function(124, background ).
function(125, background ).
function(126, background ).
function(127, background ).
function(128, background ).
```

```
function(129, conclusion ).
function(130, background ).
function(131, background ).
function(132, background ).
function(133, background ).
function(134, background ).
function(135, methodology ).
function(136, background ).
function(137, background ).
function(138, background ).
function(139, background ).
function(140, background ).
function(141, background ).
function(142, background ).
function(143, background ).
function(144, background ).
function(145, background ).
```

```
/* text5cpt.pl */
/* prolog database produced by txtcpt.spt */
/* concept(concept identity, concept) */
concept( 1, [ fault, diagnosis] ).
concept( 2, [ responsibility] ).
concept( 3, [  introduction] ).
concept( 4, [    testing] ).
concept( 5, [ fault, diagnosis, of, printed, circuit,  cards] ).
concept( 6, [ very, important, task] ).
concept( 7, [  times, each, day] ).
concept( 8, [ performed] ).
concept( 9, [  pre-determined] ).
concept( 10, [  static] ).
concept( 11, [ structured, tests] ).
concept( 12, [  result] ).
concept( 13, [ testing, tends] ).
concept( 14, [ inefficient] ).
concept( 15, [  missing] ).
concept( 16, [  faulted,  components] ).
concept( 17, [  isolate, faults] ).
concept( 18, [ large, group, of, components] ).
concept( 19, [ introduce] ).
concept( 20, [ 'theory, of, responsibilities'] ).
concept( 21, [ approach] ).
concept( 22, [  automated,  troubleshooting] ).
concept( 23, [  approach] ).
concept( 24, [  understanding, of, how] ).
concept( 25, [ circuit,  works] ).
concept( 26, [ recorded] ).
concept( 27, [ assigning, responsibilities] ).
concept( 28, [  parts, of, the, output, waveform] ).
concept( 29, [ subsections, of, the,  circuit] ).
concept( 30, [ assigned] ).
concept( 31, [ ro, derived] ).
concept( 32, [ casual, simulation] ).
concept( 33, [ current, efforts] ).
concept( 34, [ directed] ).
concept( 35, [ automated,  troubleshooting, of, analog, circuit, cards] ).
concept( 36, [ overall,  system] ).
concept( 37, [  described] ).
concept( 38, [ milne] ).
concept( 39, [ ramsey] ).
concept( 40, [ designed] ).
concept( 41, [ test] ).
concept( 42, [identify] ).
concept( 43, [  faults] ).
concept( 44, [  analog,  card] ).
concept( 45, [  use,  of,  automatic, test, equipment] ).
concept( 46, [  past,   implementation] ).
concept( 47, [ papers] ).
concept( 48, [ conducted, testing, based] ).
concept( 49, [  structural, description, of, the, circuit] ).
concept( 50, [  paper] ).
concept( 51, [ approach] ).
concept( 52, [ functional, testing] ).
concept( 53, [  described] ).
concept( 54, [  work,  differs] ).
concept( 55, [  others] ).
concept( 56, [  several,  significant, ways] ).
concept( 57, [ work, of] ).
concept( 58, [ cantone] ).
concept( 59, [ structural, area] ).
concept( 60, [  algorithm] ).
concept( 61, [ deciding] ).
concept( 62, [ test] ).
```

```
concept( 63, [ perform, based] ).
concept( 64, [ information, gained] ).
concept( 65, [ possible, cost] ).
concept( 66, [  very, helpful] ).
concept( 67, [ structural, reasoning] ).
concept( 68, [ can't, help] ).
concept( 69, [ no, longer, probe] ).
concept( 70, [ subcircuit] ).
concept( 71, [  cantone] ).
concept( 72, [ uses, structural, information] ).
concept( 73, [ isolate] ).
concept( 74, [ fault] ).
concept( 75, [ single, functional, area] ).
concept( 76, [ paper] ).
concept( 77, [  we] ).
concept( 78, [show] ).
concept( 79, [ structural,  information] ).
concept( 80, [ used] ).
concept( 81, [ further, propose, faults] ).
concept( 82, [ de, kleer] ).
concept( 83, [ working] ).
concept( 84, [ diagnose, faults] ).
concept( 85, [  analog, circuit, cards] ).
concept( 86, [ 'first, principles'] ).
concept( 87, [  given] ).
concept( 88, [ low, level, electronic, description, of,  how] ).
concept( 89, [  capacitor,  works] ).
concept( 90, [ possible] ).
concept( 91, [  deduce] ).
concept( 92, [ filter] ).
concept( 93, [ work] ).
concept( 94, [  diagnose,  faults] ).
concept( 95, [ agreed] ).
concept( 96, [  desirable,  approach] ).
concept( 97, [  author,  feels] ).
concept( 98, [ work, still] ).
concept( 99, [ order] ).
concept( 100, [ use] ).
concept( 101, [ approach] ).
concept( 102, [ testing,  today] ).
concept( 103, [ work] ).
concept( 104, [ start] ).
concept( 105, [ 'second, principles'] ).
concept( 106, [ type, of, description] ).
concept( 107, [ electronics,  engineer, uses] ).
concept( 108, [ describes, various, building, blocks,  of, circuits] ).
concept( 109, [  several, examples] ).
concept( 110, [ contained] ).
concept( 111, [ davis] ).
concept( 112, [ work] ).
concept( 113, [ area, of,  digital, troubleshooting, based] ).
concept( 114, [ function, of, the,  components] ).
concept( 115, [ relies] ).
concept( 116, [ computing] ).
concept( 117, [ function] ).
concept( 118, [  inverse, of, each, sub-component] ).
concept( 119, [ general] ).
concept( 120, [ not,  possible] ).
concept( 121, [ compute] ).
concept( 122, [ inverse, of, analog, functions] ).
concept( 123, [ digital, domain] ).
concept( 124, [ very, simple, output,  form] ).
concept( 125, [  while] ).
concept( 126, [ analog, domain] ).
concept( 127, [  very, rich, signal] ).
concept( 128, [  hence, different, techniques] ).
```

```
concept( 130, [ fact] ).
concept( 131, [ more, complex, output, signal] ).
concept( 132, [ one, of, the, key, differences] ).
concept( 133, [ analog] ).
concept( 134, [ digital, domains] ).
concept( 135, [ work, of] ).
concept( 136, [ chandrasekaran] ).
concept( 137, [ own] ).
concept( 138, [ describes] ).
concept( 139, [ object] ).
concept( 140, [ diagnosed] ).
concept( 141, [ formal, language] ).
concept( 142, [ compiles] ).
concept( 143, [  description] ).
concept( 144, [ set, of, production, rules] ).
concept( 145, [  perform] ).
concept( 146, [ diagnosis] ).
concept( 147, [ approach] ).
concept( 148, [ use] ).
concept( 149, [  different, formal, description] ).
concept( 150, [ four, simple, rules,  of, diagnosis] ).
concept( 151, [ compiling] ).
concept( 152, [ system] ).
concept( 153, [  production, rules] ).
concept( 154, [  chandrasekaran's, work] ).
concept( 155, [ not,  defined] ).
concept( 156, [ clear, role] ).
concept( 157, [ interaction, of,  functionai] ).
concept( 158, [  structural, reasoning] ).
concept( 159, [ work] ).
concept( 160, [  structural, reasoning] ).
concept( 161, [plays] ).
concept( 162, [ dual, role] ).
concept( 163, [  used] ).
concept( 164, [ isolate] ).
concept( 165, [ possible, fault] ).
concept( 166, [ single, functional, area] ).
concept( 167, [ output] ).
concept( 168, [  zero] ).
concept( 169, [ giving, no, information] ).
concept( 170, [  structural, rules] ).
concept( 171, [  used] ).
concept( 172, [ propose, possible, faults] ).
concept( 173, [   qualitative, reasoning] ).
concept( 174, [ forbus] ).
concept( 175, [ important] ).
concept( 176, [ justify] ).
concept( 177, [ of, our, rules] ).
concept( 178, [ approaches] ).
concept( 179, [ working] ).
concept( 180, [ second, principles] ).
concept( 181, [ of,  these, results] ).
concept( 182, [ compiled] ).
concept( 183, [ descriptions] ).
concept( 184, [ structure] ).
concept( 185, [ traditional, role, of, structure] ).
concept( 186, [ isolate] ).
concept( 187, [ possible, fault] ).
concept( 188, [ one, functional, module] ).
concept( 189, [  system] ).
concept( 190, [ checks, if] ).
concept( 191, [ output] ).
concept( 192, [ correct] ).
concept( 193, [ not] ).
concept( 194, [ path, of, the, signal] ).
```

concept(196, [structure, of, the, circuit]).
concept(197, [test]).
concept(198, [chosen]).
concept(199, [split]).
concept(200, [possible, fault, path]).
concept(201, [half]).
concept(202, [one, functional, module]).
concept(203, [left]).
concept(204, [work]).
concept(205, [structure]).
concept(206, [used]).
concept(207, [output]).
concept(208, [zero]).
concept(209, [situation]).
concept(210, [no, information]).
concept(211, [base]).
concept(212, [functional, diagnosis]).
concept(213, [structural, reasoning]).
concept(214, [used]).
concept(215, [analog, circuit]).
concept(216, [interested]).
concept(217, [output, current]).
concept(218, [voltage, drop]).
concept(219, [relationship]).
concept(220, [controlled]).
concept(221, [ohm's, law]).
concept(222, [e,=ir]).
concept(223, [e]).
concept(224, [zero]).
concept(225, [zero]).
concept(226, [zero]).
concept(227, [e]).
concept(228, [zero]).
concept(229, [know]).
concept(230, [r]).
concept(231, [zero]).
concept(232, [e]).
concept(233, [zero]).
concept(234, [of, the, product, of, ir]).
concept(235, [e]).
concept(236, [zero]).
concept(237, [need, two, rules]).
concept(238, [r]).
concept(239, [zero]).
concept(240, [let]).
concept(241, [look]).
concept(242, [voltage]).
concept(243, [know]).
concept(244, [need, an, r]).
concept(245, [order]).
concept(246, [get]).
concept(247, [voltage, drop]).
concept(248, [output]).
concept(249, [shorted]).
concept(250, [ground]).
concept(251, [r]).
concept(252, [zero]).
concept(253, [hence, no, output, voltage]).
concept(254, [translate]).
concept(255, [diagnosis, rule]).
concept(256, [voltage, short, rule]).
concept(257, [one, component, connects]).
concept(258, [output]).
concept(259, [ground]).
concept(260, [output]).

```
concept( 262, [ component] ).
concept( 263, [ shorted] ).
concept( 264, [ derive] ).
concept( 265, [ rule] ).
concept( 266, [ e,=ir] ).
concept( 267, [ 'compiled'] ).
concept( 268, [ form] ).
concept( 269, [  compilation] ).
concept( 270, [ chandra] ).
concept( 271, [ needs] ).
concept( 272, [ fault, diagnosis, system] ).
concept( 273, [ rule] ).
concept( 274, [ example, of, the, high, level, knowledge] ).
concept( 275, [  electronics, engineer] ).
concept( 276, [  use] ).
concept( 277, [  call] ).
concept( 278, [  example, of, a, 'second, principle'] ).
concept( 279, [ krichoff's, current, law] ).
concept( 280, [ know] ).
concept( 281, [  current, flowing] ).
concept( 282, [ of, a, node] ).
concept( 283, [ zero] ).
concept( 284, [ no, current, flowing] ).
concept( 285, [ node] ).
concept( 286, [ no, current, flowing] ).
concept( 287, [ of, the, node] ).
concept( 288, [ compile] ).
concept( 289, [ facts] ).
concept( 290, [ 'second,  principle'] ).
concept( 291, [  current, open, rule] ).
concept( 292, [ one, component, connects] ).
concept( 293, [ input] ).
concept( 294, [ output] ).
concept( 295, [ output] ).
concept( 296, [ zero] ).
concept( 297, [ component] ).
concept( 298, [ open] ).
concept( 299, [ know] ).
concept( 300, [ no, current] ).
concept( 301, [ not,  get] ).
concept( 302, [ voltage, drop] ).
concept( 303, [ output] ).
concept( 304, [ zero] ).
concept( 305, [ no, r] ).
concept( 306, [ no] ).
concept( 307, [  based, upon, ohm's, law] ).
concept( 308, [  of, the, dependency] ).
concept( 309, [ current] ).
concept( 310, [ voltage] ).
concept( 311, [ cannot] ).
concept( 312, [ sure] ).
concept( 313, [  based, upon] ).
concept( 314, [ single, zero, output] ).
concept( 315, [  short] ).
concept( 316, [  open] ).
concept( 317, [  several,  hypotheses] ).
concept( 318, [ produced] ).
concept( 319, [ use] ).
concept( 320, [ simple, voltage, divider] ).
concept( 321, [  example, of, these, structure, rules] ).
concept( 322, [ circuit] ).
concept( 323, [ working] ).
concept( 324, [ outputs] ).
concept( 325, [ respective, values] ).
concept( 326, [ determined] ).
```

```
concept( 328, [ r] ).
concept( 329, [ value, of, 02 is, wrong] ).
concept( 330, [ correct] ).
concept( 331, [ ratio, r,1/r] ).
concept( 332, [ wrong] ).
concept( 333, [ cannot, tell] ).
concept( 334, [ value] ).
concept( 335, [ wrong] ).
concept( 336, [ r] ).
concept( 337, [ short] ).
concept( 338, [ equal] ).
concept( 339, [ not] ).
concept( 340, [ zero] ).
concept( 341, [ predicted] ).
concept( 342, [ application, of, the, voltage, short, rule] ).
concept( 343, [ note] ).
concept( 344, [ ground] ).
concept( 345, [  zero] ).
concept( 346, [ special, case] ).
concept( 347, [ r] ).
concept( 348, [ open] ).
concept( 349, [ zero] ).
concept( 350, [ current, open, rule] ).
concept( 351, [ not] ).
concept( 352, [ zero, depending, upon] ).
concept( 353, [ circuit] ).
concept( 354, [ r] ).
concept( 355, [ shorted] ).
concept( 356, [ zero] ).
concept( 357, [ voltage,  short,  rule] ).
concept( 358, [ r] ).
concept( 359, [ open] ).
concept( 360, [ zero] ).
concept( 361, [ current, open, rule] ).
concept( 362, [ section] ).
concept( 363, [ presented] ).
concept( 364, [ approach] ).
concept( 365, [ use, of, structure] ).
concept( 366, [ fault, diagnosis] ).
concept( 367, [  traditional, way] ).
concept( 368, [ used] ).
concept( 369, [ isolate] ).
concept( 370, [ fault] ).
concept( 371, [ a,  single, functional, module] ).
concept( 372, [ use] ).
concept( 373, [ propose,  faults, based, upon, shorts] ).
concept( 374, [ opens] ).
concept( 375, [ two, rules] ).
concept( 376, [ very, powerful] ).
concept( 377, [ propose] ).
concept( 378, [ correct,  fault] ).
concept( 379, [ of, the, examples] ).
concept( 380, [ encountered] ).
concept( 381, [ function] ).
concept( 382, [ rectifier, circuit] ).
concept( 383, [ structural, rules] ).
concept( 384, [  isolated] ).
concept( 385, [ fault] ).
concept( 386, [ single, functional, module] ).
concept( 387, [ turn] ).
concept( 388, [  functional, reasoning] ).
concept( 389, [ illustrate] ).
concept( 390, [ theory, of,  responsibilities, presented] ).
concept( 391, [ paper] ).
concept( 392, [ simple,  rectifier,  circuit] ).
```

```
concept( 394, [  second,  principles] ).
concept( 395, [  rectifier, circuit] ).
concept( 396, [ described] ).
concept( 397, [ follows] ).
concept( 398, [ build] ).
concept( 399, [ rectifier, circuit] ).
concept( 400, [  use] ).
concept( 401, [  diode] ).
concept( 402, [ convert, each, peak] ).
concept( 403, [ input, waveform] ).
concept( 404, [ positive, output, peak] ).
concept( 405, [  typical,  sine,  wave] ).
concept( 406, [  this, means, two, diodes] ).
concept( 407, [  one] ).
concept( 408, [ positive] ).
concept( 409, [ one] ).
concept( 410, [ negative, peak] ).
concept( 411, [ output] ).
concept( 412, [ filtered] ).
concept( 413, [ capacitor] ).
concept( 414, [ used] ).
concept( 415, [ store, energy] ).
concept( 416, [  resistor] ).
concept( 417, [ drain] ).
concept( 418, [ energy] ).
concept( 419, [ description, responsibilities] ).
concept( 420, [  assigned] ).
concept( 421, [   each, diode, produces] ).
concept( 422, [ peak] ).
concept( 423, [ output] ).
concept( 424, [  one] ).
concept( 425, [ positive, peak] ).
concept( 426, [ one] ).
concept( 427, [ negative,  peak] ).
concept( 428, [  capacitor, charges] ).
concept( 429, [  giving] ).
concept( 430, [ rising,  ramp] ).
concept( 431, [ resistor, discharges] ).
concept( 432, [ ramp] ).
concept( 433, [  write] ).
concept( 434, [ circuit, rectifier] ).
concept( 435, [ varies] ).
concept( 436, [ time, t,0-t] ).
concept( 437, [            t,0-t,1   output] ).
concept( 438, [       peak] ).
concept( 439, [ positive-max] ).
concept( 440, [ diode,1            t,1-t,2   output] ).
concept( 441, [        peak] ).
concept( 442, [ negative-max] ).
concept( 443, [ diode] ).
concept( 444, [ sub-circuit, peak] ).
concept( 445, [ varies] ).
concept( 446, [ time, t,0-t] ).
concept( 447, [            t,0-t,1  output] ).
concept( 448, [        rising, ramp] ).
concept( 449, [ capacitor,          t,1       output] ).
concept( 450, [       peak] ).
concept( 451, [ diodex,           t,1-t,2   output] ).
concept( 452, [        falling, ramp] ).
concept( 453, [ resistor,           t] ).
concept( 454, [     input-max] ).
concept( 455, [ diagnosis, rules] ).
concept( 456, [ x] ).
concept( 457, [ y] ).
concept( 458, [ z] ).
```

```
concept( 460, [ y] ).
concept( 461, [ z] ).
concept( 462, [ bad] ).
concept( 463, [ x] ).
concept( 464, [ y] ).
concept( 465, [ z] ).
concept( 466, [ not] ).
concept( 467, [ x] ).
concept( 468, [ use, structure, rules] ).
concept( 469, [ brief, paper] ).
concept( 470, [ implementation, details] ).
concept( 471, [ left] ).
concept( 472, [ prolog, user] ).
concept( 473, [  recognise] ).
concept( 474, [ role] ).
concept( 475, [ unification] ).
concept( 476, [ troubleshooting, rules] ).
concept( 477, [ need] ).
concept( 478, [ diagnose] ).
concept( 479, [ fault] ).
concept( 480, [ output, waveform] ).
concept( 481, [ compared] ).
concept( 482, [ input, waveform] ).
concept( 483, [ rule] ).
concept( 484, [identify] ).
concept( 485, [ component] ).
concept( 486, [ faulted] ).
concept( 487, [ description] ).
concept( 488, [ tested] ).
concept( 489, [ actual,  circuit] ).
concept( 490, [ resistor] ).
concept( 491, [ faulted] ).
concept( 492, [ open,  position] ).
concept( 493, [ output] ).
concept( 494, [ constant, level] ).
concept( 495, [  comparing] ).
concept( 496, [ desired, output] ).
concept( 497, [ real, output] ).
concept( 498, [ missing] ).
concept( 499, [ declining, ramp] ).
concept( 500, [ resistor] ).
concept( 501, [ bad] ).
concept( 502, [ dc, voltage, levels] ).
concept( 503, [ important] ).
concept( 504, [  diagnosis] ).
concept( 505, [ omitted] ).
concept( 506, [ paper] ).
concept( 507, [ one, of, the, diodes] ).
concept( 508, [ opened] ).
concept( 509, [ output] ).
concept( 510, [  missing, one, of, its, peaks] ).
concept( 511, [ comparing] ).
concept( 512, [ desired, output] ).
concept( 513, [ real, output] ).
concept( 514, [ input] ).
concept( 515, [ rule] ).
concept( 516, [identify] ).
concept( 517, [ diode] ).
concept( 518, [ opened] ).
concept( 519, [ capacitor] ).
concept( 520, [ opened] ).
concept( 521, [ rising,  output] ).
concept( 522, [  not] ).
concept( 523, [  ramp] ).
concept( 524, [ capacitor] ).
```

```
concept( 526, [ resistor] ).
concept( 527, [ shorted] ).
concept( 528, [ output] ).
concept( 529, [ zero] ).
concept( 530, [  structure, rules] ).
concept( 531, [ propose] ).
concept( 532, [ fault] ).
concept( 533, [ note] ).
concept( 534, [ lucky] ).
concept( 535, [ case] ).
concept( 536, [ each, component, manifests] ).
concept( 537, [ single, part, of,  the,  output,  waveform] ).
concept( 538, [  brief,  paper] ).
concept( 539, [  voltage, levels] ).
concept( 540, [ omitted] ).
concept( 541, [ critical] ).
concept( 542, [ diagnose] ).
concept( 543, [ faults] ).
concept( 544, [ deep, functional, reasoning] ).
concept( 545, [ sections] ).
concept( 546, [ assumed] ).
concept( 547, [  understand] ).
concept( 548, [ assign, responsibilities] ).
concept( 549, [ parts, of, the, output, waveform] ).
concept( 550, [ components,  of,  the,  circuit] ).
concept( 551, [ section] ).
concept( 552, [ derived] ).
concept( 553, [ basic, strategy] ).
concept( 554, [ simulate] ).
concept( 555, [ working, of,  the,  circuit] ).
concept( 556, [ 'second, principles'] ).
concept( 557, [  each, component] ).
concept( 558, [makes] ).
concept( 559, [ contribution] ).
concept( 560, [ overall,  output] ).
concept( 561, [ responsibility] ).
concept( 562, [ assigned] ).
concept( 563, [ use] ).
concept( 564, [ troubleshooting, technique] ).
concept( 565, [ trace] ).
concept( 566, [  responsibilities] ).
concept( 567, [ faulted, component] ).
concept( 568, [  use] ).
concept( 569, [  rectifier,  circuit] ).
concept( 570, [  illustration] ).
concept( 571, [ start] ).
concept( 572, [ following, second,  principles] ).
concept( 573, [  diode] ).
concept( 574, [ input] ).
concept( 575, [ positive] ).
concept( 576, [ output, equals] ).
concept( 577, [ input] ).
concept( 578, [  capacitor] ).
concept( 579, [ capacitor] ).
concept( 580, [ rising, input] ).
concept( 581, [ charge] ).
concept( 582, [  capacitor] ).
concept( 583, [ charged, capacitor] ).
concept( 584, [ load] ).
concept( 585, [ not] ).
concept( 586, [ rising, input] ).
concept( 587, [ discharge] ).
concept( 588, [  wire] ).
concept( 589, [ two, wires, meet] ).
concept( 590, [ output] ).
```

```
concept( 592, [assume] ).
concept( 593, [ low, level, rules] ).
concept( 594, [provide] ).
concept( 595, [ addition, of, signal, waveforms] ).
concept( 596, [  testing] ).
concept( 597, [  rising] ).
concept( 598, [  falling,  voltage,  levels] ).
concept( 599, [ give] ).
concept( 600, [  voltage] ).
concept( 601, [  capacitor] ).
concept( 602, [  input] ).
concept( 603, [  clipper] ).
concept( 604, [ form, of, a, sine, wave] ).
concept( 605, [  transformer] ).
concept( 606, [ diode, rule] ).
concept( 607, [ sine,  wave] ).
concept( 608, [    transformed] ).
concept( 609, [  postive,  half-wave,  followed] ).
concept( 610, [ zero] ).
concept( 611, [ second, diode] ).
concept( 612, [ reverse,  polarity] ).
concept( 613, [ produces] ).
concept( 614, [ zero, output, followed] ).
concept( 615, [  half-wave] ).
concept( 616, [ responsibility] ).
concept( 617, [ each, half-wave] ).
concept( 618, [  assigned] ).
concept( 619, [    next] ).
concept( 620, [  two,  wires,  add] ).
concept( 621, [  giving] ).
concept( 622, [  half-wave, rectified, waveform] ).
concept( 623, [ signal, arrives] ).
concept( 624, [ capacitor] ).
concept( 625, [  input, signal] ).
concept( 626, [ rising] ).
concept( 627, [ capacitor, rule] ).
concept( 628, [  capacitor,  charges] ).
concept( 629, [  rise] ).
concept( 630, [  output,  voltage] ).
concept( 631, [ assigned] ).
concept( 632, [ capacitor] ).
concept( 633, [ signal,  stops,  rising] ).
concept( 634, [  capactitor] ).
concept( 635, [    load] ).
concept( 636, [  discharges] ).
concept( 637, [ decreasing, output, voltage] ).
concept( 638, [  assigned] ).
concept( 639, [ resistor] ).
concept( 640, [ very, brief, explanation] ).
concept( 641, [ seen] ).
concept( 642, [ waveform, gets, built] ).
concept( 643, [ second, principles] ).
concept( 644, [ responsibilities] ).
concept( 645, [ assigned] ).
concept( 646, [  technique] ).
concept( 647, [ possible] ).
concept( 648, [ derive] ).
concept( 649, [  responsiblities] ).
concept( 650, [ circuit] ).
concept( 651, [ perform,  troubleshooting, based, upon] ).
concept( 652, [ conclusion] ).
concept( 653, [ paper] ).
concept( 654, [ 'theory, of, responsibilities'] ).
concept( 655, [ outlined] ).
concept( 656, [ states] ).
```

```
concept( 658, [ assigning, responsibilities] ).
concept( 659, [  parts, of, the, output] ).
concept( 660, [ various, parts, of, the, circuit] ).
concept( 661, [ responsibilities] ).
concept( 662, [ derived] ).
concept( 663, [ casual,  simulation, of, the, function, of, the, circuit] ).
concept( 664, [ demonstrated] ).
concept( 665, [ works] ).
concept( 666, [ typical,  building, block, circuit] ).
concept( 667, [ work] ).
concept( 668, [ started] ).
concept( 669, [ 'second,  principles'] ).
concept( 670, [ principles] ).
concept( 671, [ electronic, engineer] ).
concept( 672, [ use] ).
concept( 673, [ describes,  each, sub-circuit] ).
concept( 674, [ approach] ).
concept( 675, [ proved] ).
concept( 676, [  very, effective] ).
concept( 677, [ diagnosing, faults] ).
concept( 678, [ analog,  circuits] ).
concept( 679, [ one] ).
concept( 680, [ limited, understanding, of, the,  circuit] ).
concept( 681, [ responsibilities] ).
concept( 682, [ assigned] ).
concept( 683, [ limited, way] ).
concept( 684, [ hence] ).
concept( 685, [ capability] ).
concept( 686, [  diagnose, faults] ).
concept( 687, [  limited] ).
concept( 688, [  one] ).
concept( 689, [  thorough, understanding, of, the, circuit] ).
concept( 690, [  better,  job,  of,  assigning] ).
concept( 691, [    responsibilities] ).
concept( 692, [ of, limitations] ).
concept( 693, [ domain] ).
concept( 694, [ not, always] ).
concept( 695, [ possible] ).
concept( 696, [describe] ).
concept( 697, [  circuit] ).
concept( 698, [ way] ).
concept( 699, [ isolate, faults] ).
concept( 700, [ one,  component] ).
concept( 701, [ limitation] ).
concept( 702, [  primarily] ).
concept( 703, [comes] ).
concept( 704, [ limits, of, being, able] ).
concept( 705, [ diagnose] ).
concept( 706, [  circuit] ).
concept( 707, [ black, box] ).
concept( 708, [ technique] ).
concept( 709, [  applied] ).
concept( 710, [ filters] ).
concept( 711, [  rectifiers] ).
concept( 712, [  voltage, doublers] ).
concept( 713, [  wave, generators] ).
concept( 714, [ of, the, modularity, of, most,  circuits] ).
concept( 715, [ works] ).
concept( 716, [ complex,  circuits] ).
concept( 717, [ dealing] ).
concept( 718, [ sub-circuits] ).
concept( 719, [ technique] ).
concept( 720, [ very, powerful] ).
concept( 721, [ possible] ).
concept( 722, [ derive] ).
```

```
concept( 724, [ principles] ).
concept( 725, [ found] ).
concept( 726, [ prove] ).
concept( 727, [ second, principles] ).
concept( 728, [ way] ).
concept( 729, [ can't] ).
concept( 730, [ yet] ).
concept( 731, [ more, detailed, description, of, this, work] ).
concept( 732, [ milne] ).
/* heading(no,heading) */
heading( 0,' FAULT DIAGNOSIS THROUGH RESPONSIBILITY ' ).
heading( 1,' INTRODUCTION ' ).
heading( 2,' STRUCTURE ' ).
heading( 3,' FUNCTION ' ).
heading( 4,' DEEP FUNCTIONAL REASONING ' ).
heading( 5,' CONCLUSION ' ).
/* summary ( sentence no., no. of concepts identified ) */
summary( 1,2).
summary( 2,1).
summary( 3,4).
summary( 4,4).
summary( 5,7).
summary( 6,4).
summary( 7,7).
summary( 8,3).
summary( 9,3).
summary( 10,10).
summary( 11,4).
summary( 12,4).
summary( 13,3).
summary( 14,3).
summary( 15,11).
summary( 16,5).
summary( 17,6).
summary( 18,5).
summary( 19,8).
summary( 20,8).
summary( 21,6).
summary( 22,2).
summary( 23,4).
summary( 24,4).
summary( 25,4).
summary( 26,5).
summary( 27,2).
summary( 28,5).
summary( 29,3).
summary( 30,9).
summary( 31,7).
summary( 32,5).
summary( 33,4).
summary( 34,4).
summary( 35,6).
summary( 36,11).
summary( 37,1).
summary( 38,4).
summary( 39,4).
summary( 40,9).
summary( 41,2).
summary( 42,5).
summary( 43,6).
summary( 44,4).
summary( 45,4).
summary( 46,6).
summary( 47,5).
summary( 48,6).
```

summary(50,5).
summary(51,6).
summary(52,2).
summary(53,8).
summary(54,5).
summary(55,4).
summary(56,4).
summary(57,2).
summary(58,5).
summary(59,4).
summary(60,3).
summary(61,8).
summary(62,4).
summary(63,5).
summary(64,11).
summary(65,3).
summary(66,3).
summary(67,4).
summary(68,4).
summary(69,3).
summary(70,5).
summary(71,2).
summary(72,4).
summary(73,4).
summary(74,3).
summary(75,4).
summary(76,4).
summary(77,5).
summary(78,5).
summary(79,3).
summary(80,6).
summary(81,1).
summary(82,7).
summary(83,5).
summary(84,4).
summary(85,7).
summary(86,6).
summary(87,2).
summary(88,6).
summary(89,9).
summary(90,5).
summary(91,22).
summary(92,14).
summary(93,7).
summary(94,2).
summary(95,5).
summary(96,4).
summary(97,3).
summary(98,5).
summary(99,7).
summary(100,5).
summary(101,4).
summary(102,8).
summary(103,7).
summary(104,7).
summary(105,5).
summary(106,3).
summary(107,3).
summary(108,1).
summary(109,6).
summary(110,2).
summary(111,4).
summary(112,6).
summary(113,5).
summary(114,3).

```
summary( 116,10).
summary( 117,4).
summary( 118,5).
summary( 119,5).
summary( 120,3).
summary( 121,4).
summary( 122,4).
summary( 123,2).
summary( 124,4).
summary( 125,4).
summary( 126,3).
summary( 127,6).
summary( 128,6).
summary( 129,1).
summary( 130,3).
summary( 131,5).
summary( 132,3).
summary( 133,3).
summary( 134,7).
summary( 135,5).
summary( 136,9).
summary( 137,4).
summary( 138,9).
summary( 139,7).
summary( 140,6).
summary( 141,5).
summary( 142,2).
summary( 143,4).
summary( 144,6).
summary( 145,2).
/* textstats (no.of sentences,no. of headings,no.of concepts,no.of words,no. of paragraphs ) */
textstats( 145, 6, 732, 2343, 32 ).
```

```
/* prolog database produced by the program called txtcpt.spt*/
/* the format of the database is as follows: */
/* (note: title set to heading no = 0) */
/* textconcept(heading no, paragragh no,paragraphsentence no, sentence no, concept no, concept identit
textconcept( 1,0,0,1,1,  1 ).
textconcept( 1,0,0,1,2,  2 ).
textconcept( 2,0,0,2,1,  3 ).
textconcept( 2,1,1,3,1,  4 ).
textconcept( 2,1,1,3,2,  5 ).
textconcept( 2,1,1,3,3,  6 ).
textconcept( 2,1,1,3,4,  7 ).
textconcept( 2,1,2,4,1,  8 ).
textconcept( 2,1,2,4,2,  9 ).
textconcept( 2,1,2,4,3,  10 ).
textconcept( 2,1,2,4,4,  11 ).
textconcept( 2,1,3,5,1,  12 ).
textconcept( 2,1,3,5,2,  13 ).
textconcept( 2,1,3,5,3,  14 ).
textconcept( 2,1,3,5,4,  15 ).
textconcept( 2,1,3,5,5,  16 ).
textconcept( 2,1,3,5,6,  17 ).
textconcept( 2,1,3,5,7,  18 ).
textconcept( 2,2,1,6,1,  19 ).
textconcept( 2,2,1,6,2,  20 ).
textconcept( 2,2,1,6,3,  21 ).
textconcept( 2,2,1,6,4,  22 ).
textconcept( 2,2,2,7,1,  23 ).
textconcept( 2,2,2,7,2,  24 ).
textconcept( 2,2,2,7,3,  25 ).
textconcept( 2,2,2,7,4,  26 ).
textconcept( 2,2,2,7,5,  27 ).
textconcept( 2,2,2,7,6,  28 ).
textconcept( 2,2,2,7,7,  29 ).
textconcept( 2,2,3,8,1,  30 ).
textconcept( 2,2,3,8,2,  31 ).
textconcept( 2,2,3,8,3,  32 ).
textconcept( 2,3,1,9,1,  33 ).
textconcept( 2,3,1,9,2,  34 ).
textconcept( 2,3,1,9,3,  35 ).
textconcept( 2,3,2,10,1,  36 ).
textconcept( 2,3,2,10,2,  37 ).
textconcept( 2,3,2,10,3,  38 ).
textconcept( 2,3,2,10,4,  39 ).
textconcept( 2,3,2,10,5,  40 ).
textconcept( 2,3,2,10,6,  41 ).
textconcept( 2,3,2,10,7,  42 ).
textconcept( 2,3,2,10,8,  43 ).
textconcept( 2,3,2,10,9,  44 ).
textconcept( 2,3,2,10,10,  45 ).
textconcept( 2,3,3,11,1,  46 ).
textconcept( 2,3,3,11,2,  47 ).
textconcept( 2,3,3,11,3,  48 ).
textconcept( 2,3,3,11,4,  49 ).
textconcept( 2,3,4,12,1,  50 ).
textconcept( 2,3,4,12,2,  51 ).
textconcept( 2,3,4,12,3,  52 ).
textconcept( 2,3,4,12,4,  53 ).
textconcept( 2,4,1,13,1,  54 ).
textconcept( 2,4,1,13,2,  55 ).
textconcept( 2,4,1,13,3,  56 ).
textconcept( 2,4,2,14,1,  57 ).
textconcept( 2,4,2,14,2,  58 ).
textconcept( 2,4,2,14,3,  59 ).
textconcept( 2,4,3,15,1,  60 ).
textconcept( 2,4,3,15,2,  61 ).
```

```
textconcept( 2,4,3,15,4,  63 ).
textconcept( 2,4,3,15,5,  64 ).
textconcept( 2,4,3,15,6,  65 ).
textconcept( 2,4,3,15,7,  66 ).
textconcept( 2,4,3,15,8,  67 ).
textconcept( 2,4,3,15,9,  68 ).
textconcept( 2,4,3,15,10,  69 ).
textconcept( 2,4,3,15,11,  70 ).
textconcept( 2,4,4,16,1,  71 ).
textconcept( 2,4,4,16,2,  72 ).
textconcept( 2,4,4,16,3,  73 ).
textconcept( 2,4,4,16,4,  74 ).
textconcept( 2,4,4,16,5,  75 ).
textconcept( 2,4,5,17,1,  76 ).
textconcept( 2,4,5,17,2,  77 ).
textconcept( 2,4,5,17,3,  78 ).
textconcept( 2,4,5,17,4,  79 ).
textconcept( 2,4,5,17,5,  80 ).
textconcept( 2,4,5,17,6,  81 ).
textconcept( 2,5,1,18,1,  82 ).
textconcept( 2,5,1,18,2,  83 ).
textconcept( 2,5,1,18,3,  84 ).
textconcept( 2,5,1,18,4,  85 ).
textconcept( 2,5,1,18,5,  86 ).
textconcept( 2,5,2,19,1,  87 ).
textconcept( 2,5,2,19,2,  88 ).
textconcept( 2,5,2,19,3,  89 ).
textconcept( 2,5,2,19,4,  90 ).
textconcept( 2,5,2,19,5,  91 ).
textconcept( 2,5,2,19,6,  92 ).
textconcept( 2,5,2,19,7,  93 ).
textconcept( 2,5,2,19,8,  94 ).
textconcept( 2,5,3,20,1,  95 ).
textconcept( 2,5,3,20,2,  96 ).
textconcept( 2,5,3,20,3,  97 ).
textconcept( 2,5,3,20,4,  98 ).
textconcept( 2,5,3,20,5,  99 ).
textconcept( 2,5,3,20,6,  100 ).
textconcept( 2,5,3,20,7,  101 ).
textconcept( 2,5,3,20,8,  102 ).
textconcept( 2,5,4,21,1,  103 ).
textconcept( 2,5,4,21,2,  104 ).
textconcept( 2,5,4,21,3,  105 ).
textconcept( 2,5,4,21,4,  106 ).
textconcept( 2,5,4,21,5,  107 ).
textconcept( 2,5,4,21,6,  108 ).
textconcept( 2,5,5,22,1,  109 ).
textconcept( 2,5,5,22,2,  110 ).
textconcept( 2,6,1,23,1,  111 ).
textconcept( 2,6,1,23,2,  112 ).
textconcept( 2,6,1,23,3,  113 ).
textconcept( 2,6,1,23,4,  114 ).
textconcept( 2,6,2,24,1,  115 ).
textconcept( 2,6,2,24,2,  116 ).
textconcept( 2,6,2,24,3,  117 ).
textconcept( 2,6,2,24,4,  118 ).
textconcept( 2,6,3,25,1,  119 ).
textconcept( 2,6,3,25,2,  120 ).
textconcept( 2,6,3,25,3,  121 ).
textconcept( 2,6,3,25,4,  122 ).
textconcept( 2,6,4,26,1,  123 ).
textconcept( 2,6,4,26,2,  124 ).
textconcept( 2,6,4,26,3,  125 ).
textconcept( 2,6,4,26,4,  126 ).
textconcept( 2,6,4,26,5,  127 ).
textconcept( 2,6,5,27,1,  128 ).
textconcept( 2,6,5,27,2,  129 ).
textconcept( 2,6,6,28,1,  130 ).
textconcept( 2,6,6,28,2,  131 ).
textconcept( 2,6,6,28,3,  132 ).
textconcept( 2,6,6,28,4,  133 ).
textconcept( 2,6,6,28,5,  134 ).
textconcept( 2,7,1,29,1,  135 ).
textconcept( 2,7,1,29,2,  136 ).
textconcept( 2,7,1,29,3,  137 ).
textconcept( 2,7,2,30,1,  138 ).
textconcept( 2,7,2,30,2,  139 ).
textconcept( 2,7,2,30,3,  140 ).
textconcept( 2,7,2,30,4,  141 ).
textconcept( 2,7,2,30,5,  142 ).
textconcept( 2,7,2,30,6,  143 ).
textconcept( 2,7,2,30,7,  144 ).
textconcept( 2,7,2,30,8,  145 ).
textconcept( 2,7,2,30,9,  146 ).
textconcept( 2,7,3,31,1,  147 ).
textconcept( 2,7,3,31,2,  148 ).
textconcept( 2,7,3,31,3,  149 ).
textconcept( 2,7,3,31,4,  150 ).
textconcept( 2,7,3,31,5,  151 ).
textconcept( 2,7,3,31,6,  152 ).
textconcept( 2,7,3,31,7,  153 ).
textconcept( 2,7,4,32,1,  154 ).
textconcept( 2,7,4,32,2,  155 ).
textconcept( 2,7,4,32,3,  156 ).
textconcept( 2,7,4,32,4,  157 ).
textconcept( 2,7,4,32,5,  158 ).
textconcept( 2,7,5,33,1,  159 ).
textconcept( 2,7,5,33,2,  160 ).
textconcept( 2,7,5,33,3,  161 ).
textconcept( 2,7,5,33,4,  162 ).
textconcept( 2,7,6,34,1,  163 ).
textconcept( 2,7,6,34,2,  164 ).
textconcept( 2,7,6,34,3,  165 ).
textconcept( 2,7,6,34,4,  166 ).
textconcept( 2,7,7,35,1,  167 ).
textconcept( 2,7,7,35,2,  168 ).
textconcept( 2,7,7,35,3,  169 ).
textconcept( 2,7,7,35,4,  170 ).
textconcept( 2,7,7,35,5,  171 ).
textconcept( 2,7,7,35,6,  172 ).
textconcept( 2,8,1,36,1,  173 ).
textconcept( 2,8,1,36,2,  174 ).
textconcept( 2,8,1,36,3,  175 ).
textconcept( 2,8,1,36,4,  176 ).
textconcept( 2,8,1,36,5,  177 ).
textconcept( 2,8,1,36,6,  178 ).
textconcept( 2,8,1,36,7,  179 ).
textconcept( 2,8,1,36,8,  180 ).
textconcept( 2,8,1,36,9,  181 ).
textconcept( 2,8,1,36,10,  182 ).
textconcept( 2,8,1,36,11,  183 ).
textconcept( 3,0,0,37,1,  184 ).
textconcept( 3,1,1,38,1,  185 ).
textconcept( 3,1,1,38,2,  186 ).
textconcept( 3,1,1,38,3,  187 ).
textconcept( 3,1,1,38,4,  188 ).
textconcept( 3,1,2,39,1,  189 ).
textconcept( 3,1,2,39,2,  190 ).
textconcept( 3,1,2,39,3,  191 ).
textconcept( 3,1,2,39,4,  192 ).
textconcept( 3,1,3,40,1,  193 ).
```

```
textconcept( 3,1,3,40,3, 195 ).
textconcept( 3,1,3,40,4, 196 ).
textconcept( 3,1,3,40,5, 197 ).
textconcept( 3,1,3,40,6, 198 ).
textconcept( 3,1,3,40,7, 199 ).
textconcept( 3,1,3,40,8, 200 ).
textconcept( 3,1,3,40,9, 201 ).
textconcept( 3,1,4,41,1, 202 ).
textconcept( 3,1,4,41,2, 203 ).
textconcept( 3,2,1,42,1, 204 ).
textconcept( 3,2,1,42,2, 205 ).
textconcept( 3,2,1,42,3, 206 ).
textconcept( 3,2,1,42,4, 207 ).
textconcept( 3,2,1,42,5, 208 ).
textconcept( 3,2,2,43,1, 209 ).
textconcept( 3,2,2,43,2, 210 ).
textconcept( 3,2,2,43,3, 211 ).
textconcept( 3,2,2,43,4, 212 ).
textconcept( 3,2,2,43,5, 213 ).
textconcept( 3,2,2,43,6, 214 ).
textconcept( 3,2,3,44,1, 215 ).
textconcept( 3,2,3,44,2, 216 ).
textconcept( 3,2,3,44,3, 217 ).
textconcept( 3,2,3,44,4, 218 ).
textconcept( 3,2,4,45,1, 219 ).
textconcept( 3,2,4,45,2, 220 ).
textconcept( 3,2,4,45,3, 221 ).
textconcept( 3,2,4,45,4, 222 ).
textconcept( 3,2,5,46,1, 223 ).
textconcept( 3,2,5,46,2, 224 ).
textconcept( 3,2,5,46,3, 225 ).
textconcept( 3,2,5,46,4, 226 ).
textconcept( 3,2,5,46,5, 227 ).
textconcept( 3,2,5,46,6, 228 ).
textconcept( 3,2,6,47,1, 229 ).
textconcept( 3,2,6,47,2, 230 ).
textconcept( 3,2,6,47,3, 231 ).
textconcept( 3,2,6,47,4, 232 ).
textconcept( 3,2,6,47,5, 233 ).
textconcept( 3,2,7,48,1, 234 ).
textconcept( 3,2,7,48,2, 235 ).
textconcept( 3,2,7,48,3, 236 ).
textconcept( 3,2,7,48,4, 237 ).
textconcept( 3,2,7,48,5, 238 ).
textconcept( 3,2,7,48,6, 239 ).
textconcept( 3,3,1,49,1, 240 ).
textconcept( 3,3,1,49,2, 241 ).
textconcept( 3,3,1,49,3, 242 ).
textconcept( 3,3,2,50,1, 243 ).
textconcept( 3,3,2,50,2, 244 ).
textconcept( 3,3,2,50,3, 245 ).
textconcept( 3,3,2,50,4, 246 ).
textconcept( 3,3,2,50,5, 247 ).
textconcept( 3,3,3,51,1, 248 ).
textconcept( 3,3,3,51,2, 249 ).
textconcept( 3,3,3,51,3, 250 ).
textconcept( 3,3,3,51,4, 251 ).
textconcept( 3,3,3,51,5, 252 ).
textconcept( 3,3,3,51,6, 253 ).
textconcept( 3,3,4,52,1, 254 ).
textconcept( 3,3,4,52,2, 255 ).
textconcept( 3,3,5,53,1, 256 ).
textconcept( 3,3,5,53,2, 257 ).
textconcept( 3,3,5,53,3, 258 ).
textconcept( 3,3,5,53,4, 259 ).
textconcept( 3,3,5,53,5, 260 ).
textconcept( 3,3,5,53,6, 261 ).
textconcept( 3,3,5,53,7, 262 ).
textconcept( 3,3,5,53,8, 263 ).
textconcept( 3,4,1,54,1, 264 ).
textconcept( 3,4,1,54,2, 265 ).
textconcept( 3,4,1,54,3, 266 ).
textconcept( 3,4,1,54,4, 267 ).
textconcept( 3,4,1,54,5, 268 ).
textconcept( 3,4,2,55,1, 269 ).
textconcept( 3,4,2,55,2, 270 ).
textconcept( 3,4,2,55,3, 271 ).
textconcept( 3,4,2,55,4, 272 ).
textconcept( 3,4,3,56,1, 273 ).
textconcept( 3,4,3,56,2, 274 ).
textconcept( 3,4,3,56,3, 275 ).
textconcept( 3,4,3,56,4, 276 ).
textconcept( 3,4,4,57,1, 277 ).
textconcept( 3,4,4,57,2, 278 ).
textconcept( 3,5,1,58,1, 279 ).
textconcept( 3,5,1,58,2, 280 ).
textconcept( 3,5,1,58,3, 281 ).
textconcept( 3,5,1,58,4, 282 ).
textconcept( 3,5,1,58,5, 283 ).
textconcept( 3,5,2,59,1, 284 ).
textconcept( 3,5,2,59,2, 285 ).
textconcept( 3,5,2,59,3, 286 ).
textconcept( 3,5,2,59,4, 287 ).
textconcept( 3,5,3,60,1, 288 ).
textconcept( 3,5,3,60,2, 289 ).
textconcept( 3,5,3,60,3, 290 ).
textconcept( 3,5,4,61,1, 291 ).
textconcept( 3,5,4,61,2, 292 ).
textconcept( 3,5,4,61,3, 293 ).
textconcept( 3,5,4,61,4, 294 ).
textconcept( 3,5,4,61,5, 295 ).
textconcept( 3,5,4,61,6, 296 ).
textconcept( 3,5,4,61,7, 297 ).
textconcept( 3,5,4,61,8, 298 ).
textconcept( 3,6,1,62,1, 299 ).
textconcept( 3,6,1,62,2, 300 ).
textconcept( 3,6,1,62,3, 301 ).
textconcept( 3,6,1,62,4, 302 ).
textconcept( 3,6,2,63,1, 303 ).
textconcept( 3,6,2,63,2, 304 ).
textconcept( 3,6,2,63,3, 305 ).
textconcept( 3,6,2,63,4, 306 ).
textconcept( 3,6,2,63,5, 307 ).
textconcept( 3,6,3,64,1, 308 ).
textconcept( 3,6,3,64,2, 309 ).
textconcept( 3,6,3,64,3, 310 ).
textconcept( 3,6,3,64,4, 311 ).
textconcept( 3,6,3,64,5, 312 ).
textconcept( 3,6,3,64,6, 313 ).
textconcept( 3,6,3,64,7, 314 ).
textconcept( 3,6,3,64,8, 315 ).
textconcept( 3,6,3,64,9, 316 ).
textconcept( 3,6,3,64,10, 317 ).
textconcept( 3,6,3,64,11, 318 ).
textconcept( 3,7,1,65,1, 319 ).
textconcept( 3,7,1,65,2, 320 ).
textconcept( 3,7,1,65,3, 321 ).
textconcept( 3,7,2,66,1, 322 ).
textconcept( 3,7,2,66,2, 323 ).
textconcept( 3,7,2,66,3, 324 ).
textconcept( 3,7,3,67,1, 325 ).
```

```
textconcept( 3,7,3,67,3, 327 ).
textconcept( 3,7,3,67,4, 328 ).
textconcept( 3,7,4,68,1, 329 ).
textconcept( 3,7,4,68,2, 330 ).
textconcept( 3,7,4,68,3, 331 ).
textconcept( 3,7,4,68,4, 332 ).
textconcept( 3,7,5,69,1, 333 ).
textconcept( 3,7,5,69,2, 334 ).
textconcept( 3,7,5,69,3, 335 ).
textconcept( 3,8,1,70,1, 336 ).
textconcept( 3,8,1,70,2, 337 ).
textconcept( 3,8,1,70,3, 338 ).
textconcept( 3,8,1,70,4, 339 ).
textconcept( 3,8,1,70,5, 340 ).
textconcept( 3,8,2,71,1, 341 ).
textconcept( 3,8,2,71,2, 342 ).
textconcept( 3,8,3,72,1, 343 ).
textconcept( 3,8,3,72,2, 344 ).
textconcept( 3,8,3,72,3, 345 ).
textconcept( 3,8,3,72,4, 346 ).
textconcept( 3,8,4,73,1, 347 ).
textconcept( 3,8,4,73,2, 348 ).
textconcept( 3,8,4,73,3, 349 ).
textconcept( 3,8,4,73,4, 350 ).
textconcept( 3,8,5,74,1, 351 ).
textconcept( 3,8,5,74,2, 352 ).
textconcept( 3,8,5,74,3, 353 ).
textconcept( 3,8,6,75,1, 354 ).
textconcept( 3,8,6,75,2, 355 ).
textconcept( 3,8,6,75,3, 356 ).
textconcept( 3,8,6,75,4, 357 ).
textconcept( 3,8,7,76,1, 358 ).
textconcept( 3,8,7,76,2, 359 ).
textconcept( 3,8,7,76,3, 360 ).
textconcept( 3,8,7,76,4, 361 ).
textconcept( 3,9,1,77,1, 362 ).
textconcept( 3,9,1,77,2, 363 ).
textconcept( 3,9,1,77,3, 364 ).
textconcept( 3,9,1,77,4, 365 ).
textconcept( 3,9,1,77,5, 366 ).
textconcept( 3,9,2,78,1, 367 ).
textconcept( 3,9,2,78,2, 368 ).
textconcept( 3,9,2,78,3, 369 ).
textconcept( 3,9,2,78,4, 370 ).
textconcept( 3,9,2,78,5, 371 ).
textconcept( 3,9,3,79,1, 372 ).
textconcept( 3,9,3,79,2, 373 ).
textconcept( 3,9,3,79,3, 374 ).
textconcept( 3,9,4,80,1, 375 ).
textconcept( 3,9,4,80,2, 376 ).
textconcept( 3,9,4,80,3, 377 ).
textconcept( 3,9,4,80,4, 378 ).
textconcept( 3,9,4,80,5, 379 ).
textconcept( 3,9,4,80,6, 380 ).
textconcept( 4,0,0,81,1, 381 ).
textconcept( 4,1,1,82,1, 382 ).
textconcept( 4,1,1,82,2, 383 ).
textconcept( 4,1,1,82,3, 384 ).
textconcept( 4,1,1,82,4, 385 ).
textconcept( 4,1,1,82,5, 386 ).
textconcept( 4,1,1,82,6, 387 ).
textconcept( 4,1,1,82,7, 388 ).
textconcept( 4,1,2,83,1, 389 ).
textconcept( 4,1,2,83,2, 390 ).
textconcept( 4,1,2,83,3, 391 ).
textconcept( 4,1,2,83,4, 392 ).
textconcept( 4,1,2,83,5, 393 ).
textconcept( 4,1,3,84,1, 394 ).
textconcept( 4,1,3,84,2, 395 ).
textconcept( 4,1,3,84,3, 396 ).
textconcept( 4,1,3,84,4, 397 ).
textconcept( 4,1,4,85,1, 398 ).
textconcept( 4,1,4,85,2, 399 ).
textconcept( 4,1,4,85,3, 400 ).
textconcept( 4,1,4,85,4, 401 ).
textconcept( 4,1,4,85,5, 402 ).
textconcept( 4,1,4,85,6, 403 ).
textconcept( 4,1,4,85,7, 404 ).
textconcept( 4,1,5,86,1, 405 ).
textconcept( 4,1,5,86,2, 406 ).
textconcept( 4,1,5,86,3, 407 ).
textconcept( 4,1,5,86,4, 408 ).
textconcept( 4,1,5,86,5, 409 ).
textconcept( 4,1,5,86,6, 410 ).
textconcept( 4,1,6,87,1, 411 ).
textconcept( 4,1,6,87,2, 412 ).
textconcept( 4,1,7,88,1, 413 ).
textconcept( 4,1,7,88,2, 414 ).
textconcept( 4,1,7,88,3, 415 ).
textconcept( 4,1,7,88,4, 416 ).
textconcept( 4,1,7,88,5, 417 ).
textconcept( 4,1,7,88,6, 418 ).
textconcept( 4,2,1,89,1, 419 ).
textconcept( 4,2,1,89,2, 420 ).
textconcept( 4,2,1,89,3, 421 ).
textconcept( 4,2,1,89,4, 422 ).
textconcept( 4,2,1,89,5, 423 ).
textconcept( 4,2,1,89,6, 424 ).
textconcept( 4,2,1,89,7, 425 ).
textconcept( 4,2,1,89,8, 426 ).
textconcept( 4,2,1,89,9, 427 ).
textconcept( 4,2,2,90,1, 428 ).
textconcept( 4,2,2,90,2, 429 ).
textconcept( 4,2,2,90,3, 430 ).
textconcept( 4,2,2,90,4, 431 ).
textconcept( 4,2,2,90,5, 432 ).
textconcept( 4,2,3,91,1, 433 ).
textconcept( 4,2,3,91,2, 434 ).
textconcept( 4,2,3,91,3, 435 ).
textconcept( 4,2,3,91,4, 436 ).
textconcept( 4,2,3,91,5, 437 ).
textconcept( 4,2,3,91,6, 438 ).
textconcept( 4,2,3,91,7, 439 ).
textconcept( 4,2,3,91,8, 440 ).
textconcept( 4,2,3,91,9, 441 ).
textconcept( 4,2,3,91,10, 442 ).
textconcept( 4,2,3,91,11, 443 ).
textconcept( 4,2,3,91,12, 444 ).
textconcept( 4,2,3,91,13, 445 ).
textconcept( 4,2,3,91,14, 446 ).
textconcept( 4,2,3,91,15, 447 ).
textconcept( 4,2,3,91,16, 448 ).
textconcept( 4,2,3,91,17, 449 ).
textconcept( 4,2,3,91,18, 450 ).
textconcept( 4,2,3,91,19, 451 ).
textconcept( 4,2,3,91,20, 452 ).
textconcept( 4,2,3,91,21, 453 ).
textconcept( 4,2,3,91,22, 454 ).
textconcept( 4,2,4,92,1, 455 ).
textconcept( 4,2,4,92,2, 456 ).
textconcept( 4,2,4,92,3, 457 ).
```

```
textconcept( 4,2,4,92,5,  459 ).
textconcept( 4,2,4,92,6,  460 ).
textconcept( 4,2,4,92,7,  461 ).
textconcept( 4,2,4,92,8,  462 ).
textconcept( 4,2,4,92,9,  463 ).
textconcept( 4,2,4,92,10,  464 ).
textconcept( 4,2,4,92,11,  465 ).
textconcept( 4,2,4,92,12,  466 ).
textconcept( 4,2,4,92,13,  467 ).
textconcept( 4,2,4,92,14,  468 ).
textconcept( 4,3,1,93,1,  469 ).
textconcept( 4,3,1,93,2,  470 ).
textconcept( 4,3,1,93,3,  471 ).
textconcept( 4,3,1,93,4,  472 ).
textconcept( 4,3,1,93,5,  473 ).
textconcept( 4,3,1,93,6,  474 ).
textconcept( 4,3,1,93,7,  475 ).
textconcept( 4,4,1,94,1,  476 ).
textconcept( 4,4,1,94,2,  477 ).
textconcept( 4,4,2,95,1,  478 ).
textconcept( 4,4,2,95,2,  479 ).
textconcept( 4,4,2,95,3,  480 ).
textconcept( 4,4,2,95,4,  481 ).
textconcept( 4,4,2,95,5,  482 ).
textconcept( 4,4,3,96,1,  483 ).
textconcept( 4,4,3,96,2,  484 ).
textconcept( 4,4,3,96,3,  485 ).
textconcept( 4,4,3,96,4,  486 ).
textconcept( 4,5,1,97,1,  487 ).
textconcept( 4,5,1,97,2,  488 ).
textconcept( 4,5,1,97,3,  489 ).
textconcept( 4,5,2,98,1,  490 ).
textconcept( 4,5,2,98,2,  491 ).
textconcept( 4,5,2,98,3,  492 ).
textconcept( 4,5,2,98,4,  493 ).
textconcept( 4,5,2,98,5,  494 ).
textconcept( 4,5,3,99,1,  495 ).
textconcept( 4,5,3,99,2,  496 ).
textconcept( 4,5,3,99,3,  497 ).
textconcept( 4,5,3,99,4,  498 ).
textconcept( 4,5,3,99,5,  499 ).
textconcept( 4,5,3,99,6,  500 ).
textconcept( 4,5,3,99,7,  501 ).
textconcept( 4,5,4,100,1,  502 ).
textconcept( 4,5,4,100,2,  503 ).
textconcept( 4,5,4,100,3,  504 ).
textconcept( 4,5,4,100,4,  505 ).
textconcept( 4,5,4,100,5,  506 ).
textconcept( 4,6,1,101,1,  507 ).
textconcept( 4,6,1,101,2,  508 ).
textconcept( 4,6,1,101,3,  509 ).
textconcept( 4,6,1,101,4,  510 ).
textconcept( 4,6,2,102,1,  511 ).
textconcept( 4,6,2,102,2,  512 ).
textconcept( 4,6,2,102,3,  513 ).
textconcept( 4,6,2,102,4,  514 ).
textconcept( 4,6,2,102,5,  515 ).
textconcept( 4,6,2,102,6,  516 ).
textconcept( 4,6,2,102,7,  517 ).
textconcept( 4,6,2,102,8,  518 ).
textconcept( 4,6,3,103,1,  519 ).
textconcept( 4,6,3,103,2,  520 ).
textconcept( 4,6,3,103,3,  521 ).
textconcept( 4,6,3,103,4,  522 ).
textconcept( 4,6,3,103,5,  523 ).
textconcept( 4,6,3,103,6,  524 ).
textconcept( 4,6,3,103,7,  525 ).
textconcept( 4,6,4,104,1,  526 ).
textconcept( 4,6,4,104,2,  527 ).
textconcept( 4,6,4,104,3,  528 ).
textconcept( 4,6,4,104,4,  529 ).
textconcept( 4,6,4,104,5,  530 ).
textconcept( 4,6,4,104,6,  531 ).
textconcept( 4,6,4,104,7,  532 ).
textconcept( 4,6,5,105,1,  533 ).
textconcept( 4,6,5,105,2,  534 ).
textconcept( 4,6,5,105,3,  535 ).
textconcept( 4,6,5,105,4,  536 ).
textconcept( 4,6,5,105,5,  537 ).
textconcept( 4,6,6,106,1,  538 ).
textconcept( 4,6,6,106,2,  539 ).
textconcept( 4,6,6,106,3,  540 ).
textconcept( 4,6,7,107,1,  541 ).
textconcept( 4,6,7,107,2,  542 ).
textconcept( 4,6,7,107,3,  543 ).
textconcept( 5,0,0,108,1,  544 ).
textconcept( 5,1,1,109,1,  545 ).
textconcept( 5,1,1,109,2,  546 ).
textconcept( 5,1,1,109,3,  547 ).
textconcept( 5,1,1,109,4,  548 ).
textconcept( 5,1,1,109,5,  549 ).
textconcept( 5,1,1,109,6,  550 ).
textconcept( 5,1,2,110,1,  551 ).
textconcept( 5,1,2,110,2,  552 ).
textconcept( 5,2,1,111,1,  553 ).
textconcept( 5,2,1,111,2,  554 ).
textconcept( 5,2,1,111,3,  555 ).
textconcept( 5,2,1,111,4,  556 ).
textconcept( 5,2,2,112,1,  557 ).
textconcept( 5,2,2,112,2,  558 ).
textconcept( 5,2,2,112,3,  559 ).
textconcept( 5,2,2,112,4,  560 ).
textconcept( 5,2,2,112,5,  561 ).
textconcept( 5,2,2,112,6,  562 ).
textconcept( 5,2,3,113,1,  563 ).
textconcept( 5,2,3,113,2,  564 ).
textconcept( 5,2,3,113,3,  565 ).
textconcept( 5,2,3,113,4,  566 ).
textconcept( 5,2,3,113,5,  567 ).
textconcept( 5,3,1,114,1,  568 ).
textconcept( 5,3,1,114,2,  569 ).
textconcept( 5,3,1,114,3,  570 ).
textconcept( 5,3,2,115,1,  571 ).
textconcept( 5,3,2,115,2,  572 ).
textconcept( 5,3,2,115,3,  573 ).
textconcept( 5,3,2,115,4,  574 ).
textconcept( 5,3,2,115,5,  575 ).
textconcept( 5,3,2,115,6,  576 ).
textconcept( 5,3,2,115,7,  577 ).
textconcept( 5,3,2,115,8,  578 ).
textconcept( 5,3,2,115,9,  579 ).
textconcept( 5,3,2,115,10,  580 ).
textconcept( 5,3,2,115,11,  581 ).
textconcept( 5,3,2,115,12,  582 ).
textconcept( 5,3,2,115,13,  583 ).
textconcept( 5,3,2,115,14,  584 ).
textconcept( 5,3,2,115,15,  585 ).
textconcept( 5,3,2,115,16,  586 ).
textconcept( 5,3,2,115,17,  587 ).
textconcept( 5,3,2,115,18,  588 ).
textconcept( 5,3,2,115,19,  589 ).
```

```
textconcept( 5,3,2,115,21, 591 ).
textconcept( 5,4,1,116,1, 592 ).
textconcept( 5,4,1,116,2, 593 ).
textconcept( 5,4,1,116,3, 594 ).
textconcept( 5,4,1,116,4, 595 ).
textconcept( 5,4,1,116,5, 596 ).
textconcept( 5,4,1,116,6, 597 ).
textconcept( 5,4,1,116,7, 598 ).
textconcept( 5,4,1,116,8, 599 ).
textconcept( 5,4,1,116,9, 600 ).
textconcept( 5,4,1,116,10, 601 ).
textconcept( 5,4,2,117,1, 602 ).
textconcept( 5,4,2,117,2, 603 ).
textconcept( 5,4,2,117,3, 604 ).
textconcept( 5,4,2,117,4, 605 ).
textconcept( 5,4,3,118,1, 606 ).
textconcept( 5,4,3,118,2, 607 ).
textconcept( 5,4,3,118,3, 608 ).
textconcept( 5,4,3,118,4, 609 ).
textconcept( 5,4,3,118,5, 610 ).
textconcept( 5,4,4,119,1, 611 ).
textconcept( 5,4,4,119,2, 612 ).
textconcept( 5,4,4,119,3, 613 ).
textconcept( 5,4,4,119,4, 614 ).
textconcept( 5,4,4,119,5, 615 ).
textconcept( 5,4,5,120,1, 616 ).
textconcept( 5,4,5,120,2, 617 ).
textconcept( 5,4,5,120,3, 618 ).
textconcept( 5,4,6,121,1, 619 ).
textconcept( 5,4,6,121,2, 620 ).
textconcept( 5,4,6,121,3, 621 ).
textconcept( 5,4,6,121,4, 622 ).
textconcept( 5,5,1,122,1, 623 ).
textconcept( 5,5,1,122,2, 624 ).
textconcept( 5,5,1,122,3, 625 ).
textconcept( 5,5,1,122,4, 626 ).
textconcept( 5,5,2,123,1, 627 ).
textconcept( 5,5,2,123,2, 628 ).
textconcept( 5,5,3,124,1, 629 ).
textconcept( 5,5,3,124,2, 630 ).
textconcept( 5,5,3,124,3, 631 ).
textconcept( 5,5,3,124,4, 632 ).
textconcept( 5,5,4,125,1, 633 ).
textconcept( 5,5,4,125,2, 634 ).
textconcept( 5,5,4,125,3, 635 ).
textconcept( 5,5,4,125,4, 636 ).
textconcept( 5,5,5,126,1, 637 ).
textconcept( 5,5,5,126,2, 638 ).
textconcept( 5,5,5,126,3, 639 ).
textconcept( 5,6,1,127,1, 640 ).
textconcept( 5,6,1,127,2, 641 ).
textconcept( 5,6,1,127,3, 642 ).
textconcept( 5,6,1,127,4, 643 ).
textconcept( 5,6,1,127,5, 644 ).
textconcept( 5,6,1,127,6, 645 ).
textconcept( 5,6,2,128,1, 646 ).
textconcept( 5,6,2,128,2, 647 ).
textconcept( 5,6,2,128,3, 648 ).
textconcept( 5,6,2,128,4, 649 ).
textconcept( 5,6,2,128,5, 650 ).
textconcept( 5,6,2,128,6, 651 ).
textconcept( 6,0,0,129,1, 652 ).
textconcept( 6,1,1,130,1, 653 ).
textconcept( 6,1,1,130,2, 654 ).
textconcept( 6,1,1,130,3, 655 ).
textconcept( 6,1,2,131,1, 656 ).
textconcept( 6,1,2,131,2, 657 ).
textconcept( 6,1,2,131,3, 658 ).
textconcept( 6,1,2,131,4, 659 ).
textconcept( 6,1,2,131,5, 660 ).
textconcept( 6,1,3,132,1, 661 ).
textconcept( 6,1,3,132,2, 662 ).
textconcept( 6,1,3,132,3, 663 ).
textconcept( 6,1,4,133,1, 664 ).
textconcept( 6,1,4,133,2, 665 ).
textconcept( 6,1,4,133,3, 666 ).
textconcept( 6,1,5,134,1, 667 ).
textconcept( 6,1,5,134,2, 668 ).
textconcept( 6,1,5,134,3, 669 ).
textconcept( 6,1,5,134,4, 670 ).
textconcept( 6,1,5,134,5, 671 ).
textconcept( 6,1,5,134,6, 672 ).
textconcept( 6,1,5,134,7, 673 ).
textconcept( 6,1,6,135,1, 674 ).
textconcept( 6,1,6,135,2, 675 ).
textconcept( 6,1,6,135,3, 676 ).
textconcept( 6,1,6,135,4, 677 ).
textconcept( 6,1,6,135,5, 678 ).
textconcept( 6,2,1,136,1, 679 ).
textconcept( 6,2,1,136,2, 680 ).
textconcept( 6,2,1,136,3, 681 ).
textconcept( 6,2,1,136,4, 682 ).
textconcept( 6,2,1,136,5, 683 ).
textconcept( 6,2,1,136,6, 684 ).
textconcept( 6,2,1,136,7, 685 ).
textconcept( 6,2,1,136,8, 686 ).
textconcept( 6,2,1,136,9, 687 ).
textconcept( 6,2,2,137,1, 688 ).
textconcept( 6,2,2,137,2, 689 ).
textconcept( 6,2,2,137,3, 690 ).
textconcept( 6,2,2,137,4, 691 ).
textconcept( 6,2,3,138,1, 692 ).
textconcept( 6,2,3,138,2, 693 ).
textconcept( 6,2,3,138,3, 694 ).
textconcept( 6,2,3,138,4, 695 ).
textconcept( 6,2,3,138,5, 696 ).
textconcept( 6,2,3,138,6, 697 ).
textconcept( 6,2,3,138,7, 698 ).
textconcept( 6,2,3,138,8, 699 ).
textconcept( 6,2,3,138,9, 700 ).
textconcept( 6,2,4,139,1, 701 ).
textconcept( 6,2,4,139,2, 702 ).
textconcept( 6,2,4,139,3, 703 ).
textconcept( 6,2,4,139,4, 704 ).
textconcept( 6,2,4,139,5, 705 ).
textconcept( 6,2,4,139,6, 706 ).
textconcept( 6,2,4,139,7, 707 ).
textconcept( 6,2,5,140,1, 708 ).
textconcept( 6,2,5,140,2, 709 ).
textconcept( 6,2,5,140,3, 710 ).
textconcept( 6,2,5,140,4, 711 ).
textconcept( 6,2,5,140,5, 712 ).
textconcept( 6,2,5,140,6, 713 ).
textconcept( 6,2,6,141,1, 714 ).
textconcept( 6,2,6,141,2, 715 ).
textconcept( 6,2,6,141,3, 716 ).
textconcept( 6,2,6,141,4, 717 ).
textconcept( 6,2,6,141,5, 718 ).
textconcept( 6,3,1,142,1, 719 ).
textconcept( 6,3,1,142,2, 720 ).
textconcept( 6,3,2,143,1, 721 ).
```

```
extconcept( 6,3,2,143,3,  723 ).
extconcept( 6,3,2,143,4,  724 ).
extconcept( 6,3,3,144,1,  725 ).
extconcept( 6,3,3,144,2,  726 ).
textconcept( 6,3,3,144,3,  727 ).
textconcept( 6,3,3,144,4,  728 ).
textconcept( 6,3,3,144,5,  729 ).
textconcept( 6,3,3,144,6,  730 ).
textconcept( 6,3,4,145,1,  731 ).
textconcept( 6,3,4,145,2,  732 ).
```

RESULTS OF THE EXPERT CLASSIFIER MODULE.

TEXT 1: THE CHARACTERISTIC ERROR APPROACH TO CONFLICT RESOLUTION.

List of sentences which are highly important:

1,2,3,7,16,20,9,21,22,24,55,61,122,123,125,126,128,

List of sentences which are of medium importance:

4,5,6,17,18,19,23,56,57,58,59,60,124,127,

List of sentences which have a low importance:

8,10,11,12,13,14,15,25,26,27,28,29,30,31,32,33,34,35,
36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,

List of sentences which have zero importance:

62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,
80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,
99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,
114,115,116,117,118,119,120,121,

RESULTS OF PAPER 1 GIVEN AS PROLOG FACTS

```
/* text1imp_a.pl */
importance(1,high).
importance(2,high).
importance(3,high).
importance(7,high).
importance(4,medium).
importance(5,medium).
importance(6,medium).
importance(16,high).
importance(20,high).
importance(17,medium).
importance(18,medium).
importance(19,medium).
importance(8,low).
importance(9,high).
importance(10,low).
importance(11,low).
importance(12,low).
importance(13,low).
importance(14,low).
importance(15,low).
importance(21,high).
importance(22,high).
importance(24,high).
importance(23,medium).
importance(55,high).
```

importance(79,zero).
importance(80,zero).
importance(81,zero).
importance(82,zero).
importance(83,zero).
importance(84,zero).
importance(85,zero).
importance(86,zero).
importance(87,zero).
importance(88,zero).
importance(89,zero).
importance(90,zero).
importance(91,zero).
importance(92,zero).
importance(93,zero).
importance(94,zero).
importance(95,zero).
importance(96,zero).
importance(97,zero).
importance(98,zero).
importance(99,zero).
importance(100,zero).
importance(101,zero).
importance(102,zero).
importance(103,zero).
importance(104,zero).
importance(105,zero).
importance(106,zero).
importance(107,zero).
importance(108,zero).
importance(109,zero).
importance(110,zero).
importance(111,zero).
importance(112,zero).
importance(113,zero).
importance(114,zero).
importance(115,zero).
importance(116,zero).
importance(117,zero).
importance(118,zero).
importance(119,zero).
importance(120,zero).
importance(121,zero).
importance(122,high).
importance(123,high).
importance(125,high).
importance(124,medium).
importance(126,high).
importance(128,high).
importance(127,medium).

NOTE:- Please see over & Film P407 first.

RESULTS OF THE EXPERT CLASSIFIER MODULE.

TEXT 2: COMODEL: A LANGUAGE FOR THE REPRESENTATION OF TECHNICAL KNOWLEDGE.

List of sentences which are highly important:
1,2,3,9,10,13,110,111,112,114,116,

List of sentences which are of medium importance:
4,5,6,7,8,11,12,115,

List of sentences which have a low importance:
113,

List of sentences which have zero importance:
14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,
30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,
47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,
98,99,100,101,102,103,104,105,106,107,108,109,

RESULTS OF PAPER 2 GIVEN AS PROLOG FACTS

```
/* text2imp_a.pl */
importance(1,high).
importance(2,high).
importance(3,high).
importance(9,high).
importance(4,medium).
importance(5,medium).
importance(6,medium).
importance(7,medium).
importance(8,medium).
importance(10,high).
importance(13,high).
importance(11,medium).
importance(12,medium).
importance(14,zero).
importance(15,zero).
importance(16,zero).
importance(17,zero).
importance(18,zero).
importance(19,zero).
importance(20,zero).
importance(21,zero).
importance(22,zero).
importance(23,zero).
```

```
importance(24,zero).
importance(25,zero).
importance(26,zero).
importance(27,zero).
importance(28,zero).
importance(29,zero).
importance(30,zero).
importance(31,zero).
importance(32,zero).
importance(33,zero).
importance(34,zero).
importance(35,zero).
importance(36,zero).
importance(37,zero).
importance(38,zero).
importance(39,zero).
importance(40,zero).
importance(41,zero).
importance(42,zero).
importance(43,zero).
importance(44,zero).
importance(45,zero).
importance(46,zero).
importance(47,zero).
importance(48,zero).
importance(49,zero).
importance(50,zero).
importance(51,zero).
importance(52,zero).
importance(53,zero).
importance(54,zero).
importance(55,zero).
importance(56,zero).
importance(57,zero).
importance(58,zero).
importance(59,zero).
importance(60,zero).
importance(61,zero).
importance(62,zero).
importance(63,zero).
importance(64,zero).
importance(65,zero).
importance(66,zero).
importance(67,zero).
importance(68,zero).
importance(69,zero).
importance(70,zero).
importance(71,zero).
importance(72,zero).
importance(73,zero).
importance(74,zero).
importance(75,zero).
importance(76,zero).
```

```
importance(77,zero).
importance(78,zero).
importance(79,zero).
importance(80,zero).
importance(81,zero).
importance(82,zero).
importance(83,zero).
importance(84,zero).
importance(85,zero).
importance(86,zero).
importance(87,zero).
importance(88,zero).
importance(89,zero).
importance(90,zero).
importance(91,zero).
importance(92,zero).
importance(93,zero).
importance(94,zero).
importance(95,zero).
importance(96,zero).
importance(97,zero).
importance(98,zero).
importance(99,zero).
importance(100,zero).
importance(101,zero).
importance(102,zero).
importance(103,zero).
importance(104,zero).
importance(105,zero).
importance(106,zero).
importance(107,zero).
importance(108,zero).
importance(109,zero).
importance(110,high).
importance(111,high).
importance(112,high).
importance(114,high).
importance(116,high).
importance(115,medium).
importance(113,low).
```

RESULTS OF THE EXPERT CLASSIFIER MODULE.

TEXT 3: Checking An Expert Systems Knowledge Base For Consistency And Completeness.

List of sentences which are highly important:
1,2,3,10,11,17,74,75,79,82,82,

List of sentences which are of medium importance:
4,5,6,7,8,9,12,13,14,15,16,76,77,78,

List of sentences which have a low importance:
80,81,

List of sentences which have zero importance:
18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,
35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,
53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,

RESULTS OF PAPER 3 GIVEN AS PROLOG FACTS.

```
/* text3imp_a.pl */
importance(1,high).
importance(2,high).
importance(3,high).
importance(10,high).
importance(4,medium).
importance(5,medium).
importance(6,medium).
importance(7,medium).
importance(8,medium).
importance(9,medium).
importance(11,high).
importance(17,high).
importance(12,medium).
importance(13,medium).
importance(14,medium).
importance(15,medium).
importance(16,medium).
importance(18,zero).
importance(19,zero).
importance(20,zero).
importance(21,zero).
importance(22,zero).
importance(23,zero).
importance(24,zero).
importance(25,zero).
importance(26,zero).
importance(27,zero).
```

```
importance(28,zero).
importance(29,zero).
importance(30,zero).
importance(31,zero).
importance(32,zero).
importance(33,zero).
importance(34,zero).
importance(35,zero).
importance(36,zero).
importance(37,zero).
importance(38,zero).
importance(39,zero).
importance(40,zero).
importance(41,zero).
importance(42,zero).
importance(43,zero).
importance(44,zero).
importance(45,zero).
importance(46,zero).
importance(47,zero).
importance(48,zero).
importance(49,zero).
importance(50,zero).
importance(51,zero).
importance(52,zero).
importance(53,zero).
importance(54,zero).
importance(55,zero).
importance(56,zero).
importance(57,zero).
importance(58,zero).
importance(59,zero).
importance(60,zero).
importance(61,zero).
importance(62,zero).
importance(63,zero).
importance(64,zero).
importance(65,zero).
importance(66,zero).
importance(67,zero).
importance(68,zero).
importance(69,zero).
importance(70,zero).
importance(71,zero).
importance(72,zero).
importance(73,zero).
importance(74,high).
importance(75,high).
importance(79,high).
importance(76,medium).
importance(77,medium).
importance(78,medium).
importance(82,high).
```

```
importance(82,high).
importance(80,low).
importance(81,low).
```

RESULTS OF THE EXPERT CLASSIFIER MODLE.

TEXT 4: AN EXPERT ADVISOR FOR PHOTOLITHOGRAPHY.

List of sentences which are highly important:
1,2,3,6,16,20,86,87,88,104,108,

List of sentences which are of medium importance:
4,5,17,18,19,105,106,107,

List of sentences which have a low importance:
7,8,9,10,11,12,13,14,15,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,

List of sentences which have zero importance:
21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,
45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,
70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,

RESULTS OF PAPER 4 GIVEN AS PROLOG FACTS

```
/* text4imp_a.pl*/
importance(1,high).
importance(2,high).
importance(3,high).
importance(6,high).
importance(4,medium).
importance(5,medium).
importance(16,high).
importance(20,high).
importance(17,medium).
importance(18,medium).
importance(19,medium).
importance(7,low).
importance(8,low).
importance(9,low).
importance(10,low).
importance(11,low).
importance(12,low).
importance(13,low).
importance(14,low).
importance(15,low).
importance(21,zero).
importance(22,zero).
importance(23,zero).
importance(24,zero).
importance(25,zero).
importance(26,zero).
importance(27,zero).
importance(28,zero).
importance(29,zero).
importance(30,zero).
importance(31,zero).
importance(32,zero).
importance(33,zero).
importance(34,zero).
importance(35,zero).
importance(36,zero).
importance(37,zero).
importance(38,zero).
importance(39,zero).
importance(40,zero).
importance(41,zero).
importance(42,zero).
importance(43,zero).
```

importance(44,zero).
importance(45,zero).
importance(46,zero).
importance(47,zero).
importance(48,zero).
importance(49,zero).
importance(50,zero).
importance(51,zero).
importance(52,zero).
importance(53,zero).
importance(54,zero).
importance(55,zero).
importance(56,zero).
importance(57,zero).
importance(58,zero).
importance(59,zero).
importance(60,zero).
importance(61,zero).
importance(62,zero).
importance(63,zero).
importance(64,zero).
importance(65,zero).
importance(66,zero).
importance(67,zero).
importance(68,zero).
importance(69,zero).
importance(70,zero).
importance(71,zero).
importance(72,zero).
importance(73,zero).
importance(74,zero).
importance(75,zero).
importance(76,zero).
importance(77,zero).
importance(78,zero).
importance(79,zero).
importance(80,zero).
importance(81,zero).
importance(82,zero).
importance(83,zero).
importance(84,zero).
importance(85,zero).
importance(86,high).
importance(87,high).
importance(88,high).
importance(104,high).
importance(108,high).
importance(105,medium).
importance(106,medium).
importance(107,medium).
importance(89,low).
importance(90,low).
importance(91,low).
importance(92,low).
importance(93,low).
importance(94,low).
importance(95,low).
importance(96,low).
importance(97,low).
importance(98,low).
importance(99,low).
importance(100,low).
importance(101,low).
importance(102,low).
importance(103,low).

RESULTS OF THE EXPERT CLASSIFIER MODULE.

PAPER 5: FAULT DIAGNOSIS THROUGH RESPONSIBILITY.

List of sentences which are highly important:
1,2,3,5,36,36,12,20,31,129,130,135,142,145,

List of sentences which are of medium importance:
4,17,131,132,133,134,143,144,

List of sentences which have a low importance:
6,7,8,9,10,11,13,14,15,16,18,19,21,22,23,24,25,26,
27,28,29,30,32,33,34,35,136, 137,138,139,140,141,

List of sentences which have zero importance:
37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,
54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,
71,72,73,74,75,76,78,79,80,81,82,83,84,85,86,87,88,
89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,
105,106,107,108,109,110,111,112,113,114,115,116,117,
118,119,120,121,122,123,124,125,126,127,128,

RESULTS OF PAPER5 GIVEN AS PROLOG FACTS.

```
/* text5imp_a.pl */
importance(1,high).
importance(2,high).
importance(3,high).
importance(5,high).
importance(4,medium).
importance(36,high).
importance(36,high).
importance(6,low).
importance(7,low).
importance(8,low).
importance(9,low).
importance(10,low).
importance(11,low).
importance(12,high).
importance(13,low).
importance(14,low).
importance(15,low).
importance(16,low).
importance(17,low).
importance(18,low).
importance(19,low).
importance(20,high).
importance(21,low).
importance(22,low).
```

```
importance(23,low).
importance(24,low).
importance(25,low).
importance(26,low).
importance(27,low).
importance(28,low).
importance(29,low).
importance(30,low).
importance(31,high).
importance(32,low).
importance(33,low).
importance(34,low).
importance(35,low).
importance(37,zero).
importance(38,zero).
importance(39,zero).
importance(40,zero).
importance(41,zero).
importance(42,zero).
importance(43,zero).
importance(44,zero).
importance(45,zero).
importance(46,zero).
importance(47,zero).
importance(48,zero).
importance(49,zero).
importance(50,zero).
importance(51,zero).
importance(52,zero).
importance(53,zero).
importance(54,zero).
importance(55,zero).
importance(56,zero).
importance(57,zero).
importance(58,zero).
importance(59,zero).
importance(60,zero).
importance(61,zero).
importance(62,zero).
importance(63,zero).
importance(64,zero).
importance(65,zero).
importance(66,zero).
importance(67,zero).
importance(68,zero).
importance(69,zero).
importance(70,zero).
importance(71,zero).
importance(72,zero).
importance(73,zero).
importance(74,zero).
importance(75,zero).
importance(76,zero).
```

```
importance(77,zero).
importance(78,zero).
importance(79,zero).
importance(80,zero).
importance(81,zero).
importance(82,zero).
importance(83,zero).
importance(84,zero).
importance(85,zero).
importance(86,zero).
importance(87,zero).
importance(88,zero).
importance(89,zero).
importance(90,zero).
importance(91,zero).
importance(92,zero).
importance(93,zero).
importance(94,zero).
importance(95,zero).
importance(96,zero).
importance(97,zero).
importance(98,zero).
importance(99,zero).
importance(100,zero).
importance(101,zero).
importance(102,zero).
importance(103,zero).
importance(104,zero).
importance(105,zero).
importance(106,zero).
importance(107,zero).
importance(108,zero).
importance(109,zero).
importance(110,zero).
importance(111,zero):
importance(112,zero).
importance(113,zero).
importance(114,zero).
importance(115,zero).
importance(116,zero).
importance(117,zero).
importance(118,zero).
importance(119,zero).
importance(120,zero).
importance(121,zero).
importance(122,zero).
importance(123,zero).
importance(124,zero).
importance(125,zero).
importance(126,zero).
importance(128,zero).
importance(129,high).
importance(130,high).
```

```
importance(135,high).
importance(131,medium).
importance(132,medium).
importance(133,medium).
importance(134,medium).
importance(142,high).
importance(145,high).
importance(143,medium).
importance(144,medium).
importance(136,low).
importance(137,low).
importance(138,low).
importance(139,low).
importance(140,low).
importance(141,low).
```

APPENDIX I

SENTENCES CLASSIFIED AS OF LOW/ZERO IMPORTANCE

Paper 1: The Characteristic Error Approach to Conflict Resolution.

Conflict resolution has been treated by a number of different authors who use various techniques and terminology such as reasoning about uncertainty or 'the integration of knowledge from disparate sources ' [4]. Most of this work takes a numeric approach to conflict resolution using, for example, Bayesian or Dempster-Schaefer techniques [5]. However, Bayesian techniques assume the ability to generate a priori a table of conditional probabilities (i.e. the probability of A given B), and only allow the source to posit a single probability at a time. Dempster-Schafer techniques, on the other hand, while allowing multiple beliefs, assume that all possibilities are independent, an assumption not often satisfied in reality. Other approaches to conflict resolution represent a more semantic knowledge based approach and do not rely exclusively upon a numeric algorithm to resolve conflicts. Examples of these approaches include Cohen's theory of endorsements [2] and Tenenbaum and Barrow's Interpretation Guided Segmentation [6]. Even these approaches however, exhibit some limitations. They lack extensibility, function in limited domains and require either extensive preprogramming or interactive assistance for operation.

Characteristic error is based upon the idea that, within a multi-source system, each individual knowledge source is, to some extent, the best judge of its own performance. The motivation for this becomes apparent if one considers the factors which are normally embedded in the process of developing any given expert system.

The knowledge of the context in which it expects to operate; ie., an expectation of the form which the environment will take and the types of interrelationships that may occur.

The type of boundary or limiting conditions it expects to encounter; i.e., what information can occur in the environment which is close to what is expected but is actually different.

The type of misclassifications it postulates can occur; ie., what information may be present in the environment that is often confused with the desired information.

Implementing the concept of characteristic error requires that each knowledge source posit an a priori estimate of the error classes which it might encounter. This estimate is an explicit ranking of its expectations for the immediately local context, and is independent of the strength of the knowledge source's opinion in any particular instance of its application. It is based only upon the expert's knowledge of the underlying structure of its environment and decisions. For example, a region growing knowledge source, designed for road finding within an image understanding system, will know that local classification error may occur because a portion of the road may be partially obscured by trees. By the same token, it will also be aware of the fact that it is very uncharacteristic for any region designated as deep water to actually be a road. The same source will, however, have no real opinion as to the likely misclassification of shallow water. These considerations are independent of its specific belief of the existence of roads in any particular instance.

This error class information is dependent only upon the context assumed by the knowledge source. It is independent of the actual context of any particular instance in which the knowledge source is utilized. In other words, the knowledge source's characteristic error list explicitly states the context which was assumed to be part of the underlying structure of the environment during knowledge acquisition. Therefore, no a priori controls need be imposed upon the overall system using the knowledge source, since all context information is explicitly stated by the knowledge source itself.

To illustrate these points, Table 1 shows the structure of two road finders. The road finder which anticipates being called upon to process satellite information might expect partial road covering by treetops. The ground level road finder on the other hand, has no expectation of such a conflict.

Such a priori identification of errors also promotes the graceful performance degradation of a group of knowledge sources, if and when any knowledge source is operating outside of its area of expertise. A source operating in an unfamiliar area should not have the conflicting class appearing anywhere in its characteristic error list. This absence serves as an immediate flag for the conflict resolver to investigate the situation further.

The minimum implementation of the characteristic error technique requires the following information:

The characteristic a priori errors expected to be encountered by the knowledge source in the environment

The knowledge source's strength of opinion in a particular instance

A methodology for evaluating the error list and determining the context.

The system works by comparing the characteristic error lists of the conflicting knowledge sources. This determines the conflict's local context. This context is, in essence, an indicator of the similarity and structure of these error lists. The implementation is such that similar error lists create a context controlling the relative strength of opinion necessary from the knowledge sources to alter a decision. For example, if all sources agree that a given error is characteristic of the situation, it will not take a large (absolute) opinion strength to shift the decision in another direction. By the same token, if the context is relatively incoherent, a knowledge source's opinion must be stronger for the opinion to change. The interaction of these two factors is the basis for the implementation of the most likely error approach.

~3 AN APPLICATION OF CHARACTERISTIC ERROR.

This section presents an example application of the characteristic error concept. The focus of our application is TASC's Multi-Spectral Image Analysis System (MSIAS) [3]. MSIAS is a knowledge based system that interprets cartographic surface attributes using information obtained from remotely sensed imagery.

The MSIAS classifier utilizes the relative spectral signature of an individual pixel location (i.e., its relative intensity in various spectral bands) to determine its surface material class e.g. concrete, vegetation, water. Conflicts will only occur when new knowledge sources, using other criteria, are introduced into the system. These knowledge sources may start with the latter spectral identification, but may base their own classification opinion upon some alternate criteria (e.g. structural measures). The important feature of these knowledge sources is that they use the same initial information (the spectral classification), but apply independent criteria to obtain their individual results. These results generate alternate possible classifications for any given pixel/region. In the example below, alternate classifications represent the conflicts. The process of obtaining a final classification is conflict resolution. Using the nomenclature of Clancy [1] this is a problem of heuristic classification.

It is planned that the conflict resolver will be activated late in the image classification process. The late activation occurs because a number of different knowledge sources must be polled for a conflict to occur. Conflict resolution is only applied to those objects which are the subject of conflicting opinions. The frequency of these conflicts will increase as the semantic level of the program increases. That is, the use of the resolver will be lower, on a percentage basis for pixels than for regions. Upon activation, the resolver first looks at the pixels/regions immediately adjacent to the one currentiy under investigation and searches for other opinions as to this pixel's classification. Every adjacent pixel is marked with both a specific classification and characteristic error list. The list summarizes the characteristic error classes posited by the knowledge source which identified it. This information serves as the basis for conflict resolution, attempting to address such issues as:
How do the other conflicts and their characteristic error classes relate to the identification of this pixel? Do they reinforce the current designation or contest it and in what way.
The error classes may also be completely mute about this possible pixel classification.

As shown in Figure 1, a pixel can have up to eight nearest neighbors in a digitized image.

There are two extreme cases for the possible opinions of these neighbors: First the original pixel classification (at x) may be a member of the 'uncharacteristic' error class of each and every neighboring pixel; second, the pixel's class appears in the 'characteristic' error class of all neighboring pixels.

In the first case, the conflict resolver would not, under any circumstances, alter the pixel's classification, as as all of the knowledge sources agree that it is unlikely that this pixel is misclassified. Therefore, even if all knowledge sources felt that their regions were worthy of extension, no regions would be extended as they all have preagreed to abide by this pixel designation. That is, they have all agreed that an 'x' class pixel is both something which they expect to see in the environment and something which could be a boundary of their regions. This corresponds to the general case in which an expert would say 'I use x as an indicator of this condition, y is an indicator of an other condition, x and y are rarely confused'.

In the second case however, the conflict resolver will change the classification of this pixel to the competing class with the highest relative likelihood (opinion strength). Such a change will occur even if the relative likelihood is fairly low in an absolute sense. This is because there is complete agreement that initial misclassification of this pixel is often characteristic of the situation. This corresponds to the general case in which an expert would say ' I use x as an indicator of this condition, however y is often confused with x, so if you have doubts about y you may also use it as an indicator'.

Fundamentally, the characteristic error paradigm gives one a methodology for juggling cases intermediate between these two extremes. For example, for seven neighbors to agree is less strong a case than all eight, or if one has the class on its uncharacteristic error list it can possibly be excluded.

There are three basic ways of performing this integration. The simplest method merely tabulates frequency of appearance for the pixel class in question, combines the two frequencies (characteristic and uncharacteristic) and decides to go with the highest combination of relative frequency and opinion strength. This is a relatively simple approach with weights being heuristically determined.

The second approach judges the relative coherence of these opinions. For example, if three divergent region types had found that the class in question was on their characteristic error list, it is a stronger case for misclassification than if a single type of region had reached the same conclusion three tunes. This approach attempts to compensate for systematic bias on the part of the experts.

The third method which might be considered involves merely discounting the opinion of any region having this class on its uncharacteristic error list. This is however not a valid method as in such a situation a region is actually giving a strong vote for retaining the current classification.

The characteristic error paradigm allows the extraction of information about the characteristics of the knowledge source itself. First, it is important to recall that although we have been discussing three classes there actually are four specific class lists in existence. The fourth class being the class(es) for which the structural knowledge source (region grower) was actually designed. For example, a road grower would have concrete/asphalt in this class, some other examples appear in Table 2 .

The characteristic error class is a group of pixel identifications in which one would expect an error to occur in the environment. As shown in the table, if one was growing highways, it would be expected that some road areas may have been misclassified due to overhanging trees, oil slicks etc . The 'uncharacteristic' error classes are the ones which are expected to form a hard boundary between the given class and an alternate. These would be such classes as deep water and rock for roads. Neutral classes are classes which may exist in the expected environment but about which no intelligent statement may be made i.e. very shallow water, patchy snow for roads etc . As was discussed in section 2, manipulation of this list can be a significant action in and of itself and may also serve as an ideal place for changing context. For example, if we know that an area had recently been flooded, we would move shallow water to the characteristic error set for all low lying classes.

The explicit designation of the classes expected to occur in the environment has a very useful side effect: It is now possible to determine that a knowledge source is operating outside of its area of expertise. For example, an airport hangar area grower would expect fuel storage tanks appearing next to it as a hard boundary. The highway expert, on the other hand, would not expect such a structure in any immediately adjacent pixel. Fuel storage tanks would therefore not appear anywhere on a highway grower's list of error classes; not in 'characteristic' 'uncharacteristic' nor in 'neutral'. Therefore if the conflict resolution process retained the storage tank classification, the conflict resolver would disregard the results of the highway finder in the abutting region. It would then try to find alternate explanations of the region/pixel which it had designated as 'highway', by eliminating the highway designation and asking for alternate opinions.

~2 MODELLING OF TECHNICAL SYSTEMS.

The different works on modelling of technical systems mentioned above have in common that they do not describe a technical system globally, rather they start from context independent descriptions of single parts and their functions and try to infer descriptions of aggregates from those of the parts. The descriptions are based on variables, e.g. temperature, pressure, cross-section etc., values of variables and constraints. Forbus [2, 3, 4] takes the process, causing changes of physical situations, as the basic concept of his description, in de Kleer's and Brown's approach [5, 6] the physical component, whose behavior is characterized by confluences, plays this role, and Kuipers' and Kassirer's work [7, 8, 9] is based on 'universal' constraints, related to different types of variables. Raulefs [10] attaches importance to different degrees of granulation in the qualitative description and to reasoning about time.

Our approach has the following basic features.
1. Component oriented descripton. A technical system is regarded as composed of a finite number of components that correspond to the parts of a real system and can be accumulated to aggregates.
2. Qualitative and quantitative representation of the properties of physical objects.
3. System theoretical taxonomy for classifying the variety of components and their connections (component types, interaction types).
4. Uniform description of the function and the geometric properties of components.

With respect to 1. and 2., our approach is related to that of de Kleer and Brown and of Kuipers and Kassirer. Component orientation allows modular descriptions of large aggregates, therefore such a description can be easily modified and adapted to other configurations. The behaviour of components and aggregates is defined by means of constraints on physical 'properties" or variables. With respect to 3. and 4., we differ from the works mentioned above. As a consequence of a component oriented approach, the components can and should be classified in a systematic way according to the principles of system engineering. Geometric properties of components are of interest in so far as they are essential parts of the behaviour of the components, and for this reason they are described in the same way as the functional properties.

~3 ELEMENTS OF COMODEL.

In the COMponent Oriented DEscription Language (COMODEL) a technical system is viewed as a set of phenomenons, that may be observable by automatic measurement or by human senses.

Phenomenons can be connected by relations. This results in a network consisting of phenomenons as edges and two types of nodes, called components and interactions (cf. figure 1). Components, interactions and phenomenons are the basic objects of our description language, they are related to the parts of a technical system, the connections of these parts and the observable physical properties in the real world.

A COMODEL description of a real technical system consists of a number of object-definitions, defining the three basic objects and a compound object called aggregate.

A phenomenon is a collection of a finite number of physical properties, combined from a particular point of view.

A physical property is a part of a phenomenon. It may be observable or not.

A component is an object with a finite number of gates and relations. To each gate a phenomenon is assigned. The gates serve as links between components and interactions. The relations describe connections between values of different physical properties in the gates' phenomenons or define values of physical properties.

An interaction is an object with a finite number of gates and relations, like a component. It differs from a component by the fact that its relations cannot define values, i.e. interactions do not have characteristic or intrinsic values.

Gates are parts of components and interactions and are used as links between both. The connection of a component with an interaction by corresponding gates induces identity on the corresponding phenomenons of the gates.

Relationships between physical properties are described by relations. Simple relations are equal, unequal, proportional etc..

Connection of components and interactions results in larger entities called aggregates. In the same way aggregates are combined to form bigger aggregates up to the whole technical system, that can be viewed as one aggregate consisting of a hierarchy of (sub-) aggregates.

In the definition of phenomenons, components, interactions and aggregates one may refer either to build-in types (PHENOMENON, COMPONENT, INTERACTION, AGGREGATE) or to user-defined objects (symbolized by the preterminal symbol identifier). These objects form a hierarchy representing the taxonomy of system theoretical terms. Each object describes some aspects of a part of a real system at a particular level of generality.

-4 SAMPLE APPLICATION OF COMODEL.

paradigm for the representation of technical systems we take the nitric acid cooler of Lapp/Powers [11], shown in figure 2 . The function of this process is to cool a hot nitric acid stream before reacting it with benzene to form nitrobenzene. One top event for the system is a high temperature in the nitric acid reactor feed, since this could cause a reactor runaway.

In order to show the applicability of COMODEL to describe a broad variety of technical mechanisms we choose two parts of the nitric acid cooler - the heat exchanger as energy flow system and a piston and cylinder, taken from the cooling water pump, which is assumed to be a piston pump, as an example for geometric interactions.

~4 1 Heat Exchanger.

The heat exchanger is an aggregate with four gates, two fluid streams passing through and interchanging heat energy. lt can be viewed as constructed of two equal components, each a fluid stream with heat energy flowing off or flowing to (cf. figure 3). Therefore, such a component is an object with three gates, two fluid streams, and one energy stream (cf. figure 4). The relevant physical properties (flow direction, temperature, heat stream) are combined to phenomenons.

The heat exchanger component HE is GIVEN BELOW. The heat exchanger component has three gates: two for the fluid stream (M1, M2) and one for the heat stream (E3). The fluid stream passes through the gates M1 and M2 in opposite directions with respect to the centre of the component, i.e. at one gate it enters the component and at the other it leaves the component. This is stated by the first relation. The other relations define some relationships between the heat stream and the temperature of the fluid stream. The heat stream is proportional to the difference of temperatures, i.e. to the loss or gain of temperature of the fluid stream. If the entrance temperature is greater (less) than the exit-temperature then the heat stream leaves (enters) the component. The average temperature of the fluid stream is equal to the temperature of the heat stream.

Fluid stream and heat stream are described by interactions. The interaction FLUIDSTREAM has two gates (M1, M2). The relation states that the values of corresponding physical properties occurring at gate M1 and gate M2 respectively are equal, except for the values of the flow directions.

The interaction HEATENERGY has the two gates E1 and E2. The relations define some relationships between the physical properties included in the phenomenon ENERGYFLOW occurring at the gates of HEATENERGY. The difference of the temperatures at both gates determines the heat stream. No heat is lost between both gates, therefore E1 Q = E2 Q. The heat flows from higher to lower temperature, i.e. it enters (leaves) the interaction at E1 if E1 T > (<) E2 T. The same holds for gate E2 because of the opposite directions.

The aggregate HEATEXCHANGER (cf. figure 5) consists of two components of type HE and one interaction of type HEATENERGY. The list of gate-pairs define how the constituents of the aggregate are composed, because a gate-pair represents an identification of two gates. For example HE1 is composed with HF by identification of the gates HE1-E3 and HF-E1. A pair like (HE2-M1 M3) says that gate M1 of HE2 is 'free', i.e. not used for composition, and becomes therefore a gate of the aggregate. The description of the aggregate gates is adapted from the description of the corresponding component gates.

~4 2 Piston and cylinder.

Piston and cylinder are the basic elements of the piston pump. They are viewed as two components and modelled by the two objects full cylinder and hollow cylinder, cf. figure 6 . The phenomenons join together geometric and kinematic properties.A hollow body (e.g. a cylinder) can have no bottoms (pipe), one bottom (opened can), or two bottoms (closed can). The two components are given below.

In the definition of piston and cylinder radius, length, degrees of freedom and number of bottoms are bound to particular values (XR, XL, TRANS-FR,_ ,1). Every piston and cylinder has six degrees of freedom: translation along the R-axis, along the L-axis, and along the R-axis after a 90o-rotation, rotation around the R-axis, around the L-axis, and around the R-axis after a 90o rotation. These definitions describe aspects of individual objects from the class of all fullcylinders and all hollowcylinders respectively, only the values for the position in space are left open.

The FULLCYLINDER-IN-HOLLOWCYLINDER-interaction describes the geometric relationship, if the full cylinder is put into the hollow cylinder. It creates a new hollow cylinder, called pump chamber, with the length depending on the position of the piston. The piston retains two degrees of freedom in the resulting aggregate, cf. figure 7 .

The FULLCYLINDER-IN-HOLLOWCYLINDER-interaction has three gates (F, H, R). It combines the phenomenons FULLCYLINDERPROPERTIES and HOLLOWCYLINDERPROPERTIES to form a new phenomenon HOLLOWCYLINDERPROPERTIES.
Gates R and H have the same radius, i.e. the radius of the interaction as a whole depends on the hollowcylinder which it is composed with. A similar relation holds for the length. The number of bottoms is 2 . Only two degrees of freedom are left: translation along and rotation around the L-axis.
The aggregate PUMPCHAMBER consists of two components of type CYLINDER and PISTON respectively and one interaction of type FULLCYLINDER-IN-HOLLOWCYLINDER. Here, the interaction plays a central role. Gate FIT_R is the gate of the aggregate. The composition of the three constituents is obvious.

The definition of the aggregate reflects the fact that the behaviour of a pumpchamber can be described mainly by the interaction of a piston and a cylinder.

It is our intention to use COMMODEL as a technical description language for expert systems for the area of technical systems, in particular for interpretation of the states of those systems and diagnosis of failures.

~2 CHECKING FOR CONSISTENCY AND COMPLETENESS.

A static analysis of the rules can detect many potential problems and gaps that exist in a rule base. We now identify and give definitions for seven criteria that are used by CHECK to perform static analysis of any rule base constructed for use with LES. The first four criteria are concerned with potential problems, whereas the last three criteria are concerned with gaps in the knowledge base.

~2 1 POTENCIAL PROBLEMS IN A KNOWLEDGE BASE.

By statically analyzing the logical semantics of the rules represented in LES's case grammar format, CHECK can detect redundant rules, conflicting rules, rules that are subsumed by other rules, and circular-rule chains. The following definitions for these four potential problems are used in CHECK.
Redundant rules: two rules succeed in the same situation and have the same results. In LES, this means that the IF parts of the two rules are equivalent, and one or more THEN clauses are also equivalent. Because LES allows variables in rules, equivalent means that the same specific object names can match their corresponding variables. For example the rule 'p(x) --> q(x)' is equivalent to the rule 'p(y) --> q(y)', where x and y are variables.
Conflicting rules: two rules succeed in the same situation but with conflicting results. In LES, this means that the IF parts of the two rules are squivalent, but one or more THEN clauses are contradictory, or one pair of IF clauses is contradictory while they have equivalent THEN clauses. For example, the rule 'p(x) --> not(q(x))' is contradictory to the rule 'p(x) --> q(x)'.
Subsumed rules: two rules have the same results, but one contains additional constraints on the situations in which it will succeed. In LES, this means one or more THEN clauses are equivalent, but the IF part of cne rule contains fewer constraints and/or clauses than the IF part of the other rule. For example, the rule '(p(x) and q(y)) --> r(z)' is subsumed by the rule 'p(x) --> r(z)'.
Circular rules: a set of rules is a circular-rule set if the chaining of those rules in the set forms a cycle. For example, if we had a set of rules

'q(x) --> r(x)', (3) 'r(x) --> p(x)', and the goal is r(A), where A is a constant, then the system will enter an infinite loop at run time, unless the system has a special way of handling circular rules.

~2 2 POTENTIAL GAPS IN A KNOWLEDGE BASE.

The development of a knowledge-based system is an iterative process in which knowledge is encoded, tested, added, changed, and refined. This iterative process often leaves gaps in the knowledge base which both the knowledge engineer and the expert may have overlooked during the knowledge acquisition process. In LES, we have found three situations indicative of gaps in the knowledge base. These three situations, called (1) missing rules, (2) unreachable clauses, and (3) deadend clauses are described below.

Missing rules: a situation in which some values in the set of possible values (called legal values) of an object's attribute are not covered by any rule's IF clauses (i.e., the legal values in the set are covered only partially or not at all). A partially covered attribute can prohibit the system from attaining a conclusion or cause it to make a wrong conclusion when an uncovered attribute value is encountered during run time.

Unreachable clauses: in a goal-driven production system, a THEN clause of a rule should either match a goal clause or match an IF clause of another rule (in the same rule set). Otherwise, the THEN clause is unreachable.

Deadend clauses: to achieve a goal (or subgoal) in LES, it is required that either: (1) the attributes of the goal clause are askable (user provides needed information) or (2) that the goal clause is matched by a THEN clause of one of the rules in the rule sets applying to that goal. If neither of these conditions is satisfied then the goal clause can not be achieved, i.e., it is a 'deadend clause'. Similarly, the IF clauses of a rule also must meet one of these two conditions, or they are 'deadend clauses'.

~2 3 DEPENDENCY CHART AND CIRCULAR-RULE CHAINS DETECTION.

As a by-product of the rule checking, CHECK generates a dependency chart which shows the interactions among the rules and between the rules and the goal clauses. An example of a dependency chart for a small problem is shown in Figure 1 . A '*' indicates that one or more clauses in the matches one or more clauses in the THEN part of a rule. The dependency chart is very useful when the knowledge engineer deletes, modifies, or adds rules to the rule base.

Note that in Figure 1, the '*''s indicate the dependencies for the original rule set. By adding a clause to Rule 2, the '*2' dependencies appeared. Note, Rule 2 now references itself - a self-circular rule. By the addition of one clause to Rule 1, the '*1' dependencies appeared. This also causes the rule set to be circular, since an IF clause of Rule 1 is matched by THEN clauses of Rule 7 and Rule 8 which in turn match an IF clause of Rule 1 . Circular rules should be avoided since they can lead to an infinite loop at run time. Some expert systems, such as EMYCIN, handle circular rules in a special way. Nevertheless, the knowledge engineer will want to know which rules are circular. So, CHECK uses the dependency chart to generate graphs representing the interactions between rules, and uses a cyclic graph detection algorithm to detect circular rule chains.

~3 IMPLEMENTATION OF RULE CHECKER.

In solving a problem, the knowledge engineer may write several sets of goal-driven rules with each set having a unique subject category. (In LES it is convenient to put rules in different subject categories so that the system can solve different goals using only those rule sets which apply to that goal.) To solve a particular goal, often he will select several goal-driven rule sets and WHEN rules (demons) . Since these rule sets are generated over a period of time, it is quite possible that their interaction will cause some problems. Thus, for each goal it is necessary to compare the rules (in the rule sets specified by that goal) against each other and against the clauses of that goal. We now show an algorithm (in an Algol-like notation) which CHECK uses. The algorithm does the checking for a set of subject categories with N rules and a goal with G clauses.

Because an IF part or a THEN part can have more than one clause, the comparison between one part and another is handled by comparing a clause of one part to every clause in the other part.
These algorithms also work for forward-chaining rules, which are called WHEN rules in LES. However, the criterion unreachable clauses is not applicable to forward-chaining rules.

From our experiences with constructing different knowledge bases, we find that many changes and additions to the rule sets occur during the development of a knowledge base. Thus, a tool such as CHECK that can detect many potential problems and gaps in the knowledge base should be very useful to the knowledge engineer in helping him to develop a knowledge base rapidly and accurately.

When fire fighting, the process engineer's first priority is to get the line up and running again as quickly as possible. Fire fighting consumes about half his day, conflicting with his main responsiblities of coordinating line personnel and monitoring and controlling the process to meet specifications on uniformity of pattern thickness, develop, exposure, baking, and many other variables. Under these circumstances, temporary solutions usually result, with no time left for designing long-term solutions to recurrent problems.

An expert system, like the Photolithography Advisor, can help alleviate fire fighting, monitor and control the process, record process data that can help determine what the recurrent problems are, and provide information needed to coordinate line personnel on different work shifts.

The complex procedures needed to produce correct patterns by photolithography are many and varied. For an overview see (Thompson, et al, 1983). In addition, the number of ways a defective pattern can be produced is very large. When armed with this knowledge, the Advisor can help the technician focus on the data pertinent to isolating the defect and determining its cause and can recommend corrective action. Without the Advisor or the process engineer, the technician may search blindly for a very long time and become overwhelmed and frustrated.

~2 The Photolithography Advisor.

The Photolithography Advisor is written in HP-RL (Rosenberg, 1983), a knowledge representation language that supports structured declarative objects (frames), message-passing, active values (daemons), procedural values, a rule system for logic programming with agenda-based inference engines for forward and backward chaining, and a smooth interface with the host LISP environment. HP-RL runs on HP model 9000 series 200 workstations under both PSL (Griss, 1981) and Common-Lisp (Steele, 1984). The Advisor is a small to medium sized expert system. It builds on a number of well-known medical diagnosis programs, borrowing in particular some of their architectural ideas; (Shortliffe, 1976), (Aikins, 1984), (Kunz, et al, 1979), and (Smith and Clayton, 1980). The program currently contains 30 class-frames, with up to 100 slots each, and 300 rules, 30 forward-chaining and the rest backward-chaining.

The Advisor is structured as a team of three cooperating specialists that work together to solve problems posed by the user. An inspector works with the user to collect symptoms, passing this information on to a diagnostician. The diagnostician uses knowledge of chemistry and photolithography equipment to diagnose wafer problems and their causes and tells the repairman about them. The repairman then advises the user on how to correct the diagnosed problems. The specialists communicate through the frame database. In this sense, the Advisor uses a blackboard architecture (Erman et. al, 1980).

Any specialist can ask for more information when needed. This allows the consultation to be focused, eliminates asking for extraneous or irrelevant information that may be expensive in time and effort to gather, and results in a more human-like interaction.

The basic diagnostic strategy is to work with visual symptoms first. A visual inspection revealing no symptoms does not mean the wafer is free of defects. It only means that no gross defects can be detected by the naked eye. In

... Advisor requests the user to examine the wafer under the microscope at low magnification. If symptoms are found by this inspection, they must be verified using high magnification.

The user interface has turned out to be one of the most important components. Initially, we were more interested in problem-solving performance than ease of use, so a simple gloss tty interface was used. This was unsatisfactory. In response, we implemented an interface which required very little typing (all questions being asked through pop-up menus) and recorded all interaction in a transcript window. The user could then answer most questions by using either a single-key command, pressing a softkey, or selecting with the mouse. The consultation could be reviewed by scrolling back in the transcript window. We are now building another interface that implements a virtual advisor's notebook. It uses graphical representations of the wafer and processing equipment, sophisticated browsers for examining the knowledge-base, and a laser-disc database of symptom and defect images. While being as easy to use as the current interface, this interface will use more cognitively stimulating media to provide considerably more explanatory power.

~3 Performance and Evaluation.

The first version of the Photolithography Advisor was installed in a low-volume research lab and was used for about a year. At the time of this writing, we have just installed a second version of the program in a manufacturing line for additional tasting.

The test in the research lab helped us debug the knowledge base and obtain useful ideas on what was needed in the user interface. Overall, the program was considered quite useful and generally diagnosed problems as well as the available process engineers (the same ones who helped us build the program).

Some of the benefits of the program were.

Substantial time was saved in processing wafers. If a process error is not detected, reprocessing could take up to 8 hours; whereas running the program required less than 10 minutes.

The Advisor was always available, even during the night shift. Process engineers like to sleep at night.

The hardware emitted less particles than an engineer and could be installed in the clean room where it was convenient and available.

The hardware cost was less than $35,000; cheaper than a full-time engineer.

Some of the needs uncovered during this informal test were.

Most users were naive computer users. The interface must eliminate most typing and make correcting errors easy when typing is needed. We dealt with this in the next version by extensively using popup menus and customized softkeys, eliminating most typing. Where typing was still needed, customized editor modes were designed that limited the number of ways a user could get into trouble.

Questions asked by the Advisor used jargon to describe wafer defects. Menu items could easily be misinterpreted. We dealt with this terminology problem by adding alternate definitions for defects and coilected photographs to illustrate defects. By showing a photograph of the defect we were trying to describe,

...a misinterpretation of the text.

Experienced users were frustrated having to answer questions about visual symptoms when they knew a high-power inspection was needed. We accomodated them by structuring the menus to have default answers that provided an express-lane to more detailed inspection.

Despite our attempt to make the problem-solving knowledge generic, most fabrication lines will want to customize for local practice. Domain experts (with training from us) can now edit the knowledge base using a sophisticated set of browsing tools that do not require knowing the syntax of either HP-RL or LISP.

The program's problem-solving ability was inadequately tested. Measuring this ability was difficult since the research processes were continually modified.

Primarily because of the last point, we have started an extensive field test on a production line to more fairly and completely test it in its target environment than could be done in a research facility. This evaluation consists of an in vivo use of the program on a negative resist fabrication line, a user survey, and a comparision of key process statistics using before and after data.

The user survey has two parts. The first is administered with each consultation. Here we are trying to ascertain the program's accuracy and obtain detailed comments about the wording on menus, of advice, etc. The second survey is administered every month to collect more general comments about the friendliness and usefulness of the system.

Statistics measuring process performance include yield histories, rework rates, process hold times, and equipment downtime. Many of these are collected automatically by existing process management software. Thus, a large database of data on past performance of the process line exists. In cases where the necessary data is not logged by existing software, the Advisor automatically collects it so that it can be compared with manually reconstructed past data.

We have found that peopie not trained in AI can quickly learn to modify an expert system in their domain of expertise if given the right tools. But we have also found that they are limited if they are not an experienced AI programmer.

Domain experts tend to introspect about their problem-solving strategies in terms of plans. Even simple planning constructs are useful in helping them articulate their expertise.

Building an expert system has much in common with other types of software engineering. The programming paradigm is different, but experimentation, iteration, and testing are as important here as in other types of programming.

Meta-level rules can be useful initially to capture an expert's problem-solving heuristics. Once the problem is better understood, however, they can usually be replaced by specific object-level rules that solve the problem much better (and more efficiently).

Experts disagree. It helps to sample a lot of them. In our application, many problems can be solved several ways. Which solution is used often depends on personal preference or on established local procedures. Vocabulary can also be a problem. Different people will call the same thing by different names. This problem becomes acute when trying to design the user interface.

Paper 5: FAULT DIAGNOSIS THROUGH RESPONSIBILITY.

We introduce the 'theory of responsibilities' as an approach to automated troubleshooting. Using this approach, the understanding of how a circuit works is recorded by assigning responsibilities for parts of the output waveform to subsections of the circuit. These can be assigned manually ro derived from casual simulation.

Our current efforts are directed at automated troubleshooting of analog circuit cards. Our overall system, described in (Milne 1984) and (Ramsey 1984), is designed to automatically test and identify faults in an analog card through the use of automatic test equipment. Our past implementation and papers have conducted testing based on the structural description of the circuit. In this paper, our approach to functional testing is described.

This work differs from others in several significant ways. The work of (Cantone 1985) has been entirely within the structural area. His algorithm for deciding which test to perform based on the most information gained and possible cost is very helpful to structural reasoning, but can't help us once we can no longer probe within an sub-circuit. Cantone only uses structural information to isolate the fault to a single functional area. In this paper, we will show how structural information can be used to further propose faults.

(de Kleer 1983) is working to diagnose faults in analog circuit cards from 'first principles'. That is, given the low level electronic description of how a capacitor works, it should be possible to deduce how a filter would work, and consequently, diagnose faults in it. Although it is agreed that this is the most desirable approach, the author feels that there is too much work still to be done in order to use this approach in testing today. In our work, we start from 'second principles', that is, the type of description that an electronics engineer uses to describes various building blocks of circuits. Several examples are contained below.

(Davis 1983) has done much work in the area of digital troubleshooting based on the function of the components. He relies on computing the function and inverse of each sub-component. In general, it is not possible to compute the inverse of analog functions. Also the digital domain has a very simple output form (1 or 0), while the analog domain may have a very rich signal. Hence different techniques are called for. In fact the more complex output signal is one of the key differences between the analog and digital domains.

The work of (Chandrasekaran 1985) is most similar to our own. He describes the object to be diagnosed in a formal language and then compiles this description into a set of production rules to perform the diagnosis. In our approach, we use a different formal description and four simple rules production rules. Chandrasekaran's work has not defined a clear role for the interaction of functional and structural reasoning. In our work, structural reasoning plays a dual role. It is first used to isolate the possible fault to a single functional area. Secondly, whenever the output is zero, (giving no information), structural rules are used to propose possible faults.

~2 STRUCTURE.

The traditional role of structure is to isolate the possible fault to one functional module. The system first checks if the output is correct. If it is not, the path of the signal is traced back through the structure of the circuit and a test is chosen which will split the possible fault path in half. This is done until only one functional module is left.

In our work, structure is also used when the output is zero. In this situation, we have no information on which to base the functional diagnosis, so structural reasoning must be used. In an analog circuit, we are interested in the output current and voltage drop. Their relationship is controlled by Ohm's Law: E=IR. Qualitatively, we can see that if E is zero, then I will be zero; and that if l is zero, then E will be zero. We can also know that if R is zero, then E and I will be zero. Because of the product of IR, if E is zero we will need two rules, since I or R could be zero.

Let us look at voltage. We know that we need an R in order to get a voltage drop. If an output is shorted to ground, then R is zero, and hence no output voltage. We can translate this into a diagnosis rule.
Voltage Short Rule:
If one component connects the output and ground,
and the output is zero,
then that component may be shorted.

We could derive this rule from E=IR, but it can be 'compiled' to the above form. This compilation is similar to (Chandra 1985) and only needs to be done once for the fault diagnosis system. This rule is an example of the high level knowledge that an electronics engineer may use. We call this an example of a 'second principle'.

From Krichoff's Current Law, we know that the current flowing into and out of a node is zero. If there is no current flowing into a node, then there can be no current flowing out of the node. As before, we can compile these facts into a 'second principle'.
Current Open Rule:
If one component connects the input and output,
and the output is zero,
then that component could be open.

We also know that with no current, we will not get a voltage drop. If the output is zero, it could be from no R or no I, based upon Ohm's Law. Because of the dependency between the current and voltage, we cannot be sure, based upon a single zero output, whether we have a short or open, so several hypotheses may be produced.

We will use a simple voltage divider as an example of these structure rules. When the circuit is working properly, the outputs are 01 and 02. Their respective values are determined by the ratio of R1 and R2. If the value of 02 is wrong when 01 is correct, then the ratio R1/R2 is wrong. Qualitatively, we cannot tell which value is wrong.

If R1 is a short, then 02 will equal 01 and not be zero. This can be predicted by the application of the Voltage Short rule. (Note that ground being zero is only a special case). If R1 is open, then 02 will be zero by the Current Open rule. 01 may or may not be zero depending upon the circuit. If R2 is shorted, then 02 will be zero by the Voltage Short rule. If R2 is open, then 02 will be zero by the Current Open rule.

In this section, we have presented our approach to the use of structure for fault diagnosis. In the traditional way, it is used to isolate a fault to a single functional module. We also use it to propose faults based upon shorts and opens. These two rules are very powerful and alone will propose the correct fault for most of the examples we have encountered.

~3 FUNCTION.

The Rectifier Circuit: When the structural rules have isolated a fault to a single functional module, we turn to functional reasoning. To illustrate the theory of responsibilities presented in this paper, a simple rectifier circuit will be used. In second principles, the rectifier circuit can be described as follows. To build a rectifier circuit, use a diode to convert each peak in the input waveform to a positive output peak. For a typical sine wave, this means two diodes, one for the positive, and one for the negative peak. The output is then filtered. To do this a capacitor is used to store energy and a resistor to drain that energy.

From this description responsibilities can be assigned: each diode produces a peak in the output, one at the positive peak and one at the negative peak. The capacitor charges, giving us the rising ramp, and the resistor discharges the ramp. We can write this formally as follows.
The diagnosis rules:
If X is Y by Z and not(Y), then Z is bad;
If X is Y by Z and not(X) then use structure rules.

In this brief paper, the implementation details have been left out, although the PROLOG user will recognise the role for unification.

These are the only troubleshooting rules we need. To diagnose a fault, the output waveform is compared with the input waveform. The first rule will then identify which component is faulted.

The above description was tested on an actual circuit. When the resistor was faulted to the open position, the output was a constant level. Comparing the desired output from the real output, we are missing the declining ramp, so the resistor is bad. The DC voltage levels are important in this diagnosis, but have been omitted for this paper.

When one of the diodes was opened, the output was missing one of its peaks. By comparing the desired output with the real output and the input, the rule can identify correctly which diode has been opened. When the capacitor is opened, the rising output is not a ramp, so the capacitor is predicted bad. If the resistor is shorted, the output is zero, and the structure rules will correctly propose the fault. It should be note that we are lucky in this case that each component manifests it to a single part of the output waveform. In this brief paper, the voltage levels have been omitted. These are critical to diagnose some faults.

~4 DEEP FUNCTIONAL REASONING.

In the above sections, we have assumed that we understand how to assign responsibilities between the parts of the output waveform and the components of the circuit. In this section, we will see how this can be derived automatically.

The basic strategy is to simulate the working of the circuit through our 'second principles'. As each component makes a contribution to the overall output, a responsibility is assigned. We then use the above troubleshooting technique to trace the responsibilities to the faulted component.

We will use the rectifier circuit as an illustration. We start with the following second principles:
Diode: If the input is positive, the output equals the input;
Capacitor: A capacitor with a rising input will charge;
Capacitor: A charged capacitor with a load, and not a rising input will discharge;
Wire: When two wires meet, the output is the sum of the two inputs.

We also assume low level rules that provide for the addition of signal waveforms, the testing for rising and falling voltage levels and give the voltage across a capacitor. The input to the clipper is in the form of a sine wave from a transformer. When we apply the diode rule, The sine wave is transformed into a postive half-wave followed by zero. The second diode is in reverse polarity, so it produces a zero output followed by a half-wave. Responsibility to each half-wave is then assigned. Next, the two wires add, giving us a half-wave rectified waveform.

Now the signal arrives at the capacitor and the input signal is rising. By the capacitor rule, the capacitor now charges. The rise in the output voltage is assigned the capacitor. When the signal stops rising, the capacitor has a load, so discharges. This decreasing output voltage is assigned to the resistor.

In this very brief explanation, we have seen how the waveform gets built up from second principles and the responsibilities are assigned. Using this technique, it is possible to derive the responsiblities from a circuit and then perform troubleshooting based upon these.

If one has only a limited understanding of the circuit, then responsibilities can only be assigned in a limited way, and hence the capability to diagnose faults will be limited. If one has thorough understanding of the circuit, can do a better job of assigning the responsibilities. However, because of limitations within the domain, it will not always be possible to describe the circuit in such a way as to isolate faults to one component. This limitation, however, primarily comes from the limits of being able to diagnose a circuit as a black box. This technique has been applied to filters, rectifiers, voltage doublers and wave generators. Because of the modularity of most circuits, it works even on seemingly complex circuits, by dealing with the sub-circuits.