COMPUTATIONAL SEMANTICS

A Study of a Class of Verbs

John Nicholas Dorian Fisher

Submitted for the Degree of

Doctor of Philosophy

University of Aston in Birmingham

October 1974

ABSTRACT

This thesis starts by showing that the dominant linguistic theory of the last fifteen years, Transformational Generative Theory, has now been so criticized that it is no longer viable as a Theory of Language. An alternative theory is proposed, called the Functional Theory of Language (FTL). This theory should be extremely useful for the field of Artificial Intelligence, since it would allow computers to understand Natural Language in any context. FTL proceeds from the realization that Language is used to convey information from one person to another. A class of verbs, explicitly performative verbs, is distinguished. Use of such verbs in sentences displays unambiguously the intentions of the speaker (the information he wishes to convey). This information is carried by the presuppositions inherent in the verb, which limit the choice of verb to that which accurately reveals the attitudes of the speaker (his cognitive structure). Two computer programs have been written to test FTL, both of which only accept sentences containing explicitly performative verbs. (It is theoretically possible for any English sentence to be 'reduced' to sentences containing explicitly performative verbs.) The first program detects inconsistencies in the speech of various 'people'; that is to say, it tests how well understanding of English can be achieved according to FTL. The second program 'converses' with the user; that is to say, it tests how well English can be 'generated' according to FTL. The conclusions are drawn that the programs mark a significant advance in the field of Artificial Intelligence and that, on the basis of those programs, the Functional Theory of Language should be considered further by philosophers of Language.

ACKNOWLEDGEMENTS

I would like to thank all the academic staff of the Computer Centre at the University of Aston in Birmingham for the help they have given me in completing the work in this thesis. I would particularly like to thank my supervisor, Dr. F.G.Pagan, and Dr. A.J.Harget for their many helpful comments. The programming staff of the Centre, especially Mr. E.C.Richards, have been of great help in explaining the intricacies of the GEORGE 3 operating system. Finally, I would like to thank Dr. G.J.Shute for the philosophical talks that we have had and from which I have learnt so much.

The work of this thesis has been financially supported by the Science Research Council, under grant number B/71/1369.

CONTENTS

Abstract ii

Acknowledgements iv

I. Literary Diversions 1

1. TG Theory Expounded 2

2. TG Theory Attacked 13

3. Theory of Intentions 19

4. Presupposition 28

5. Conclusions 32

II. Computers and Language 35

1. Terminology 35

2. Syntax 37

3. Semantics 48

4. Conclusions 57

III. The Functional Theory of Language 59

1. Language 59

2. Cognitive Structures and Understanding 63

3. Presuppositions 68

4. Philosophical Implications of FTL 72

5. The Functional Theory and AI 81

IV. PIP 86

1. Facilities Available 86

2. Aims and Objectives 88

3. Syntactic Analysis 92

4. Semantic Analysis 102

5. The Verb "Implies" 113

6. An Example 116

V. PIP Talks 122

1. Aims and Objectives 122

2. PIP's Cognitive Structure 124

3. Responses 129

4. An Example 136

VI. Conclusions 141

1. Analysis of PIP's Performance 141

2. Consequences for Artificial Intelligence 145

3. Consequences for Linguistics 150

4. Consequences for the Philosophy of Language 159

Bibliography 166

Appendix A 175

Control Structure for Program 1 176

Alphabetical List of Functions Used in Program 1 180

Alphabetical List of Global Variables

Used in Program 1 184

Listing of Program 1 189

Sample Run of Program 1 221

Appendix B 228

Control Structure for Program 2 229

Alphabetical List of Functions Used in Program 2 231

Alphabetical List of Global Variables

Used in Program 2 233

Listing of Program 2 235

Sample Run of Program 2 252

"Philosophers try to bully us into saying that we know a number of things. And when on reflection we find that we do not, we pacify the philosophers by professing that we do know them, but only implicitly: thus having it both ways."

H.H.Price

I. LITERARY DIVERSIONS

The motivation behind the work contained in this thesis was a desire to propose and develop a Theory of Language. Part of a possible theory will be tested, the result of which will hopefully show that the whole approach is worth pursuing. Central to such a theory is, of course, the notion of Language. Language is the means by which human beings communicate with each other. The basic method of communication is speech: one person speaking either to another person, or to a group of people. The purpose of the speaker in speaking is to make his audience understand his thoughts, emotions, arguments or beliefs. This 'definition' of Language was not acceptable to the followers of the dominant linguistic trend during the last decade and a half, namely the Transformational Generative (TG) Theories of Noam Chomsky. These TG Theories had repercussions in such diverse fields as Philosophy of Language, Human Biology, Psychology, Psycholinguistics and Natural Language Processing by computer. The theory to be developed[1] arises out of the faults in TG Theory, and out of some theories of other philosophers that perhaps deserve more attention than has been

---

[1] See chapter III.

paid them. Section 1 of this chapter will therefore outline TG Theory as Chomsky and his colleagues have developed it,[2] whilst section 2 will show what is wrong with it. The remainder of the chapter will introduce, as an alternative to TG Theory, the concepts, terminology and ideas already proposed by other philosophers, which will form the basis of the theory to be developed later.

1. TG Theory Expounded

Aspects of the Theory of Syntax[3] is an account of TG work in progress at that time, taking regard of various criticisms of the position proposed in Chomsky [10] and advances in the realm of semantics as expressed in Katz and Fodor [38] and Katz and Postal [40]. It was to help contribute to a comprehensive description of Natural Language that would embrace the three main traditional departments of linguistics: syntax, semantics and phonology. The book was furthermore an attempt to correct misunderstandings of Chomsky's earlier position, particularly with regard to the notion of 'grammar'. "A grammar of a language," says Chomsky, "purports to be a description of the ideal speaker-hearer's intrinsic competence."[4] It is therefore with competence, said to be "the speaker-hearer's knowledge of

---

[2] The interested reader is directed to the following accounts for a more detailed exposition: Botha [6], especially pp. 18 - 47; Katz [41], chapter 4; Katz [42], pp. 31 - 122.

[3] Chomsky [12].

[4] Ibid., p. 4.

his language", as opposed to performance, said to be "the actual use of language in concrete situations", that Chomsky is interested.

"A fully adequate grammar must assign to each of an infinite range of sentences a structural description indicating how this sentence is understood by the ideal speaker-hearer."[5] Chomsky comes to the belief that for all sentences (grammatical strings of symbols) of a language there is some underlying deep structure which is this structural description. Thus, although the surface structure of the two sentences

(1)a. The boy saw the dog

b. The dog was seen by the boy

is different, their underlying structure is, at some point, the same. This belief is grounded first on the notion of 'simplicity', that is that it is simpler to describe Language in this way, and secondly on the fact that two superficially similar strings may have a different underlying structure, for example

(2)a. John is eager to please

b. John is easy to please.

For Chomsky, it is the syntax that provides this structural description, the deep structure underlying Natural Language. Semantics and phonology merely interpret the deep structure in such a way that the hearer understands a particular sentence. They "play no part in the recursive generation of sentence structures".[6] In fact, Chomsky's theory, although being an

---

[5] Chomsky [12], p. 4.

[6] Ibid., p. 141.

attempt to integrate the various aspects of linguistic description, actually confines each of them in a separate component. The syntactic component of the theory is primary, providing the input to both the semantic component and the phonological component.

1.1 The syntactic component of Chomsky's theory is divided into two sub-components: the Base sub-component and the Transformational sub-component. The Base consists of a series of (unordered) rewrite rules, known as Phrase-Structure (PS) rules or Immediate Constituent (IC) rules. These rules are of the form:-

(3) XAY $\rightarrow$ XBY

where A represents a single symbol (in this case some kind of syntactic category), B any non-null string of symbols, X and Y any string of symbols, possibly null, and "$\rightarrow$" is to be read as "can be rewritten as". In the special case where X and Y are null for all the rules of the Base, then the grammar is said to be context-free, otherwise it is context-sensitive.[7] Syntactic strings can be generated by following these rules through until no more apply. From the derivation of such a string, a tree diagram can be constructed showing how each symbol is replaced by others lower down the tree. This construction is known as the Base (underlying) Phrase-marker (P-marker) of some sentence which has its structure.

---

[7] There are some restrictions on the set of PS rules that may be chosen for any grammar. These are explained in Bach [4], pp. 35 - 36.

1.1.1 As well as PS rules the Base also contains a <u>lexicon</u>. In it are so-called lexical formatives which (after appropriate action by the phonological component) will become the words of the language. Associated with each lexical formative is a set of <u>syntactic features</u>. These are of three types:-

(4)a. Category Features e.g. +N

b. Inherent Features e.g. +Common -Abstract

c. Contextual Features e.g. + —— NP

Types (4)b. and (4)c. are also known as sub-categorization rules. Each lexical entry is thus a pair (D,C) where D is a lexical formative and C is some set of syntactic features associated with D; this set (C) can also be called a <u>complex symbol</u>. The result of applying the PS rules is a <u>pre-terminal</u> string which contains grammatical formatives (N, Adj. etc.) and complex symbols. A <u>terminal</u> string is formed from it by means of the following lexical rule:- "If Q is a complex symbol of a pre-terminal string and (D,C) is a lexical entry where C is not distinct from Q, then Q can be replaced by D".[8] This process has come to be known as <u>lexical insertion</u>.

However, the theoretical justification for such a lexicon is not strong. Its inclusion in the theory of Chomsky [12] is mainly for the purpose of blocking the generation of such non-sentences as

(5) *John frightens sincerity

(where the * indicates that the sentence is not a sentence of the language). This sentence is taken to be syntactically

---

8 Chomsky [12], p. 84.

rather than semantically anomalous.[9] This means that the lexicon must contain a lot of information about each lexical formative. In fact, Chomsky says, "in general, all properties of a formative that are essentially idiosyncratic will be specified in the lexicon. In particular, the lexical entry must specify: (a) aspects of phonetic structure that are not predictable by general rule ... (b) properties relevant to the functioning of transformational rules ... (c) properties of the formative that are relevant for semantic interpretation ... (d) lexical features indicating the positions in which a lexical formative can be inserted ... in a pre-terminal string."[10] Of these four characteristics of a lexicon, (c) and (d) have come in for criticism,[11] (b) is philosophically unjustifiable, and (a) considerably weakens the claim of 'universality' for the rules in each of the components of the Integrated Theory.

1.2 The Transformational Sub-component consists of a series of ordered transformational rules, each of which maps one P-marker into another. In particular, each rule has a <u>structural description</u> (S.D.) which specifies the class of strings (in terms of analysis by P-markers) to which the rule

---

[9] For a discussion of the merits and demerits of considering (5) to be syntactically anomalous, and for a discussion of this aspect of TG Theory, see Harrison [31], pp. 185 - 209.

[10] Chomsky [12], p. 87.

[11] See Botha [6], pp. 152 - 247.

may apply, and a <u>structural</u> <u>change</u> (S.C.) which shows the structure of the new transformed P-marker. In addition, each rule is classified as either optional or obligatory. For example, the Passive Transformation which maps (1)a. into (1)b. might look like:-[12]

(6) $T_{PASSIVE}$ OPTIONAL

S.D.	NP	T	V		NP
	1	2	3		4
S.C.	4	2 + be + en + 3		+ by +	1

Transformational rules are introduced into the grammar because they are more powerful than PS rules for describing "certain relations holding between structures in a grammar. [They are] introduced because such [rules] can do things which simpler rules cannot do (or can only do in a clumsy manner)."[13]

1.3 The semantic component of TG Theory has not been defined by Chomsky. It is his colleague at M.I.T., J.J.Katz, who has put forward a semantic theory that is supposed to interpret the terminal strings from Chomsky's Base grammar in some meaningful way. Katz and Fodor define semantics in the following way: "Semantics takes over the explanation of the speaker's ability to produce and understand infinitely many sentences at the point where grammar [syntax] leaves off".[14] (Unfortunately, this is, of course, no definition at all!) In particular, Katz

---

[12] The notation is similar to that used by Bach [4].

[13] Bach [4], p. 64.

[14] Katz and Fodor [38], p. 483.

and Fodor distinguish four different factors which characterize a speaker's semantic abilities:-[15]

(7)a. the ability to detect non-syntactic ambiguities in a sentence

b. the ability to determine how many ways a sentence may be ambiguous

c. the ability to detect semantic anomalies

d. the ability to paraphrase.

It is the job of a semantic theory to describe these abilities in terms of rules which the speaker may or may not know consciously, but which, according to the given theory, he knows tacitly.

1.3.1 The semantic component, in Katz's account, is divided into two parts: a dictionary and a set of projection rules. The dictionary entries consist of a finite number of sequences of symbols, each of which consists in turn of an initial sequence of syntactic markers, followed by a sequence of 'semantic markers', then, optionally, a 'distinguisher', and finally a 'selection restriction'. Diagram 1 (on page 9) might represent the dictionary entry for the word "bachelor".[16] In it, the syntactic marker is unadorned, semantic markers are enclosed in parentheses, distinguishers in square brackets, and selection restrictions in angled brackets. It is the semantic markers that are supposed to be the means of expressing the commonality

---

[15] Katz and Fodor [38], pp. 485 - 486.

[16] Diagram 1 is taken from Katz [39], p. 523.

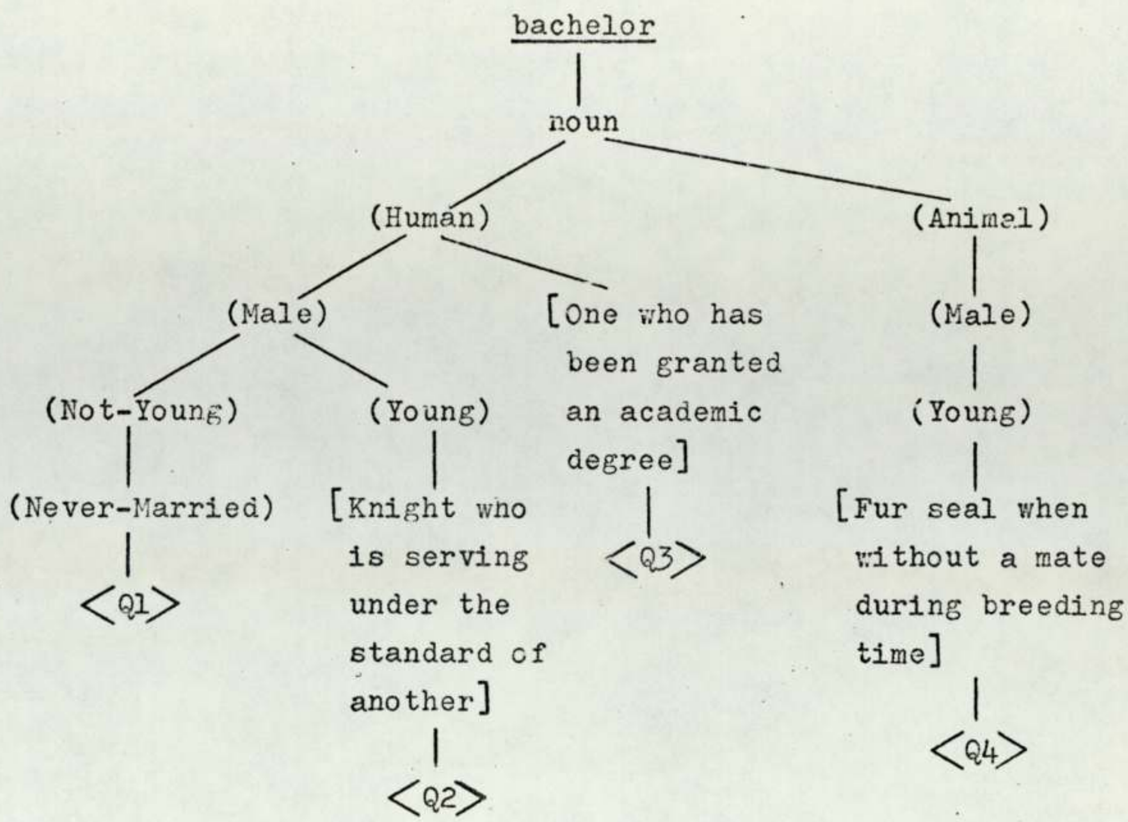


DIAGRAM 1

between words, and the distinguisher marks the idiosyncracies of each meaning.

1.3.2 The projection rules permit the readings for any one dictionary entry to be meaningfully amalgamated to those of another, following the information given in the underlying P-marker. No path will be chosen for amalgamation where the selection restriction would prevent it. The selection restrictions are usually composed, not of dummy symbols as in Diagram 1, but of syntactic or semantic markers, often combined in some way using logical operators. They are the means of determining semantically acceptable combinations (cf. the role of the syntactic features of the lexicon). In the sentence

(8) The bachelor put on her gown

it is clear that only the third 'meaning' of "bachelor" in Diagram 1 is intended. It is the selection restrictions which must ensure that only that reading is finally accepted as a legitimate reading for (8).

1.4 That concludes this review of what one might call the standard version of TG Theory. It has not altered substantially since then. The reason for going to such lengths is that anyone wishing to propose a theory of language that might have what has been called 'psychological reality' must take note of an anomaly in the literature regarding this aspect of TG Theory. Chomsky is at pains to make clear that TG Theory is only one way of describing the linguistic data at hand. He says, "to avoid what has been a continuing misunderstanding, it is ... worth while to reiterate that a generative grammar is not a model for a speaker or for a hearer."[17] Bach makes a similar point when he says "transformations are often erroneously conceived to be direct descriptions of processes that a speaker follows in constructing sentences .... The investigation of how speakers actually construct or understand sentences is properly the concern of psychology or psycholinguistics."[18]

On the other hand, however, Chomsky also says "no doubt

---

[17] Chomsky [12], p. 9.

[18] Bach [4], p. 64.

a reasonable model of language use will incorporate, as a basic component, the generative grammar that expresses the speaker-hearer's knowledge of the language."[19] Katz talks about a child in terms of a "rationalist-transformationalist" theory about the child's innate ideas. He says "[the child] is assumed to know, innately, that the grammar of the language has the form of a transformational grammar as given in phonological, syntactic, and semantic theory, i.e. the form of the rules in each component, the constructs out of which actual rules can be formulated, and the principles for organizing such rules into a system."[20]

1.5 Indeed, it is in dealing with the child that TG Theory has some of its strongest adherents among psychologists. McNeill [62] assumes, even before he has written a word, that a child has 'grammar'. He even goes so far as to say that children formulate one 'grammar' by the age of 28 months and another by the age of 36 months. This would not be that bad, if it were not for the fact that the second 'grammar' bears no relationship to the first except that both use PS rules. Moreover, he says, "How are transformations learned? The process remains one of the major mysteries in the acquisition of language. Children seem unable to avoid forming relations between underlying and surface structures."[21] Yet no evidence

---

[19] Chomsky [12], p. 9, italics added.

[20] Katz [42], p. 140.

[21] McNeill [62], p. 82.

has been brought forward by McNeill for stating so categorically that children do work with transformational grammars. He has merely shown that adults can characterize the utterances of children using transformational grammars (albeit not very well). Menyuk, working with the same assumptions as McNeill, says, "when the child decides that an utterance is a possible sentence of the language, he then analyzes the sentence by attempting to match the structure of the utterance to structures in his grammar."[22] This requires that a child "decide" upon some non-linguistic criteria whether an utterance is to be analyzed for a "decision" as to its grammaticality, which is surely one decision too many, at the very least!

Nevertheless, despite these intrinsic objections to both McNeill's and Menyuk's arguments, they and other psycholinguists obviously believe that the child, and therefore the adult, interpret language in a transformationally generative way. Thus, for anyone wishing to propose an alternative theory of language, it is very necessary to demonstrate the faults in TG Theory as a viable theory of how humans do use language. It is not denied that the syntactic component of Chomsky's grammar is an excellent way of characterizing some of the relationships between utterances. What is denied is that it can be an excellent way of characterizing man's understanding of his fellow man.

---

[22] Menyuk [55], p. 160.

2. TG Theory Attacked

There are two kinds of objections to the standard TG Theory: those that are internal to the Theory, and those that are external to the Theory (stating, in fact, that it has no claim to the name 'Theory of Language'). The kernel of most of the internal objections (i.e. objections to the way in which parts of the Theory work) is supposed autonomy and/or primacy of syntax. The objectors, particularly Lakoff and McCawley, have tried to show that no distinct boundary can be drawn between syntax and semantics and hence that there is no entity which can be called 'deep structure'. (A corollary of this is that the argument as to whether semantics should or should not be interpretive becomes totally irrelevant.)[23] Other objections to TG Theory are concerned with the argument that the Theory is not 'elegant', that it has areas of duplication.

2.1 The most obvious of these areas of duplication is pointed out by Weinreich [85]. According to the Chomsky and Katz position it is necessary to have a dictionary in the syntactic component (the lexicon) and another (separate) dictionary in the semantic component. If one considers the vast amount of data that has to be stored in a dictionary, then such a duplication is clearly undesirable. Admittedly, this will not cause TG adherents to abandon their position wholesale, but it

---

[23] For a discussion of some of the problems raised by maintaining that the semantic component should be purely interpretive, see Partee [65].

should be remembered that one of Chomsky's criteria for adopting one grammar over another (assuming them to be equally adequate at describing any given language) is that of 'simplicity'.[24]

2.1.1 A further duplication, pointed out by Langendoen [47] among others, is that the Base Sub-component of the syntax generates many 'deep structures' that have no surface structure representation and must therefore be blocked by restrictions placed on the application of certain transformational rules. This process is usually called 'filtering'.[25] It is, however, also the case that some sort of similar filtering must take place in the semantic component, in order that semantic anomalies should not occur (the work, in fact, of the projection rules).

2.1.2 Weinreich [85] also attacks the notion of a semantic marker, arguing that the set of Katz's semantic markers is in fact infinite. Weinreich's example is that of the activity denoted by the verb "to eat", which would require different semantic markers according to whether one were eating bread, soup (in some dialects), apples, peanuts, peas or spaghetti. He calls this '<u>Infinite Polysemy</u>'. It is a disease not of a small subset of English words, but of the vast majority. Although some solutions have been proposed (e.g. by Ziff [93]),

---

[24] See, for example, Chomsky [12], pp. 37 - 40.

[25] <u>Ibid.</u>, pp. 138 - 139.

they are themselves not free from objection according to Weinreich. Moreover, this particular disease leaves the status of the semantic marker in extreme jeopardy. There remains no empirical method of discovering what should and what should not constitute such markers. Yet if they are not to be independently justified in some such manner, then no conclusions can be drawn from them about the 'meanings' of words. They will have merely been chosen arbitrarily.

2.2 The arguments of McCawley and Lakoff are at the same time both more detailed and more fundamental. They attack the belief that there is some natural breaking point between syntax and semantics and that that point marks off 'deep structure'. McCawley [62] shows first that Chomsky would consider the 'sentence'

(9) *That idea is green with orange stripes

as syntactically deviant, violating the sub-categorization rules (see (4)c.). He further points out that if such 'rules' really were syntactic in nature then

(10) I dreamed that my toothbrush was pregnant

should also be a syntactically deviant sentence (since the word "pregnant" should require that its noun-phrase have the associated complex symbol +Human). This shows that the sub-categorization rules do not restrict one's use of Language in the way claimed by Chomsky, "since 'violations' of them are quite normal in reports of dreams, reports of other people's beliefs and science fiction stories".[26] McCawley thus believes

---

[26] McCawley [62], p. 219.

that the restrictions on the well-formedness of (9) and (10) are semantic in nature, whilst noting that even dreams are not completely free as to the possibilities of combining semantic material. For example, only a quantity of time can elapse, thus

(11) *I dreamed that my toothbrush elapsed

is still a deviant sentence.

2.2.1 This is McCawley's first line of argument. The second lies in his claim that both the syntactic and semantic derivational trees can be represented by a form of symbolic logic. Lakoff [45] showed how traditional categories of symbolic logic could be reduced to others; for example, it is possible to consider quantifiers as two-place predicates, where one place corresponds to a propositional function and the other to a set. McCawley, following Lakoff, shows that many of the so-called syntactic categories of the Base have no independent justification for their existence (cf. Weinreich and semantic markers), other than the fact that they 'trigger' certain transformations. When these categories have been eradicated, there is, according to McCawley, a one-to-one relationship between the remaining categories and the categories of symbolic logic. Since he also claims that semantics can be represented in the same way, syntax and semantics can be 'generated' together rather than sequentially.

2.2.2 To show that there is no level of 'deep structure' is also the purpose of Lakoff [46]. However, there is no room here to go into his arguments. Suffice it to say that he concludes:-

(12)a. that there is no level of 'deep structure'

b. that lexical insertion cannot be completed 'all in one go' either before or after the application of the transformational rules

c. that semantics has a better claim to primacy among linguistic categories than syntax.

2.3 The external objections to TG Theory have been pointed out by Shute [77]. In order to understand them, it is necessary to determine what kind of attitude one has to a theory; namely, whether one believes it to be a set of true statements about the world, and hence whether one believes that the concepts appearing within the theory designate real or existing entities. If this position is maintained, then it follows that the theory should comprise:-

(13)a. a formal calculus which is not interpreted

b. an intended interpretation for this calculus

c. correspondence rules which assign empirical content to the calculus and the intended interpretation thereof

d. a set of predictions subject to experimental verification which are the deductive consequences of (a), (b) and (c) together.

It is clear from sections 1.4 and 1.5 of this chapter that Katz, McNeill and Menyuk, at least, believe that TG Theory does indeed make true statements about the world, and that the concepts of TG Theory ('deep structure', 'transformational rules', 'semantic marker' etc.) do refer to actual entities. Language itself corresponds to (13)a, and TG Theory corresponds to (13)b. There is, however, a problem with regard to the assigning of empirical content to the description of Language. The syntax has never been anything but a descriptive mechanism. Chomsky says, "By a generative grammar I mean simply a system of rules that in some explicit and well-defined way assigns structural descriptions to sentences."[27] That leaves the semantics to provide the empirical content of Language. Weinreich, however, has shown convincingly that there is no empirical justification for the semantic markers, which in turn means that there is no empirical 'meaning' that can be attached to them. Hence (13)c is not satisfied, and therefore (13)d cannot, by definition, be satisfied. TG Theory cannot therefore be regarded as saying anything about the world: it is at best a (good) formal system for describing linguistic data.

2.4 All of which goes to show that if TG Theory is considered as a formal system for describing linguistic data, then hardly anybody is going to argue, but that as soon as it is claimed to have the status of a Theory of Language, then it has got to prove a great deal more than it has at present. All of which also means that a viable Theory of Language will have to be sought elsewhere.

---

[27] Chomsky [12], p. 8.

3. Theory of Intentions

TG Theorists, whether syntax-oriented like Chomsky, or semantics-oriented like McCawley, have one common belief: the 'meaning' of a sentence can be 'discovered' by looking solely at that sentence itself. The 'underlying deep structure' that most of them study is still a part of that sentence. Yet many philosophers believe that there is something even more fundamentally relevant to the understanding of a sentence; that is the message (information) that a speaker or writer wishes to convey to his audience. Language for these philosophers is not an interesting though isolated phenomenon, but rather the means by which human beings communicate with one another. For them, the desire to communicate is just as important as the communication itself. This desire to communicate is often called the intention on the part of the speaker to impart certain information. Thus Searle says, "The unit of linguistic communication is not, as has generally been supposed, the symbol, word, or sentence ... but rather the production or issuance of the symbol or word or sentence in the performance of the speech act .... Furthermore, not only must I assume the noise or mark to have been produced as a result of intentional behaviour, but I must also assume that the intentions are of a special kind peculiar to speech acts."[28]

3.1 Grice [27] (although himself believing that language is

---

[28] Searle [75], pp. 16 - 17.

self-justifying) is led from a study of the verb "to mean" to an analysis in terms of intentions. He contrasts such pairs of sentences as:-

(14)a. Those spots mean measles

b. Those three rings on the bell (of the bus) mean that the bus is full.

In cases like (14)a. it is not possible to say (truthfully) "those spots mean measles, but he hasn't got measles". That is to say that in sentences of type (14)a. "x means p" entails p. On the other hand, it is perfectly possible to say "those three rings on the bell mean that the bus is full; but it isn't full - the conductor made a mistake". Thus in sentences of type (14)b. "x means p" does not entail p. This is just one of five distinctions that Grice draws between the two types of sentence. He calls the first type of "mean" 'natural meaning' and the second type 'non-natural meaning'. It is of non-natural meaning that Grice's subsequent 'analysis' treats.

3.1.1 This 'analysis' has been very neatly condensed by Strawson [81]. Grice says, according to Strawson that a speaker, S, non-naturally means something by an utterance x if S intends (i_1) to produce by uttering x a certain response (r) in an audience, A, and intends (i_2) that A shall recognize S's intention (i_1) and intends (i_3) that this recognition on the part of A shall function as A's reason, or a part of his reason, for his response r. Strawson himself, however, believes that one must add the further intention (i_4) on the part of S that A should recognize S's intention (i_2).

3.2 Austin [3] arrives at a similar position, also by considering utterances and verbs. He begins by questioning an antithesis — between constative utterances (statements) which have the property of being either true or false, and performative utterances which can never be either. The performative utterance "has its own special job, it is used to perform an action. To issue such an utterance is to perform the action".[28] Examples of performative utterances are:-

(15)a. I name this sword "Excalibur".

b. Shut the door!

c. I promise to take you to the zoo tomorrow.

Although performative utterances cannot themselves be said to be true or false, they are nevertheless not "exempt from all criticism",[29] as Austin puts it. He distinguishes three ways in which the situation might not be appropriate for the utterance of a particular performative. The utterance is then said to be 'unhappy' (or 'infelicitous', as he also called it).

3.2.1 First, a performative utterance may be 'null and void', as when the speaker is not in a position to perform the said act, or when the object of the act is unsuitable for the purported performative. Thus, for instance, a bigamist does not get married a second time, he only goes through the motions; similarly, a table cannot be baptised.

Secondly, a performative can be 'unhappy' if it is uttered

---

[28] Austin [3], p. 242, italics in the original.

[29] Ibid., p. 243.

insincerely. If one utters the sentence (15)c. but without in the least intending to carry out the promise, then one has "abused the formula"[30] of promising.

According to Austin, the third way in which a performative utterance can be 'unhappy' is 'breach of commitment'. The performative act may have been performed normally and sincerely, but some later events may happen which are not in order. Thus, if John utters (15)c. but on the morrow breaks his leg, so that he cannot take you to the zoo, then he will have broken his commitment, despite having made the promise quite sincerely at the time.

3.2.2 Having now become acquainted with the notion of a performative, it would be useful if some criterion, either grammatical or lexical, could be found that would determine whether a given utterance were performative or not. Unfortunately, there is none. What do exist, in English at least, are two, so to speak, 'normal forms' in which the performative is expressed. One is the first person singular of the present indicative active (e.g. "I promise ..."); the other is the third person present indicative passive (e.g. "Passengers are requested ..."). Austin calls this the explicit form of the performative, and verbs like "promise" (where to say "I promise ..." is to perform the act of promising) can be called explicitly performative verbs. Although agreeing that not all performative utterances are explicit (cf. (15)b.),

---

[30] Austin [3], p. 243.

Austin does say "We may hope, all the same, that any utterance which is in fact performative will be reducible (in some sense of that word) to an utterance in one or other of our normal forms".[31]

As an answer to his original question, Austin concludes that it is in fact wrong to draw a dividing line between constative and performative utterances, that the two are very similar.

3.2.3 The author now wishes to turn to Austin [2] and to introduce more new terminology. The act of saying something, uttering certain noises or uttering words in a certain construction, Austin calls the performance of a '<u>locutionary</u>' act. This is thus equivalent to a (not necessarily meaningful) utterance in other terminology.

When a locutionary act is performed, speech is used, but used for a certain purpose; for telling stories, promising, pleading, threatening, joking, reprimanding and so on (cf. Wittgenstein [90]). The particular use of an utterance (locution) is the '<u>illocutionary</u>' act of that utterance. Austin explains this kind of speech act as "the performance of an act <u>in</u> saying something as opposed to performance of an act <u>of</u> saying something; ... and I shall refer to the doctrine of the different types of function of language here in question as the doctrine of 'illocutionary forces'."[32]

---

[31] Austin [3], p. 245.

[32] Austin [2], p. 99, italics in the original.

There is yet another kind of speech act that Austin distinguishes, namely a '<u>perlocutionary</u>' act. Saying something will often have a certain effect upon the feelings or thoughts of the person who hears the utterance; this effect may have been intended by the speaker, but the speaker may refer only obliquely or even not at all to the performance of the illocutionary act. It could be said that this is the performance of an act <u>by</u> saying something.

It will be seen that both illocutionary and perlocutionary acts are proper, mutually exclusive subsets of locutionary acts. An example of the different kinds of act (as performed by John) might be given by the following:-

(16)a.	Locutionary Act	John said "Shoot her!"
b.	Illocutionary Act	John urged (advised, ordered etc.) me to shoot her.
c.	Perlocutionary Act	John's saying "Shoot her!" caused me to shoot her.

One might differentiate (16)b. from (16)c. by saying that a perlocutionary act causes the state of mind of the hearer to alter, thus causing him to do something (in this case, shoot her), whereas an illocutionary act is merely an expression of the speaker's attitudes, even though they may include a wish that the hearer's attitudes should alter. It is now, however, necessary to distinguish between the illocutionary <u>force</u> of an utterance and the illocutionary <u>act</u> itself.

3.2.4 It has already been shown in what ways performative utterances might be 'unhappy'. There is, however, yet another way in which the (now) illocutionary acts can be 'unhappy', and that is if the hearer of the locutionary act does not understand the speaker's intentions and therefore the force of the locutionary act. Thus, in Austin's words, "the performance of an illocutionary act involves the securing of uptake".[33] The illocutionary force of a performative utterance is therefore always present; the illocutionary act, on the other hand, will only be successfully completed when the audience has understood that force. Although the illocutionary force of a performative utterance is not always clear, the illocutionary force of an explicitly performative utterance is perfectly obvious. If one utters "I promise ...", then the illocutionary force is that of promising (unless, of course, the utterance has been made insincerely). Securing of uptake is therefore easier for the hearer, if the speaker uses explicitly performative verbs.

The wheel has now moved full circle, for the illocutionary force of an utterance can be seen as Strawson's intention (i_1) and the successful performance of the illocutionary act requires that at least Strawson's intention (i_2) be satisfied, if not (i_3) and (i_4) as well.

3.3 Searle [75] takes the doctrine of illocutionary forces even further. He agrees with TG Theorists that Language is rule-based, but his rules are very different from TG rules.

---

[33] Austin [2], p. 116.

He distinguishes between regulative and constitutive rules. Regulative rules, he says, "regulate ... independently existing forms of behaviour", whereas constitutive rules "create or define new forms of behaviour".[34] Examples of regulative rules are "Officers must wear ties at dinner", or "When submitting a thesis, type on one side of each leaf only, with a margin at the binding edge of at least 35mm. and margins elsewhere of at least 15mm.". As an example of a set of constitutive rules Searle cites the rules of football or chess, which, as it were, do not merely govern the actual playing of those games, but in fact create the very possibility of playing them.

Searle's hypothesis is then clearly stated: "The semantic structure of a language may be regarded as a conventional realization of a series of sets of underlying constitutive rules, and that speech acts are acts characteristically performed by uttering expressions in accordance with these sets of constitutive rules".[35] To show what he means by the constitutive rules underlying Language and how he derives them, Searle gives an extended example (that of promising). These are those "semantical rules for the use of any illocutionary force indicating device Pr for promising":-[36]

Rule 1. Pr is to be uttered only in the context of a sentence (or larger stretch of discourse) T, the utterance of which predicates some future act A of the speaker S.

---

[34] Searle [75], p. 33.

[35] Ibid., p. 37.

[36] Ibid., pp. 62 - 63.

Rule 2. Pr is to be uttered only if the hearer H would prefer S's doing A to his not doing A, and S believes H would prefer S's doing A to his not doing A.

Rule 3. Pr is to be uttered only if it is not obvious to both S and H that S will do A in the normal course of events.

Rule 4. Pr is to be uttered only if S intends to do A.

Rule 5. The utterance of Pr counts as the undertaking of an obligation to do A.

One of the important consequences of analyzing illocutionary acts in this way is that Searle is in a position to show (where Austin [3] merely believed) that the act of referring (cf. constative utterances) is just as much an illocutionary act as is that of promising. He is able to deduce the rules for the use of any illocutionary force indicating device R for referring in exactly the same way as he did for Pr for promising.

3.3.1 Thus, a theory of intentions is the beginning of a Theory of Language which states that the meaning of an utterance is not completely determined by the way in which each individual element within that utterance is syntactically combined with the other elements of that utterance. Rather, the meaning of an utterance can only be understood when one has secured uptake of the various illocutionary force indicating devices implicit in that utterance (where each device has rules to help one recognize them).

4. Presupposition

Katz and Fodor [38] were convinced that any theory about the 'meaning' of an utterance could not take into account the whole of the socio-physical context in which the utterance was made. This was mainly because any such theory would, in their opinion, have to represent the total sum of knowledge about the world that both speaker and hearer possessed. This is, however, not the case. What is necessary is some method of representing how each utterance reflects some part of the knowledge about the world that both the speaker and the hearer possess. It is most important that a semantic theory should be capable of just such an explanation.

It has already been said that an utterance is 'null and void' if made in inappropriate circumstances. To say, for example, "I take thee for my lawful wedded wife" presupposes that certain conventions hold, viz.:-

(17) a. the speaker is male and the hearer female
b. (in the West) the speaker is not already married
c. (in the West) the hearer is not already married
d. both speaker and hearer are in a location where they can be married
e. there is somebody else present who is empowered to marry the speaker to the hearer.

All this is understood as involved with the illocutionary force of the act of marrying. '<u>Presupposition</u>' has therefore come to be studied rather more deeply by recent linguists than was previously the case.

4.1 Fillmore [18] in fact makes a distinction between the presuppositional and the illocutionary level of analyzing an utterance. These are, respectively, the implicit and explicit levels of communication. He goes on to analyze a number of verbs of judging in this way, using various new terms for identifying the 'entities' needed in the description of the conditions under which it would be appropriate to use the particular verb.

The term '<u>situation</u>' is used when it is necessary to refer to any situation, action, deed or state of affairs.

If the situation referred to can ~~be~~ favourably or unfavourably affect some individual, then that person is the '<u>affected</u>'.

If it is relevant to ask whether some individual is responsible for either bringing about the situation, or allowing the situation to occur, then he is called the '<u>defendant</u>'.

There may be some person who makes some kind of moral judgement about either the situation or the defendant's responsibility for it. He is called the '<u>judge</u>'.

If a verb refers to a locutionary act, then the actual verbal content of that act is included in the analysis as 'X' in quotation marks.

One of Fillmore's examples is:-

(18) ACCUSE [Judge, Defendant, Situation (for)]

Meaning: SAY [Judge, 'X', Addressee]

X = RESPONSIBLE [Defendant, Situation]

Presupposition: BAD [Situation]

This is contrasted with:-

(19) CRITICIZE [Judge, Defendant, Situation (for)]

Meaning: SAY [Judge, 'X', Addressee]

X = BAD [Situation]

Presupposition: RESPONSIBLE [Defendant, Situation]

Presupposition: ACTUAL [Situation]

These analyses[37] show that a speaker of English uses the verb "accuse" when talking about a situation which is unquestionably bad and when he wants to claim that a certain person was responsible for that situation. "Criticize", however, is used when there is no doubt as to the identity of the person responsible for the situation, but the speaker wishes to claim that the situation was blameworthy. Thus, the difference in 'meaning' between the two verbs is seen to be a matter of distinguishing between the implicit and explicit levels of communication.

L.2 Elsewhere, Fillmore[38] has called these presuppositions <u>properties</u> of the verb. For this reason, he came under attack from Garner [26]. Fillmore says that one of the happiness conditions of the sentence

(20) Please open the door

is that the door shall be closed at the time the sentence was uttered. If not, then the illocutionary act "has gone wrong in

---

[37] Taken from Fillmore [18], p. 288.

[38] Fillmore [20].

some way".[39] However, says Garner, if this is a property of the verb "open", then how is it possible to account for (21)?

(21) If the door is not already open, go and open it.

According to Garner, there is no presupposition in (21) that the door is closed.

This argument assumes, however, that both parts of (21) must be interpreted at one and the same time, and that the interpretation of the whole sentence can be meaningful only when both parts have been so interpreted. Quite clearly, sentences like (21) are not interpreted in one step; it is necessary to semantically analyze the if-clause first. Let P(open) stand for "the properties of the verb 'open'". Now (21) can be interpreted by the hearer as

(22) If P(open) satisfies situation, then DO(open), else NIL.

This formula, which is very similar in structure to that of some high-level computer programming languages, can be interpreted as the algorithm represented by the flowchart in Diagram 2, on page 32. The symbol 'DO' in (22) represents the illocutionary force indicating device of commanding, hence the box labelled A in Diagram 2. It can be seen that this box will only be reached if the presuppositions of the verb "open" are met, thus the illocutionary act has not misfired, even if the presuppositions of "open" are not met. In that case, one simply does nothing ('NIL' in (22)). This, after all, is only what one would expect to understand by (21) in the first place.

---

39 Garner [26], p. 40.

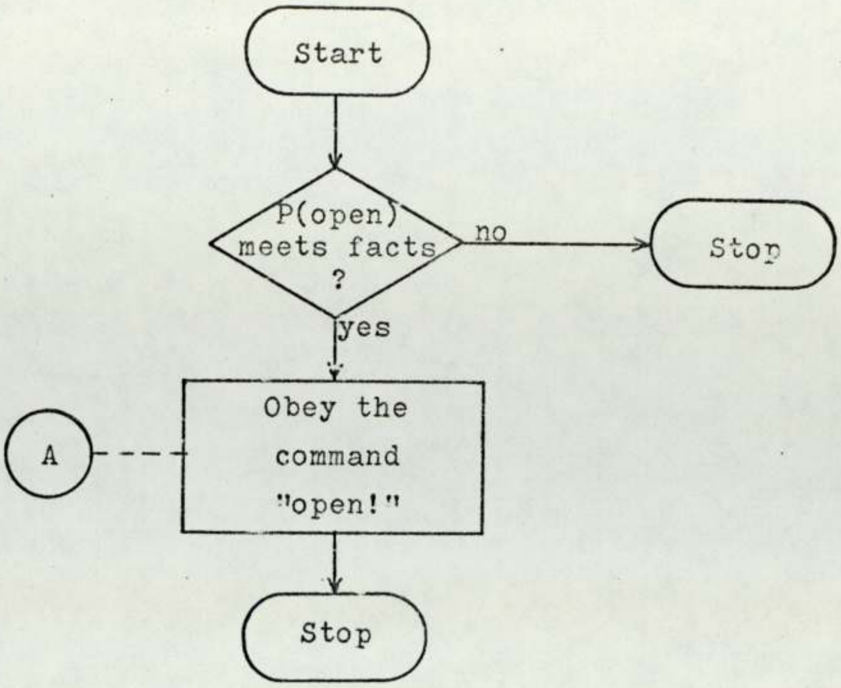


DIAGRAM 2

5. Conclusions

It has been shown that TG Theory is not a viable Theory of Language. There are two main reasons for this. First, the syntactic and semantic components of the theory cannot be separated in the required way. Secondly, there is a lack of empirical content to the theory, since the semantic markers have no independent justification. The proposed theory fails, therefore, both from within and without, to be a characterization of the way in which human beings understand Natural Language.

It has also been shown that an alternative method of explaining people's understanding of Language already exists. On this view, Language is seen as a means of communicating the attitudes of one person, the speaker, to one or more people, his audience. The particular terms in which the speaker couches his intentions may not be understood by his audience, but, provided they have been uttered 'correctly', they will be a representation of his attitudes (the illocutionary force of his utterance). The notion of presuppositions inherent in verbs is extremely important, when one remembers that the illocutionary force of an utterance is most clearly spelled out in explicitly performative utterances. The presuppositions inherent in the explicitly performative verb of such an utterance will be equivalent to rules for such an illocutionary force indicating device.

An attempt to combine the two theories has been made by Travis [84]. He attempts to establish a generative theory of illocutions. However, his 'rules' are context-sensitive rewrite rules. The 'generative' part of the theory is restricted to this sense of a directed branching from an initial node I for Illocution. The whole theory suffers from the basic defect that it could not possibly be used by anybody without a prior knowledge of the language concerned. There is no way of determining when a given utterance has been used with one particular illocutionary force, or with another. Indeed, there seems no way to stop any sentence, on Travis' theory, from being uttered with every possible combination of illocutionary forces that he can imagine. This is clearly not the way in which speakers of a language understand that language.

This thesis will not attempt to combine the two theories, but will concentrate on developing a theory based on the notion of presuppositions inherent in explicitly performative verbs.[40] It will be developed in conjunction with a computer program representing such a theory. It is hoped that this will

(a) help to systematize the details of such a theory

(b) provide some evidence that understanding, even by a machine, can be accomplished on the basis of such a theory.

Since the theory will be modelled by a computer, it will have relevance for the fields of Artificial Intelligence and Computational Linguistics. The next chapter will show what has already been achieved in those fields with regard to Natural Language processing, and why a theory of illocutionary forces is just as important for those fields as for the Philosophy of Language.

---

[40] See chapter III.

II. COMPUTERS AND LANGUAGE

The first section of this chapter is devoted to a clarification of what is meant by Artificial Intelligence throughout this thesis. The rest of the chapter will review work in the various fields of computers and language in as far as it has a bearing on Artificial Intelligence.

1. Terminology

Since the early nineteen-fifties people working with computers have been intrigued by the possibilities of processing natural languages by means of their machines. Several badly defined, overlapping fields have grown up with names such as Machine Translation (MT), Computational Linguistics (CL), and (part of) Artificial Intelligence (AI).[1] The whole subject might even be amalgamated under the heading of Natural Language Processing!

1.1 "The scientific goal of research work in artificial

---

[1] For good surveys of early work in the various fields, see Josselson [35], Montgomery [61] and Simmons [78].

intelligence," says Michie, "is the development of a systematic theory of intelligent processes wherever they may be found."[2] Thus, included under AI is the theory of game playing by computer, computer modelling of neural networks, generalized problem solving, theorem proving, pattern recognition and machine understanding of Natural Language. Indeed, any work whereby machines achieve certain goals is said to be part of that 'intelligence' which distinguishes Man from other animals, such as the ability to manipulate one's hands to manufacture tools, buildings, statues and so on. Lighthill [50] has argued that the future of AI is dead if it continues to try and build a General-Purpose Robot, which is an avowed aim of two of the main centres of AI in the world, Edinburgh and Stanford. It is this author's belief that Lighthill's views are probably correct, unless computers can be taught to understand Natural Language (or to model an understanding of Natural Language) in all contexts. Michie, talking about MT, says, "Language interpretation has been the graveyard of many well-financed projects for 'machine translation'. The trouble proved to be the assumption that it is not necessary for the machine to 'understand' the domain of discourse".[3] It is just as important that general-purpose robots 'understand' discourse in their domain, that is <u>any</u> domain. For this reason, the term AI will be used throughout the remainder of this thesis solely as a shorthand for the machine 'understanding' of Natural

---

[2] Michie [57], p. 507.

[3] Ibid., p. 508.

Language, or for that body of research which seeks inferences about human understanding of Language from computational models of understanding.

Computational Linguistics, on the other hand, would appear to be the study of the structural relationships appertaining between different parts of Natural Language, where the relationships are systematized for use by a computer and are thereby checked by it. Both syntax and semantics have been treated quite extensively in this way (phonology has been treated to a much lesser extent). In this sense, MT has therefore been a subset of CL, since no claims are made as to the psychological reality of the methods used in analyzing and generating each of the given languages.

2. Syntax

Of course, computers have to be able to syntactically analyze high-level programming languages before compiling them into machine code.[4] Although for Chomsky there is no difference between the grammar required to represent the speaker's language activity and that required to represent the hearer's language activity (that is to say that the encoding and decoding processes are based on the same methodology), there is a great deal of difference for the computer scientist. The decoding process has a separate significance for him from

---

[4] A useful general survey of methodology can be found in Foster [24].

that of encoding, which is seen as the actual writing of programs by people.

2.1 Many of the grammars used in the initial stages of compiling are context-free PS grammars.[5] The process of compiling a programming language into machine code is similar to the process of analysing a natural language to determine its syntactic structure. An early example of one method of computing a PS grammar is given by Kuno and Oettinger [44]. This is also of interest since they do claim that analyzers of the type they favour "suggest themselves as potential mechanisms for speakers and hearers".[6] Since they are interested in analyzing actual sentences, their program takes the form of a <u>directed production analyzer</u> (dpa). Each production of the dpa is of the form

(23) $(P,c) \rightarrow c\ P_1 \ldots P_n$

where c is a terminal symbol (the syntactic word class of the word being scanned) and each P stands for some intermediate syntactic structure. Each of P_1 to P_n is used as a prediction of what the syntactic structure might be one level lower in the parse tree. The whole operation proceeds as follows. The topmost prediction P in a production pool (which is a pushdown storage area) is used to form a couple (P,c) with the word class c of the word being scanned. If there is no such couple in the grammar, then the whole pool is abandoned; otherwise

---

[5] See chapter I, section 1.1.

[6] Kuno and Oettinger [44], p. 416.

that P is deleted from the pool. If there is more than one rule in the grammar with the couple (P,c) on the left-hand side, then as many copies of the pool are made as there are rules, and into each of the pools the elements P_1 to P_n are loaded. The process moves on to look at the next word with each of the new pools in turn. The whole operation is started with one pool containing the distinguished symbol S and finishes when either no pools are left (the sentence is ungrammatical) or a full-stop is reached. If there is more than one pool left when the process terminates, then the sentence is syntactically ambiguous. However, in terms of AI, this is not a good model, since, despite the 2100 rules of Kuno and Oettinger's system, so many sentences of natural languages remain ambiguous when only context-free grammars are considered, although they are not ambiguous in fact.

2.2 It will be remembered that one of Chomsky's claims for TG Theory was that it was highly systematic, this being, indeed, one of its advantages. This led, not unnaturally, to the belief that TG Theory could easily be implemented on computers. IBM did some research and in 1966 published their first report.[7] The first section, written by Rosenbaum and Lochak, contains what was called the IBM Core Grammar of English (CG). This grammar, as its name suggests, does not set out to derive all the sentences of English, but merely the majority. It consists of ten PS rules, many of which have

---

[7] Lieberman [49].

either choices or options in the manner of their expansion, or indeed both; 41 cyclic transformational rules, of which 12 are optional; and 32 post-cyclic transformational rules, of which only two are optional. The lexicon contains seven thousand of the most commonly used words of English (drawn from the Thorndike-Lorge list)[8] and all the types of sub-categorization rules. Later workers with computerized TG grammars have more often than not taken this grammar as their working model, but it does in fact have one or two drawbacks.

For instance, although it is readily admitted that the CG cannot derive all the sentences of English, it does not, however, derive only meaningful sentences of English. It cannot, for example, block the derivation of such a non-sentence as

(24) *The teapot elapses.

Moreover, the present trend of TG grammar writing[9] is towards having as few PS rules as possible, in fact only those necessary to generate recursion of the distinguished symbol.

2.2.1 Perhaps the most substantial work on utilizing the computer as an aid to the TG linguist has been carried out by Joyce Friedman [25] and her colleagues at Stanford University. She has not actually written any given TG grammar, but has rather provided the TG linguist with a computerized system which, she claims, will enable any TG grammar to be tested.

---

8 Thorndike and Lorge [82].

9 See for example Friedman et al. [25].

Unfortunately, and perhaps understandably, this system only deals with the syntactic component of a TG grammar, semantics hardly being mentioned.

In the course of her research, Friedman did find it necessary to add to the standard TG position and has made decisions on one or two points where Chomsky only made suggestions. Thus, in the Base, she allows recursion on symbols other than the distinguished symbol in the PS rules. In the lexicon, she introduces two extra notational characters in order to represent all the possible complex symbols in an adequate way. She does not choose the method of lexical insertion that is associated with the standard TG Theory,[10] but rather an alternative method hinted at by Chomsky[11] and simply attaches complex symbols to lexical category nodes in the parse tree.

In dealing with the Transformational Sub-component of a TG grammar, Friedman has developed a control language[12] which allows the TG linguist to:-

(a) group transformations into ordered sets and apply transformations either individually or by transformation set

(b) specify the order in which transformation sets are to be considered

(c) specify the subtrees in which a transformation set is to be applied

---

[10] See chapter I, section 1.1.1.

[11] Chomsky [12], pp. 120 - 123.

[12] Friedman [25], p. 100 and pp. 106 - 125.

(d) allow the order of application to depend on which transformations have previously modified the tree

(e) apply a transformation set either once or repeatedly.

While these improvements and alterations to the standard TG Theory make the syntactic component even more useful as an analytical tool in the hands of linguists, they do not make TG Theory itself any more viable as a Theory of Language (thus not being useful in the field of AI), and, in fact, they make the syntactic component even larger and more unwieldy, thus making it unsuitable for MT. Nevertheless, it is important to notice that Friedman's work is a good example of how the process of translating a linguistic theory into a computer program enforces rigour and exactitude from it.

2.2.2 There is one author, Petrick [66], who has tried to show how the decoding process might be achieved for a TG grammar. For each of the transformational rules in the original grammar Petrick adduces an Inverse Transformational Rule in the recognition grammar. This inverse rule will be the same as the original except that the S.C. of the original is now the Inverse Structural Description and the S.D. (with appropriate alterations) becomes the Inverse Structural Change. However, there are space and time problems in putting this grammar on the computer, so that Petrick has to suggest a linguistically unjustified set of auxiliary context-free PS rules in the Base, which enable the fruitfulness of proceeding from any I.S.C. to be checked by some method such as Kuno and Oettinger's. However, this would seem to be a departure from the notion that encoding and decoding (production and recognition) are really accomplished by the same grammar.

2.2.3 Three related parsing systems based on TG Theory have recently been developed.[13] The basic idea of these systems is the 'augmented transition network', that is to say that the parser is seen as a transition network similar to a finite-state recognizer used in automata theory, but extended twice. First, the networks may recursively call either themselves or other networks. The second addition is to allow the network to "make changes in the contents of a set of registers associated with the network, and whose transitions can be conditional on the contents of those registers".[14] This is similar to the action of calling a subroutine (in a low-level programming language), remembering the address of the first location after the execution of that subroutine, only to alter that address during the execution of the subroutine, thus exiting to some other part of the main program. It is interesting to look at the comments of Winograd [89] on the differences between his parser and the augmented transition network parsers, which will be done in section 2.3.1.

2.3 The most impressive recent work in the field of AI has undoubtedly been that of Winograd [89]. He has written a

---

[13] Thorne, Bratley and Dewar [83], Bobrow and Fraser [5], and Woods [91].

[14] Woods [91], p. iv.

computer system that understands English in a limited context. His system "answers questions, executes commands, and accepts information in an interactive English dialog".[15] His parser is based on the systemic grammar of Halliday.[16] Winograd believes, as does the author, that "what is needed is an approach which can deal meaningfully with the question 'How is language organized to convey meaning?' rather than 'How are syntactic structures organized when viewed in isolation?'".[17] Systemic grammars deal with "'system networks' describing the way different features interact and depend on each other" rather than "placing emphasis on a 'deep structure' tree".[18] Syntactic structure is organized into 'groupings' of phrases which are meaningfully connected, rather than into the more common form of parse tree. Thus, the sentence

(25) The three big red dogs ate a raw steak

would be parsed into some tree like Diagram 3,[19] on page 45. This set of features, which Winograd calls 'deep structure', is related to surface structure by 'realization' rules. These are very similar to transformation rules, except that in systemic grammars the notion of, for example, "PASSIVE" or "QUESTION" is already present in what is the equivalent of Chomsky's underlying P-marker. An example of Winograd's network for

---

[15] Winograd [89], p. 1.

[16] See Halliday [29] and Halliday [30].

[17] Winograd [89], p. 16.

[18] Loc. cit.

[19] Taken from Winograd [89], p. 17.

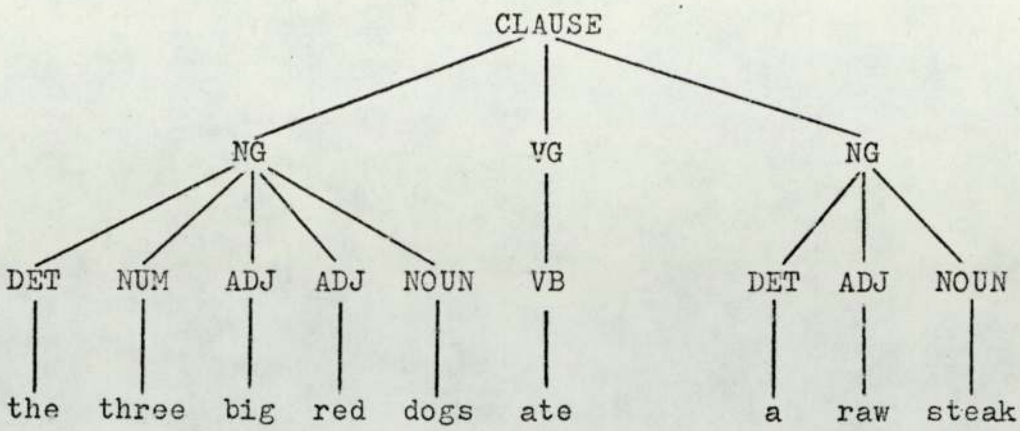


DIAGRAM 3

"CLAUSE" is given in Diagram 4,[20] on page 46. In that Diagram, a disjunct is represented by a vertical bar, a conjunct by a bracket and "null" or "nothing" by a dotted line; an asterisk may be replaced by either 'Q' or 'REL' depending on the particular expansion of the network under consideration.

Winograd's parser is basically a top-down, left-to-right parser, but syntax and semantics are integrated, thus, for instance, in the sentence

(26) He gave the boy plants to water

the parser would never generate an interpretation akin to

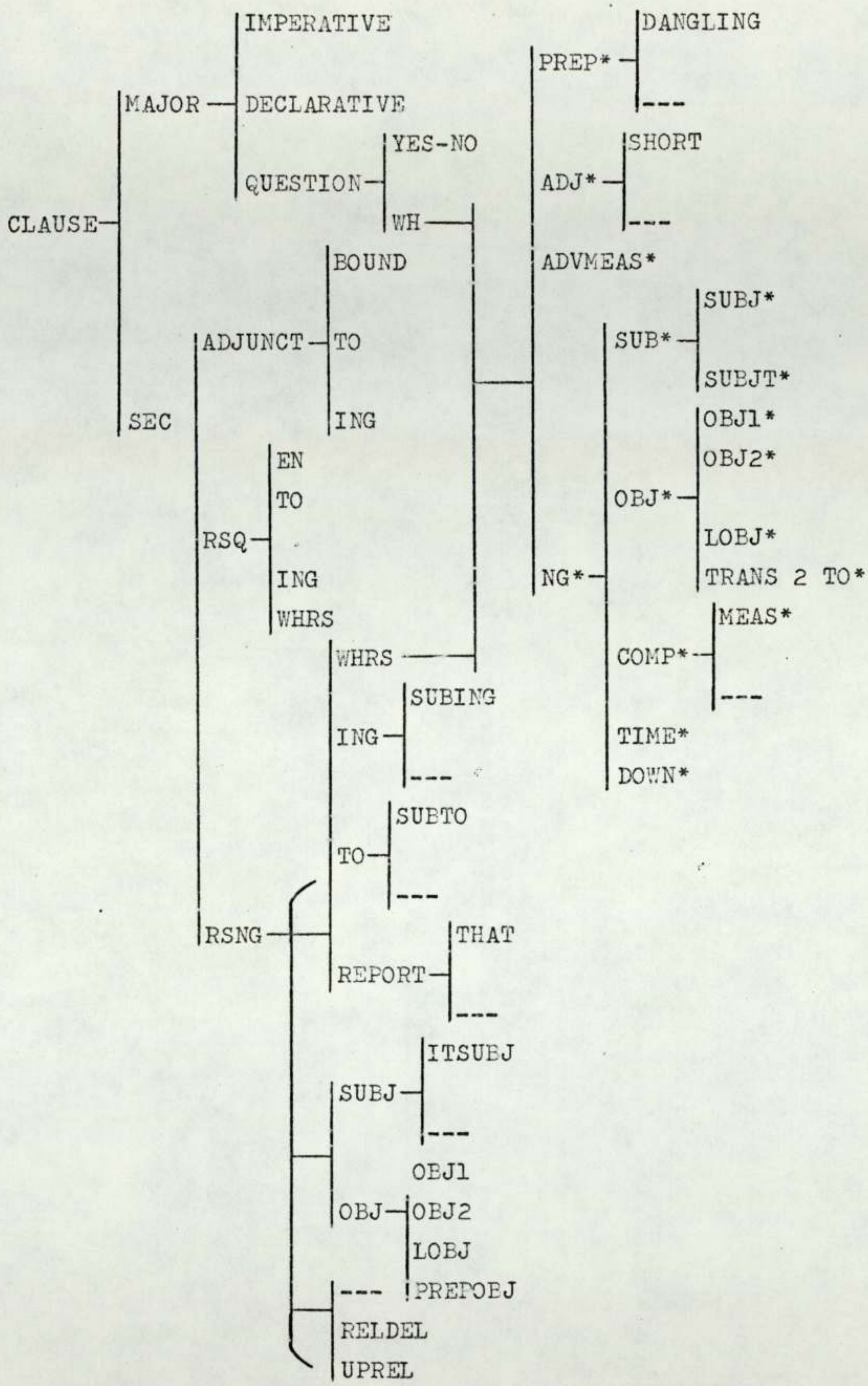
(27) He gave the house plants to charity

since the phrase "boy plants" makes no sense (as against "house plants"). Although Winograd's system does not carry all possible parsings in parallel,[21] it does not follow a blind

---

[20] Taken from Winograd [89], p. 48.

[21] Cf. Kuno and Oettinger [44], in section 2.1.


CLAUSE
MAJOR
IMPERATIVE
DECLARATIVE
QUESTION
YES-NO
WH
SEC
ADJUNCT
BOUND
TO
ING
RSQ
EN
TO
ING
WHRS
RSNG
WHRS
ING
SUBING

TO
SUBTO

REPORT
THAT

SUBJ
ITSUBJ

OBJ
OBJ1
OBJ2
LOBJ
PREPOBJ

RELDEL
UPREL
PREP*
DANGLING

ADJ*
SHORT

ADVMEAS*
NG*
SUB*
SUBJ*
SUBJT*
OBJ*
OBJ1*
OBJ2*
LOBJ*
TRANS 2 TO*
COMP*
MEAS*

TIME*
DOWN*

DIAGRAM 4

automatic backup procedure if one parsing fails, but will rather report what kind of failure has occurred, thus enabling the program to 'decide' for itself what specific backup measures are necessary.

2.3.1 Winograd compares his system with, specifically, that of Woods [91],[22] but his remarks apply equally to the other systems outlined in section 2.2.3. He first of all shows that Woods' discussion of the advantages of networks as grammars is the same as Winograd's own discussion of the advantages of programs. (Winograd's parser is composed of various programs (subroutines) for finding the syntactic structure of each sentence presented to it.) He then points out three basic differences:-

(a) all the network type grammars are based on TG Theory, his own on systemic grammar

(b) Winograd has implemented special additions to the basic parser, such as his method of dealing with the word "and". Whenever "and" is encountered, the parsing is interrupted (cf. programmed interrupts in operating systems) and a special program for interpreting conjoined structures is started. This feature is not in the network model

(c) network parsers use automatic backup procedures rather than the 'intelligent' backup procedures used by Winograd.

---

[22] Winograd [89], pp. 44 - 46.

3. Semantics

Whereas most of the texts reviewed in section 2 were written with CL in mind, it is from workers in the field of AI that the semantics of Natural Language has received attention. This is not surprising, since the best method of determining whether the computer has 'understood' what has been input to it is for it to respond 'intelligently'.

3.1 Winograd [89] claims that "a semantic theory must describe relationships at three different levels. First, it must define the meanings of words .... Secondly, we need an analysis of the ways in which English structures convey meaning, and the roles the words and syntactic structures play .... Finally, ... a semantic theory must describe how the meaning of a sentence depends on its context".[23] Just as Winograd's syntactic analysis is accomplished by means of subroutines,[24] so his semantic analysis does the same. Winograd calls these subroutines '<u>semantic experts</u>' or '<u>semantic specialists</u>'. These specialists build up either '<u>object structures</u>' (basically for noun groups) or '<u>relationship structures</u>' (for the other groups). However, these structures are based very much on Katz type semantic markers. Winograd claims that this is one of the methods for filtering out useless semantic interpretations, the second

---

[23] Winograd [89], p. 28.

[24] See section 2.3.1 of this chapter.

method being to do the interpretation continuously (constantly interacting with the parser). The third method of filtering occurs only when the sentence is truly ambiguous, in which case the semantic system looks at the context of that sentence to settle the ambiguity. Winograd distinguishes three types of context:-

(28)a. local discourse context: the previous sentence (possibly two)

b. overall discourse context: the general subject matter of the discourse

c. context of knowledge about the world.

(28)c. will differentiate between (29)a. and (29)b.:-

(29)a. He hit the car with the rock

b. He hit the car with the dented bumper

since it is known that cars have bumpers but not rocks.

3.2 One of the aims of Yorick Wilks [88] is to "construct a theory that enables us to detect semantic forms directly, and not via a strong and conventional syntax analysis".[25] He distinguishes four meta-linguistic notions: word-sense, message-form, text-fragment and semantic compatibility. Word-senses are represented by semantic formulae which have elements, such as BE, COUNT, GRAIN, THINK. Fifty-three "primitive semantic classifiers"[26] of this kind are given and have been selected not on the basis that they are the 'right'

---

[25] Wilks [88], p. 94.

[26] Ibid., p. 104.

set, but that they produce a reasonable result. Message-forms are represented by templates ("used ... only as experimental devices in their own right")[27] such as MAN+BE+KIND (where '+' means 'to-the-left-of') which can be interpreted as 'a man is a man of a certain sort'. This is, in particular, a bare template, which consists of the heads of three semantic formulae. It is these bare templates which are used by the sequence rules, which achieve compatibility within a text-fragment. Sequence rules use the notion of 'semantic closeness' in deciding between interpretations of an ambiguous sentence.

An example[28] may serve to make the foregoing a little clearer. In the phrase

(30) the old salt is damp

there is an ambiguity between "salt" as 'sailor' and "salt" as 'chemical'. Thus, in the sentence

(31) The old salt is damp but the biscuits are still dry

one would expect "salt" to have the sense of 'chemical'. Wilks' program would generate two sets of bare templates:-

(32) STUFF+BE+KIND THING+BE+KIND

(33) MAN+BE+KIND THING+BE+KIND

The left-hand template in (32) and (33) represents "the old salt is damp" and the right-hand template represents "the biscuits are dry". An attempt is made to compare the heads of each bare template (STUFF with THING, and MAN with THING). Since they are not the same in either case, the sequence rules

---

[27] Wilks [88], p. 14.

[28] Taken from Wilks [88], pp. 113 - 114.

will choose (32) as the better sense for the text-fragment as a whole, since STUFF is 'semantically closer' to THING than MAN is.

3.2.1 Wilks' system is in fact more complicated than the above example makes explicit, since the sequence rules work in general with full templates after an attempt has been made to match bare templates. The ideas are, however, still similar to those shown above. Despite the fact that this system is undoubtedly capable of disambiguating text-fragments and generating 'meanings' for them, it nevertheless relies heavily on a pre-set dictionary,[29] consisting of <u>sense-pairs</u>, where the first member is a (well-formed) semantic formula, and the second member is a sense description. Thus, the sense-pair for the word "compass" is given as:-[30]

```
(34) (COMPASS ((((((THIS POINT) TO) SIGN) THING)
              (COMPASS AS INSTRUMENT POINTING NORTH))
              ((COUNT DO)
              (COMPASS AS TO MEASURE NUMERICALLY))))
```

Wilks admits that this is very similar to the Katz position on semantics, without his belief in a prior syntactic analysis. However, Wilks disclaims the 'correctness' of his system by saying, "There is no intended suggestion that the CSD [Computable Semantic Derivations] system I have described is

---

[29] The system does allow for the automatic construction of concepts, but that is not an important part of the system.

[30] Wilks [88], p. 106.

the linguistic system for analysing natural language. Contemporary philosophy, and the older linguistics, should have by now inhibited the search for any such Grail".[31] Against that, it can and should be argued that there must be (scientifically) at least one method by which at least some people do understand Natural Language and do formulate utterances of their own. It is not a foregone conclusion that a search for such a method is doomed to failure.

3.3 Roger Schank and Charles Rieger [74] have looked at a different aspect of AI, namely the notion of 'inference'. They "consider an inference to be a new piece of information which is generated from other pieces of information, and which may or may not be true."[32] In order for these inferences to be made, a conceptual dependency theory is postulated. This theory is built up from six categories and a number of rules which define the valid relationships between them. Schank and Rieger claim that fourteen primitive actions are all that are necessary "to account for the action part of a large class of natural language sentences".[33] Again, these are chosen because of their effectiveness, not for any a priori reasons. An example of the kind of inference made by their program[34] is given if one considers the sentence (35):-

---

[31] Wilks [88], p. 133, italics in the original.

[32] Schank and Rieger [74], p. 2.

[33] Ibid., p. 13.

[34] Ibid., pp. 15 - 16.

(35) John likes chocolate.

Schank and Rieger claim that there is a missing 'action' in (35), that is to say that people "like chocolate" because they like to INGEST it (where INGEST is one of their primitives). Even if this is not actually the case for John's liking chocolate, a missing 'action' will still be predicted, since the analysis of "like" (which, on their interpretation, is represented as a state) gives an 'actor' and an 'act' and no 'action' to link them. Schank and Rieger believe (without actually quoting any sources) that psychological evidence backs them up in their belief that human beings do understand this kind of inference.

While this may very well turn out to be an accurate description of how human beings extract more information from an utterance than is there in the bald statement of that utterance, this kind of analysis does not help in determining what caused the speaker to utter the given sentence. Indeed, Schank and Rieger's examples tend to read like sentences in a narrative novel, rather than sentences uttered as part of a discourse.

3.4 Weizenbaum's ELIZA[35] is a program in which the semantic analysis is dependent upon the recognition of keywords in the input sentence. Thus, if

(36) He says I'm depressed much of the time

is the input sentence, then the keyword found by ELIZA is

---

[35] Weizenbaum [86] and Weizenbaum [87].

"depressed", thus generating the response

(37) I am sorry to hear that you are depressed.[36]

The program is supposed to model the replies of a particular school of psychologists: the Rogerians. The problem with it is that there is no theoretical justification for the particular words that Weizenbaum has chosen to be keywords, apart from the fact that they are psychologically significant (e.g. "mother", "depressed").

Replies from the system are generated by responding in pre-determined (even if randomly chosen) sentential 'frames'. However, Weizenbaum explores a line of argument that will be returned to in chapter III, section 5. He considers the sentence

(38) I am very unhappy these days

as heard by a foreigner, whose English is only good enough to allow him to understand the words "I am". This foreigner, wishing to appear interested, might reply:

(39) How long have you been very unhappy these days?

where "how long have you been ...?" is obviously a stock sentential 'frame' for replying to "I am ...", irrespective of what follows. It will be argued later that it is this ability to 'split' a sentence into parts, semantically as well as syntactically, that will enable the computer to 'understand' Natural Language in <u>any</u> context.

3.5 The setting up of a network of semantic material has

---

[36] Weizenbaum [86], p. 36.

been the aim of M. Ross Quillian.[37] His model consists basically of a mass of nodes interconnected by various kinds (six in all) of associative links. Each separate concept (not each separate word) is depicted as being on one plane. Each concept is modified (or 'defined') by the logical-type links in its own plane, but each of the nodes of the links will have its own 'meaning' and is itself defined in some other plane. There have therefore to be links from one plane to other planes. From a theoretical point of view, this method of 'defining' concepts has one drawback. There is, in theory, no end to the definition of any concept. The links from each node all lead to other nodes, most of which are on other planes; those nodes have their own links, and the process of following any given path never terminates. However, the (psycho-)linguist (or indeed Quillian) might object that it is precisely the same for human beings. They can only 'define' concepts in terms of other concepts, each of which also needs to be 'defined'.

Quillian's program, TLC, sets out "to extract and somehow retain meaning from natural language text it has not seen before, at a level of skill comparable with that of human readers".[38] It is interesting, therefore, to learn how Quillian believes that human beings understand what they are reading. "Natural language text communicates by causing a reader to recall mental concepts that he already has. It refers him to such already known concepts either with isolated words,

---

37 Quillian [70] and Quillian [71].

38 Quillian [71], p. 459.

or with short phrases, and then specifies or implies particular relations between these. In this way the text may be said to direct the reader to form new concepts."[39] Curiously enough, this is open to the same objection as earlier, only in reverse. Previously, the objection was raised that Quillian's definitions never end; now one can object that there is nowhere for man's concept formation to begin! What is the first concept without which man could learn nothing else? How are the earliest 'new concepts' formed?

However, in the same article, Quillian says, "in TLC, the function of text is to be viewed not as explicitly stating information for a reader but rather as directing the reader to construct for himself <u>various cognitive structures</u>".[40] This is an extremely important point, which will be further elucidated in chapter III. Suffice it to say here that it points out the error of insisting on an attempt to discover what a 'meaning' is. The 'meanings' of sentences or words no more exist in this world than do unicorns or dragons. Meaning is conveyed by the speaker to the hearer when the hearer has either built up in his own mind the relevant cognitive structures that caused the utterance, or understood that the speaker has those cognitive structures.

---

[39] Quillian [71], p. 464.

[40] Ibid., p. 460.

4. Conclusions

For the purposes of AI as defined in section 1.1, the choice of a suitable syntax would appear to lie between TG grammars and the systemic grammars of Halliday and Winograd. In terms of programming techniques, systemic grammars would appear to be superior, since the networks are 'easily' programmable and are interchangeable between understanding and generating utterances. TG grammars, with the exception of the transition networks, are more cumbersome and are, apparently, difficult to handle when understanding is the goal. As far as semantics is concerned, nobody has really shown how computers can understand Natural Language in any and all contexts. Winograd admits that a proper semantic theory must take account of knowledge about the world, but his system only knows the highly restricted context of one table, one box and some blocks of wood. Apart from Schank and Rieger, everybody has concentrated more on nouns than verbs, although it will be shown later that general knowledge depends more on understanding the verbs.[41]

There seem to be no published works where people in the field of AI have taken the Theory of Intentions (or the notion of presupposition) and used it to enable computers to model and understand Natural Language. This is an important omission, both in the sense that, as has been seen, there is good reason

---

[41] It has been assumed throughout this paragraph that, contrary to Wilks, syntax and semantics are 'separate' in some definable way.

to believe that the speaker's intentions when uttering a sentence matter for that sentence's comprehension, and in the sense that it is through this kind of theory that knowledge of the world can be 'learnt'.[42] The concept of learning is also important, since AI programs should be able to learn from past experience, this being what human beings are supposed to do. Thus, the program to be developed in the next chapters will be able to learn and remember, and should be able to deal with any context.

---

[42] That is not to say that other programs have omitted the 'learning' operation.

III. THE FUNCTIONAL THEORY OF LANGUAGE

It was argued at the end of chapter I and the end of chapter II that a theory based upon Austin's notion of 'intention' and Fillmore's notion of 'presupposition' would be of benefit to both AI and the Philosophy of Language. This chapter will show the possible structure of such a theory.

1. Language

Language is the means by which human beings communicate with each other. The basic method of communication is speech: one person speaking either to another person or to a group of people (the audience). The purpose of the speaker in speaking is to make his audience understand his thoughts, emotions, arguments or beliefs. The phonetic arrangements which actually form the sounds of each individual word are purely arbitrary. The same can be said of the so-called syntactic 'rules' that are supposed to govern Language. If these really were rules (in the sense of rules of a programming language), then to break them should (a) have the most undesirable results (the audience would not understand the speaker), or (b) be impossible. Neither of these alternatives is actually the case.

In actual speech situations, people are constantly interrupted, thus being unable to finish their sentence, or they produce such a long sentence that syntactic relationships are 'broken', or they simply speak 'ungrammatically'. Indeed, so often do people speak ungrammatically that they can in fact change the 'grammar' of the language. Thus

(40) I wish he were here

is still supposed to be 'correct' English, but so many people (including television announcers) are inclined to say

(41) I wish he was here

that the subjunctive may soon pass out of English 'grammar' entirely, going the way of Case before it. The fact that sentences such as (41) can be uttered meaningfully shows that whatever kind of syntax human beings actually 'possess', it is not constructed from hard and fast rules.

1.1 Although Language is the means by which human beings communicate with each other, this does not mean that a 'thought' is necessarily prior to its linguistic representation. One may distinguish three kinds of thinking:-

(a) thinking that requires a search of the long-term memory. Searching for the linguistic representation (word) of a concept that one 'knows' is an example of this kind of thought process. This process is obviously <u>not</u> linguistic, although the result of such thinking often will be linguistic, for example, "Ah, 'pragmatism' was the word I was looking for".

(b) the thinking involved when one is arguing, often over some academic matter. This kind of thought process usually <u>is</u>

linguistic. Thus, one prepares arguments and counter-arguments in one's mind, both those that one is intending to produce and those that one hopes will not be produced on the other side.

(c) what might be called 'situational' thinking; the kind of thought processes that lead up to most speech situations. Although this process is not obviously linguistic, it can be argued that it is linguistic in form, but not necessarily at the level of Natural Language. If one considers a computer's 'brain', it can be seen that a computer with a core memory can only recognize, in the last analysis, the presence or absence of a magnetic field at a particular location in its memory; this recognition is accomplished by passing an electric current through the relevant 'cell'. The presence or absence of magnetic field is usually represented by the binary digits (bits) 1 and 0; the 'language' built round these bits is called Machine Code. This is the true Language of the Machine. On the other hand, compilers written (usually) in a mnemonic form of Machine Code enable the computer to 'translate' high-level languages (which may be similar to restricted subsets of Natural Language) into Machine Code. The human brain can be said to be similar in many ways. In human brain cells, electricity causes changes of their chemical contents; Legéndy [48] claims that "the effect of information on the brain is to cause the neurons to form groups",[1] which is similar to bits being grouped into computer words[2] and 'data

---

[1] Legéndy [48], p. 556.

[2] Cf. also the "cell assemblies" of Hebb [32].

structures'. Thus, the concept of, say, a Brain Code is similar to the concept of Machine Code. One might say that Brain Code is the proper Language of the Mind. High-level languages such as English can be translated into Brain Code, and *vice versa*. Suppose Jones says

(42) I accuse Smith of robbing the bank

then Jones might or might not be conscious of the thought

(43) Smith is responsible for robbing the bank

but he will have 'had' the thought nevertheless.

Despite the fact that high-level programming languages must be translated into Machine Code before a computer can 'understand' them, English does not have to be translated into Brain Code to be comprehended by the brain. Brain Code is more properly described as the Language of the Subconscious Mind. Natural Language is the Language of the Conscious Mind. The same 'thoughts' can be made in either language form without being translated from the other, in exactly the same way that a bi-lingual person is quite happy to talk in either of his two languages without going through a translation process. (Indeed, one is not supposed to have achieved mastery of a foreign language until one can stop translating out of and into the mother tongue.) This distinction between Brain Code and Natural Language would also account for the apparently contradictory results obtained by asking people whether they "think in Language or not". Some people believe that they do think (type (c) above) in Language; others are equally convinced that they do not. (The author has made this discovery over six or seven years of casual conversation with many people in many walks of life.) The truth might well be that the latter think in Brain Code (i.e. subconsciously), while the former do think in Natural Language (i.e. consciously).

2. Cognitive Structures and Understanding

Consider (43) again. This could be reformulated as "RESPONSIBLE [Smith, robbing the bank]". Looking at example (18) in chapter I, it will be seen that this is exactly what Fillmore believes is said by the 'judge' when uttering a sentence, such as (42), using the verb "accuse". Thus, it is a condition for the use of "accuse" by Jones that (43) is thought by Jones. According to Fillmore, it is a presupposition of "accuse" that the 'situation' is BAD. It would be logical to assume that some thought such as

(44) Robbing the bank is a bad thing to do

would also be required before Jones could utter (42), but this is not so. This thought does not have to be formulated even in Brain Code, for it is not part of the information that Jones wishes to convey to his audience, whereas (43) is precisely that information. What is necessary is that Jones' cognitive structure includes BAD [robbing the bank].

2.1 The term 'cognitive structure' requires explanation. It is first necessary to abrogate any notion of there being an actual 'mind' which is something distinct from a brain. Such metaphysical 'entities' have no place in a scientific thesis. Talk about minds is really only a convenient shorthand for talk about brains. It is the brain that is at the centre of Language processing activities, and it is in the brain, therefore, that all the requisite aids to Language comprehension and generation are situated. Memory, then, being a function of the brain, is realized by a large number of brain cells. A human being's knowledge is stored in cells somewhat as a computer stores its data. It is not the case, however, that each cell is a discrete entity, having life of its own and its own bit of knowledge. Cells are linked together, and it is the linked pattern of cells that forms each piece of knowledge. However, knowledge is not merely restricted to 'facts' about the world; it also embraces moral beliefs and attitudes in general. It is perfectly acceptable to say "I know that robbing banks is immoral" or "I know that God is benevolent", where the argument of neither sentence represents a 'fact' in the world. Each set of linked brain cells forming a piece of knowledge or an attitude is to be called 'a cognitive structure'. The set of all cognitive structures of a given person is to be called 'the cognitive structure' of that person.[3]

2.2 Presuppositions inherent in the use of a given verb (that is, the implicit level of communication)[4] play a different role for Fillmore from what he calls the 'meaning' of the verb, that is conveying the information intended by the speaker to his audience. It was shown in section 4.2 of

---

[3] Cf. Quillian in chapter II, section 3.5.

[4] Cf. chapter I, section 4.1.

chapter I that presuppositions can be considered as properties of the verb with which they are associated. It is equally possible to think of them as functions of the argument, x, of a given sentence. Indeed, Fillmore's notation implies this attitude towards them. As far as the speaker is concerned, these functions act as checking mechanisms on the speaker's cognitive structure. For example, suppose Jones wishes (intends) to convey the information contained in (43). Jones may know a number of verbs which would convey this information, including, say, "accuse" and "congratulate". However, a presupposition of "congratulate" is GOOD [x]. If this is applied to "robbing the bank", then there will be a clash between the structure propcsed by the function GOOD and the already existing cognitive structure formed by BAD. Since GOOD and BAD are mutually exclusive, the check fails; "congratulate" will not be chosen by Jones whereas "accuse" (or "reprove", "indict", "impeach", "reproach" etc.) will be chosen. It is important to note that Jones might have had no views whatsoever on the subject of robbing the bank. In that case, not only could he not have used the verb "congratulate", but nor could he have used "accuse".

Clearly, the explicit level of communication may also be analyzed in terms of functions in the same way as the presuppositions. Indeed, the same set of functions is used for both the presuppositions and the 'information content' associated with verbs. Consider (43) again. This is the prompt for Jones to utter sentence (42). It is also the information content associated with the verb "accuse". It might be the case that Jones already has the cognitive structure

(45) RESPONSIBLE [Smith, robbing the bank].

In that case, use of the verb "accuse" is 'generated' from the existing cognitive structure and there is no change to Jones' total knowledge. However, (43) might be a completely 'new' thought to Jones, either in his high-level language, or in Brain Code, prompted, say, by having seen Smith emerge from the bank last Thursday carrying a large bag, and by hearing later that the bank had been robbed. In that case (45) will become a new cognitive structure and will thereby add to Jones' knowledge. This reasoning process is one way whereby one's stock of knowledge may be increased. The second way will be shown in section 2.3 below.

2.3 A sentence is understood by the hearer when he has recognized (a) the presuppositions[5] inherent in the verb in that sentence, and therefore (b) the cognitive structures of the speaker that led to the uttering of the sentence. The hearer models the cognitive structures of the speaker, after which the presuppositions act as checking mechanisms on the hearer's cognitive structure. Any clash is likely to produce a reply such as "Wait a minute; that means that you think that such and such, but that's not right". If there is no clash, then the hearer is likely to reply to the information content of the sentence, as shown by the explicit level function

---

[5] Henceforward, the term 'presupposition' will be used to include both implicit <u>and</u> explicit levels of communication.

associated with the verb. (Needless to say, the sentence can have been understood without any such reply whatsoever.) If the hearer's cognitive structure includes nothing at all to do with the argument of the sentence uttered, and if the hearer believes the speaker to be reliable, then <u>all</u> of the speaker's cognitive structures shown by the utterance may be adopted by the hearer: thus being the second way in which one's stock of knowledge can be increased.

It might, of course, be possible that one adds to one's stock of knowledge through reading. In this case, one could say that when reading (a serious work) uncritically, one simply transfers the cognitive structures of the writer of the piece one is reading, as shown by what he says, into one's own cognitive structure in the long-term memory. Critical reading, however, would produce the same check on those cognitive structures, while they are still in the short-term memory, as has been outlined above. This check is performed both on the reader's cognitive structure and on the reader's model of that part of the writer's cognitive structure that has been revealed in the work. It follows from this kind of analysis of the acquisition of knowledge, that some kind of verbalization is necessary before any 'new' thoughts are retained by the long-term memory. This ensures that the right presuppositions form the individual's new cognitive structures. The verbalization may well only be at the Brain Code level, but it must nevertheless have taken place.

Since understanding is dependent upon a recognition of the presuppositions inherent in the verb of an utterance, it can be said that at least part of the illocutionary force[6] of the utterance is contained in those presuppositions. It is also the case that all the illocutionary force of an explicitly performative utterance is contained in the presuppositions inherent in the explicitly performative verb of the utterance.

Let this theory of how Language is understood be called the '<u>Functional Theory of Language</u>' (FTL).

3. Presuppositions

This section will outline the kinds of presupposition that can be found to be inherent in performative verbs. There is no theoretical reasoning behind the choice of presuppositions; they have all been 'discovered' by actually analyzing performative verbs (and occasionally by the necessity to alter an analysis from experience of using the computer program to be explained in chapter IV). The names of the presuppositions have been chosen arbitrarily and do not necessarily have any cognitive significance. It will be seen that the presuppositions go in pairs, one positive and one negative in intent. This is essential, since verbs can be either positive or negative in intent (compare "assert" with "deny").

3.1 Consider verbs such as "assert", "state" and "believe". As well as being performative verbs, they are verbs which report on the state of the individual's cognitive structure.

---

[6] See chapter I, section 3.2.4.

They are verbs which say (at the very least) that the speaker, S, believes that x, the argument of the verb, is true. This presuppositional function will be called VALIDBLF, with a corresponding negative (for verbs like "deny") called NVALIDBF. However, this is thorny philosophical ground. There is a raging debate as to what 'belief' is, and, in particular, what the difference between 'belief' and 'knowledge' might be.[7] The only part of that debate that is of relevance here is that part which refers to "I know ... " as uttered by S. To say "I know ... " is to claim to know something, not actually to know it. This distinction is brought out by Woozley [92] who says "the question whether I know that something is the case is the same as the question whether I can *truly* claim to know that it is the case; but it is not the same as the question whether I can *justifiably* claim to know that it is the case. The question whether I can truly claim to know that it is the case is the question whether I know that it is the case; the question whether I can justifiably claim to know that it is the case is the question whether my reasons for saying that I know that it is the case are good reasons".[8] FTL's concern is with the justification of the claim. Thus two further presuppositions are necessary: CONFDENT, meaning that S has good evidence for believing x, and NCONF, meaning that S does not have good evidence. This evidence, moreover, is not what

---

[7] See Griffiths [28] for a presentation of many sides of the argument.

[8] Woozley [92], pp. 83 - 84, italics in the original.

might be called 'direct' evidence, e.g. when S has seen something happen; it is rather either inferential evidence or, for instance, evidence from books.

Consider, however, the verbs "confirm" and "affirm". Here, to say "I confirm that ... " is to make a claim by S that he was present when x occurred, or that he has first-hand knowledge of the existence of a state of affairs. This requires the presuppositions VALID and NVALID, meaning that x occurred and x did not occur, respectively. These are really the only presuppositions that refer to facts in the outside world; all the others (with the possible exceptions of PASTUTT and NPASTUTT,[9] which are in a totally different category) refer to individuals' cognitive structures.

3.1.1 Verbs like "credit" or "accuse" require two different pairs of presuppositions. First, to credit I with x presupposes that x is a good thing, whilst to accuse I of x presupposes that x is a bad thing; hence the functions GOOD and BAD. At the explicit level, both "credit" and "accuse" show that S believes that I was responsible for causing x to occur. This is represented by the function RSPONSBL; its corresponding negative is NRSPNSBL, meaning that S believes that I was not responsible for x.

3.1.2 FUTACT, meaning that S intends to do x in the future, and its opposite NFUTACT (S does not intend to do x in the

---

9 See section 3.1.3, below.

future) are presuppositions of verbs like "promise" and "refuse" respectively. Similarly, WANTS (S wants x) and NWANTS (S does not want x) are presuppositions of "wish" and "reject" respectively.

3.1.3 Some verbs, for example "confirm", presuppose that x, the argument of the verb, has been uttered at some previous time in the conversation, usually, though not always, by some person other than the current speaker. This is represented by the function PASTUTT. Similarly, NPASTUTT means that x has not been uttered previously. This is a presupposition of a verb such as "wonder".

3.1.4 Consider "promise" again. In section 3.3 of chapter I, it was shown that Searle believed part of one of the 'rules' underlying the illocutionary force indicating device Pr for promising to be: "Pr is to be uttered only if the hearer H would prefer S's doing A to his not doing A" where A is the future act that S is promising to do. It is therefore necessary to have presuppositions that refer to the addressee's cognitive structures. These end in the letter H and are WANTH and NWANTH (the addressee wants or does not want x) and VALIDBFH and NVALDBFH (the addressee does or does not believe that x is true).

3.1.5 Consider the sentence

(46) I order you to attack the enemy.

This presupposes that the speaker is in a position of authority over the addressee(s). Hence the presupposition AUTHORTY [S,I], meaning that S is in authority over I. Its corresponding negative is NAUTHRTY. These last two functions are two-place presuppositions; the only other pair of two-place presuppositions is RSPONSBL and NRSPNSBL. Moreover, AUTHORTY and NAUTHRTY are the only two functions that do not refer to x in some way.

3.2 There is one presupposition that is different in kind from the presuppositions outlined in section 3.1 above. This is the presupposition VALUE which is inherent in some of the verbs that Austin called 'verdictives',[10] for example, "value", "deem", "rank" and "rate". These verbs make a comparative judgement (verdict) on some article or argument. For obvious reasons, VALUE has no negative complement.

4. Philosophical Implications of FTL

If the Functional Theory of language comprehension is correct, then a child cannot be born with a 'tabula rasa' for a mind, nor is Language to be learnt by ostensive definition alone. No utterances in a language are considered to be 'basic', either in the Lockean notion of 'concept', or in the early Chomsky sense of 'kernel sentence'.[11] In fact, none of the empiricist views of Language and Language acquisition is

---

[10] Austin [2], pp. 152 - 154.

[11] Chomsky [10], especially pp. 45 - 48.

upheld. This is perhaps just as well, since Harrison [31] has shown very clearly how any Empiricist Theory of Language, based as it must be upon an associative-referential theory of meaning, becomes wholly vacuous.[12]

FTL is a Rationalist Theory of Language. Human beings are born with a propensity, not necessarily for learning the whole of a given language, but at least for associating functions (presuppositions) with particular verbs (assuming that infants have some method of differentiating discrete sequences of sound into particular words). This is not an empiricist type of association, since no reference is made to any object in the outside world. From data published by McNeill [63] (following Roger Brown), it can be seen that a child of 28 months uses a verb in about 60% of two-word utterances and about 76% of three-word utterances.[13] The majority of utterances where the verb is omitted would appear to be cases where the verb would be the copula, for which the presupposition would be the rather trivial one of VALIDBLF; thus the utterance "dress pretty" is short for "[I believe that] the dress is pretty". Other common types of utterance, like "more milk" or "Mummy get ladder" imply the presupposition WANTS. The more abstract, less self-centred presuppositions, such as those that imply moral or ethical judgements, only appear much later in a child's development. The omission of such verbs as "think" or "want" might well be due to the same reason that children omit

---

[12] Harrison [31], especially Part I, pp. 1 - 109.

[13] McNeill [63], Table 3, p. 28.

mentioning themselves as the subject of sentences, namely that infantile egocentricism (a term introduced by Piaget [67]) makes a child think that everyone knows that he is talking about himself. Nevertheless, this must not be seen as evidence for presuppositions; it is merely that FTL enables one to account for the child's linguistic behaviour in a consistent manner. The argument in section 4.3.1, below, will explain why linguistic data from the child cannot count as evidence for FTL.

4.1 Not only is a child born with this propensity for associating presuppositions with verbs, but he must also be born with a fully functioning Brain Code. No computer can work properly if its CPU fails to process accurately the low-level instructions input to it. A child has no engineers to come and put his brain right if part of it is not functioning correctly. A child cannot learn its Brain Code from anybody else, since Brain Code is not public property in the way that natural languages are. Therefore, a child's processor and the low-level language that it processes must be intact at birth. Indeed, there is evidence from brain-damaged patients that the brain has duplicated some of its processes, just in case its original 'channels' should become inoperative for any reason. This being the case, it is quite possible that babies cry for food, not simply because they are hungry, but because they have had a low-level thought corresponding to the high-level "I am hungry", but are unable to communicate this thought to anybody else.

4.2 TG Theorists would say that FTL is a theory of performance and, as such, not directly contrary to TG Theory, which they claim is a theory of competence.[14] However, the distinction between competence and performance,[15] far from being a "major methodological clarification",[16] has had the effect of pushing philosophical discourse into the realms of fantasy. Human beings are said to have a 'knowledge' of their own natural language, amounting to a 'knowledge' of the grammar underlying that language. This 'knowledge' is hardly ever put into practice: indeed, the only apparent justification for stating that this 'knowledge' does exist is the fact that human beings can answer "yes" or "no" to the question whether an utterance 'x' is part of the language, L, or not. Chomsky himself says "... we cannot avoid being struck by the enormous disparity between knowledge and experience — in the case of language, between the generative grammar that expresses the linguistic competence of the native speaker and the degenerate data on the basis of which he has constructed this grammar for himself".[17] Indeed, the data on which Chomsky has constructed his own grammar for a fragment of English is just as meagre and degenerate, because all the data provided for this grammar is data abstracted from actual linguistic performance, which need

---

[14] See chapter I, section 1.

[15] For an account of many of the arguments in favour of the distinction, see Fodor and Garrett [23].

[16] Ibid., p. 137.

[17] Chomsky [13], p. 68.

not have been generated by the competence grammar 'known' by the speaker at all.[18]

Why must there be this disparity between the sentences produced by the grammar that an adult 'knows' and the sentences which he uses? The obvious answer is that language users do not always act in accordance with 'rules' in the way that TG Theorists would like. Indeed, the need to account for so-called "semi-sentences"[19] has led Katz to postulate a further set of 'rules', "transfer rules", which correspond to either kind of 'rule' in a standard TG grammar, but are, so to speak, incorrect rules. It is these "transfer rules" that are followed when sentences are uttered that do not accord with the competence grammar.[20] Is it the case that semi-sentences are not part of the language L? If they are not, then how do speakers of L understand them? On the other hand, if semi-sentences are part of L, then why are they called semi-sentences and treated differently, being designated 'ungrammatical' and therefore not derivable by the competence grammar? To postulate transfer rules alongside the other rules is just as 'inelegant', just as theoretically undesirable, as the areas of duplication in TG Theory pointed out in chapter I, section 2.1.

Since the competence 'rules' can only be formulated by looking at the results of performance, which is said to be

---

[18] See Chomsky [12], p. 9.

[19] See Katz [37].

[20] Cf. section 1 of this chapter.

"degenerate", any 'conclusions' drawn about these 'rules' can only be arbitrary; there can be no evidence for their accuracy or otherwise. The drawing up of a competence grammar is therefore pure fantasy, and is open to just the same line of attack that Katz and Fodor themselves used on the arbitrariness of the artificial languages used by Logical Atomists as idealizations of Natural Language.[21] Moreover, as Campbell and Wales [8] point out, TG Theorists have, in general, not tried to characterize the nature of competence seen as "the nature of those human abilities that are specific to language" but have attempted rather to characterize "a more restricted competence ... from which by far the most important ability has been omitted — the ability to produce or understand utterances which are not so much grammatical but, more important, appropriate to the context in which they are made".[22] Katz and Fodor claim that "a sentence cannot have readings in a setting which it does not have in isolation",[23] whereas it would be more correct to say that a sentence has no meaning out of a context. Although Campbell and Wales still believe that the distinction between competence and performance can be maintained, it should be quite clear from the preceding arguments that the nearer one comes to the context in which a sentence is uttered, the more impossible it becomes to draw the line between 'competence' and other knowledge.

---

[21] See Katz and Fodor [36].

[22] Campbell and Wales [8], p. 247, italics in the original.

[23] Katz and Fodor [38], p. 488.

4.3 In section 2.3 of chapter I, it was stated that the reason why TG Theory was merely a formal system for describing linguistic data and not a properly scientific Theory of Language was the lack of correspondence between empirical content and the (supposedly) interpretive system. It is, in theory, possible to test empirically for the kinds of cognitive structures outlined above. Unfortunately, the technical equipment and knowledge is not available at the present time. In general, neurophysiologists and neuropsychologists work with people who have suffered brain damage in some way, hardly ever with 'normal', healthy human beings; moreover, their results are limited to marking the general locations in the brain of linguistic activities, such as reading, writing and speaking, as well as showing how the two sides of the brain interact with each other.

"[Modern psychology] has undoubtedly made considerable progress in the study of the genesis of psychological processes, and in their changes in the course of development. It has described the structure of human mental activity. It now has clear ideas on the structure of higher psychological actions and complex conscious activities that cannot in any way be compared with the classical schemes of associationism or with the general ideas of Gestalt psychology, with the simplified phenomenology of behaviourism, or with the pretensions of 'depth psychology'. Despite all these advances, our knowledge of the psychophysiological structure of mental processes and of their internal intimate mechanisms is still grossly inadequate. We still know very little about the internal nature and the neurological structure of complex forms of conscious activity although their course is now reasonably well understood. We know almost nothing about the factors comprising the structure of this activity, and how these factors change in the successive stages of mental development and with the acquisition of the complex devices facilitating the course of these processes."[24] Thus says Luria at the end of his book The Working Brain, which is a good survey of forty years' work in the (relatively) new field of neuropsychology. It is clear from that statement that our knowledge of the way that the human brain works is extremely meagre. Those neuropsychologists and neurophysiologists who work with the language processing parts of the brain have either found it unnecessary or impossible (as yet) to work with individual brain cells or groups of cells. Nevertheless, it is intrinsically possible for evidence from such a source to be found.

4.3.1 Furthermore, it is only evidence from the brain that can really count as empirical evidence for or against Theories of Language. Quite clearly, no linguistic evidence is outside the system to be explained; indeed to argue that a particular part of a linguistic theory is 'correct' because there is linguistic evidence for it is a circular argument (cf. McNeill's various transformational grammars for children of different ages[25] and

---

[24] Luria [51] , pp. 342 - 343.

[25] McNeill [63], especially pp. 15 - 36.

Menyuk's description of the child's 'acquisition' of transformational syntax).[26] Similarly, experiments, such as those by Johnson [34] and Fodor and Bever [22], which attempted to show that the immediate constituents of PS rules (or the rules themselves) have 'psychological reality', managed merely to 'prove' what has been known for some time; namely, that phrases are larger syntactic units than words and that, since the short-term memory can process only seven (plus or minus two) items at once,[27] the brain has to interpret the phrase as one 'item'. This does not, despite the conclusions drawn by the experimenters, 'prove' the 'psychological reality' of anything, far less of 'rules'. In the same way, experiments which attempt to determine how semantic memory is organized by doing time trials are doomed to failure. One such published attempt, Collins and Quillian [15], which in any case uses far too small a sample population to draw significant statistical conclusions, manages to 'prove' that it takes longer to process false sentences than it does to process true ones. Yet, despite their attempts, it is quite impossible to say anything, based on such evidence, as to how the semantic memory is organized. First, it is impossible to set the experimental constraints tightly enough to ensure that only that which one wants to test is being tested. Secondly, it is quite probable that <u>any</u> theory of semantic memory organization would predict that false sentences will take longer to process than true ones.

---

[26] Menyuk [55], especially pp. 93 - 161.

[27] See Miller [59].

The only acceptable evidence for what happens when a piece of language activity is heard and understood must come from the brain which is the true receptor of the utterance. With the advance of technology, it may be that experiments on individual brain cells or groups of cells will be possible in the near future.

5. The Functional Theory and AI

In chapter II it was argued that the most urgent requisite for AI was a computer that was able to understand Natural Language in any context. FTL can provide the basis for just such a comprehension of Natural Language by a machine. Consider (42) again. It has been said that Jones' cognitive structure must include BAD [robbing the bank] before he could utter (42). Now consider

(47) I accuse Smith of putting apartheid policies into practice

(48) I accuse Smith of mugging the shopkeeper

(49) I accuse Smith of adultery

all as said by Jones. In these cases Jones' cognitive structure would have to include BAD [putting apartheid policies into practice], BAD [mugging the shopkeeper] or BAD [adultery]. Thus, irrespective of what actually follows the "of" in sentences containing the verb "accuse", it is known that it must be a bad thing to do (in Jones' view). The context could be robbery, politics, murder, ethics, morals, justice, or any other. In terms of Weizenbaum's sentential 'frames' (explained in section 3.4 of chapter II), it could be maintained that

(50) "I accuse 'x' of 'y'"

is a stock sentential 'frame' for stating that BAD ['y'] (implicitly) and RSPONSBL ['x','y'] (explicitly) belong to the utterer's cognitive structure. This is the semantic analysis. It is dependent, at least in part, on syntax, since 'x' and 'y' are separated by the preposition "of" which is associated with the verb "accuse".

5.1 There are two reasons why only the verb in each sentence will be semantically analyzed by the computer. First, there is a sense in which the argument of (47), namely "putting apartheid policies into practice", is one indivisible concept, not an aggregation of the meanings of each separate word in the phrase. If that is the case, then it is obviously folly to semantically analyze a coherent whole into a group of smaller 'entities' which do not exist. The concept itself can be stored just as easily in its original form as in the form of, say, n where n is an integer pointing to the position of such a concept in an 'array' of concepts.

5.1.1 However, if one does not believe the above argument, or fails to see its applicability to sentences such as (49) where "adultery" is the one word argument of "accuse", and if one therefore requires a semantic analysis of at least nouns and adjectives in the sentence, then there is still a good reason for not doing so. As far as the computer is concerned, words can only be defined in terms of other words, whether as a straight dictionary definition, or by some scheme such as Quillian's.[28] It cannot know what a human being is, far less what apartheid might be. Even if the computer has a hand and eye 'attached' like Winograd's SHRDLU, terms like "apartheid" or "adultery" will be meaningless. Nevertheless, it is sufficient for a computer to 'hear' Jones utter (49) to know that Jones believes adultery to be bad.

5.2 The notion of computers being used to model cognitive structures is not new. Abelson and Carroll [1] modelled "individual belief systems", by which they meant "interrelated [sets] of affect-laden cognitions concerning some aspects of the psychological world of a single individual".[29] (Weizenbaum's ELIZA uses their kind of belief systems when trying to build up a 'picture' of the person inputting sentences to it.) Their system deals only with a corpus of knowledge derived (in FTL) from the presupposition VALIDBLF. For their program "the standard linguistic unit in which beliefs are stored is a sentence, consisting of a concept followed by a predicate. A predicate typically consists of a verb followed by a concept".[30] The cognitive processes that are modelled by this computer program are the "credibility test", a check on the plausibility of new input sentences compared with the current data base, and the "rationalization

---

[28] See chapter II, section 3.5.

[29] Abelson and Carroll [1], p. 24.

[30] Ibid., p. 25, italics in the original.

attempt", which is an attempt to explain away sentences which are internally inconsistent. Unfortunately, no indication is given of what semantic analysis, if any, is performed by Abelson and Carroll's computer. For instance, "left-wingers mistreat U.S. friends abroad" is taken to be an instance of "liberals support anti-colonial policies".[31] They do say that "a substantial portion of [the] corpus of sentences is devoted to definitional information",[32] but that still does not explain how the verbs in each sentence are treated.

The artificial belief system (ABS_1) of Colby and Smith [14] was set up to "study certain properties of credibility functions in a synthesized artificial system whose structure and starting conditions were entirely under [their] control".[33] This system is similar to Abelson and Carroll's in that the data input to the artificial belief system would be derived from VALIDELF, and in that there are what Colby and Smith call "expectancy or implication rules"[34] which give definitional information to the system. ABS_1 is different in that it caters for different strengths of belief; from certainty to possibility. ABS_1 also allows the user to put questions to it which it then attempts to answer with some relevant strength of probability. When a session has finished, ABS_1 attempts to give its informant a credibility rating, using

---

31 Abelson and Carroll [1], p. 26.

32 Ibid., p. 25.

33 Colby and Smith [14], p. 319.

34 Ibid., p. 320.

one person's set of beliefs as highly credible[35] and comparing those beliefs with the beliefs of the current informant.

The computer program based on FTL would be able to incorporate the results and procedures of either of the above models of belief systems. What is not taken into account by either of them, however, is the vast variety of attitudes which make up a person's cognitive structure. To limit these to the single attitude of VALIDBLF is to make the same kind of mistake that the Logical Atomists made in thinking that Language is restricted to sentences that can be either true or false. The mind, like the language that expresses the mind's attitudes, is more complex than that. It is necessary to model all those complexities if the machine is to be capable of understanding Natural Language in any context.

---

[35] Cf. chapter V, section 2.

IV. PIP

This chapter will detail the program written to demonstrate that a computer can understand at least one language (in this case, English) in any context. It will also outline the limitations imposed on that program by outside influences that are not under the author's control. The program is called PIP (Presuppositions Inherent in Performatives), for reasons that will become apparent later.

1. Facilities Available

The University of Aston has an ICL 1905E computer with 96K words of computer memory, where each word has a length of 24 bits. The system is designed for file-handling under the control of the GEORGE 3 operating system. The files that are currently in use are kept on three EDS 60 high-speed discs, with a number of magnetic tape units as backing store. Seven on-line terminals are linked to the system, mainly for users to carry out routine file-updating; but, in certain circumstances, users may also run programs from these terminals. There is, however, a restriction on those users who do run on-line programs: the program size must be less than

24K words of (virtual) store.[1]

1.1 The languages available at the time work began on the program were PLAN, BASIC, ALGOL 60, SNOBOL3, FORTRAN and POP-2.[2] Of these six languages, the first three did not come into serious contention: PLAN and BASIC are low-level languages and, apart from the complexities involved in writing a really long program at low-level, English as a high-level language (in the conscious brain) should have a high-level processor (simulator); ALGOL 60 is simply unsuitable for character/word handling. Of the remaining three languages, POP-2 was chosen for one overriding reason: it has excellent facilities for handling functions, their definition, notation and use; in particular, it allows the user to define and manipulate function variables (i.e. variables that can take on the 'value' (definition) of any previously defined function). Since the Functional Theory of Language is based on functions, this capability is a necessity for any program that hopes to model cognitive structures built by functions. Moreover, POP-2 is the only truly interactive language of the three concerned.

There is, however, one drawback to the use of POP-2. Being an on-line language, it requires part of the compiler to be in store at run-time. This takes up about 10K words of store.

---

[1] This restriction has recently been relaxed slightly, but was in operation for most of the time that PIP was being developed.

[2] ALGOL 68 has been implemented in the last year, but this was also too late for the current program.

Since only 24K are available to on-line users in the first place, that leaves the user just 14K words of store for the actual program to use. In comparison with Winograd's program for understanding Natural Language which uses 80K words of 36-bit computer store,[3] this is a pitiful amount. Nevertheless, even within these restraints, much can be achieved.

2. Aims and Objectives

PIP has been written with the aim of demonstrating that a computer (even a 1905E!) can understand English in any context. Since FTL is closely interrelated with the notion of 'cognitive structure', PIP will be 'asked' to listen in on a conversation between several 'people', building up a model of each person's cognitive structure as more information is input to it. Understanding will, for the time being,[4] be said to have occurred when PIP can detect all and only those clashes of attitude between the current input sentence and previous input sentences concerning a particular individual. Thus, if Jones says

(51) "I accuse Smith of stealing the money"

and some time later says

(52) "I praise my son for stealing the money"

---

[3] See Winograd [89], p. 7.

[4] See chapter V, section 1.1, for a revised 'definition' of understanding.

then PIP should pick up the clash between BAD [stealing the money] as a cognitive structure of Jones from (51) and the putative cognitive structure of GOOD [stealing the money] evinced by (52). Similarly, if Jones utters (51), but Robinson utters (52), then PIP should make no comment, as the two would then be perfectly compatible.

2.1 Since only 14K words of store are available for PIP, it is obvious that there must be some restrictions on the English which PIP will accept. There is no space for an elaborate syntactic analysis of the data input to PIP, so sentences must be kept simple; that is, they must include one main clause and either one subordinate clause or one adverbial phrase. The main clause must consist of a performative verb, in the explicitly performative form starting "I ...", and a direct object where required by the verb. Since PIP must also know who the speaker is, the format of a 'correct' sentence input to PIP is:-

(53) "<utterance>" said <speaker>.

where <speaker> is a one-word name, such as John or Jones, and the <utterance> is of the form:-

(54) I <verb> <optional direct object> <argument>

where <argument> is the subordinate clause or adverbial phrase as required by the verb. Thus, a 'correct' input sentence might be:-

(55) "I congratulate Marion on passing her exams" said John.

PIP's memory contains the presuppositions associated with each performative verb that it has met before. The user is asked to define any new verb in terms of those presuppositions that it 'knows'.

These particular restrictions were decided upon for two reasons. First, the illocutionary force of an explicitly performative utterance is perfectly clear. Assuming that the sentences are uttered sincerely, as PIP does,[5] then if Jones says "I promise ...", the illocutionary force is that of 'promising'; thus uptake is easy for PIP to secure.[6] Secondly, it must be remembered that Austin believed all performative utterances capable of 'reduction' to explicitly performative forms.[7] All the language activities in which human beings indulge (joking, ordering, praying etc.) can easily be seen as illocutionary acts, with the possible exception of the act of referring. Searle has shown[8] that the act of referring is also an illocutionary act. Illocutionary acts are completed when the hearer has understood the illocutionary force of the speaker's utterance. At least part of the illocutionary force is carried by the performative verb. Thus, in theory and Austin's belief, all utterances should be capable of 'reduction' to explicitly performative forms, in which all of the illocutionary force is carried. Therefore, if PIP can understand the explicitly performative forms, then, at some

---

[5] Cf. chapter I, section 3.2.1 and section 4.2 of this chapter.

[6] Cf. chapter I, section 3.2.4.

[7] See chapter I, section 3.2.2.

[8] See chapter I, section 3.3.

later stage when more memory is available, PIP should be able to understand sentences in any form. Valuable space is thus not wasted at this experimental stage in transforming sentences into their equivalents simply to be able to claim that the program can cope with Language in any form. Thus, where Winograd [89] has written a program showing that computers can understand any sentence of English in a restricted context, PIP will attempt to show that computers can understand a restricted sub-set of English in any context.

2.1.1 It is now plain that the class of verbs that is basically the object of study in this thesis is the class of explicitly performative verbs. However, that class will be expanded slightly. Since FTL is founded upon the notion of cognitive structures, those verbs that report on the state of an individual's cognitive structure, for example "know", "believe" or "think", will also be included, provided that they are also used in the explicitly performative form. They are not true explicitly performative verbs only in the sense that to say "I believe that ..." is not itself to perform the act of believing. In the fact that the illocutionary force of an utterance using such verbs is perfectly clear, they act in exactly the same way as 'ordinary' explicitly performative verbs like "promise".

2.2 It must be made quite clear that no amount of experimentation will ever 'prove' that FTL is the 'correct' solution to the problems in the Philosophy of Language. Karl

Popper [68] has shown very clearly that theories remain theories, however probable they might appear, in the same way that general universal statements can only be shown to be false, never true. He says, for instance, "By the method of elimination [of false theories] we may hit upon a true theory. But in no case can the method establish its truth, even if it is true; for the number of possibly true theories remains infinite, at any time and after any number of crucial tests".[9] It can be shown that FTL is false (incorrect), if, for example, PIP palpably does not understand the sentences input to it. A positive result from the experiment will only serve to make the theory more probable. On the other hand, AI has no need of such theoretical rectitude, provided it gets results. Therefore, any positive result from PIP will be a step forward in the field of Artificial Intelligence.

3. Syntactic Analysis

The programming of PIP is separated into two parts: syntax and semantics.[10] The purpose of the syntax is to provide the semantic analyzer with information about who the speaker is, to whom the utterance is addressed, what the various components of the utterance are, and the 'names' of the presuppositions inherent in the verb used in the utterance. In a system

---

[9] Popper [68], p. 15, italics in the original.

[10] A diagram of the overall control structure can be found in Appendix A.

containing a complete grammar (of some specified kind) and a complete semantic analyzer (that is, one which 'discovers' the meanings of nouns and adjectives as well as verbs), there would have to be some degree of interaction between the grammar and the semantics, but in the limited case of PIP that is not necessary.

3.1 A flowchart of the function PARSE[11] appears on pages 94 - 97. This function controls the syntactic analysis that PIP performs. First of all, the input sentence is read. A full-stop is taken to be the terminator of each sentence. WHOSAID then finds the name of the speaker. This must be one word only: if more than one word is used, then PIP will assume that only the first word was meant. There is therefore no difference for PIP between Ted Heath and Ted Ray. Again, this is a restriction introduced to save computer space. In an expanded version this restriction would be removed. If no speaker is found, that is the sentence consists simply of an utterance,[12] then PIP assumes that the speaker is the person who was previously speaking. This is consistent with normal English usage.

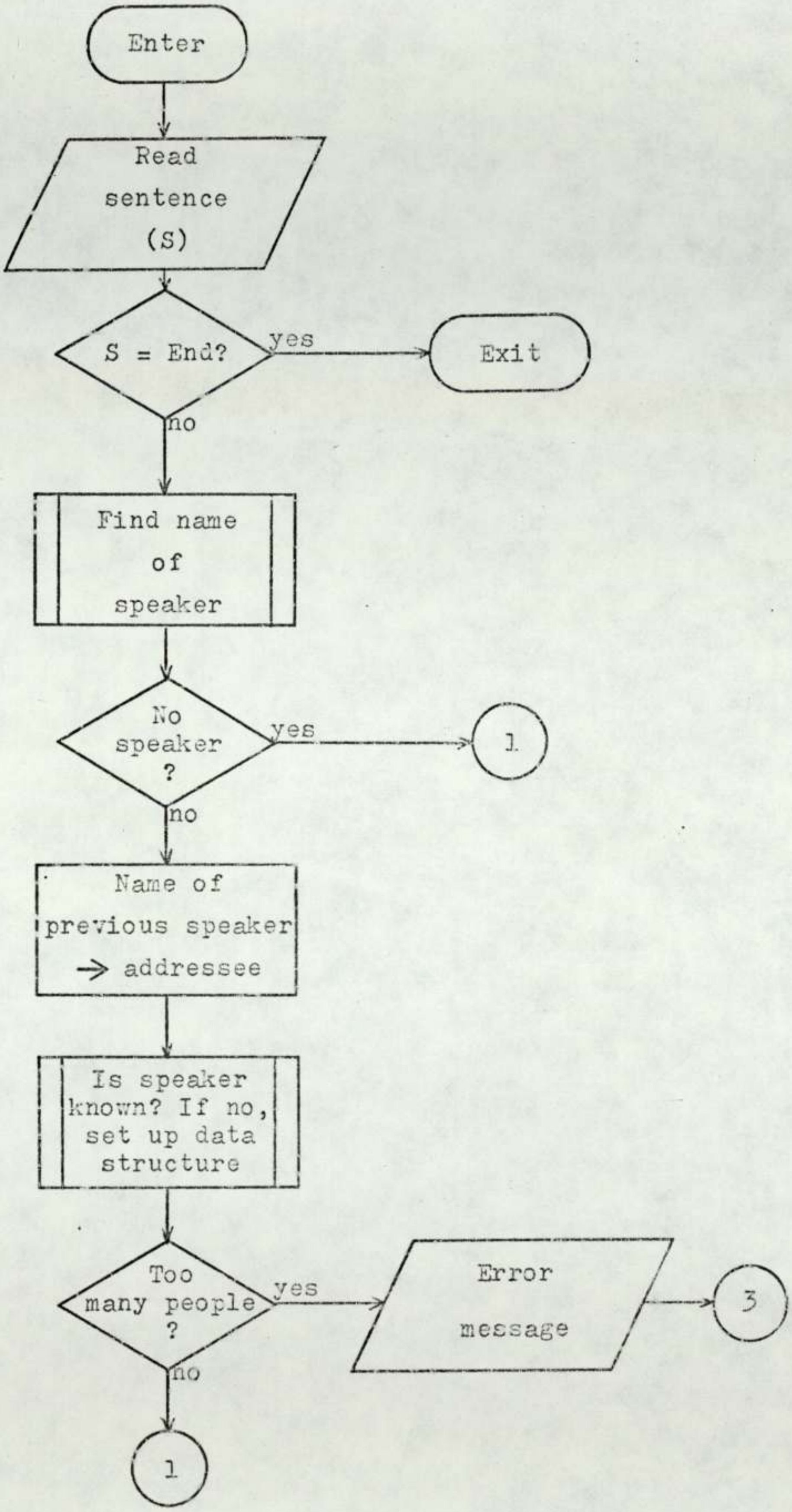
One of the theoretical problems raised by having PIP listen in to conversations in this manner is that of deciding

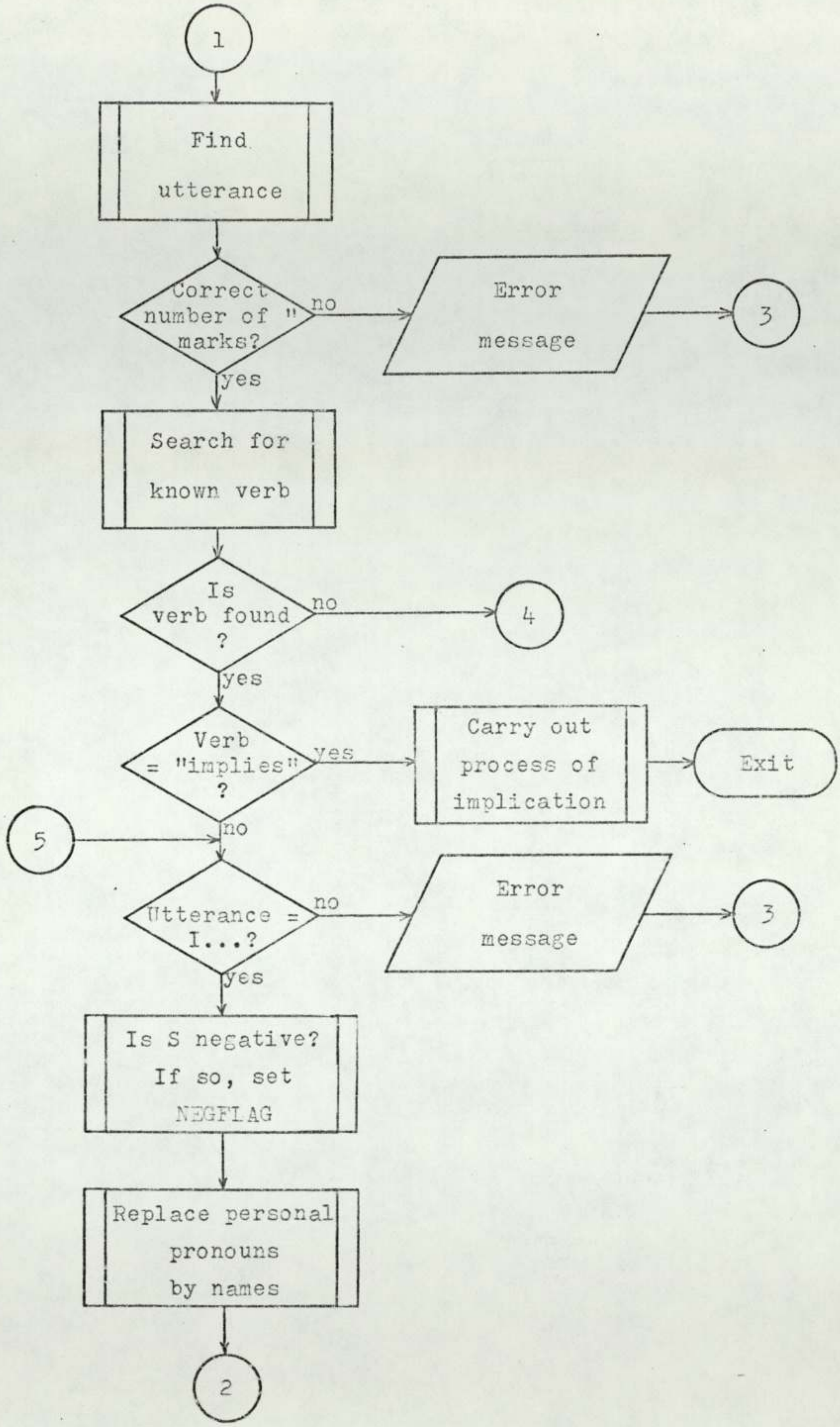
---

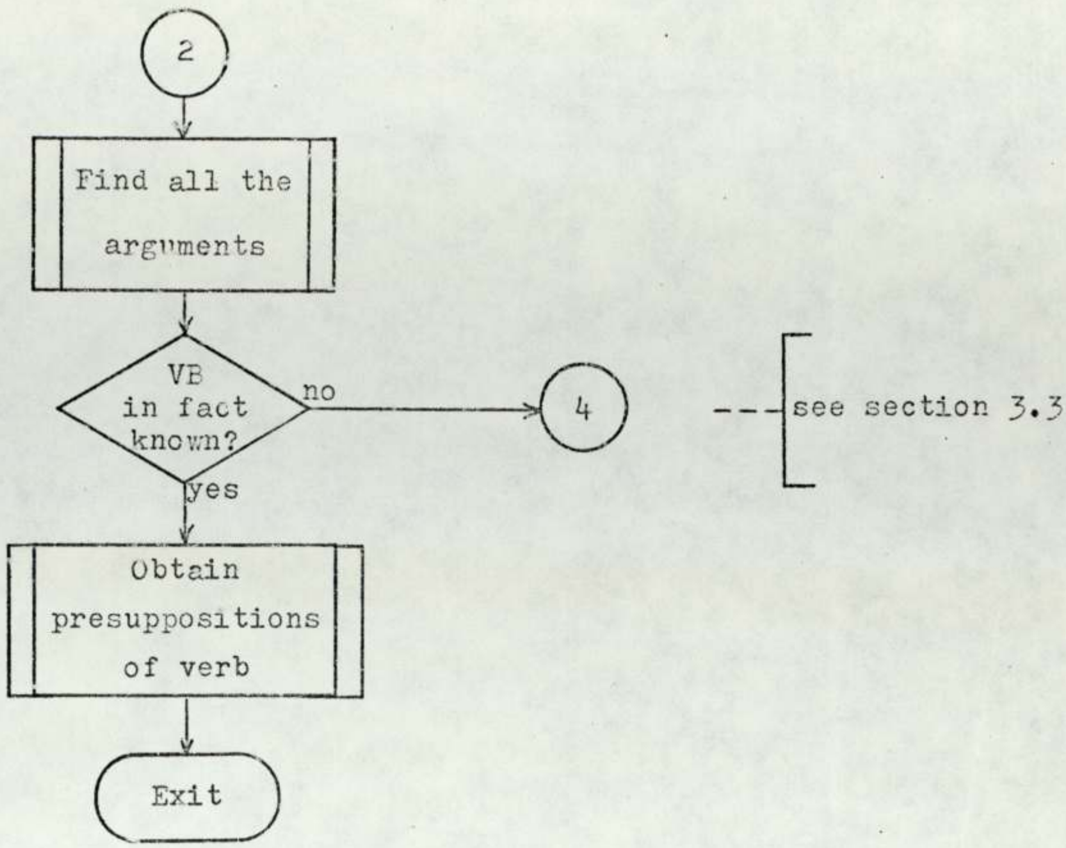
[11] All function names will be written in capital letters and underlined; global variables will also be written in capital letters, but not underlined.

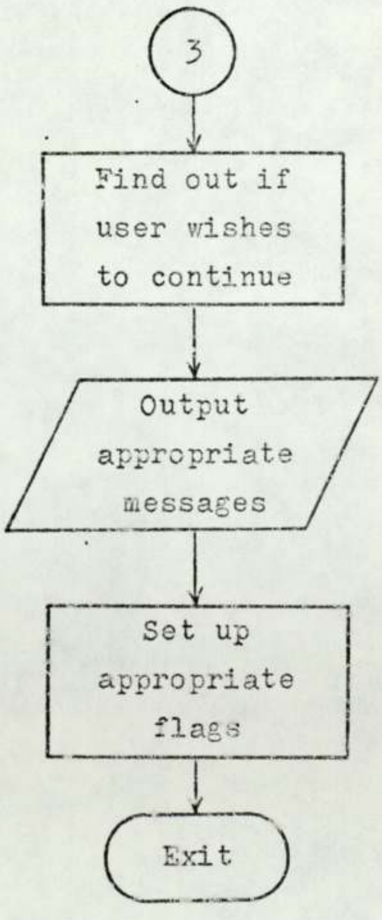
[12] A 'sentence' is the whole of the input stimulus to PIP, whereas an 'utterance' is what is contained within quotation marks.

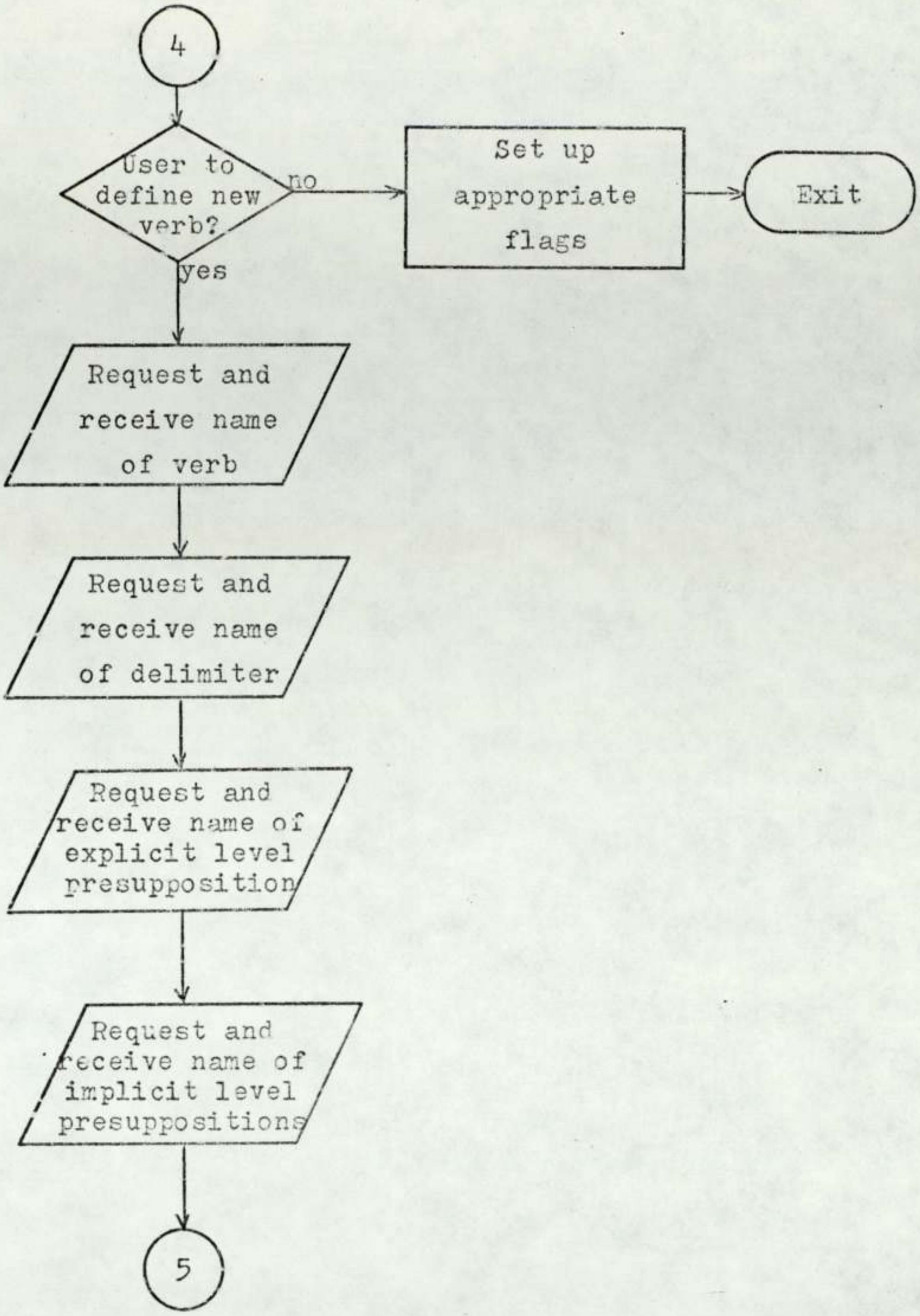
FUNCTION PARSE




1
Find
utterance
Correct
number of "
marks?
no
Error
message
3
yes
Search for
known verb
Is
verb found
?
no
4
yes
Verb
= "implies"
?
yes
Carry out
process of
implication
Exit
5
no
Utterance =
I...?
no
Error
message
3
yes
Is S negative?
If so, set
NEGFLAG
Replace personal
pronouns
by names
2


2
Find all the arguments
VB in fact known?
no
yes
4
see section 3.3
Obtain presuppositions of verb
Exit


3
Find out if user wishes to continue
Output appropriate messages
Set up appropriate flags
Exit


4
User to define new verb?
no
Set up appropriate flags
Exit
yes
Request and receive name of verb
Request and receive name of delimiter
Request and receive name of explicit level presupposition
Request and receive name of implicit level presuppositions
5

to whom each utterance is addressed. Utterances might be addressed to everybody present, or to only one person. If they are addressed to only one person, that person might not necessarily be the previous speaker. However, PIP does not have eyes to see whether the current speaker has turned to any one person, nor does the acceptable sentence structure allow the addressee to be named. This being the case, PIP always assumes that the addressee is the person who was the speaker of the previously input sentence. It is necessary to know who the addressee is, since some verbs (e.g. "promise") have presuppositions which refer to the cognitive structures of the addressee as well as those of the speaker.[13] A maximum of nine speakers is allowed in any one conversation; to save space.

3.2 Having removed the two double quotation marks surrounding the utterance (and checked that both were present), PARSE now concentrates on the utterance itself. The most important word in the utterance is the (performative) verb. This is found by straightforward table look-up techniques. Before the user is allowed to input any sentence to PIP, the program 'remembers' those verbs that have been defined in previous runs of the program, reading the verbs, the 'delimiter'[14] and the presuppositions associated with each verb into a table, each entry of which is a record[15] of four

---

[13] See chapter III, section 3.1.4.

[14] See below for an explanation of the term 'delimiter'.

[15] Burstall et al. [7], pp. 33 - 35 and pp. 103 - 104; this book will henceforward be referred to simply as POP-2.

parameters. If no verb is found, then PIP reports that fact to the user, who is asked if he wishes to define the verb.[16] If not, the program passes on to the next sentence. If the user does wish to define the new verb, then he must state what the verb is and what the delimiter associated with that verb is. Consider the following utterances:-

(56) I believe that Fred killed the postman

(57) I accuse Fred of killing the postman

(58) I blame the powerful rulers of the Arab countries for the rise in world prices of oil.

In (56) the delimiter is the word "that", in (57) it is "of", in (58) it is "for"; that is to say, the delimiter is either a preposition when the <argument> of (54) is an adverbial phrase introduced by that preposition, or a conjunction introducing a subordinate clause as the <argument> of (54). The user is then asked to state the explicit and implicit level presuppositions inherent in the given verb. These must be in terms of those presuppositions previously known by and defined in the system.[17] It might have been possible for PIP to define the verbs for itself, thus modelling human acquisition of verbal 'meaning', but this would have required PIP's asking a lot of questions of the user in terms of verbs that it did know. This would have taken up a considerable amount of actual time on a not particularly fast computer; it was therefore decided to let the user simply define any new verb, the

---

[16] This is an example of PIP's learning capabilities.

[17] See section 4.4 of this chapter, and chapter III, section 3.

presence or absence of such a self-defining process not being vital for the overall success or failure of the program. One particular verb that PIP knows, namely "implies", is not an explicitly performative verb. The reason for its presence is both semantic and syntactic, but since its main purpose is semantic, it will be explained later, in section 5.

PARSE now proceeds to do some syntactic 'housekeeping'. It checks that the utterance is explicitly performative (that is that the first word is "I"). It checks for the presence of the word "not", indicating that the utterance is negative. If it is found, then it is removed and an appropriate flag is set. Personal pronouns (with the exception of "they" or "them") are replaced by the appropriate names of 'people' (notice that "you" will always be replaced by the name of the addressee; PIP has no way of dealing with utterances aimed at more than one person).

3.3 The last main task of PARSE is to calculate all the arguments of the utterance. Consider:-

(59) I believe that all men are mortal.

(60) I accuse Fred of spilling the beer.

(61) I confirm that Charles did not help the wounded man.

In (59), the argument of the verb "believe" is quite clear. It is the clause "all men are mortal" which immediately follows the delimiter. Any presuppositional function will apply to that argument. However, in (60), it is necessary that PIP should recognize that the speaker's cognitive structure includes at least the following:-

(62)a. RSPONSBL [Fred, spilling the beer]

b. BAD [spilling the beer]

c. VALIDBLF [the beer was spilled]

d. VALIDBLF [Fred spilled the beer].

Each of the arguments of the various presuppositional functions in (62) is different. These arguments must be identified before the functions can be applied to them. Since this is not a semantic operation in PIP, it is performed by PARSE. In (61), once the word "not" has been removed, the argument is "Charles did help the wounded man". However, this must be capable of a contrast with the argument of

(63) I believe that Charles helped the wounded man.

"Did help" in (61) is therefore transformed into "helped". All these processes are carried out by the function FINDALL, which finds all the possible arguments from the utterance, even if they are not in fact used by the semantic analyzer. During this process, it might be discovered that the verb used was not the verb PIP had previously found. This can only occur if one lexical item can have two different senses, where the only difference is in the delimiter,[18] for example, "believe that" and "believe in". If this is the case, then FINDALL checks whether the verb and its new delimiter are known; otherwise the user is asked to define the verb as if it were totally unknown to PIP.

---

[18] PIP cannot deal with two or more senses of one verb without this difference, but explicitly performative verbs should never, in theory, be capable of two interpretations.

Finally, the presuppositions inherent in the verb used in the utterance are obtained and placed on a list, PRESUPL, which is used by the semantic analyzer. The processes outlined above are the only syntactic analysis and syntactic checks that are performed in this program.

4. Semantic Analysis

It is the purpose of the semantics section of PIP to model the cognitive structure of the speaker and addressee of any given utterance; to check for and remark on any inconsistencies between the cognitive structures of those individuals and the implied structure of the utterance; to add any new information contained in that utterance to those cognitive structures.

4.1 It is important to say at this stage exactly how cognitive structures are modelled by PIP. Each individual's cognitive structure is held on a list[19] each element of which is also a list. Each of these inner lists is composed of one of the arguments used by that individual in utterances that PIP has 'overheard', prefixed by an '<u>attitude</u> <u>marker</u>'. This attitude marker is a number comprising eleven pairs of binary digits (the maximum allowable on the ICL 1905E in one computer word). For example, PIP's model of the cognitive structure of the character 'John' might be as given in example (64) on page 103. The various presuppositions inherent in the verbs used in

---

19 <u>POP-2</u>, pp. 20 - 24, 28 - 33, and 106 - 108.

(64) [[0101010101010101010100 Fred is innocent]
0101010101010101011001 robbing banks]

the utterance act (as functions) on different pairs of binary digits. The binary digits are grouped in pairs in order to have "yes" (00), "no" (10) and "don't-know" (01) as possible replies by the individual to a question concerning any attitude to any argument, and also in order to allow later expansions of the program to use the rules of three-value logic to determine the relationships between various arguments. It is the attitude marker which is at the heart of the understanding which PIP shows.

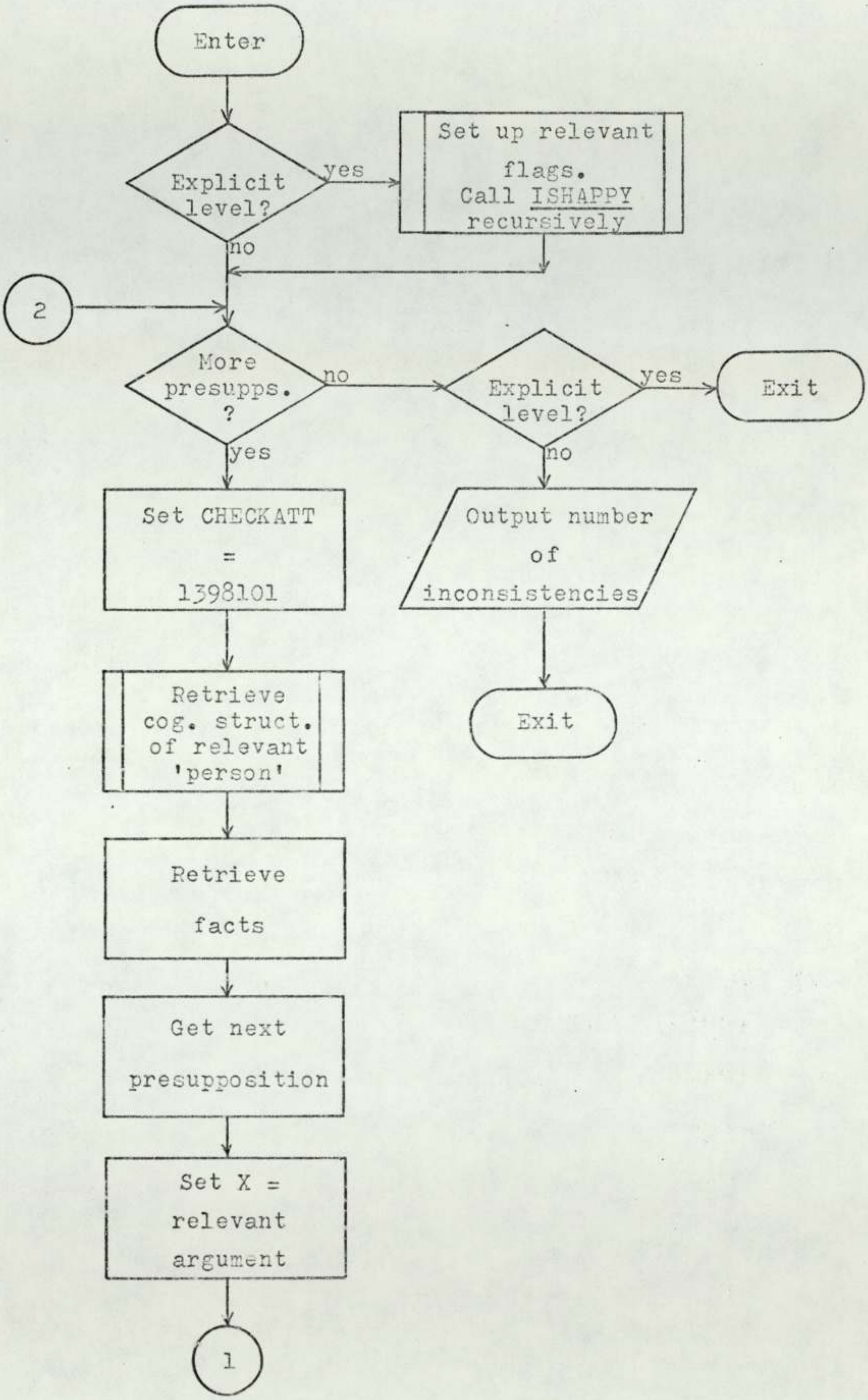
4.2 The function controlling the semantic analysis is called ISHAPPY, since it checks the happiness conditions for the use of a particular verb.[20] A flowchart of ISHAPPY appears on pages 104 - 105. It first checks whether it is dealing with the explicit level of communication or not. Although the method of proceeding is the same at each level, it is necessary for PIP to know which level it is working at, in order that appropriate messages are output to the user. When ISHAPPY has finished processing each utterance, it outputs a total number of inconsistencies to the user and then awaits further input.

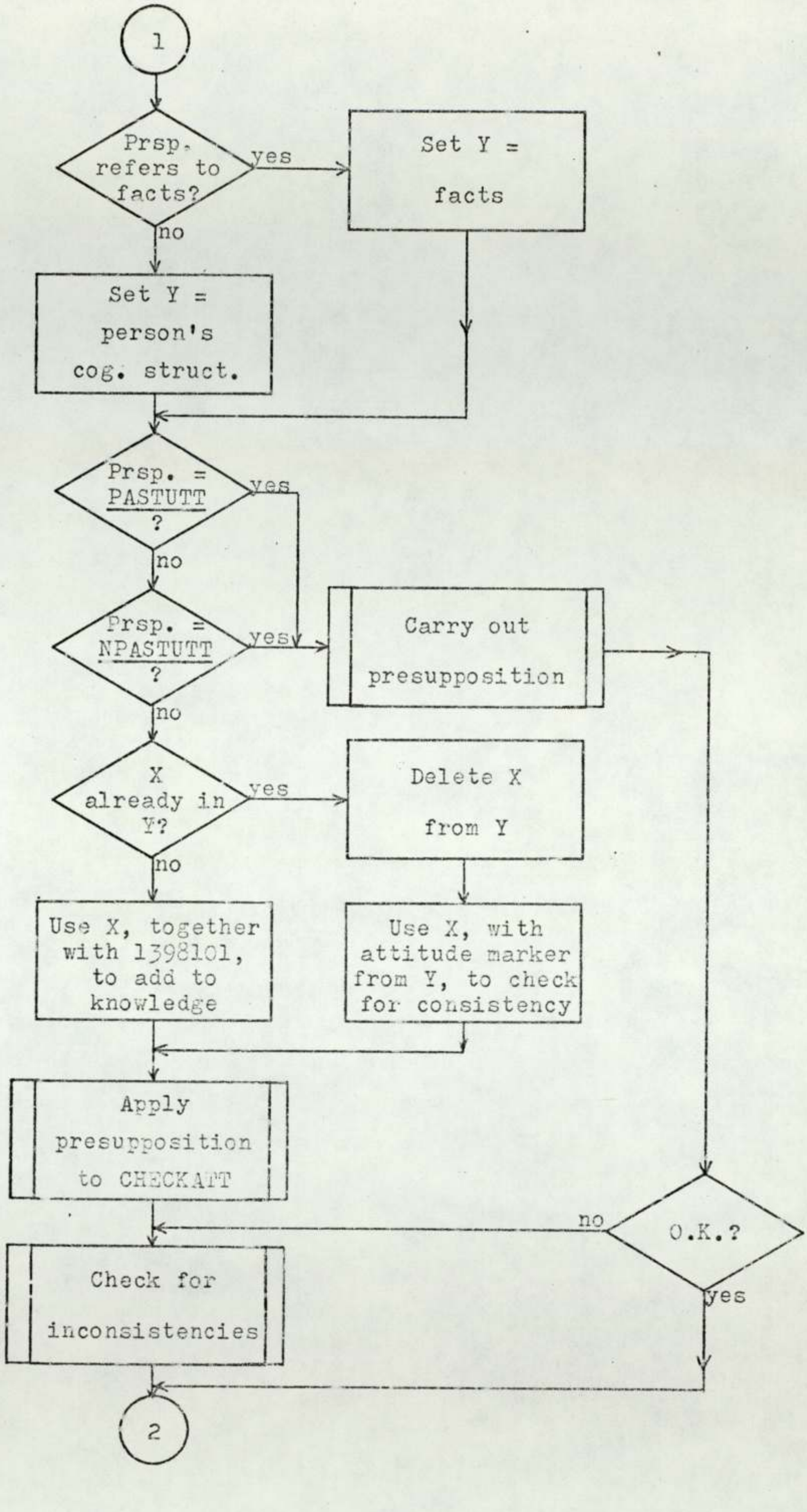
The body of the function begins by setting the global variable CHECKATT equal to the decimal number 1398101 which is equivalent to the binary number 0101010101010101010101. It is

---

[20] Cf. chapter I, section 3.2.1.

FUNCTION ISHAPPY




1
Prsp. refers to facts?
yes
Set Y = facts
no
Set Y = person's cog. struct.
Prsp. = PASTUTT ?
yes
no
Prsp. = NPASTUTT ?
yes
Carry out presupposition
no
X already in Y?
yes
Delete X from Y
no
Use X, together with 1398101, to add to knowledge
Use X, with attitude marker from Y, to check for consistency
Apply presupposition to CHECKATT
no
O.K.?
Check for inconsistencies
yes
2

this variable which will be altered by the presupposition that ISHAPPY is about to consider. ISHAPPY then retrieves the cognitive structure of the speaker (this will be the NULL or empty list if the speaker has not previously spoken) and whatever facts about the world it might know. Some verbs, such as "confirm", imply not only that the speaker believes the argument to be true, but that it is true in actual fact. PIP tries to build up a model of these 'facts' as well as a model of individuals' belief structures. This is only possible because PIP assumes that speakers are sincere in their utterances and that they know the meanings of each verb. This assumption must be made at this stage for substantially the same reason that PIP concentrates on explicitly performative verbs, namely that it is important to ensure that straightforward utterances can be understood before attempting to deal with any of the more complex aspects of Language. Until one can show that sincere utterances can be understood according to FTL, it is theoretically undesirable to let PIP attempt to understand jokes, untruths, or verbs used with unusual overtones, such as "I promise ..." to mean "I threaten ...".

4.3 ISHAPPY now takes the next presupposition from PRESUPL and finds out whether it refers to facts or personal beliefs, forgetting whichever is inappropriate. It then retrieves the argument that is relevant to the particular presupposition. In POP-2, it is possible to associate extra information with a function by use of a standard function. This function is called

FNPROPS.[21] As far as the presuppositions are concerned, this extra information is a list consisting of the name of the presupposition itself and the name of the variable which will contain the argument to which the function applies. If the presupposition is either PASTUTT or NPASTUTT, then it is applied to the argument yielding a value of true or false. If the value is true, then there is no inconsistency; if, however, the value is false, then there is an inconsistency and (part of) the function BINARY is called, which outputs relevant information to the user.

For the rest of this explanation of ISHAPPY it will be assumed that the presupposition refers to personal beliefs, although exactly the same process is followed with facts. ISHAPPY checks whether the argument to which the presupposition is applicable already exists in the speaker's cognitive structure. If it does exist, then it is deleted, together with its attitude marker, from the speaker's cognitive structure, to be replaced (by the function BINARY) with an appropriately altered attitude marker if necessary. In general, if the argument is found in a cognitive structure, then it will be used for checking purposes. If it is not found, then ISHAPPY makes up a list composed of the number 1398101 and the relevant argument which, after appropriate action by the presupposition, will be added to the speaker's cognitive structures, thus representing an increase in PIP's knowledge of that speaker.

The presuppositional function is then called, which alters

---

[21] POP-2, pp. 58 and 111.

CHECKATT appropriately. The function BINARY is then called, which compares CHECKATT with the attitude marker (or 1398101) of the argument found in the previous stage (see previous paragraph). If there are any inconsistencies, then appropriate comments are made. In this case, the original cognitive structures remain unaltered. Thus, 'people' cannot change their minds. If the particular function now under examination has not previously been applied to the argument, then the attitude marker is altered accordingly. After the argument has been replaced by BINARY as part of the speaker's cognitive structure, control passes back to ISHAPPY which returns to pick up the next presupposition and begin again.

4.3.1 It is a matter of fact that PIP represents each person's cognitive structure separately, without any cross-referencing to other people. Thus, although several people might have some attitude to the argument

(65) [stealing money from the government]

PIP will not know this. Moreover, the list (65) is repeated for each person who has an attitude to it. This is wasteful of space. This particular problem was only highlighted during the latter stages of developing PIP and it was not thought necessary to alter the programming. Nevertheless, should anyone else wish to develop PIP for their own purposes, it is suggested that the arguments only be kept on a list, say the list GESAGT, which already exists. It contains all the arguments used in a run, for the purposes of PASTUTT and NPASTUTT. It is also suggested that the model of each person's cognitive structure contain merely a list of attitude markers and pointers to the appropriate element of GESAGT. This should save space in the computer memory.

4.4 The two types of presupposition mentioned in chapter III, section 3.1.5 and section 3.2, namely VALUE and AUTHORTY, have not been programmed. This was mainly for lack of space, but VALUE was omitted for a further reason. The judgement that is made when using a verb, one of whose presuppositions is VALUE, is not apparent solely from the verb used.

(66) "I value that jewel at £500."

(67) "I value that jewel at £5."

The verb used in each of the utterances (66) and (67) is the same, but the judgement is different. It is therefore necessary to know what value the speaker places upon the object. However, the syntax is more complicated than in the sentences described above in section 3. Compare (66) with (68):-

(68) "I value that jewel highly."

An adverb follows the article valued in (68), not an adverbial phrase introduced by a preposition; there is therefore no 'easy' way of separating the utterance into its various parts without 'knowing' any syntactic rules. Although PIP cannot therefore handle such a construction, there is no theoretical bar to its being handled by any system with a powerful syntactic analyzer.

4.4.1 One further point ought to be mentioned in connection with presuppositions. The logical rule of implication holds between some of them. Consider

(69) "I accuse Seamus of helping the IRA."

One of the presuppositions of "accuse" in (69) is

(70) RSPONSBL [Seamus, helping the IRA]

but this also means that

(71) VALIDBLF [Seamus helped the IRA]

is part of the speaker's cognitive structure. It is true in general that RSPONSBL implies VALIDBLF. Therefore, RSPONSBL has been written so that VALIDBLF is added to the list of presuppositions; it does not have to be part of the presupposition list remembered by PIP. In a similar way, NRSPNSBL implies NVALIDBF, VALID implies VALIDBLF and NVALID implies NVALIDBF.

4.5 The following is a table of the verbs that PIP knows at the current time, together with the delimiter, explicit level presupposition and implicit level presuppositions (if any) associated with each verb.

accuse	of	RSPONSBL	BAD VALID
acquit	of	NRSPNSBL	BAD
admit	that	RSPONSBL	BAD CONFDENT VALID PASTUTT
advocate	that	GOOD	WANTS
affirm	that	VALID	CONFDENT PASTUTT
approve	of	GOOD	
ask	for	WANTS	
assert	that	VALIDBLF	CONFDENT

believe	that	VALIDBLF	
blame	for	RSPONSBL	BAD VALID
claim	that	VALIDBLF	CONFDENT
compliment	on	GOOD	RSPONSBL
condemn	for	RSPONSBL	BAD
confess	that	RSPONSBL	BAD CONFDENT VALID PASTUTT
confirm	that	VALID	CONFDENT PASTUTT
congratulate	on	GOOD	RSPONSBL VALID
contend	that	VALIDBLF	NVALDBFH PASTUTT
credit	with	RSPONSBL	GOOD VALID
declare	that	VALIDBLF	CONFDENT
deny	that	NVALIDBF	CONFDENT
despise	for	BAD	RSPONSBL
disapprove	of	BAD	
disbelieve	that	NVALIDBF	
dislike		NWANTS	BAD
doubt	whether	NVALIDBF	NCONF
dread	that	NWANTS	BAD
grant		WANTH	NWANTS PASTUTT
guess	that	VALIDBLF	NCONF
hate		NWANTS	BAD
hold	that	VALIDBLF	
imagine	that	VALIDBLF	NCONF NPASTUTT
implies			
indict	for	RSPONSBL	BAD VALIDBLF
intend	to	FUTACT	
know	that	VALIDBLF	CONFDENT
like		WANTS	GOOD
love		WANTS	GOOD

maintain	that	VALIDBLF	
pardon	for	NRSPNSBL	FUTACT BAD
pledge		FUTACT	WANTH
praise	for	GOOD	RSPONSBL VALID
presume	that	VALIDBLF	NPASTUTT
proclaim	that	VALID	CONFDENT
promise	to	FUTACT	WANTH
protest	that	RSPONSBL	NVALDBFH GOOD PASTUTT
punish	for	RSPONSBL	BAD VALID
question	whether	NVALIDBF	PASTUTT
refuse	to	NFUTACT	NWANTS
refute		NVALIDBF	VALIDBFH PASTUTT
reject		NVALIDBF	VALIDBFH PASTUTT
reprimand	for	BAD	RSPONSBL
reproach	for	RSPONSBL	BAD
reprove	for	RSPONSBL	BAD
request	that	WANTS	
state	that	VALIDBLF	CONFDENT
suggest	that	VALIDBLF	CONFDENT NPASTUTT
suppose	that	VALIDBLF	NCONF
surmise	that	VALIDBLF	NPASTUTT
suspect	that	VALIDBLF	NCONF NPASTUTT
swear	that	VALIDBLF	CONFDENT
think	that	NCONF	VALIDBLF
vow	that	VALIDBLF	CONFDENT
want	to	WANTS	
want		WANTS	
wish	for	WANTS	
wonder	whether	NCONF	VALIDBLF NPASTUTT

5. The Verb "Implies"

There are two occasions on which one might want to inform PIP that some individual's attitude to the statement "x" is the same as that person's attitude to the statement "y". Consider

(72) John said "I believe that Harry is innocent".[22]

(73) "I believe that Harry is guilty" said John.

It is important that PIP recognize that (72) and (73) are mutually incompatible. In order to accomplish this, the attitude marker of "Harry is innocent" must be transferred to NOT ("Harry is guilty") and both replaced as part of John's cognitive structure. This is achieved by the sentence

(74) "Harry is innocent implies Harry is not guilty".

This causes PIP to take the following actions:-

1. PIP searches the linguistic world of each individual it 'knows', trying to find the argument on the left-hand side of the verb "implies". If it is not found, then PIP carries on to the next individual.

2. If the argument on the left-hand side of "implies" has been found for any individual, then PIP searches that individual's cognitive structure a second time, looking for the argument on the right-hand side of "implies". If it is not found, then PIP adds the appropriate attitude marker to this second argument and adds it to the individual's cognitive structure; PIP then passes on to the next individual.

---

[22] Although (53) states that 'said <speaker>' should follow the utterance, PIP will accept these two in any combination that English will allow.

3. If both arguments already belong to an individual's cognitive structure, then PIP checks their respective attitude markers for consistency. If they are consistent, then PIP carries on to the next individual. If they are not, PIP types a message stating

(75) FOR 'x' THERE WERE 'n' INCONSISTENCIES

where 'x' is the individual concerned and 'n' is the number of attitudes which were inconsistent. The argument on the right-hand side of "implies", together with its attitude marker, is then deleted from that person's cognitive structure. This is the only occasion on which PIP alters cognitive structures once they have been constructed.

Thus, if (74) is inserted between (72) and (73), then PIP will show an inconsistency after (73) has been input. If it follows (73), however, then no inconsistency will be found after (73), but PIP will reply

(76) FOR JOHN THERE WAS 1 INCONSISTENCY

after (74) has been input. These implications are not remembered after they have been input, since that would be costly in terms of both space and time. That is to say, if individuals use the expression "Harry is innocent" who were not known to PIP at the time (74) was input, then (74) will have to be input again; the alteration to the cognitive structures of those individuals who were known to PIP on the first occasion (74) was input is, of course, permanent for the duration of the session.

5.1 The second occasion on which "implies" might be used arises because of the lack of syntactic analysis performed by PIP. In particular, it does not know what 'strong' (irregular) verbs are. Consider

(77) John said "I accuse Fred of stealing five pounds".

Two cognitive structures belonging to John will be modelled by PIP in this way:-

(78)a. VALIDBLF [five pounds was stealed]

b. VALIDBLF [Fred stealed five pounds].

It might, however, be necessary (or desirable) that PIP notice the anomaly between (77) and

(79) "I do not believe that Fred stole five pounds" John said.

In that case, the user must at some stage type the sentence

(80) "Fred stealed five pounds implies Fred stole five pounds".

This use of "implies" is subject to the same restrictions on PIP's 'memory' as above, and would, of course, be totally unnecessary if PIP had any syntactic (or possibly even lexical) analysis to tell it about 'strong' verbs.

5.2 There is one kind of generalized implication that has not been programmed. Consider

(81) "'i' is guilty implies 'i' is not innocent".

where 'i.' is intended to be a variable. There are a number of reasons why this kind of 'rule' has not been programmed. First, implications are not remembered, and (81) is essentially an implication that is intended to be constant throughout a run of PIP. Secondly, and perhaps more importantly, such an implication would require a fair amount of 'semantic' analysis, for which neither the time nor the space is available. Furthermore, such analysis would be within one argument, and arguments in PIP are whole and inviolate. No analysis of the arguments themselves is performed anywhere else in PIP. Considering that implication is a minor part of the system and not of prime interest, it was decided to forego the facility of allowing general implications.

6. An Example

In the following example, which will illustrate some, but not all of the points made above, the input sentences will be typed normally and PIP's replies and comments will be in block capitals. Numbers in parentheses indicate comments which follow the example.

PIP IS NOW READY FOR USE

DO YOU REQUIRE INFORMATION?

No (1)

NOW FOR THE FIRST SENTENCE

"I doubt whether Fred is innocent" said John.

O.K.

"I promise to acquit Charles".

THERE IS NO ADDRESSEE FOR THIS SENTENCE (2)

THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE

Fred said "I deny that I stole the money".

O.K.

"Fred stole the money implies Fred is not innocent".

O.K.

"Fred is innocent implies Fred did not steal the money".

O.K.

"Fred is not innocent implies Fred is guilty".

THE LEFT-HAND-SIDE OF AN IMPLICATION MUST BE POSITIVE (3)
TRY AGAIN

"Fred is innocent implies Fred is not guilty".

O.K.

"I do not believe that Fred is guilty" John said.

THE MEANING OF BELIEVE IS:-
VALIDBLF (NOT [FRED IS GUILTY])
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF JOHN (4)

THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE

"I blame Fred for stealing the money".

O.K.

"Fred stealed the money implies Fred stole the money". (5)

O.K.

"I confess that I am guilty" Fred said.

A PRESUPPOSITION OF CONFESS IS:-
VALIDBLF [FRED IS GUILTY]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF FRED

THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE

Mary said "I refuse to go down to the pub tonight".

O.K.

"Go down to the pub tonight implies take me to the pub tonight". (6)

O.K.

"I wish for a peaceful evening".

O.K.

"I promise to take you to the pub tonight" said John.

A PRESUPPOSITION OF PROMISE IS:-
    WANTH [TAKE MARY TO THE PUB TONIGHT]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF MARY (7)

THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE

"A peaceful evening implies go down to the pub tonight".

FOR MARY THERE WAS 1 INCONSISTENCY (8)

"I praise Fred for not stealing the money" John said.

A PRESUPPOSITION OF PRAISE IS:-
    VALID (NOT [THE MONEY WAS STEALED])
THIS IS INCONSISTENT WITH A PREVIOUS FACT (9)

A PRESUPPOSITION OF PRAISE IS:-
    RSPONSBL (FRED , NOT STEALING THE MONEY)
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF JOHN

A PRESUPPOSITION OF PRAISE IS:-
    VALIDBLF (NOT [THE MONEY WAS STEALED])
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF JOHN (9)

A PRESUPPOSITION OF PRAISE IS:-
    VALIDBLF (NOT [FRED STEALED THE MONEY])
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF JOHN

THERE WERE 4 INCONSISTENCIES IN THAT UTTERANCE

"I accuse Roy of spilling the coffee" said Lynne.

O.K.

"Roy spilled the coffee implies the fact that Roy spilled the coffee".

O.K.

"I do not believe that Roy spilled the coffee".

THE MEANING OF BELIEVE IS:-
    VALIDBLF (NOT [ROY SPILLED THE COFFEE])
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF LYNNE

THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE

George said "I refute the fact that Roy did not spill the coffee".

A PRESUPPOSITION OF REFUTE IS:-
VALIDBFH (NOT [THE FACT THAT ROY SPILLED THE COFFEE])
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF LYNNE (10)
THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE
End.
THE ACTUAL NUMBER OF COMPUTER WORDS USED ON THIS RUN WAS:-
12058 (11)

(1) If "yes" is answered to this question, then PIP gives some advice on its own use and a list of the verbs (with their delimiters) that PIP knows. However, this can take some time when sitting at a terminal (it has been known to take as long as one hour). This not being very practical, a version of PIP has been written allowing PIP to be run from cards when this information is automatically given. This version also prints all the arguments used by each sentence. The user is not allowed to define any new verbs in this version.

(2) Since PIP has only 'met' one speaker, John, it does not know to whom John is addressing this utterance and of whose cognitive structure WANTH [acquit Charles] should be part. PIP therefore assumes that John is in an empty room and has therefore made a defective promise.

(3) In first order logic 'not(p) implies q' is not the same as 'q implies not(p)'. However, "implies" is used by PIP to transfer attitude markers from one attitude to another; that is to say, it uses the rule of logical equivalence. It is true

that 'not(p) is equivalent to q' is the same as 'q is equivalent to not(p)'. Why not then allow the left-hand argument to be negative, since they are equivalent? Because the verb "implies" has been used, and, in the limited case where p is positive (and true), 'p implies q' is equivalent to 'p is equivalent to q' and 'p implies not(q)' is equivalent to 'p is equivalent to not(q)'. Therefore, p is kept positive.

(4) First, the word "meaning" is used to express the explicit level of communication; "presupposition" is used to denote what is implicit in the utterance. Secondly, note that PIP has followed the reasoning from the previous implication statements.

(5) This is an example of the verb "implies" being used to correct PIP's (lack of) grammar.

(6) First, note that PIP considers Mary to be the speaker, since she was last to speak. Therefore, the word "me" will be changed to "Mary". The reason for having the sentence at all is that "go to the pub", "be taken to the pub" and "have someone take me to the pub" are all more or less synonymous (provided one assumes that Mary does not go to the pub on her own). The next comment will show the necessity of this sentence.

(7) Here it can be seen that "me" (comment (6)) has indeed been changed to "Mary", and that a promise is defective if the addressee does not want what is being promised. It would, however, have been slightly odd if John had said "I promise to go to the pub tonight" and PIP had still picked up an inconsistency. PIP could not tell that the action "go to the pub" as predicated of John or Mary would be different. Hence

the sentence commented by (6).

(8) This sentence shows that the user may realize that an inconsistency has not previously been found by PIP because PIP did not have all the relevant information. PIP's action will be as described in section 5.

(9) These inconsistencies should not really have been found, since they are not really presuppositions of the utterance. However, the word "not" and the whole notion of negation are extremely complex and outside the scope of this investigation.[23] Negation for PIP is simply propositional negation, applied to the arguments of all the presuppositions inherent in the particular verb. In this utterance, it should only apply to the presuppositions RSPONSBL and VALIDBLF with its second argument (the fourth inconsistency found here, but not the third). In an expanded version of PIP with more storage available this would have to be improved.

(10) One cannot "refute" something that is not believed by someone else, in this case the addressee.

(11) This program has been running in 23K words of store. The total mill time for the run was 51 seconds, of which approximately 8 seconds were required by the operating system.

An extended (uncommented) example of a run of PIP will be given in Appendix A.

---

[23] See Klima [43] for a review of the many different facets of negation; see also section 1.1 of chapter VI.

V. PIP TALKS

Since utterances are not merely understood by human beings, but are also produced by them, PIP should be capable not only of passively understanding other 'people', but also of producing its own sentences. This chapter will detail how PIP achieves this linguistic ability.

1. Aims and Objectives

First and foremost, PIP must be given a cognitive structure of its own. Without such a structure, according to the Functional Theory of Language, PIP should not be able to generate sentences in 'any context'. The term 'any context' must be more clearly defined. When PIP is simply monitoring other people's conversations, then 'any context' does literally mean 'any possible context'. However, when PIP has been given a cognitive structure and is asked to generate its own sentences, then this is not the case. It is first necessary to ensure that there are no theoretical restraints precluding any attitudes from being part of PIP's cognitive structure; that is to say, it should be as 'random' as a human being's cognitive structure is. Having made sure of this, it is then sufficient to restrict 'any context' to 'any context of which PIP has knowledge (for which PIP has an attitude)'. It will still be valid to draw general conclusions from any success PIP might have. The actual method by which PIP is given its cognitive structure will be explained in section 2. For the time being, it will be assumed that PIP has been given such a structure in some theoretically appropriate fashion.

1.1 In chapter IV, understanding was deemed to have occurred if PIP could detect clashes of attitude for individuals. In this second program, PIP is only in contact with one person, the user. There are three types of sentence that can be input: questions about PIP's attitudes, questions about the user's attitudes and statements by the user. For example:-

(82) I accuse Roy of spilling the coffee.

is a statement by the user, in exactly the same format as explained in section 2.1 of chapter IV. (PIP will still check for inconsistencies in the user's cognitive structure.)

(83) Do I believe that Roy spilled the coffee?

is a question about the user's cognitive structure (attitudes) and

(84) Would you accuse Charles of killing the postman?

is a question about PIP's attitudes. In all cases, understanding will be said to have occurred if PIP can reply correctly to the input sentence (or question), generating sentences using its own (explicitly performative) verbs wherever possible. For example, a reply to (84) might be

(85) No. I would blame Fred for killing the postman.

In this way, the criterion of comprehension to which PIP has to submit becomes more akin to the criterion on which children are judged. That is to say that instead of a mere detection of anomaly (important though that is), an actual reply to each sentence or question can be given. In particular, the ability to answer questions about its own cognitive structure is an important step forward for PIP.

It must be emphasized that all the restrictions[1] that applied to the first program with regard to the facilities available for running programs at the University of Aston apply here. Similarly, it must be re-emphasized that this second program of PIP does not seek to model human understanding of any sentence of English any more than the first program did.[2] Here, too, it is PIP's ability to handle any context that is important.

2. PIP's Cognitive Structure

It would have been possible for the author to 'invent' a cognitive structure and present it as PIP's. This was thought, however, to be unethical and not conducive to conclusive results. It was therefore decided to allow first-year undergraduates to participate in the experiment, albeit unwittingly. They were allowed to run the first PIP program. It was difficult to monitor exactly how often students did run

---

[1] See section 1 of chapter IV.

[2] Cf. chapter IV, section 2.1.

PIP, since they could abandon a run before any information was output to the author. However, before the students became bored, approximately forty runs were made, about half of which were useful for this second program. At the end of a run, PIP would check to see if one of the speakers were called "John". If not, then nothing further would be done. If "John" had been used, however, then PIP would read in a file which contained a list in the form of a cognitive structure.[3] This list and "John"'s cognitive structure would be compared in such a way that any argument in "John"'s cognitive structure that was not contained in the list would be added to that list; where "John" had used arguments that were already on the list, then PIP would check that there was no contradiction in the attitude markers. If there were a contradiction (inconsistency), then PIP would ignore "John"'s attitude marker (in the same way that inconsistencies were simply commented upon by PIP without altering any cognitive structure),[4] otherwise the attitude markers would be merged. Finally, the updated list would be output to a file, so that it could be used as data for the program under consideration in this chapter. PIP would go through the same process irrespective of the identity of the user. In this way, the author retained some control over PIP's cognitive structure for the second program, but not all.

The fact that about half the runs were useful for the second program of PIP can be partly attributed to good

---

[3] Cf. chapter IV, section 4.1.

[4] Cf. chapter IV, section 4.3.

psychology. A User's Guide to PIP was written for the benefit of the students. All the examples in the Guide have "John" as the name of the speaker. Since the whole system was unfamiliar to these first-year undergraduates, they kept their early runs to within the examples of the Guide. Hence their usefulness for PIP's cognitive structure. Nevertheless, in order to obtain a reasonable amount of data, the author was compelled to run the first program himself.

2.1 From the above discussion, it will be seen that PIP is in a position analogous to that of a child. A child hears much language activity around it, but is disposed to trust only its parents. Thus, at an early age at least, parents have some control over the way their child thinks. (It is only later that children start to question their parents' attitudes!) In this case, "John" is in the position of PIP's 'father'. PIP trusts "John" and takes "John"'s cognitive structure for its own, while it ignores everybody else. This is similar to part of the ABS_1 system of Colby and Smith [14] outlined in section 5.2 of chapter III. They also give their system a 'highly credible' set of beliefs which are used for comparison with those of the user of their system in order to give that user a credibility rating. PIP's beliefs allow it to agree or disagree with the user, as necessary.

A security problem for the future is also raised by this parent-child relationship between "John" and PIP. If this system ever becomes the basis of a large Language understanding system or of a robot's linguistic abilities, then strict control will have to be maintained over just what PIP's attitudes become. It would not be good policy to allow a robot to wander around believing that either robbery or murder were ethically good! Therefore, some kind of really safe file store security would be necessary, just in case subversive elements in society felt inclined to tamper with PIP's ethics, or those of PIP's successors. This problem has already been raised by people outside the field of computing to whom output from this second program of PIP has been shown.

2.2 The following is a table of PIP's cognitive structure as it is at the present time. About half of the arguments and attitudes were 'fathered' by the author, the other half by the students.

acquit Charles	FUTACT
Agnes is a Chinese sympathizer	VALIDBLF BAD NFUTACT WANTS
Agnes is a Russian sympathizer	NVALIDBF BAD NFUTACT NWANTS
Agnes is a spy	VALIDBLF BAD NFUTACT NWANTS
Carole is beautiful	NVALIDBF NCONF
Charles is guilty	NVALIDBF
Charles is innocent	VALIDBLF
Charles killed the postman	NVALIDBF
Charles, killing the postman	NRSPNSBL
Charles robbed the bank	NVALIDBF
empiricism is correct	NVALIDBF
Fred, honouring the queen	RSPONSBL
Fred is innocent	NVALIDBF

Fred is guilty	VALIDBLF BAD
Fred killed the postman	VALIDBLF BAD
Fred, killing the postman	RSPONSBL
Fred robbed the bank	NVALIDBF
Fred stealed the money	VALIDBLF
Fred, stealing the money	RSPONSBL
Fred stole the money	VALIDBLF
go by bus	FUTACT
go to the zoo tomorrow	FUTACT WANTS
Harold, robbing the bank	RSPONSBL
honour the queen	FUTACT WANTS
honouring the queen	GOOD
John is guilty	VALIDBLF NCONF
John is innocent	NVALIDBF NCONF
killing the postman	BAD
man is born with an empty brain	NVALIDBF
man learns everything from his environment	NVALIDBF
Mildred robbed the bank	NVALIDBF
Mildred, robbing the bank	NRSPNSBL
preaching high treason	BAD NFUTACT NWANTS
rationalism is correct	VALIDBLF
robbing the bank	BAD
stealing the money	BAD
support for the Liberal party	FUTACT WANTS
support for the Tories	NFUTACT NWANTS
support the Liberal party	FUTACT WANTS
support the Tories	NFUTACT NWANTS

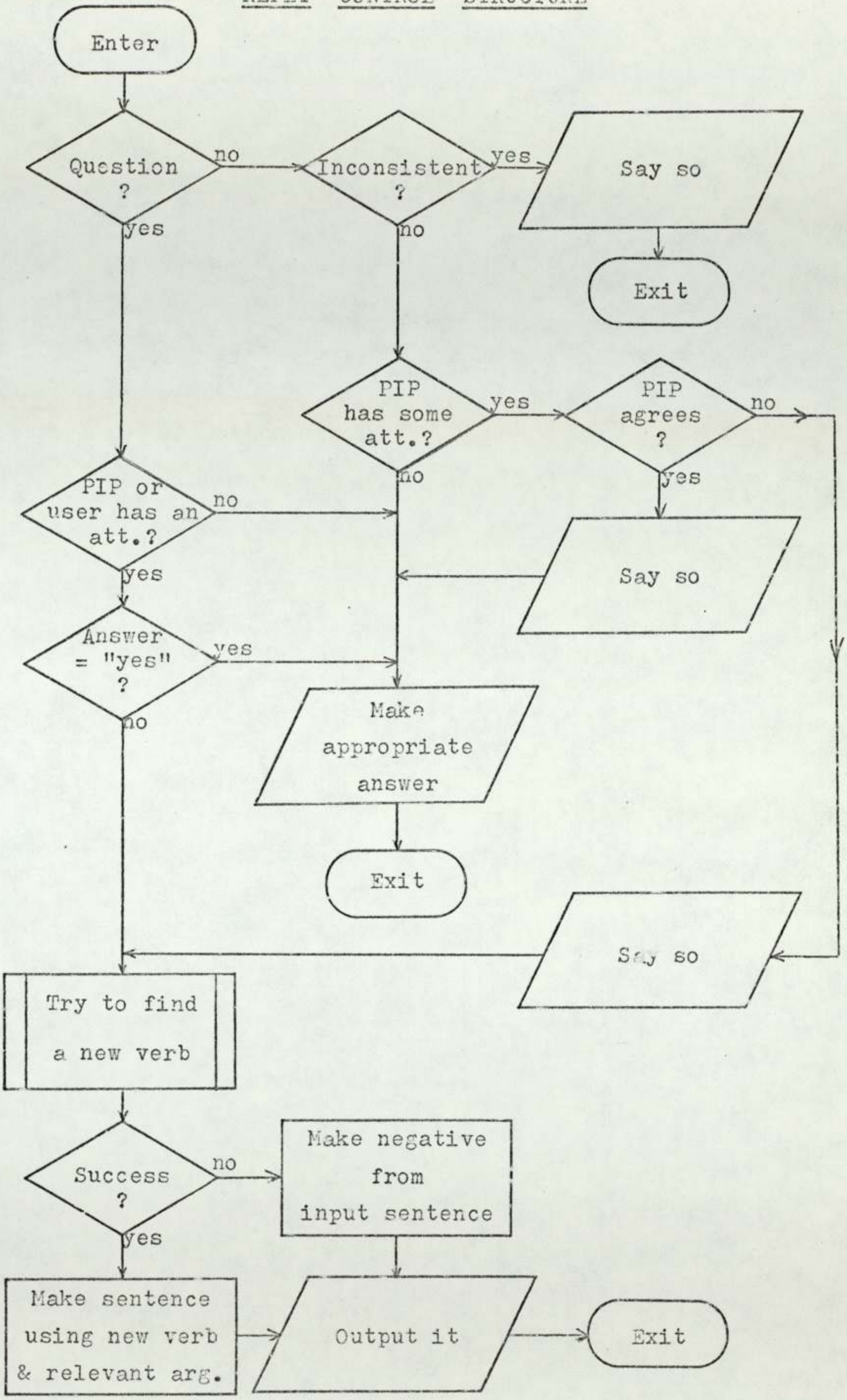
take George to the zoo tomorrow	FUTACT
take Mary to the pub tonight	FUTACT
the bank was robbed	VALIDBLF
the money was stealed	VALIDBLF
the queen was honoured	VALIDBLF

3. Responses

Needless to say, this program relies heavily on the programming for the first experiment, as outlined in sections 3, 4 and 5 of chapter IV. All these facilities are built into the program to be outlined below. The only difference is in the number of 'people' allowed. Where nine speakers were previously accepted, now only two require cognitive structures: PIP and the user. Apart from this, and unless otherwise specifically stated, all the other facilities are still available; the same presuppositions are used, the same verbs are known by PIP, the same use of the verb "implies" has been incorporated. What has been added is an ability to reply. The programming of this ability will be described below.

3.1 There is no one overall function which controls the reply structure, so that this discussion will in fact cover a number of different functions, each with its own task. A flowchart for the overall control appears on page 130. First of all, PIP decides whether the input sentence is a statement or a question. It recognizes these by the terminating character.

REPLY CONTROL STRUCTURE



If that character is a full-stop, then the input sentence is a statement; if it is an asterisk, then it is a question. READSENT, the function which reads the input stimuli, uses the POP-2 standard function ITEMREAD.[5] Unfortunately, a question-mark is not a POP-2 item.[6] Therefore, an asterisk is used instead of a question-mark. If the sentence is a statement, then PIP checks for inconsistencies in the user's cognitive structure in exactly the same way as in section 4 of chapter IV. If it finds any, then it reports those inconsistencies that it has found and returns to read the next sentence.

3.1.1 Assuming that no inconsistencies have been found, ISHAPPY[7] is called again, this time with two purposes. First, does PIP have some attitude to the arguments expressed by the user? Secondly, if the answer to the previous question is in the affirmative, does PIP agree or disagree with the attitudes expressed by the user? If the answer to the first question is in the negative, then PIP answers in some appropriate fashion. Consider

(86) I congratulate Marion on passing her exams.

as an input sentence. If PIP has no attitude to [passing her exams] or [Marion, passing her exams] or any other possible (combination of) argument(s) from this sentence, then it will

---

[5] POP-2, pp. 47 - 49, and p. 112.

[6] POP-2, pp. 81 - 83.

[7] See sections 4.2 and 4.3 of chapter IV.

examine the presuppositions inherent in "congratulate". It will find GOOD, and so will reply

(87) Good for Marion.

If BAD had been a presupposition of the verb used, then PIP would have replied

(88) That is wrong of Marion.

These are stock sentential 'frames', of the kind used by Weizenbaum's ELIZA.[8] The variation in them is in what follows the preposition. Thus, the 'frames' are "Good for 'x'" and "That is wrong of 'x'", where 'x' is in each case replaced by the subject of the presupposition RSPONSBL. PIP then returns to read the next sentence.

If PIP does have some attitude to the argument(s) expressed in the input sentence, and those attitudes are the same as those expressed by the user, then PIP says that it agrees and then replies further in the same way as described above. If PIP does not agree, then again it says so. This time, though, a sentence expressing PIP's attitude(s) is generated by a method to be outlined in section 3.3.

3.2 Consider now the case where the input is in the form of a question. PIP first determines whether the question is about itself or the user. In either case, it then checks whether the appropriate 'person' does have any attitudes at all to the argument(s) of the verb. If there is none, then it replies with either (89) or (90).

---

[8] See chapter II, section 3.4.

(89) I do not know anything about that.

(90) I do not know what you think of that.

It then returns to read the next sentence.

If the answer to the question is "yes", then PIP says so and then repeats the question in the form of an affirmative statement. Thus, for example, if the input is

(91) Would you accuse Fred of killing the postman*

then PIP's answer might be

(92) Yes. I would accuse Fred of killing the postman.

If the answer is "no", then PIP again says so, but now it tries to generate a new sentence using a verb which expresses the attitude(s) of the 'person' about whom the question was posed. This is explained in the next section.

3.3 This section will describe how the function FINDNEWV attempts to find a verb that PIP knows and which will express the attitude(s) either of PIP (if control has come from section 3.1.1) or of the subject of the question (if control has come from section 3.2). FINDNEWV utilizes the global variable OFFENDER which contains a list of the opposites of the presuppositions in the original verb that have caused inconsistencies in the original analysis. Thus, if PIP's cognitive structure includes

(93) GOOD [passing her exams]

and the user inputs

(94) Would you blame Marion for passing her exams*

then PIP will realize that BAD [passing her exams], a presupposition of "blame" in (94), does not agree with (93).

PIP knows that GOOD is the opposite of BAD because this information is part of the FNPROPS[9] associated with the function BAD. Thus, GOOD will become an element of OFFENDER.

FINDNEWV takes the elements of OFFENDER one by one, and, using a sequential search through the verbs that PIP 'knows', attempts to find a verb whose explicit level presupposition is the same as the element of OFFENDER under consideration. When such a verb has been found, then a comparison is made between the presuppositions inherent in that verb and the cognitive structure of the relevant 'person'. This is achieved using ISHAPPY; on this occasion, however, only positive or negative attitudes count, not 'don't-knows'. Thus, for example, the verb "assert" has VALIDBLF for its 'meaning' (explicit level) and CONFDENT as a presupposition. If a question is asked of PIP for which it has the attitude of VALIDBLF, but has no attitude of CONFDENT nor of NCONF, then "assert" will not be used in any sentence that PIP makes up. This is obvious, if one remembers that PIP would attribute the attitude of CONFDENT to anybody whom it 'heard' using the verb "assert". Therefore, if it does not have the attitude, it cannot use the verb itself.

Since FINDNEWV uses a sequential search, and since there are some verbs which have the same presuppositions as others, it would normally happen that only one of those verbs would be used. Thus, if "state" comes before "assert", "swear" and "know" in PIP's table of verbs, then "state" would always be chosen by FINDNEWV. Therefore, a tag is attached to each verb,

---

9 POP-2, pp. 58 and 111; see also chapter IV, section 4.3.

showing how many times each verb has been used. The verb with the lowest tag that fits the attitudes of PIP (or the 'person' about whom the question was posed) is then found by FINDNEWV. This ensures that there is at least some variety in PIP's answers.

3.3.1 There are two exceptions to the above procedure. The functions RSPONSBL and NRSPNSBL are both functions with two arguments. Suppose the input sentence is

(95) Would you accuse Charles of killing the postman*

and PIP's cognitive structure contains

(96)a. NRSPNSBL [Charles, killing the postman]

b. RSPONSBL [Fred, killing the postman].

It is not enough for PIP to answer merely "No. I would not accuse Charles of killing the postman". (cf. (98) below) It would be better if PIP could discover that it believes Fred to be responsible for the postman's having been killed, and reply accordingly. Therefore, if the first element (say) of OFFENDER is NRSPNSBL, then it is changed to RSPONSBL and an attempt is made to find some cognitive structure RSPONSBL ['x', killing the postman]. If such a structure is found, then 'x' is filled in, other arguments are altered accordingly and a comparison is made as above. The reverse procedure is followed if RSPONSBL is an element of OFFENDER. In the case of (95) and (96), PIP would find (96)b. and, provided the other presuppositions of "blame" were also met, might reply

(97) No. I would blame Fred for killing the postman.

Notice that (95) and (97) correspond to (84) and (85).

3.4 If FINDNEWV produces a new verb by the above procedure, then PIP answers in terms of the new verb with the appropriate argument (e.g. (97)). If FINDNEWV has failed to produce a new verb, then PIP constructs a negative from the original input sentence. For example, suppose that PIP had (96)a. but not (96)b. as part of its cognitive structure. A reply to (95) would then be

(98) No. I would not accuse Charles of killing the postman.

In either case, PIP returns to accept the next sentence.

It is those answers generated by the method of section 3.3 that are of most interest to both philosophers and workers in the field of AI. Sentences are generated because of attitudes held by either PIP or the user. The power that this gives to PIP will be seen in the following example.

4. An Example

The following example was designed to illustrate all the procedures outlined in section 3. The same conventions will be adhered to as in chapter IV, that is that input sentences will be typed normally and PIP's replies will be in block capitals; numbers in parentheses again denote comments which will follow the example.

WHO ARE YOU?

John

I AM READY NOW

I believe that Fred is innocent.

I DO NOT AGREE.
FOR MY PART, I QUESTION WHETHER FRED IS INNOCENT

I accuse Roy of spilling the coffee.

THAT IS WRONG OF ROY

I would therefore not accuse George of spilling the coffee.

A PRESUPPOSITION OF ACCUSE IS:-
VALID (NOT [THE COFFEE WAS SPILLED]) (1)
THIS IS INCONSISTENT WITH A PREVIOUS FACT

A PRESUPPOSITION OF ACCUSE IS:-
BAD (NOT [SPILLING THE COFFEE]) (1)
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF YOURS, JOHN

A PRESUPPOSITION OF ACCUSE IS:-
VALIDBLF (NOT [THE COFFEE WAS SPILLED]) (1)
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF YOURS, JOHN

Would I blame Fred for spilling the coffee*

NO. YOU WOULD ACCUSE ROY OF SPILLING THE COFFEE

Do I believe that George spilled the coffee*

NO. YOU DISBELIEVE THAT GEORGE SPILLED THE COFFEE (2)

Do you believe that Roy spilled the coffee*

I DO NOT KNOW ANYTHING ABOUT THAT

Would I pardon Roy for spilling the coffee*

NO. YOU WOULD BELIEVE THAT ROY SPILLED THE COFFEE

Would you accuse Charles of killing the postman*

NO. I WOULD NOT ACCUSE CHARLES OF KILLING THE POSTMAN

You believe that Fred killed the postman* (3)

YES. I BELIEVE THAT FRED KILLED THE POSTMAN

You will pardon Fred for killing the postman.

I CAN MAKE UP MY OWN MIND THANK YOU (4)

I know that John is guilty.

I AGREE. THAT IS INTERESTING

I admit that George spilled the coffee.

THE MEANING OF ADMIT IS:-
RSPONSBL (GEORGE , SPILLING THE COFFEE)
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF YOURS, JOHN

A PRESUPPOSITION OF ADMIT IS:-
VALIDBLF [GEORGE SPILLED THE COFFEE]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF YOURS, JOHN

Do you deny that Agnes is a spy*

NO. I SWEAR THAT AGNES IS A SPY (5)

Do you deny that Agnes is a spy*

NO. I STATE THAT AGNES IS A SPY (5)

Do you deny that Agnes is a spy*

NO. I ASSERT THAT AGNES IS A SPY (5)

Would I deny that Agnes is a spy*

I DO NOT KNOW WHAT YOU THINK OF THAT

I do deny that Agnes is a spy.

I DO NOT AGREE.
FOR MY PART, I KNOW THAT AGNES IS A SPY (5)

I believe that Agnes is a Russian sympathizer.

I DO NOT AGREE.
FOR MY PART, I DENY THAT AGNES IS A RUSSIAN SYMPATHIZER (6)

I praise Harold for arresting Agnes.

GOOD FOR HAROLD

Agnes was arrested implies Agnes is a spy.

O.K.

End.

THE ACTUAL NUMBER OF COMPUTER WORDS USED ON THIS RUN WAS:-
12859 (7)

(1) The reader is referred to comment (9) of the example in section 6 of chapter IV for a discussion of why these inconsistencies should not have been found, and of why they have in fact been found.

(2) The verb "disbelieve" is really an archaism, and this reply sounds odd to twentieth-century ears. This verb was given to PIP as an example of a verb which has NVALIDEF as its explicit level presupposition and no implicit level ones. It is difficult to find another such verb in the English language!

(3) A question is recognized by the asterisk at the end of the sentence, not by word-order within the sentence. Thus, the question that relies on voice pitch (in spoken language) can be handled by PIP.

(4) PIP will not be bullied into taking attitudes that the user wishes it to have. In fact, it is unbribable!

(5) These replies indicate that the tags attached to each verb are being correctly updated and that PIP will not use the same verb over and over again. Of course, PIP treats each of these input sentences as unique; it does not therefore get angry about being asked the same question many times.

(6) Agnes is presumably Chinese! When students run programs, anything can happen.

(7) Even though this program is running in 24K words of store, there are still not very many free words left. The total mill time for this twenty-sentence run was 75 seconds, of which approximately 10 seconds are required by the operating system. Thus, sentences are being processed in an average time of approximately three seconds each. Considering the number of

comparisons involved in generating some sentences and the slowness of the machine, that is no bad result.

An uncommented example of a run of this second version of PIP will be given in Appendix B.

VI. CONCLUSIONS

It is the purpose of this chapter to bring together the conclusions that can be drawn from the experiments outlined in chapters IV and V and the consequences that these have for various fields of study.

1. Analysis of PIP's Performance

The notion of what it would be for a computer to understand Natural Language is crucial to the interpretation of the results from the two programs detailed above. There is a trivial sense in which it could be said that computers never understand anything; it would be said that they merely work with ones and zeroes, that they can perform only relatively simple operations with those ones and zeroes, that they therefore cannot have 'thoughts'. In other words, it would be said that a man-made machine cannot have 'thoughts' and therefore cannot comprehend thoughts expressed as sentences in Natural Language. Yet human beings do not know what happens to linguistic input in their own brains. During the discussion of 'situational' thinking in section 1.1 of chapter III, the belief of Legéndy [48] that information presented to the brain caused neurons to form groups was mentioned. That is of relevance here too. Man does not know why particular neurons group together, nor what a particular cluster of neurons might represent. These neuron clusters, which form the Brain Code of chapter III, are just as unnecessary to a discussion of understanding at the human level as the clusters of bits which form the Machine Code of computers is irrelevant to a discussion of understanding at the machine level.

One dictionary, Macdonald [54], defines 'understanding' as "the act of comprehending" where 'comprehend' is defined as "to seize or take up with the mind, to understand", which is certainly circular. A dictionary is therefore of no help. Indeed, any non-circular definition of 'understanding' will have been arbitrarily chosen, in the same way that any technical term is defined arbitrarily by an author. Nevertheless, an attempt has been made to keep the definition similar to that used in ordinary language. Thus, the criterion of understanding is an ability to respond intelligently and intelligibly to input stimuli (sentences). Admittedly, this is a behaviouristic criterion. Admittedly, Behaviourism as a theory of human comprehension of Natural Language is incapable of being seriously maintained.[1] Nevertheless, until or unless someone can invent a psychological test for computers, it remains the only way to judge the performance of a computer.

---

[1] See Chomsky's devastating attack on Skinner in Chomsky [11]; see also Harrison on Osgood and Mowrer in Harrison [31], pp. 300 - 309.

Successful performance (based on an adequate theory) has the right to be called 'understanding'.

1.1 In the first program, therefore, PIP reacts intelligently by detecting anomalies of attitude on the part of individuals. This was given as the criterion of understanding for that program.[2] How successful is PIP in detecting all the anomalies, and how successful in detecting only the anomalies? Note (9) to the example given in section 6 of chapter IV demonstrates that too many anomalies are sometimes detected by PIP. As explained there, the whole concept of negation is extremely complex and it was not originally expected that PIP should have to cope with it. There would seem to be no theoretical reason why PIP should not eventually be capable of coping with negation. It is a moot point whether negation is a syntactic or a semantic concept. Katz says "we shall make the reasonable assumption that the scope of a negative in a sentence is determined by the grammatical analysis of the sentence".[3] Thus, although negation has important effects on the semantics of a sentence, the scope of that negative is determined by syntactic analysis. First, PIP performs as little syntactic analysis as possible. Secondly, only verbs are at present semantically analyzed. It was therefore not unpredictable that PIP would not be able to cope with the whole concept of negation (in particular the scope of the negative).

---

[2] See chapter IV, section 2.

[3] Katz [39], p. 534.

That it has not coped is consequently not a possible falsification of PIP's methods.

Apart from negation, the problem is to prove that PIP detects all and only the anomalies of attitude of different individuals. No demonstration run listed in this thesis can do that, since it could be claimed that only those runs had been chosen that showed understanding on the part of PIP, but not those that did not. No protestations on the part of the author could ever gainsay such a claim. If the reader does not believe that the examples given[4] are representative of PIP's abilities, then he is asked to use the program himself and prove it to his own satisfaction. As far as the examples given are concerned, PIP does show understanding of the input sentences. It constructs models of the various individuals' attitudes (cognitive structures) and correctly identifies all and only those anomalies that appear.[5]

1.2 The criterion of comprehension on which the second PIP program is judged is that of correctly replying to input stimuli, which can be either statements or questions, generating sentences using its own verbs to 'describe' its own cognitive structure wherever possible.[6] The correctness of the reply is not to be judged by the 'correctness' of the English,

---

[4] See chapter IV, section 6, and Appendix A.

[5] Again, with the exception of some sentences where negatives are employed.

[6] See chapter V, section 1.1.

which, because of the lack of syntactic analysis, might sometimes appear stilted, but rather by the appropriateness of the answers to both the input stimulus and the attitudes of PIP and the user. In this case, one is in the position of being able to examine PIP's cognitive structure completely, which is of course not possible with human beings. Comparing the table of PIP's attitudes given in section 2.2 of chapter V with PIP's replies both in section 4 of chapter V and in Appendix B, it will be seen that PIP's replies are indeed appropriate. The user's attitudes, expressed in his conversation with PIP in those examples, are also clearly adhered to in PIP's replies. PIP can be seen to choose those verbs that correctly express either its or the user's attitudes to the input stimulus. PIP has therefore become a program that can genuinely generate meaningful sentences (as opposed to those programs that 'generate' sentences along TG principles, sentences whose generation owes everything to correct syntax and nothing to a purposeful semantics).[7]

In both programs, therefore, PIP has shown the understanding asked of it, and the experiments have been successful.

2. Consequences for Artificial Intelligence

Perhaps the most important consequences of PIP's success are for the field of AI. Until now, the only programs that have

---

[7] Cf. Friedman et al. [25].

been written for Natural Language processing have attempted to cope with all the intricacies of Natural Language and have sacrificed the ability to let that Language deal with any context. PIP offers an excellent method of representing references to any context, at the same time as showing that this reference can be considered as independent of the meanings of nouns or noun phrases. It was shown in chapter II that most AI work has concentrated on the semantics of nouns (when it has touched upon semantics at all). Winograd, in particular, has concentrated on (fairly) complicated semantic definitions of various types of blocks, but his verbs are all concerned with the movements of his 'robot's' arm.[8] The evidence from PIP is that the verb is very important for the representation of 'knowledge'. They should not be restricted to 'movement' verbs. It will be just as important for AI as for the Philosophy of Language[9] that a method of reduction be found from 'ordinary' sentences to explicitly performative sentences. When that has happened, whether the method be found by philosophers or workers in AI, then AI will very definitely have a way of representing knowledge about anything.

2.1 Furthermore, previous AI programs have, in the main, been concerned with the physical world, the 'actual' world. PIP has shown how computers might manipulate abstract concepts.

---

[8] Nevertheless, PIP does support the notion of 'procedural semantics' as exhibited by Winograd [89] and others at M.I.T.

[9] See below, section 3.2.

Moreover, since PIP is capable of making ethical judgements, then it could be claimed that PIP represents the beginnings of a really intelligent machine that can cope with Natural Language. Previously, there have been machines that have exhibited some of the skills of human beings, such as grasping objects and manoeuvring them (Winograd [89]), acting like psychiatrists (Weizenbaum [87]), or inferring facts from fairy stories (Charniak [9]). Now, however, there is a program, which, when expanded, will be capable of making value judgements, of holding a reasonable ethical position, and of representing 'facts' in a way that is consistent with its representation of all other attitudes. For PIP, facts about the world are as intangible as they are for humans (compare Descartes' struggle to find something really tangible). PIP understands Language not as a means whereby it obeys orders to manipulate an arm, not as a series of (to it) meaningless statements designed to prolong a (to it) meaningless conversation, not as a test of disambiguation, but rather as a means of communicating one person's attitudes about anything to somebody else. This is, of course, precisely the definition of Language given in chapter III, section 1, and on the very first page of this thesis. Thus, there is now a program which will eventually (one hopes) be capable of holding a perfectly normal conversation; in fact, the beginning of what is essential if there is ever to be a general purpose robot.[10] Such robots will have to be able to discuss abstract concepts

---

[10] Cf. chapter II, section 1.1.

such as beauty, or cleanliness in order to be fully able to take part in normal conversations of the kind that are heard in every household every day.

2.2 A further consequence of PIP's success, perhaps not totally in the field of AI, but certainly within the field of Natural Language processing, is the possibility of textual study by machine. Textual study is currently limited to word counts, concordances, and (possibly) a determination of authorship on the basis of such statistics. PIP now offers the possibility of checking the internal consistency of a piece of writing; that is to say that it could do part of the job of proof checking. PIP has been 'trained' to look for inconsistencies in people's attitudes. A book or an article is simply an extended expression of one person's attitudes. Moreover, PIP should not be confused by what is said by the author and what the author quotes. Even the writing of a thesis has shown this author that it is difficult to be consistent over six chapters of writing!

When the logic of the whole system has been worked out, then there may also be the possibility of checking books or articles for the correct or incorrect conclusions having been drawn from premises, or of checking whether an argument is circular. There are some books that one can think of which would benefit from this kind of treatment! Naturally, everything that has been said in this section applies equally well to conversations, but its use is probably more practical in the literary field. One does not want to imagine computers getting absolutely everywhere, simply to check on one's verbal consistency. Moreover, some concept of time would have to be incorporated to allow for people to change their minds; which is not the case when checking for inconsistencies.

2.3 What of the future? It was argued in section 2.2 of chapter IV that any positive result from PIP would be a step forward for AI, since a model of human intelligent processes does not have to be <u>the</u> process by which human beings actually use that intelligence. PIP has shown a positive result. One therefore feels that a full-scale program, based on PIP, but expanded to take account of as many of the recommended improvements outlined in this chapter as possible, would be a productive experiment. (Such a program should also model as much of the Functional Theory of Language as possible.) In terms of AI, two improvements are immediately obvious.

2.3.1 First, although PIP does at the moment learn the 'meanings' of verbs from its users, it merely accepts them, without trying to work them out for itself.[11] Since human beings have no way of asking for the 'meanings' of verbs, not while they are still very young, at least, it seems clear that some method of enabling PIP to do its own learning and discovering must be found. At the same time, PIP currently remembers the 'meanings' of verbs from session to session, but not the cognitive structures of the various speakers. This too

---

[11] See chapter IV, section 3.2.

could be altered, allowing PIP to develop a broader and broader knowledge of many 'people'. This would already have been achieved, had more computer memory been available, but this particular facility is extremely demanding on memory space.

2.3.2 The second improvement has already been mentioned in section 4.3.1 of chapter IV. One important aspect of work in AI is a search for an adequate model of human representation of knowledge. PIP does offer a method of representation, but it is extremely inefficient in its storage of cognitive structures. If the suggestion made in chapter IV is adopted, that is that the arguments should be held on one list, with pointers to that list as part of PIP's model of individuals' cognitive structures, then a comparatively efficient, as well as theoretically good model of knowledge representation will have been constructed.

3. Consequences for Linguistics

In section 2.2 of chapter IV it was stated that no success would count as proof that the Functional Theory of Language was the correct theory to describe a human being's capabilities with regard to Language comprehension and production. That does not mean, however, that no conclusions about Language can be drawn from the success of PIP.

3.1 In the first program, PIP is able to construct a model of (part of) each speaker's cognitive structure merely on the basis of hearing that speaker utter sentences, which need not be descriptive of that cognitive structure (i.e. the sentences need not begin "I believe that ... " or "I know that ... "). Furthermore, these cognitive structures are organized in a particular way. Each cognitive structure consists of an attitude marker and an argument. Since PIP can understand what each individual is saying, on the basis of that structure, there is evidence that this might indeed be the way (or, at least, one way) that human beings understand such utterances. A theory of understanding based on such structures might be a fruitful way to proceed. The second program is also based on cognitive structures. In this case, PIP generates sentences representing its own cognitive structure by an appropriate verb. Again, the inference is that a theory of language generation based upon cognitive structures might well be a useful approach to the study of human linguistic abilities.

Both the two programs and the above comments depend on the notion of presuppositions inherent in the verb of the utterance acting as functions on the attitude marker of the cognitive structure. One would like to be able to draw a similar conclusion with regard to this aspect of FTL as to cognitive structures. This will be possible only when it has become clear that the verbs used by PIP are reasonably easy to define in terms of the given presuppositions. It was stated earlier[12] that PIP did not discover the 'meanings' of new verbs for itself, but rather that the user would define new

---

[12] Chapter IV, section 3.2, and section 2.3.1 of this chapter.

verbs for it. Thus, the problem was not one for PIP, but for the human user. In practice, the difficulty hinged not on what presuppositions should be used to define any given verb, but on which presupposition should represent the explicit level of communication. The verb "praise" is a good example of this purely human dilemma. It is quite clear that both RSPONSBL and GOOD are presuppositions inherent in "praise". However, does

(99) "I praise the bank manager for calling the police so quickly"

mean (explicitly) GOOD [calling the police so quickly] and merely imply that it was the bank manager who did the calling, or *vice versa*? The reader is probably quite capable of imagining situations in which either possibility would be the case. However, as soon as one became accustomed to defining verbs in terms of presuppositions, it became relatively easy to decide which presuppositions were inherent in which verbs. It is therefore possible to draw the conclusion that it might well be the case that a Theory of Language based on functions of some kind is worth pursuing.

3.1.1 As a corollary of the above line of argument, it is also possible to draw the conclusion from PIP's success that presuppositions are indeed adequate for characterizing the 'meanings' of at least that class of verbs that PIP has studied.[13] The inconsistencies found in the first program should have been detectable by any native speaker of English,

---

[13] But cf. section 3.3 below.

even if the explanation of such inconsistencies might not have been expressed in terms of presuppositions. Similarly, the replies generated by the second program are convincing pieces of English, taken on their own. Verbs are used correctly. Since they are used correctly only by virtue of the presuppositions deemed to be inherent in them, one can say that the presuppositions do adequately represent the 'meanings' of explicitly performative verbs.

3.2 There is therefore no evidence to disprove the Functional Theory of Language; indeed, there is much evidence to suggest that it is worth pursuing. There are a number of directions in which further investigation might be profitable. First, consider the fact that although explicitly performative verbs do exist and sentences are uttered in which they appear, the vast majority of sentences do not contain such verbs. Austin [2] believed that all sentences could be 'reduced' to sentences containing explicitly performative verbs. He did not, however, intimate how this 'reduction' was to be achieved. It is most important that this means of reducing 'ordinary' sentences to those containing explicitly performative verbs be found. Since PIP has shown how the latter can be understood, any sentence that is translatable into an explicitly performative sentence is inherently capable of being understood. It may be, taking the simplistic approach, that it is in fact the verbs in other sentences that are themselves capable of translation into explicitly performative verbs. However, this is not the place to discuss such hypotheses.

A second area of research would be to apply the notion of presuppositions in reverse, so to speak. Instead of characterizing existing verbs, it might be interesting and productive to 'invent' combinations of presuppositions and see if verbs in any language could be found to fit the particular combination. It is well known both that individual languages have concepts that cannot be expressed in other languages and that Language itself is a redundant system. This investigation might show why languages have concepts impossible of direct translation (i.e. combinations of presuppositions that have a representation in one language but not another) and might also demonstrate yet another area of redundancy (i.e. combinations of presuppositions that have no known representation).

Thirdly, it has become quite clear that some kind of syntactic parser is necessary to any model of Language. What is not clear is what kind of parser is required either to model human usage of Language or to enable programs to analyze linguistic input. Many kinds of parser were discussed in chapter II, especially in section 2. At the end of that chapter, the conclusion was advanced that systemic grammars were probably superior to TG grammars as far as programming was concerned. It might be that they also prove to be superior as far as modelling human linguistic abilities is concerned.

More important than the determination of what kind of syntax is to be preferred is the fact that the syntax and semantics will have to interact in some way. How they interact will, of course, depend to some extent on which particular kind of syntactic analyzer is used. That they must interact has become clear over the last few years both through the criticisms of TG Theory by the Generative Semanticists and through the program of Winograd [89]. PIP's syntactic analysis is almost non-existent. It merely provides the arguments on which the semantic component can work. Nevertheless, it is quite clear that for more complicated sentences than PIP handles at the moment the syntax will have to be far more rigorously defined in order to continue providing the semantics with the correct information. Moreover, at the moment PIP puts no semantic interpretation on anything other than verbs (one of its strengths). Some grammatical ambiguities can only be solved by semantics, especially ambiguity of word class. Thus, the precise nature of the interaction between syntax and semantics within FTL will have to be formulated.

It is to be hoped that, whatever kind of syntax has been finally chosen, it will be able to determine the scope of negatives.[14] In a recent paper, Davies [17], negation in English is still considered to be consistent with negation in predicate logic. It should be obvious to anybody who has read this thesis and seen the difficulty PIP has when negation is so represented, that it is about time some determined effort was made to formalize the various aspects of negation in Natural Language and, if possible, to program them in such a way that they can be incorporated into any program concerning any part of Natural Language. It is important for FTL (as for any other Theory of Language) to explain the concept of negation, or to

---

[14] Cf. section 1.1 of this chapter.

at least give some description of its structure within the framework of the theory.

3.3 The above section would call for new programming that is not present in PIP. However, there are two 'facilities' that are present in PIP that could be expanded to give FTL more generality. First, PIP's world is one of black and white. It can apply GOOD and BAD, or TRUE and FALSE (VALID and NVALID in the program) to an argument; it can, alternatively, retain an open mind by giving a "don't-know": but those are the only three values that it can apply. In other words, it uses a three-value logic. On the other hand, it is by no means clear that human beings are restricted to three-value logic for all the types of presupposition (if any) used by PIP; indeed, there is good evidence that many more values are employed. Consider GOOD and BAD again. Many school reports, for instance, have categories for describing a pupil's work as excellent, very good, good, fair, average, poor, bad, or very bad. In other words, there is a spectrum of values ranging from excellent at one end to very bad at the other. Values anywhere on the spectrum can be picked out and labelled. This argues that there must be some multi-valued logic behind at least those presuppositions other than TRUE and FALSE, although even there one wonders whether three values are enough. An example of how this can affect PIP is given by those replies commented by (5) in the example in section 4 of chapter V. PIP is quite happy to use the verbs "swear", "state", "assert" and "know" to represent the cognitive structure given in (100).

(100) VALIDBLF [Agnes is a spy]

Nevertheless, one is not entirely convinced that those verbs all express (100) with the same intensity. The ABS_1 system of Colby and Smith [14][15] catered for different strengths of belief. Something similar is necessary for PIP.

The second way in which PIP could be expanded would be by allowing functions to apply to other functions. Thus, it might be necessary to define a verb in terms of "the speaker wants that the hearer wants that the speaker intend to do 'x' in the future", that is WANTS (WANTH (FUTACT ['x'])). This is more likely to be necessary when more is known about the translation process from 'ordinary' sentences to explicitly performative sentences. It is not possible within the current system, nor indeed was it found necessary. There is, however, no theoretical reason why it cannot be incorporated into either PIP or the Functional Theory of Language.

3.4 The world of neuropsychology is by no means concerned solely with the language processing parts of the brain. However, bearing in mind the comparative ignorance of the field, as testified by the extract from Luria [51], quoted in section 4.3 of chapter III, and bearing in mind the success of PIP, it might not be too presumptuous to suggest one or two areas where neuropsychologists might look for evidence of workings within the language processing parts of the brain.

First, they might try to find evidence for function-like

---

[15] See chapter III, section 5.2.

processes. The evidence from PIP would suggest that it is at least a reasonable assumption that people do use such processes when understanding or generating Language. Should such evidence be found, then neuropsychologists might like to speculate as to how such processes come into being, or as to how much of the process is present at birth.

Secondly, PIP relies on attitude markers to determine what people's attitudes to propositions are. The problem is that one does not know what an attitude marker might look like in the human brain. Nevertheless, it is indicated that one could look for such markers.

Further, one has no real idea of how the neurons in the brain would actually combine to form cognitive structures. How, for instance, are they linked? It is possible that attitudes are grouped by sense, or that they are 'stored' haphazardly, in any pigeon-hole that happens to be free, or that they are 'stored' sequentially. Is it possible for cognitive structures to overlap? In other words, it might be possible that individual neurons or groups of neurons 'take part in' more than one cognitive structure. Evidence of such a nature from neuropsychologists and neurophysiologists would be extremely helpful to FTL, and even more helpful for a better model of human cognitive structures than PIP currently incorporates.

3.4.1 As opposed to the ways in which PIP might influence neuropsychologists, there are two ways in which neuropsychologists could be of benefit to FTL. It has been repeatedly stated that no evidence will ever 'prove' a theory correct; evidence can only prove a theory to be incorrect, in that it is inconsistent with known facts. Thus, paradoxically, the first benefit neuropsychologists might provide is evidence that FTL is wrong! However, the evidence from PIP is that this is not likely to be forthcoming. Secondly, consider again PIP's black and white world. Although it was shown[16] that human beings work with more than a three-valued logic, there are no indications of how many values are actually used (or whether it varies from person to person). It would be extremely useful if evidence could be obtained from the brain showing, if not exactly how many values, then at least a minimum number of values with which human beings work.

4. Consequences for the Philosophy of Language

If the inferences that have been drawn from the success of PIP are valid and if the Functional Theory of Language has some claim to be taken seriously by philosophers, then there are a number of consequences to which those philosophers should apply themselves. First, the notion that a Theory of Language must be grounded in a meaningful theory of syntax must be seen as a blind alley. PIP performs no syntactic analysis, in the accepted sense of that phrase, on its input sentences. The analysis, such as it is, is extremely *ad hoc*, based on no theory whatsoever; it works simply because explicitly performative sentences in English are constructed in the way

---

16 See section 3.3 of this chapter.

that they are. It must also be repeated here that one cannot separate a Theory of Language from a Theory of the Use of Language. Progress in the study of human comprehension of Language can only be impeded by a belief that there is some abstract 'knowledge' of their language that human beings possess, but which does not necessarily help them to produce 'correct' utterances in that language. Thus, the two programs of PIP both comprehend and generate utterances according to the same theory, not different ones.

Moreover, the adequacy of both TG syntax and systemic grammars to describe the grammatical structure of English as it is used must show that it is not difficult to find descriptions for what Chomsky would call our intuitive knowledge of our own language. Until such time as neurophysiology can determine how human brain cells are structured and, in particular, what the connections between cells might mean, then one is forced to rely on criteria such as 'simplicity' to decide between competing grammars. The evidence from PIP is that one does not require a prior notion of syntax in order to set up a semantic theory. Indeed, it is arguable that one should have a prior notion of semantics before setting up a syntactic theory. Some languages, Latin for instance, do not have a rigid structural frame for their sentences. Perhaps the slogan should not have been "Linguistic Description Minus Grammar Equals Semantics"[17] but rather "Linguistic Description Minus Semantics Equals Grammar", where "Grammar" in this second case is taken to be

---

[17] Katz and Fodor [38], p. 483.

that amount of semantically relevant syntactic information, for example, the scope of a negative particle.

4.1 Given that philosophers will turn (have turned) their attention to semantics, then there is a further point to be made. Very little attention has been paid to the verb. Most semantic theories depend on the noun and noun groups. Yet one of the 'grammatical truths' that most children learn in school is "every sentence must have a verb". If one is agreed that the sentence is the unit of speech (Language) that must be studied, and that clauses in isolation are ellipses for sentences, then the verb simply cries out for prior analysis. PIP also supports this view. The only semantic analysis performed is on the verb. Because of this analysis, PIP can understand English in any context. The important information carrier in the sentence is the verb. Language is the means by which human beings impart information to other human beings. Therefore, philosophers of Language should study the verb. This is the case, even if one does not wish to defend FTL.

4.2 Assuming, on the other hand, that there is some relevance for Philosophy to be found in FTL, then there are some questions that need to be answered. First, what constitutes the set of all possible presuppositions? The set of presuppositions given in section 3 of chapter III can be easily divided into nine groups, each of a different type. Thus:-

Group 1 VALID, NVALID, VALIDBLF, NVALIDBF, VALIDBFH, NVALDBFH

Group 2 CONFDENT, NCONF

Group 3 GOOD, BAD

Group 4 WANTS, NWANTS, WANTH, NWANTH

Group 5 FUTACT, NFUTACT

Group 6 RSPONSBL, NRSPNSBL

Group 7 AUTHORTY, NAUTHRTY

Group 8 VALUE

Group 9 PASTUTT, NPASTUTT

Whilst the author is quite prepared to admit that not all types (or groups) have yet been found, it is by no means admitted that they suffer from Weinreich's disease of Infinite Polysemy.[18] So few groups of presuppositions have so far been deemed necessary by comparison with Katz's semantic markers that it seems extremely unlikely that they will ever suffer from that disease. Moreover, Austin estimates that there are of the order of the third power of 10 (i.e. between 1,000 and 9,999) explicitly performative verbs in English (at a conservative estimate).[19]

On the other hand, mention of Infinite Polysemy leads to the second question that must be answered. What is the inclusion rule that decides what constitutes the set of all (groups of) presuppositions discovered by answering the first

---

18 Cf. chapter I, section 2.1.2.

19 Austin [2], p. 149.

question above? Katz's semantic markers were attacked for having been arbitrarily chosen. At the moment, that is also true of the status of presuppositions. They have so far been chosen because they work. That is, however, not totally sound philosophical reasoning. It might be, for instance, that they correspond to the basic human emotions or to the basic human urges. That would be for philosophers and psychologists to decide. In any case, it is important that some theoretical justification be found. That PIP works, that the notion of presuppositions is useful to a working program to understand and generate English utterances, is an indication that there should be some sound philosophical reason for their 'existence'.

Moreover, there is an interesting factor about the particular presuppositions that have so far appeared to be necessary. Philosophers have written many books on such topics as truth and falsity, knowledge and belief, ethics (morals), free will, what it is to make value judgements. These are precisely the points for which presuppositions have been found necessary. Thus, there are ethical presuppositions (GOOD and BAD), there are presuppositions to distinguish between knowledge and belief (CONFDENT and NCONF), and so on. One wonders whether this is merely a chance happening, or whether this explains why there has been so much philosophical controversy in these areas. It has long been known that Language should not really be used to describe itself. Dictionaries make a good attempt, but usually depend on some notion of circularity. If, however, Language is an axiomatic system (like mathematics), then one would expect that problems could be solved by applying theorems to them. On the other hand, the theorems are built on the axioms, which, being axioms, cannot be 'proved'; they must be assumed. Thus, all problems in Language could be solved by the application of theorems, or by 'logical' reasoning, except those involved with the axioms. There, controversy must rage. If they are the axioms, however, then it is from them that Language is built, and it is to them that an analysis of Language must return. Is this perhaps a justification of the particular presuppositions used, and a purely philosophical reason for why these particular presuppositions should have been chosen?

4.3 At the beginning of the thesis, it was proposed that a theory based on the notions of 'illocutionary force' (demonstrating the intention(s) of the speaker), 'presuppositions' (which, being inherent in a verb, could carry the illocutionary force) and 'explicitly performative verb' (where the illocutionary force is obvious) should be a profitable alternative to TG Theory as a viable Theory of Language. Such a theory should also profit the field of Artificial Intelligence. The theory that has in fact been developed has required one more hypothetical construct, namely the notion of 'cognitive structures', in which are 'stored' the attitudes of the individual as formed by the presuppositions. The experimentation in chapters IV and V has helped to clarify just what kinds of presupposition might be inherent in verbs and therefore what kinds of intention people have when uttering sentences sincerely. More importantly, the experimentation has demonstrated that an understanding of Natural Language in any context can be achieved on the basis of the Functional Theory of Language. Thus, it can be said that the Functional Theory of Language has benefited Artificial Intelligence, since context has previously stopped AI programs from understanding Language generally, and that the Functional Theory of Language does have a claim to be a viable Theory of Language in its own right.

BIBLIOGRAPHY

[1] Abelson, Robert P., and J. Douglas Carroll. "Computer Simulation of Individual Belief Systems," The American Behavioral Scientist, (1965), 24 - 30.

[2] Austin, John L. How to do Things with Words. Oxford: University Press, 1962.

[3] Austin, John L. "Constatives and Performatives," (1962), reprinted in Olshewsky [64], 242 - 250.

[4] Bach, Emmon. An Introduction to Transformational Grammars. New York: Holt, Rinehart and Winston, 1964.

[5] Bobrow, D.G., and J.B. Fraser. "An Augmented State Transition Network Analysis Procedure," Proceedings of the International Joint Conference on Artificial Intelligence, (1969), 557 - 568.

[6] Botha, Rudolf P. The Function of the Lexicon in Transformational Generative Grammar. The Hague: Mouton, 1968.

[7] Burstall, R.M., J.S. Collins, and R.J. Popplestone. Programming in POP-2. Edinburgh: University Press, 1971.

[8] Campbell, Robin, and Roger Wales. "The Study of Language Acquisition," (1970), in Lyons [53], 242 - 260.

[9] Charniak, Eugene. "Jack and Janet in Search of a Theory of Knowledge," Advance Papers of the Third International Joint Conference on Artificial Intelligence, Stanford Research Institute, (1973).

[10] Chomsky, Noam. Syntactic Structures. The Hague: Mouton, 1957.

[11] Chomsky, Noam. "A Review of B.F. Skinner's Verbal Behavior," Language, vol. 35, no. 1, (1959), 26 - 58. (Skinner [79].)

[12] Chomsky, Noam. Aspects of the Theory of Syntax. Cambridge, Mass.: M.I.T. Press, 1965.

[13] Chomsky, Noam. Language and Mind. New York: Harcourt, Brace and World, 1968.

[14] Colby, K.M., and D.C. Smith. "Dialogues Between Humans and an Artificial Belief System," Proceedings of the International Joint Conference on Artificial Intelligence, (1969), 319 - 324.

[15] Collins, Allan M., and M. Ross Quillian. "Retrieval Time from Semantic Memory," Journal of Verbal Learning and Verbal Behavior, vol. 8, (1969), 240 - 247.

[16] Cuadra, Carlos A., and Ann W. Luke, eds. Annual Review of Information Science and Technology. Vol. 4. Chicago: Encyclopedia Britannica, 1969.

[17] Davies, D.J.M. "Representing Negation in a PLANNER System," Advance Papers of the AISB Summer Conference, University of Sussex, (1974), 26 - 36.

[18] Fillmore, Charles J. "Verbs of Judging: an Exercise in Semantic Description," Papers in Linguistics, vol. I, no. 1, (July, 1969). Reprinted in Fillmore and Langendoen [19], 273 - 289.

[19] Fillmore, Charles J., and D. Terence Langendoen, eds. Studies in Linguistic Semantics. New York: Holt, Rinehart and Winston, 1971.

[20] Fillmore, Charles J. "Types of Lexical Information," (1971), in Steinberg and Jakobovits [80], 370 - 392.

[21] Fodor, Jerry A., and Jerrold J. Katz. The Structure of Language. Englewood Cliffs: Prentice-Hall, 1964.

[22] Fodor, Jerry A., and T.G. Bever. "The Psychological Reality of Linguistic Elements," Journal of Verbal Learning and Verbal Behavior, vol. 4, (1965), 414 - 420.

[23] Fodor, Jerry A., and M. Garrett. "Some Reflections on Competence and Performance," (1966), in Lyons and Wales [52], 135 - 154.

[24] Foster, J.M. Automatic Syntactic Analysis. London: Macdonald, 1970.

[25] Friedman, Joyce, et al. A Computer Model of Transformational Grammar. New York: Elsevier, 1971.

[26] Garner, Richard. "'Presupposition' in Philosophy and Linguistics," (1971), in Fillmore and Langendoen [19], 23 - 42.

[27] Grice, H.P. "Meaning," Philosophical Review, vol. 66, (1957), 377 - 388.

[28] Griffiths, A. Phillips, ed. Knowledge and Belief. Oxford: University Press, 1967.

[29] Halliday, M.A.K. "Notes on Transitivity and Theme in English," Journal of Linguistics, vol. 3, (1967), 37 - 81.

[30] Halliday, M.A.K. "Functional Diversity in Language as Seen from a Consideration of Modality and Mood in English," Foundations of Language, vol. 6, (1970), 322 - 361.

[31] Harrison, Bernard. Meaning and Structure: An Essay in the Philosophy of Language. London: Harper and Row, 1972.

[32] Hebb, D.O. The Organization of Behavior. New York: Wiley, 1949.

[33] Hopgood, F.R.A. Compiling Techniques. London: Macdonald, 1969.

[34] Johnson, N.F. "The Psychological Reality of Phrase-Structure Rules," Journal of Verbal Learning and Verbal Behavior, vol. 4, (1965), 469 - 475.

[35] Josselson, Harry H. "Automatic Translation of Languages Since 1960: A Linguist's View," Advances in Computers, vol. 11, (1971), 1 - 58.

[36] Katz, Jerrold J., and Jerry A. Fodor. "What's Wrong with the Philosophy of Language?" Inquiry, vol. 5, (Autumn, 1962), 197 - 237.

[37] Katz, Jerrold J. "Semi-Sentences," (1964), in Fodor and Katz [21], 400 - 416.

[38] Katz, Jerrold J., and Jerry A. Fodor. "The Structure of a Semantic Theory," (1964), in Fodor and Katz [21], 479 - 518.

[39] Katz, Jerrold J. "Analyticity and Contradiction in Natural Language," (1964), in Fodor and Katz [21], 519 - 543.

[40] Katz, Jerrold J., and Paul M. Postal. An Integrated Theory of Linguistic Descriptions. Cambridge, Mass.: M.I.T. Press, 1964.

[41] Katz, Jerrold J. The Philosophy of Language. New York: Harper and Row, 1966.

[42] Katz, Jerrold J. Linguistic Philosophy: The Underlying Reality of Language and its Philosophical Import. London: George Allen and Unwin, 1972.

[43] Klima, Edward S. "Negation in English," (1964), in Fodor and Katz [21], 246 - 323.

[44] Kuno, Susumu, and Anthony G. Oettinger. "Syntactic Structure and Ambiguity in English," AFIPS, vol. 24, (1963), 397 - 418.

[45] Lakoff, George. On the Nature of Syntactic Irregularity. Ph.D. dissertation, Indiana University, 1965.

[46] Lakoff, George. "On Generative Semantics," (1971), in Steinberg and Jakobovits [80], 232 - 296.

[47] Langendoen, D. Terence. Essentials of English Grammar. New York: Holt, Rinehart and Winston, 1970.

[48] Legéndy, C.R. "On the Scheme by Which the Human Brain Stores Information," Mathematical Biosciences, vol. 1, no. 4, (Dec., 1967), 555 - 597.

[49] Lieberman, David V., ed. "Specification and Utilization of a Transformational Grammar," IBM, Scientific Report No. 1, (Mar., 1966).

[50] Lighthill, M.J. Artificial Intelligence: A General Survey. London: Science Research Council, 1973.

[51] Luria, A.R. The Working Brain: An Introduction to Neuropsychology. [Translated by Basil Haigh]. Harmondsworth: The Penguin Press, 1973.

[52] Lyons, John, and R.J. Wales, eds. Psycholinguistic Papers: The Proceedings of the 1966 Edinburgh Conference. Edinburgh: University Press, 1966.

[53] Lyons, John, ed. New Horizons in Linguistics. Aylesbury: Hazell Watson and Viney, 1970.

[54] Macdonald, A.M., ed. Chambers Twentieth Century Dictionary. Edinburgh: W & R Chambers, 1972.

[55] Menyuk, Paula. The Acquisition and Development of Language. Englewood Cliffs: Prentice-Hall, 1971.

[56] Michie, Donald, ed. Machine Intelligence. Vol. 3. Edinburgh: University Press, 1968.

[57] Michie, Donald. "Machines and the Theory of Intelligence," Nature, vol. 241, (Feb., 1973), 507 - 512.

[58] Miller, George A. The Psychology of Communication: Seven Essays. Harmondsworth: The Penguin Press, 1967.

[59] Miller, George A. "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information," (1967), in Miller [58], 21 - 50.

[60] Minsky, Marvin L., ed. Semantic Information Processing. Cambridge, Mass.: M.I.T. Press, 1968.

[61] Montgomery, Christine A. "Automated Language Processing," (1969), in Cuadra and Luke [16], 145 - 174.

[62] McCawley, James D. "Where Do Noun Phrases Come From?" (1971), in Steinberg and Jakobovits [80], 217 - 231.

[63] McNeill, David. The Acquisition of Language: The Study of Developmental Psycholinguistics. New York: Harper and Row, 1970.

[64] Olshewsky, Thomas M., ed. Problems in the Philosophy of Language. New York: Holt, Rinehart and Winston, 1969.

[65] Partee, Barbara Hall. "On the Requirement that Transformations Preserve Meaning," (1971), in Fillmore and Langendoen [19], 1 - 21.

[66] Petrick, Stanley R. A Recognition Procedure for Transformational Grammars. Ph.D. dissertation, Massachusetts Institute of Technology, 1965.

[67] Piaget, Jean. The Language and Thought of the Child. New York: Harcourt, Brace, 1926.

[68] Popper, Karl R. Objective Knowledge: An Evolutionary Approach. Oxford: Clarendon Press, 1972.

[69] Price, H.H. "Some Considerations About Belief," Proceedings of the Aristotelian Society, vol. 35, (1935), 229 - 252.

[70] Quillian, M. Ross. "Semantic Memory," (1968), in Minsky [60], 216 - 270.

[71] Quillian, M. Ross. "The Teachable Language Comprehender: A Simulation Program and Theory of Language," CACM, vol. 12, no. 8, (Aug., 1969), 459 - 479.

[72] Rosenbaum, P., and Dorita Lochak. "The IBM Core Grammar of English," (1966), in Lieberman [49], 1 - 53.

[73] Rosenberg, Jay F., and Charles Travis, eds. Readings in the Philosophy of Language. Englewood Cliffs: Prentice-Hall, 1971.

[74] Schank, Roger C., and Charles J. Rieger. "Inference and Conceptual Memory," Presented to the 1973 International Conference on Computational Linguistics, Pisa, (1973).

[75] Searle, John R. Speech Acts.
Cambridge: University Press, 1969.

[76] Sebeok, T.A., ed. Current Trends in Linguistics.
Vol. III. The Hague: Mouton, 1966.

[77] Shute, G.J. Personal Communication.

[78] Simmons, R.F. "Natural Language Question-Answering Systems: 1969," CACM, vol. 13, no. 1, (Jan., 1970), 15 - 30.

[79] Skinner, B.F. Verbal Behavior.
New York: Appleton-Century-Crofts, 1957.

[80] Steinberg, Danny D., and Leon A. Jakobovits, eds.
Semantics. Cambridge: University Press, 1971.

[81] Strawson, P.F. "Intention and Convention in Speech Acts," The Philosophical Review, vol. 73, (1964), 439 - 460.

[82] Thorndike, E.L., and I. Lorge.
The Teacher's Word Book of 30,000 Words.
New York: Columbia University Press, 1944.

[83] Thorne, J., R. Bratley, and H. Dewar. "The Syntactic Analysis of English by Machine," (1968), in Minsky [60], 281 - 309.

[84] Travis, Charles. "A Generative Theory of Illocutions," (1971), in Rosenberg and Travis [73], 629 - 645.

[85] Weinreich, Uriel. "Explorations in Semantic Theory," (1966), in Sebeok [76], reprinted in Steinberg and Jakobovits [80], 308 - 328.

[86] Weizenbaum, Joseph. "ELIZA - A Computer Program for the Study of Natural Language Communication Between Man and Machine," CACM, vol. 9, no. 1, (Jan., 1966), 36 - 43.

[87] Weizenbaum, Joseph. "Contextual Understanding by Computers," CACM, vol. 10, no. 8, (Aug., 1967), 474 - 480.

[88] Wilks, Yorick Alexander. Grammar, Meaning and the Machine Analysis of Language. London: Routledge and Kegan Paul, 1972.

[89] Winograd, Terry. Understanding Natural Language. Edinburgh: University Press, 1972.

[90] Wittgenstein, Ludwig. Philosophical Investigations. [Translated by G.E.M. Anscombe]. Oxford: Basil Blackwell, 1953.

[91] Woods, W.A. "Augmented Transition Networks for Natural Language Analysis," Report No. CS-1, Aiken Computation Laboratory, Harvard University, (Dec., 1969).

[92] Woozley, A.D. "Knowing and not Knowing," (1967), in Griffiths [28], 82 - 99.

[93] Ziff, Paul. Semantic Analysis. Ithaca, New York: Cornell University Press, 1960.

APPENDIX A

CONTROL STRUCTURE FOR PROGRAM 1

```
TEST ──────┤ REMEMBER ──────────────┤ SETVERB
           │ WRITE ─────────────────┤ VBOL
           │
           │ PARSE (see page 178)
           │
           │ ISHAPPY (see page 179)
           │
           │ RETAIN ────────────────┤ MEM
           │ MEMORY
           │ ISJOHN
           │ FILEIN
           │ CHECKBLF
           │ FILEOUT
```

- PARSE
  - READSENT
  - WHOSAID
  - ISKNOWN
  - FINDUTT
  - FINDVERB
  - IMPLIES
    - DEPRO
    - ISNEG
    - FINDALL
      - FINDARG
        - NEWDEL
      - MOVE
    - RETAIN
      - MEM
    - TENSE
    - WHATBLF
    - CHANGE
      - DELETE
      - EQUIV
    - MEMBLF
  - ISNEG
  - TENSE
  - DEPRO
  - FINDALL
    - FINDARG
      - NEWDEL
    - MOVE
  - WHATPRSP
  - DEFVB
    - SHOW

The following functions are ubiquitous:-

APPEND, EMPSTACK, MEMBER, PRLIST

```
ISHAPPY ├──── MEAN
              │
              WHATELF
              │
              REDUCE
              │
              DELETE
              │
              'any presupposition function' ─┤ KEEP ──────┤ ISKNOWN
              │                               │
              │                               NEG
              │                               │
              │                               RELEASE
              │
              BINARY ─────┤ MEMBLF
```

The following functions are ubiquitous:-

APPEND, EMPSTACK, MEMBER, PRLIST

ALPHABETICAL LIST OF FUNCTIONS USED IN PROGRAM 1

Functions marked * are also used in Program 2

APPEND* puts an item on the end of a list.

BAD* presupposition.

BINARY* performs the checking of the attitude markers for ISHAPPY.

CHANGE* adds the new attitudes for IMPLIES.

CHECKBLF compares PIPTHINK with the attitudes of "John".

CONFDENT* presupposition.

DEFVB* called when the user wishes to define a new verb.

DELETE* deletes a list from a list of lists.

DEPRO* substitutes personal nouns for pronouns.

DUMMY* dummy presupposition for the verb "implies".

EMPSTACK* empties the stack (a POP-2 storage area).

EQUIV* checks the attitude markers for IMPLIES.

FILEIN reads PIP's attitudes for Program 2 into PIPTHINK.

FILEOUT* outputs PIP's attitudes for Program 2.

FINDALL* finds all the arguments from the utterance.

FINDARG* finds ARG.

FINDUTT finds the utterance from the input sentence.

FINDVERB* finds the verb in the utterance.

FUTACT* presupposition.

GOOD* presupposition.

IMPLIES* directs the process of implication.

ISHAPPY* directs the semantic analysis (see text).

ISJOHN returns the value TRUE, iff "John" is the name of one of the speakers.

ISKNOWN determines NUMS and sets up data structures for new speakers.

<u>ISNEG*</u> determines if the utterance is negative.

<u>KEEP*</u> keeps the speaker's attitudes, when the presupposition refers to the ADDRESSEE.

<u>MEAN*</u> directs the semantic analysis for the explicit level presupposition.

<u>MEM*</u> returns the value <u>TRUE</u>, iff a list is a member of a list of lists.

<u>MEMBER*</u> returns the value <u>TRUE</u>, iff an item is a member of a list.

<u>MEMBLF*</u> restores the individual's attitudes to the appropriate data area, after alteration.

<u>MEMORY*</u> outputs the verbs.

<u>MOVE*</u> juggles with words to form ARGONE and ARGTWO.

<u>NCONF*</u> presupposition.

<u>NEG*</u> alters attitudes from positive to negative, either if NEGFLAG is set, or if the presupposition is negative in intent.

<u>NEWDEL*</u> if the delimiter is not found in the utterance, then PIP has found the wrong verb; <u>NEWDEL</u> carries out the search for a new verb with the right delimiter.

<u>NFUTACT*</u> presupposition.

<u>NPASTUTT*</u> presupposition.

<u>NRSPNSBL*</u> presupposition.

<u>NVALID*</u> presupposition.

<u>NVALDEFH*</u> presupposition.

<u>NVALIDEF*</u> presupposition.

<u>NWANTH*</u> presupposition.

NWANTS*	presupposition.
PARSE*	directs the syntactic analysis (see text).
PASTUTT*	presupposition.
PRLIST*	prints a list without brackets.
READSENT*	reads the input sentence.
REDUCE*	produces the second argument of ARGFOUR.
RELEASE*	resets the attitudes after a presupposition has referred to the ADDRESSEE.
REMEMBER*	reads in the verbs and their presuppositions.
RETAIN*	remembers all the arguments.
RSPONSBL*	presupposition.
SETVERB*	sets up the data structure for VERBS.
SHOW	shows the names of the presuppositions if the user wishes to define a new verb and requires help.
TENSE*	standardizes tenses for comparison purposes.
TEST	directs the whole program.
VALID*	presupposition.
VALIDBFH*	presupposition.
VALIDBLF*	presupposition.
VBOL	lists the verbs and delimiters that PIP knows.
WANTH*	presupposition.
WANTS*	presupposition.
WHATBLF*	produces the speaker's attitudes.
WHATPRSP*	produces the implicit level presuppositions of the current verb.
WHOSAID	finds the name of the speaker from the input sentence.
WRITE	lists directives to the user.

ALPHABETICAL LIST OF GLOBAL VARIABLES USED IN PROGRAM 1

Variables marked * are also used in Program 2

ADDRESSEE*	contains the name of the last speaker.
ARG*	contains the argument following the delimiter.
ARGATT*	contains the attitude marker of the individual to ARGUSED.
ARGFOUR*	contains the two arguments of RSPONSEL.
ARGONE*	contains an argument.
ARGTWO*	contains an argument.
ARGUSED*	contains the argument used by ISHAPPY.
BELIEFS*	a selector function producing the beliefs of an individual.
CHECKA*	contains the result of applying logical AND to CHECKATT and MASK.
CHECKATT*	contains 1398101 modified by the presupposition under consideration.
CHECKB*	contains the result of applying logical AND to ARGATT and MASK.
CONSVERB*	constructs a record of type "verb" from a variable.
CONSWLD*	constructs a record of type "world" from a variable.
DELIMIT*	a selector function producing the delimiter associated with a verb.
DESTVERB*	not used (required by the compiler).
DESTWLD*	not used (required by the compiler).
DOER*	contains the first argument of RSPONSEL.
FACT*	used by ISHAPPY as a temporary store of known facts.

FFNLIST*	a list containing the names of those presuppositions that refer to facts.
FINISH*	contains information about whether the program is to be terminated or not.
FUNC*	takes on the value of the presupposition under consideration.
FUNCN*	takes on the name of the presupposition under consideration.
GESAGT*	a list of arguments previously uttered.
HOLDBLF*	if the presupposition refers to the ADDRESSEE, used to hold the original PERSBLF.
HOLDFC*	if the presupposition refers to the ADDRESSEE, used to hold the original FUNC.
HOLDFCN*	if the presupposition refers to the ADDRESSEE, used to hold the original FUNCN.
HOLDNUMS*	if the presupposition refers to the ADDRESSEE, used to refer to the original NUMS.
HOLDPRSP*	if the presupposition refers to the ADDRESSEE, used to hold the original PRESUPL.
HOLDSAID*	if the presupposition refers to the ADDRESSEE, used to hold the original SPEAKER.
ISAD*	logical variable, <u>TRUE</u>, iff FUNC refers to ADDRESSEE.
ISADERR	logical variable, <u>TRUE</u> iff ISAD is TRUE and ADDRESSEE contains the word "nothing".
LASTSAID	contains the name of the current speaker.
MASK*	masks CHECKATT and ARGATT to discover what has been altered from 1398101.

MEANING* a selector function producing the explicit level presupposition of a verb.

MFLAG* used to keep track of whether ISHAPPY has called MEAN.

MISTAKES* contains the number of inconsistencies in each utterance.

NAME* a selector function producing the name of a speaker.

NEGFLAG* logical variable, TRUE, iff the utterance contains the word "not".

NUM* pointer to the array VERBS.

NUMS* pointer to the array SPEAKERL.

PERFORM* a selector function producing the name of a verb.

PERSBLF* used by ISHAPPY as a temporary store of SPEAKER's attitudes.

PFLAG* logical variable, TRUE iff FUNC is PASTUTT or NPASTUTT.

PFNLIST* a list containing the names of those presuppositions that refer to individuals' attitudes.

PIPTHINK* contains PIP's attitudes for Program 2.

POWER* 2 raised to the power POWER is added to or subtracted from 1398101.

PRESUP* a selector function producing the implicit level presuppositions of a verb.

PRESUPL* a list containing the presuppositions of the current verb.

S* contains the input sentence.

SAID*	contains the word "said".
SAVEPRSP*	used by <u>MEAN</u> to hold the original PRESUPL.
SPEAKER*	contains the name of the current speaker.
SPEAKERL*	an array of the records of all known speakers (including "reality").
TEXT*	a function producing textual output.
VERBS*	an array of the records of all known verbs.
VB*	contains the name of the current verb.

LISTING OF PROGRAM 1

```
COMMENT     THIS PAGE SETS UP ALL VARIABLES;




VARS VERBS PRESUP DELIMIT PERFORM DESTVERB CONSVERB;
VARS S SPEAKER SAID VB NUM PRESUPL FUNC ISADERR;
VARS CONSWLD DESTWLD NAME BELIEFS ARGFOUR;
VARS SPEAKERL NUMS ARG ADDRESSEE PERSBLF DOER;
VARS FINISH TEXT PIPTHINK LASTSAID GESAGT;
VARS FUNCN PENLIST FFNLIST FACT NEGFLAG ARGUSED;
VARS CHECKATT ARGATT ISAD MEANING MFLAG SAVEPRSP;
VARS PFLAG MASK POWER CHECKA CHECKB MISTAKES ARGONE ARGTWO;
VARS HOLDNUMS HOLDBLF HOLDSAID HOLDPRSP HOLDFC HOLDFCN;
SUBSCR(%INIT(100)%)->VERBS;
0->ISADERR;
0->MISTAKES;
0->ISAD;
NIL->PFNLIST;
NIL->FFNLIST;
NIL->GESAGT;
"NOTHING"->LASTSAID;
"SAID"->SAID;
"CARRYON"->FINISH;
RECORDFNS("VERB",[0 0 0 0])->PRESUP->MEANING->DELIMIT->PERFORM
                              ->DESTVERB->CONSVERB;
RECORDFNS("WORLD",[0 0])->BELIEFS->NAME->DESTWLD->CONSWLD;
SUBSCR(%INIT(10)%)->SPEAKERL;
CONSWLD("REALITY",NIL)->SPEAKERL(1);
PRSTRING(%CHAROUT%)->TEXT;


FUNCTION SETVERB N;
  VARS I;
  1->I;
A:IF I > N THEN RETURN
  ELSE CONSVERB(NIL,NIL,NIL,NIL)->VERBS(I);
       I+1->I;
       GOTO A
  CLOSE;
END;
```

```
FUNCTION REMEMBER;
  VARS I L M CHARREP INLIST ITEMREP;
  0->I;
  1->M;
  NIL->L;
  SETVERB(100);
  POPMESS([PTIN MIND])->CHARREP;
  INCHARITEM(CHARREP)->ITEMREP;
A:.ITEMREP->INLIST;
  IF INLIST = "[" THEN I+1->I;
                        GOTO A
  ELSEIF INLIST = "]" THEN I-1->I;
                           IF I = 0 THEN
                              L->VERBS(M).PRESUP;
                              POPMESS([%"CLOSE",CHARREP%]);
                              RETURN
                           ELSE GOTO A
                           CLOSE
  ELSEIF INLIST.ISINTEGER THEN L->VERBS(M).PRESUP;
                              NIL->L;
                              INLIST->M;
                              .ITEMREP->VERBS(M).PERFORM;
                              .ITEMREP->VERBS(M).DELIMIT;
                              .ITEMREP->INLIST;
                        POPVAL([%INLIST,";","GOON",";"%]);
                              ->VERBS(M).MEANING;
                              GOTO A
  ELSE POPVAL([%INLIST,";","GOON",";"%]);
       L.APPEND->L;
       GOTO A
  CLOSE;
END;


FUNCTION MEMORY;
  VARS I J K;
  1->I;
  POPMESS([CDOUT MIND])->CUCHAROUT;
  16.CUCHAROUT;              COMMENT      16 = BLANK    ;
  59.CUCHAROUT;              COMMENT      59 = [    ;
A:IF I > 100 THEN GOTO B
  ELSEIF VERBS(I).PERFORM = "NIL" THEN GOTO B CLOSE;
  PR(I);
  PR(VERBS(I).PERFORM);
  PR(VERBS(I).DELIMIT);
  PR(VERBS(I).MEANING.FNPROPS.HD);
  VERBS(I).PRESUP->J;
  NIL->K;
C:IF J.HD = "NIL" THEN PR(K)
  ELSE J.HD.FNPROPS.HD::K->K;
       J.TL->J;
       GOTO C
  CLOSE;
  I+1->I;
  NL(1);
  GOTO A;
B:61.CUCHAROUT;              COMMENT      61 = ]    ;
  NL(1);
  PR(TERMIN);
  CHAROUT->CUCHAROUT;
END;
```

```
FUNCTION APPEND X XL;
  XL<>(X::NIL);
END;


FUNCTION VBOL;
  VARS I;
  1->I;
  NL(1);
  TEXT('PIP ALREADY KNOWS THE FOLLOWING VERBS:-1);
  NL(2);
A:IF VERBS(I).PERFORM = "NIL" THEN EXIT;
  PR(VERBS(I).PERFORM);
  IF VERBS(I).DELIMIT = "NOTHING" THEN GOTO B CLOSE;
  SP(5);
  PR(VERBS(I).DELIMIT);
B:NL(1);
  I+1->I;
  GOTO A;
END;


FUNCTION WRITE;
  VARS Q;
  NL(6);
  TEXT('PIP IS NOW READY FOR USE1);
  NL(2);
  TEXT('DO YOU REQUIRE INFORMATION?1);
  NL(2);
  .ITEMREAD->Q;
  NL(2);
  IF Q = "NO" THEN GOTO D CLOSE;
  TEXT('ONLY TYPE WHEN INVITED1);
  NL(2);
  TEXT('IF YOU USE A VERB THAT PIP DOES NOT KNOW1);
  TEXT(' AND YOU REQUIRE HELP1);
  NL(1);
  TEXT('    THEN PLEASE TYPE THE WORD "HELP"1);
  TEXT(' WITHOUT INVERTED COMMAS1);
  NL(2);
  TEXT('REMEMBER THAT THE VERB "IMPLIES" ONLY1);
  TEXT(' REFERS TO PAST UTTERANCES1);
  NL(1);
  TEXT('    IF YOU WISH THE IMPLICATION TO HOLD LATER1);
  TEXT(' THEN YOU MUST1);
  NL(1);
  TEXT('         RETYPE THE SENTENCE WITH "IMPLIES" IN IT1);
  NL(2);
  TEXT('WHEN YOU WISH TO FINISH THE SESSION,1);
  TEXT(' PLEASE TYPE "END."1);
  NL(1);
  TEXT('    (WITHOUT INVERTED COMMAS) FOR YOUR SENTENCE1);
  NL(1);
  .VBOL;
  NL(2);
D:TEXT('NOW FOR THE FIRST SENTENCE1);
  NL(3);
END;
```

```
FUNCTION READSENT;
  VARS X;
  NIL->S;
A:.ITEMREAD->X;
  IF X = "." THEN RETURN
  ELSE S<>[%X%]->S;
       GOTO A
  CLOSE;
END;


FUNCTION WHOSAID;
  VARS L X Y;
  S->L;
A:DEST(L)->L->X;
  L.HD->Y;
  IF Y = SAID THEN IF L.TL.NULL THEN X->SPEAKER;
                                     RETURN
                   ELSEIF L.TL.HD = """ THEN X->SPEAKER;
                                             RETURN
                   ELSEIF X = """ THEN L.TL.HD->SPEAKER;
                   EXIT
  ELSEIF L.TL.NULL THEN RETURN
  ELSE GOTO A CLOSE;
END;


FUNCTION ISKNOWN;
  VARS I;
  1->I;
A:IF DATAWORD(SPEAKERL(I)) = "WORLD" THEN GOTO C CLOSE;
  CONSWLD(SPEAKER,NIL)->SPEAKERL(I);
B:1->NUMS;
  RETURN;
C:IF SPEAKER = NAME(SPEAKERL(I)) THEN GOTO B CLOSE;
  I+1->I;
  IF I > 10 THEN TEXT('TOO MANY PEOPLE!);
                 NL(1);
                 0->NUMS;
             EXIT;
  GOTO A;
END;
```

```
FUNCTION FINDUTT;
  VARS Q K X Y;
  S->Q;
  0->NEGFLAG;
A:IF Q.NULL THEN GOTO D CLOSE;
  DEST(Q)->Q->K;
  IF K = """ THEN GOTO B ELSE GOTO A CLOSE;
B:NIL->S;
C:IF Q.NULL THEN GOTO D CLOSE;
  DEST(Q)->Q->K;
  IF K = """ THEN EXIT;
  S<>[%K%]->S;
  GOTO C;
D:NL(2);
  TEXT('THIS SENTENCE IS UNGRAMMATICAL');
  NL(2);
  NIL->S;
END;
```

```
FUNCTION FINDVERB;
  VARS I L X Z;
  1->I;
  VERBS(I).PERFORM->Z;
A:S->L;
B:IF Z = "NIL" THEN NL(1);
                    TEXT('PIP DOES NOT KNOW THIS VERB!);
                    NL(2);
                    NIL->VB;
                    EXIT;
  DEST(L)->L->X;
  IF X = Z THEN Z->VB;
                I->NUM;
                RETURN
  ELSEIF L.NULL THEN I+1->I;
                     VERBS(I).PERFORM->Z;
                     GOTO A
  ELSE GOTO B CLOSE;
END;


FUNCTION MEMBER X Y;
  VARS I J;
  Y->I;
A:IF I.NULL THEN FALSE EXIT;
  DEST(I)->I->J;
  IF J = X THEN TRUE EXIT;
  GOTO A;
END;


FUNCTION ISNEG;
  VARS L X Y;
  "NOT"->X;
  IF MEMBER(X,S) THEN 1->NEGFLAG;
                      GOTO G
  CLOSE;
  RETURN;
G;S->Y;
  NIL->S;
E:IF Y.HD = "NIL" THEN EXIT;
  DEST(Y)->Y->L;
  IF L = X THEN GOTO E
  ELSE S<>[%L%]->S;
       GOTO E
  CLOSE:
END;
```

```
FUNCTION DEPRO;
  VARS P Q R;
  S->P;
  NIL->Q;
A:IF P.NULL THEN Q->S EXIT;
  DEST(P)->P->R;
  IF R = "I" THEN SPEAKER->R
  ELSEIF R = "ME" THEN SPEAKER->R
  ELSEIF R = "YOU" THEN ADDRESSEE->R
  ELSEIF R = "WERE" THEN "WAS"->R
  ELSEIF R = "AM" THEN "IS"->R
  ELSEIF R = "ARE" THEN "IS"->R CLOSE;
  Q<>[%R%]->Q;
  GOTO A;
END;


FUNCTION NEWDEL;
  VARS I;
  NUM+1->I;
A:IF VERBS(I).PERFORM = "NIL" THEN NL(1);
                              TEXT('PIP DOES NOT KNOW THIS VERB!);
                                          NL(2);
                                          NIL->VB
                                     EXIT;
  IF VERBS(I).PERFORM = VB THEN I->NUM;
                                   .FINDARG
                                EXIT;
  I+1->I;
  GOTO A;
END;


FUNCTION FINDARG;
  VARS L X Y;
  S->L;
  NIL->ARG;
  NIL->DOER;
  VERBS(NUM).DELIMIT->Y;
A:DEST(L)->L->X;
  IF X = VB THEN GOTO B ELSE GOTO A CLOSE;
B:IF Y = "NOTHING" THEN L->ARG EXIT;
C:IF L.NULL THEN .NEWDEL EXIT;
  DEST(L)->L->X;
  IF X = Y THEN L->ARG
  ELSE DOER<>[%X%]->DOER;
       GOTO C
  CLOSE;
END;


FUNCTION EMPSTACK;
A:IF .STACKLENGTH = 0 THEN EXIT;
  .ERASE;
  GOTO A;
END;
```

```
COMMENT       IN THE NEXT 3 FUNCTIONS, THE DECIMAL CHARS. ARE;
COMMENT       36 = D, 37 = E, 39 = G, 41 = I, 46 = N, 51 = S;



FUNCTION TENSE;
  VARS L M X Y;
  ARG->L;
  NIL->M;
A:IF L.NULL THEN M->ARG EXIT;
  DEST(L)->L->X;
  IF X = "DID" THEN DEST(L)->L->X;
                    X.DESTWORD;
                    ->Y;
                    37;
                    36;
                    Y+2;
                    .CONSWORD->X
  ELSEIF X = "DOES" THEN DEST(L)->L->X;
                         X.DESTWORD;
                         ->Y;
                         51;
                         Y+1;
                         .CONSWORD->X
  ELSEIF X = "DO" THEN DEST(L)->L->X
  CLOSE;
  M<>[%X%]->M;
  GOTO A;
END;


FUNCTION MOVE X Y;
  VARS P Q R L;
  X->L;
  IF L.NULL THEN DOER<>[%Y%]->ARGTWO;
                 NIL->ARGONE
           EXIT;
A:IF L.TL.NULL THEN GOTO B CLOSE;
  L.TL->L;
  GOTO A;
B:L.HD.DESTWORD->P->Q;
  IF Q = 51 THEN "WERE"->R
           ELSE "WAS"->R
  CLOSE;
  X<>[%R%]<>[%Y%]->ARGONE;
  IF DOER.HD = "NIL" THEN ARG->ARGTWO
                     ELSE DOER<>[%Y%]<>X->ARGTWO
  CLOSE;
END;
```

```
FUNCTION FINDALL;
  VARS A B C D X Y;
  .FINDARG;
  IF VB = "NIL" THEN EXIT;
  .TENSE;
  ARG->X;
  DEST(X)->X->Y;
  Y.DESTWORD->A;
  IF A > 4 THEN ->B;
                ->C;
                ->D
  ELSE GOTO V
  CLOSE;
  IF B = 39 AND C = 46 AND D = 41 THEN GOTO W
  ELSEIF A = 8 AND B = 46 AND C = 41 THEN D;
                                          GOTO W
  ELSEIF A = 8 AND B = 41 THEN D;
                               C;
                               GOTO W
  CLOSE;
V:.EMPSTACK;
  ARG->ARGONE;
  ARG->ARGTWO;
  ARG->X;
  DEST(X)->X->Y;
  [%Y%]->DOER;
P:IF X = "NIL" THEN NIL->DOER;
                    [["NIL"] ["NIL"]]->ARGFOUR
               EXIT;
  DEST(X)->X->Y;
  Y.DESTWORD->A;
  IF A > 4 THEN ->B;
                ->C
  ELSE .EMPSTACK;
       DOER<>[%Y%]->DOER;
       GOTO P
  CLOSE;
  IF B = 36 AND C = 37 THEN
  ELSEIF A = 8 AND B = 37 THEN
  ELSE .EMPSTACK;
       DOER<>[%Y%]->DOER;
       GOTO P
  CLOSE;


COMMENT     THIS FUNCTION CONTINUES;
```

```
   IF A < 8 THEN 41;
                 46;
                 39;
                 A+1
   ELSEIF B = 36 THEN 41;
                      46;
                      A
   ELSE C;
        41;
        A
   CLOSE;
   .CONSWORD->Y;
   .EMPSTACK;
   Y::X->X;
   [%DCEP%]<>[%X%]->ARGFOUF;
   RETURN;
 W:[%DOER%]<>[%ARG%]->ARGFOUR;
   IF A < 8 THEN 37;
                 36;
                 A-1
   ELSEIF B = 39 THEN 37;
                      36;
                      A-1
   ELSEIF B = 46 THEN 37;
                      36;
                      A
   ELSE 37;
        A
   CLOSE;
   .CONSWORD->Y;
   MOVE(X,Y);
END;
```

```
FUNCTION WHATPRSP;
  VERBS(NUM).PRESUP->PRESUPL;
END;


FUNCTION MEM X Y;
  VARS I J;
  Y->I;
A:IF I.NULL THEN FALSE EXIT;
  DEST(I)->I->J;
  IF EQUAL(J,X) THEN TRUE EXIT;
  GOTO A;
END;


FUNCTION RETAIN;
  IF MEM(ARG,GESAGT) THEN
  ELSE ARG::GESAGT->GESAGT CLOSE;
  IF MEM(ARGONE,GESAGT) THEN
  ELSE ARGONE::GESAGT->GESAGT CLOSE;
  IF MEM(ARGTWO,GESAGT) THEN
  ELSE ARGTWO::GESAGT->GESAGT CLOSE;
END;


FUNCTION SHOW;
  VARS I J X Y;
  0->I;
  0->J;
  FENLIST->Y;
  NL(1);
  TEXT('THE FOLLOWING FUNCTIONS REFER TO FACTS:-');
  NL(2);
A:IF Y.NULL THEN NL(2);
                  IF J = 0 THEN GOTO B ELSE EXIT
  CLOSE;
  DEST(Y)->Y->X;
  IF X = "DUMMY" THEN GOTO A CLOSE;
  PR(X);
  I+1->I;
  IF I < 5 THEN SP(5);
                GOTO A CLOSE;
  NL(2);
  0->I;
  GOTO A;
B:0->I;
  1->J;
  PFNLIST->Y;
  TEXT('THE FOLLOWING FUNCTIONS REFER TO THE ');
  NL(1);
  TEXT('COGNITIVE STRUCTURES OF INDIVIDUALS:-');
  NL(2);
  GOTO A;
END;
```

```
FUNCTION DEFVB;
  VARS K I J;
A:TEXT('DO YOU WISH TO EXPLAIN THIS VERB?!);
  NL(1);
  .ITEMREAD->K;
  IF K = "NO" THEN "FINISH"->FINISH;
                    RETURN
  ELSEIF K = "YES" THEN GOTO B
  ELSE TEXT('I BEG YOUR PARDON? PLEASE ANSWER YES OR NO!);
       NL(1);
       GOTO A
  CLOSE;
B:NL(1);
  TEXT('WHAT IS THE VERB IN THIS SENTENCE?!);
  NL(1);
  .ITEMREAD->VB;
  0->I;
C:I+1->I;
  IF I > 100 THEN TEXT('THERE IS NO MORE ROOM FOR NEW VERBS!);
                  NL(2);
                  "SORRY"->FINISH
            EXIT;
  IF VERBS(I).PERFORM = "NIL" THEN VB->VERBS(I).PERFORM;
                                   I->NUM;
                                   GOTO D
  ELSE GOTO C CLOSE;
D:NL(1);
  TEXT('WHAT IS THE DELIMITER IN THIS SENTENCE?!);
  NL(1);
  .ITEMREAD->VERBS(NUM).DELIMIT;
  NL(1);
G:TEXT('WHAT IS THE MEANING OF!);
  PR(VB);
  NL(1);
  .ITEMREAD->K;
  IF NOT(MEMBER(K,PFNLIST)) AND NOT(MEMBER(K,FFNLIST))
                              THEN GOTO F CLOSE;
  POPVAL([%K,";","GOON",";"%]);
  ->VERBS(NUM).MEANING;
  NL(1);
  TEXT('PLEASE LIST THE PRESUPPOSITIONS INHERENT IN !);
  PR(VB);
  NL(1);
  TEXT('PLEASE FINISH THE LIST WITH A ".."!);
  NL(2);
  NIL->J;


COMMENT      THIS FUNCTION CONTINUES;
```

```
E:.ITEMREAD->K;
  IF K = "HELP" THEN .SHOW;
          TEXT('YOU MAY NOW LIST THE PRESUPPOSITIONS!);
                          NL(1);
                          GOTO E
  ELSEIF K = "NONE" THEN GOTO E
  ELSEIF K = "." THEN J->VERBS(NUM).PRESUP EXIT;
  IF NOT(MEMBER(K,PFNLIST)) AND NOT(MEMBER(K,FFNLIST))
                                  THEN GOTO F CLOSE;
  POPVAL([%K,";","GOON",";"%]);
  J.APPEND->J;
  GOTO E;
F:NL(1);
  PR(K);
  TEXT(' IS NOT A VALID PRESUPPOSITION!);
  NL(1);
  IF VERBS(NUM).MEANING = "NIL" THEN GOTO G
  ELSE GOTO F CLOSE;
END;
```

```
FUNCTION WHATBLF;
  SPEAKERL(NUMS).BELIEFS->PERSBLF;
END;


FUNCTION MEMBLF;
  PERSBLF->SPEAKERL(NUMS).BELIEFS;
END;


FUNCTION DELETE X Y;
  VARS L Z;
  NIL->L;
A:IF Y.HD = "NIL" THEN L EXIT;
  IF Y.HD.ISINTEGER THEN Y->Z;
                         NIL->Y
  ELSE DEST(Y)->Y->Z
  CLOSE;
  IF EQUAL(X,Z) THEN GOTO B CLOSE;
  IF L.HD = "NIL" THEN Z->L
  ELSE Z::L->L
  CLOSE;
B:GOTO A;
END;


FUNCTION EQUIV A B;
  VARS I J;
  1->I;
  1->J;
  0->POWER;
  3->MASK;
D:IF I = 12 THEN B;
                 RETURN
  ELSEIF I > 1 THEN POWER+2->POWER;
                    MASK*4->MASK;
                    INTOF(2↑POWER)->J
  CLOSE;
  LOGAND(A,MASK)->CHECKA;
  IF CHECKA = J THEN GOTO F CLOSE;
  LOGAND(B,MASK)->CHECKB;
  IF NEGFLAG AND NOT(POWER = 4) THEN GOTO E
  ELSEIF CHECKB = J THEN IF CHECKA = 0 THEN B-J->B
                                       ELSE B+J->B
                         CLOSE
  ELSEIF NOT(CHECKA = CHECKB) THEN MISTAKES+1->MISTAKES
  CLOSE;
  GOTO F;
E:IF CHECKB = J THEN IF CHECKA = 0 THEN B+J->B
                                   ELSE B-J->B
                     CLOSE
  ELSEIF CHECKA = CHECKB THEN MISTAKES+1->MISTAKES
  CLOSE;
F:I+1->I;
  GOTO D;
END;
```

```
FUNCTION CHANGE X Y:
  VARS J L M W I Q;
  X.HD->J;
  1->I;
  0->POWER;
  3->MASK;
  1->L;
  PERSBLF->Q;
A:IF Q.HD = "NIL" THEN GOTO B CLOSE;
  DEST(Q)->Q->M;
  IF M.ISINTEGER THEN M::Q ->M CLOSE;
  IF M.ATOM THEN GOTO B CLOSE;
  IF EQUAL(M.TL,Y) THEN GOTO C
  ELSE GOTO A CLOSE;
B:IF NEGFLAG THEN 1398101->W;
                  1398101::Y->M;
                  GOTO D
            CLOSE;
  IF PERSBLF.HD = "NIL" THEN J::Y->PERSBLF
  ELSE J::Y->Y;
       Y::PERSBLF->PERSBLF
  CLOSE;
  RETURN;
C:DELETE(M,PERSBLF);
  M.HD->W;
D:EQUIV(J,W)->M.HD;
  IF PERSBLF.HD = "NIL" THEN M->PERSBLF
  ELSE M::PERSBLF->PERSBLF
  CLOSE;
END;
```

```
FUNCTION IMPLIES;
   VARS I L M X Y Z;
   NIL->M;
   .DEPRO;
   S->L;
   .ISNEG;
   .FINDALL;
   .RETAIN;
   ARG->Y;
   NL(1);
A:DEST(L)->L->Z;
   IF Z = "IMPLIES" THEN M->ARG;
                         .TENSE;
                         ARG->X;
                         GOTO B
                     CLOSE;
   M<>[%Z%]->M;
   GOTO A;
B:"NOT"->M;
   IF MEMBER(M,X) THEN
TEXT('THE LEFT-HAND-SIDE OF AN IMPLICATION MUST BE POSITIVE!);
      NL(1);
      TEXT('    TRY AGAIN!);
      NL(1);
      0->NEGFLAG
   EXIT;
   1->I;
   0->L;
C:0->MISTAKES;
   IF I = 11 THEN IF L = 0 THEN TEXT('O.K.!) CLOSE;
                  NL(1)
              EXIT;
   IF NOT(DATAWORD(SPEAKERL(I)) = "WORLD") THEN
            IF L = 0 THEN TEXT('O.K.!) CLOSE;
            NL(1)
   EXIT;
   I->NUMS;
   .WHATBLF;
   PERSBLF->Z;
   NIL->M;


COMMENT      THIS FUNCTION CONTINUES;
```

```
D:IF Z.HD = "NIL" THEN GOTO E CLOSE;
  DEST(Z)->Z->M;
  IF M.ISINTEGER THEN M::Z ->M CLOSE;
  IF M.ATOM THEN GOTO E CLOSE;
  IF EQUAL(M.TL,X) THEN CHANGE(M,Y);
                        GOTO E
  ELSE GOTO D CLOSE;
E:.MEMBLE;
  IF MISTAKES = 0 THEN I+1->I;
                       GOTO C
                  CLOSE;
  L+1->L;
  TEXT('FOR!);
  PR(SPEAKERL(NUMS).NAME);
  IF MISTAKES = 1 THEN TEXT(' THERE WAS 1 INCONSISTENCY!)
  ELSE TEXT(' THERE WERE!);
       PR(MISTAKES);
       TEXT(' INCONSISTENCIES!)
  CLOSE;
  I+1->I;
  GOTO C;
END;
```

```
FUNCTION PARSE;
  VARS Q;
  .READSENT;
  IF S.HD = "END" THEN EXIT;
  .WHOSAID;
  IF NOT(SPEAKER = LASTSAID) THEN LASTSAID -> ADDRESSEE;
                                   .ISKNOWN;
                               IF NUMS = 0 THEN GOTO R CLOSE
                                   CLOSE;
  .FINDUTT;
  IF S.NULL THEN GOTO R CLOSE;
  .FINDVERB;
  IF VB = "NIL" THEN GOTO H
  ELSEIF VB = "IMPLIES" THEN .IMPLIES;
                                   NL(1)
                             EXIT;
X:IF NOT(S.HD = "I") THEN
     TEXT('AN UTTERANCE MUST BEGIN WITH THE WORD "I"!);
                              NL(2);
                              GOTO R
                         CLOSE;
  .ISNEG;
  .DEPRO;
  .FINDALL;
  IF VB = "NIL" THEN GOTO H CLOSE;
  .WHATPRSP;
  RETURN;
R:TEXT('DO YOU WISH TO CARRY ON WITH THE TEXT?!);
  NL(2);
  .ITEMREAD->Q;
  IF Q = "YES" THEN GOTO Z
  ELSEIF Q = "NO" THEN TEXT('O.K.!);
                             NL(1);
           TEXT(' THANK YOU FOR THE SESSION. GOODBYE.!);
                             NL(1);
                             "FINISH"->FINISH;
                             RETURN
  ELSE TEXT('PARDON?!);
       NL(1);
       GOTO R;
  CLOSE;
H:.DEFVB;
  IF FINISH = "FINISH" THEN GOTO Z
  ELSEIF FINISH = "SORRY" THEN GOTO K CLOSE;
  GOTO X;
Z:NL(1);
  TEXT('O.K.!);
  NL(1);
K:TEXT('PLEASE TYPE THE NEXT SENTENCE!);
  NL(2);
  "AGAIN"->FINISH;
END;
```

```
FUNCTION MEAN;
   PRESUPL->SAVEPRSP;
   [%VERBS(NUM),MEANING%]->PRESUPL;
   1->MFLAG;
   .ISHAPPY;
   SAVEPRSP->PRESUPL;
   2->MFLAG;
END;


FUNCTION PRLIST LIST;
   VARS X Y;
   LIST->Y;
   IF Y.ATOM THEN PR(Y) EXIT;
A:DEST(Y)->Y->X;
   PR(X);
   IF Y.HD = "NIL" THEN EXIT;
   GOTO A;
END;


FUNCTION REDUCE LIST;
   VARS L M;
   LIST->L;
   DEST(L)->L->M;
   IF NOT(M.ISCOMPND) THEN TEXT('ILLEGAL ARGFOUR');
                                   ARGFOUR=>
                              EXIT;
   L;
END;
```

```
FUNCTION BINARY;
   VARS X Y Z JLOOP;
   IF PFLAG = "NIL" THEN GOTO A
   ELSEIF PFLAG THEN RETURN
   ELSE GOTO J CLOSE;
A:0->JLOOP;
   0->POWER;
   3->MASK;
   1->Z;
B:LOGAND(CHECKATT,MASK)->CHECKA;
   IF CHECKA = Z THEN GOTO C CLOSE;
   LOGAND(ARGATT,MASK)->CHECKB;
   IF CHECKB = Z THEN
                     IF CHECKATT < 1596101 THEN ARGATT-Z->ARGATT
                                           ELSE ARGATT+Z->ARGATT
                     CLOSE;
                     GOTO H
   ELSEIF CHECKA = CHECKB THEN GOTO H
   CLOSE;
J:MISTAKES+1->MISTAKES;
   1->JLOOP;
   NL(1);
   IF MFLAG = 1 THEN TEXT('THE MEANING OF !);
                          GOTO D
                     CLOSE;
   TEXT('A PRESUPPOSITION OF !);
D:PR(VB);
   TEXT('  IS:-!);
   NL(1);
   SP(5);
   IF ISAD THEN PR(HOLDFC")
   ELSE PR(FUNCN)
   CLOSE;


COMMENT      THIS FUNCTION CONTINUES;
```

```
   IF PFLAG = 0 THEN
            16.CHAROUT;             COMMENT      16 = BLANK    ;
            26.CHAROUT;             COMMENT      26 = *    ;
            24.CHAROUT;             COMMENT      24 = (    ;
            PR(ARGUSED);
            16.CHAROUT;
            25.CHAROUT              COMMENT      25 = )    ;
   ELSEIF POWER = 10 THEN
            16.CHAROUT;
            24.CHAROUT;
            PRLIST(DOER);
            16.CHAROUT;
            28.CHAROUT;             COMMENT      28 = ,    ;
            IF NEGFLAG THEN TEXT(' NOT!) CLOSE;
            PRLIST(ARG);
            16.CHAROUT;
            25.CHAROUT
   ELSE IF NEGFLAG THEN 16.CHAROUT;
                        24.CHAROUT;
                        TEXT(' NOT!)
        CLOSE;
        PR(ARGUSED.TL);
        IF NEGFLAG THEN 16.CHAROUT;
                        25.CHAROUT
        CLOSE
   CLOSE;
   NL(1);
   IF MEMBER(FUNCN,FFNLIST) THEN
          TEXT('THIS IS INCONSISTENT WITH A PREVIOUS FACT!)
   ELSE TEXT('THIS IS INCONSISTENT WITH !);
        TEXT('A PREVIOUS ATTITUDE OF !);
        PR(SPEAKER)
   CLOSE;
   NL(1);
   IF PFLAG = "NIL" THEN GOTO H
   ELSE EXIT;
H:IF JLOOP THEN
   ELSEIF DOER = "NIL" THEN
   ELSEIF POWER = 10 AND NOT(EQUAL(DOER,ARGUSED.TL,HD))
                      THEN GOTO J CLOSE;
   ARGATT->ARGUSED.HD;
   IF MEMBER(FUNCN,PFNLIST) THEN IF PERSBLF.HD = "NIL" THEN
                                              ARGUSED->PERSBLF
                                 ELSE
                                 ARGUSED::PERSBLF->PERSBLF
                                 CLOSE;
                                 .MEMBLF
   ELSE IF FACT.HD = "NIL" THEN ARGUSED->FACT
        ELSE ARGUSED::FACT->FACT
        CLOSE:
        FACT->SPEAKERL(1).BELIEFS
   CLOSE;
   RETURN;
C:POWER+2->POWER;
   INTOF(2↑POWER)->Z;
   MASK * 4 ->MASK;
   GOTO B;
END;
```

```
FUNCTION ISHAPPY;
   VARS X Y Z I J HOLDZ;
   0->J;
   IF MFLAG = 2 THEN GOTO A
   ELSEIF MFLAG = 1 THEN GOTO C
   ELSE .MEAN
   CLOSE;
A:IF PRESUPL.HD = "NIL" THEN GOTO D CLOSE;
C:NIL->PFLAG;
   1308101->CHECKATT;
   .WHATBLF;
   SPEAKERL(1).BELIEFS->FACT;
   DEST(PRESUPL)->PRESUPL->FUNC;
   IF FUNC = "NIL" THEN "THERE"->FUNCN
   ELSE FUNC.FNPROPS.HD->FUNCN
   CLOSE;
   IF MEMBER(FUNCN,PFNLIST) THEN PERSBLF->Y;
   ELSEIF MEMBER(FUNCN,FFNLIST) THEN FACT->Y
   ELSE PR(FUNCN);
        TEXT(' WOULD APPEAR TO BE AN INVALID PRESUPPOSITION!);
        NL(1);
        GOTO A
   CLOSE;
   FUNC.FNPROPS.TL.HD->I;
   IF EQUAL(ARGONE,ARGTWO) THEN 1->J CLOSE;
   IF I = "ARG" THEN ARG->X
   ELSEIF I = "ARGONE" THEN ARGONE->X
   ELSEIF I = "ARGTWO" THEN ARGTWO->X
   ELSEIF I = "ARGTHREE" THEN IF J = 0 THEN ARGONE->X;
                                            1->J;
                                     APPEND(FUNC,PRESUPL)->PRESUPL
                                ELSE ARGTWO->X;
                                     0->J
                                CLOSE
   ELSEIF I = "ARGFOUR" THEN ARGFOUR.REDUCE->X
   ELSE TEXT('PROGRAMMING ERROR!);
        NL(2)
   EXIT;
   IF X.HD = "NIL" THEN GOTO A CLOSE;
   FUNC.FNPROPS.HD -> Z;
   IF Z = "PASTUFF" OR Z = "NPASTUFF" THEN X->ARGUSED;
                                            .FUNC->PFLAG;
                                            .BINARY;
                                            GOTO A
   CLOSE;


COMMENT      THIS FUNCTION CONTINUES;
```

```
F:IF Y.HD = "NIL" THEN GOTO E CLOSE;
  DEST(Y)->Y->Z;
  IF Z.ISINTEGER THEN Z::Y ->Z CLOSE;
  IF Z.ATOM THEN GOTO E CLOSE;
  IF NOT(I = "ARGFOUR") THEN GOTO B
  ELSEIF Z.HD = 1397077 THEN
  ELSEIF Z.HD = 1399125 THEN
  ELSE GOTO F
  CLOSE;
  Z->HOLDZ;
  Z.HD::(REDUCE(Z.TL))->Z;
B:IF NOT(EQUAL(Z.TL,X)) THEN GOTO F CLOSE;
  IF I = "ARGFOUR" THEN HOLDZ->Z CLOSE;
  IF MEMBER(FUNCN,PENLIST) THEN DELETE(Z,PERSBLF)->PERSBLF
  ELSE DELETE(Z,FACT)->FACT
  CLOSE;
  Z->ARGUSED;
  GOTO G;
E:IF I = "ARGFOUR" THEN ARGFOUR->X
  CLOSE;
  1398101::X->ARGUSED;
G:.FUNC;
  IF ISADERR = 1 THEN 0->ISADERR;
                       GOTO A
                  CLOSE;
  FUNCN.DESTWORD;
  .ERASE;
  ->I;
  .EMPSTACK;
  IF I = 40 THEN EXIT;
  ARGUSED.HD->ARGAT;
   BINARY;
  GOTO A;
D:IF MFLAG = 1 THEN EXIT;
  MISTAKES->X;
  0->MISTAKES;
  NL(1);
  IF X = 0 THEN TEXT('O.K.!);
                NL(2)
  ELSEIF X = 1 THEN TEXT('THERE WAS 1 INCONSISTENCY!);
                    TEXT(' IN THAT UTTERANCE!);
                    NL(2)
  ELSE TEXT('THERE WERE!);
       PR(X);
       TEXT(' INCONSISTENCIES IN THAT UTTERANCE!);
       NL(2)
  CLOSE;
END;
```

```
FUNCTION ISJOHN;
  VARS I;
  2->I;
A:IF I > 10 THEN FALSE;
                 RETURN
  ELSEIF NOT(SPEAKERL(I).DATAWORD = "WORLD") THEN FALSE;
                                                  RETURN
  ELSEIF SPEAKERL(I).NAME = "JOHN" THEN I->NUMS;
                                        TRUE
                                    EXIT;
  I+1->I;
  GOTO A;
END;


FUNCTION FILEIN;
  COMPILE(POPMESS([CDIN PIPBLF]));
END;


FUNCTION FILEOUT;
  VARS OUT;
  POPMESS([CDOUT PIPBLF])->CUCHAROUT;
  PRSTRING(%CUCHAROUT%)->OUT;
  PR(PIPTHINK);
  OUT('->PIPTHINK;');
  NL(1);
  PR(TERMIN);
  CHAROUT->CUCHAROUT;
END;
```

```
FUNCTION CHECKBLF;
  VARS I LJ LP LJARG LPARG M N;
  PIPTHINK->LP;
  .WHATBLF;
  PERSBLF->LJ;
A:IF LJ.HD = "NIL" THEN EXIT;
  DEST(LJ)->LJ->M;
  IF M.ISINTEGER THEN M::LJ->M;
                      NIL->LJ
                CLOSE;
  IF M.HD = 1397077 THEN GOTO C
  ELSEIF M.HD = 1599125 THEN GOTO C
  CLOSE;
  M.TL->LJARG;
B:IF LP.HD = "NIL" THEN M::PIPTHINK->PIPTHINK;
                        PIPTHINK->LP;
                        GOTO A
                   CLOSE;
  DEST(LP)->LP->N;
  IF N.ISINTEGER THEN N::LP->N;
                      NIL->LP
                CLOSE;
  IF N.HD = 1397077 THEN GOTO B
  ELSEIF N.HD = 1599125 THEN GOTO B
  CLOSE;
  N.TL->LPARG;
  IF NOT(EQUAL(LJARG,LPARG)) THEN GOTO B CLOSE;
  0->MISTAKES;
  EQUIV(M.HD,N.HD)->I;
  IF MISTAKES = 0 THEN DELETE(N,PIPTHINK)->PIPTHINK;
                        I->N.HD;
                        N::PIPTHINK->PIPTHINK
                  CLOSE;
  PIPTHINK->LP;
  GOTO A;
C:IF LP.HD = "NIL" THEN M::PIPTHINK->PIPTHINK;
                        PIPTHINK->LP;
                        GOTO A
                   CLOSE;
  DEST(LP)->LP->N;
  IF N.ISINTEGER THEN N::LP->N;
                      NIL->LP
                CLOSE;
  IF N.HD = 1397077 THEN
  ELSEIF N.HD = 1599125 THEN
  ELSE GOTO C
  CLOSE:
  IF NOT(EQUAL(M.TL.HD,N.TL.HD)) THEN GOTO C CLOSE;
  IF NOT(EQUAL(M.TL.TL,N.TL.TL)) THEN GOTO C CLOSE;
  PIPTHINK->LP;
  GOTO A;
END:
```

```
FUNCTION KEEP;
  FUNC->HOLDFC;
  FUNCN->HOLDFCN;
  NUMS->HOLDNUMS;
  PERSBLF->HOLDBLF;
  SPEAKER->HOLDSAID;
  PRESUPL->HOLDPRSP;
  IF ADDRESSEE = "NOTHING" THEN 0->ISAD;
                                1->ISADERR;
                                MISTAKES+1->MISTAKES;
                                NL(1);
  TEXT('THERE IS NO ADDRESSEE FOR THIS SENTENCE');
                                NL(1)
                            EXIT;
  ADDRESSEE->SPEAKER;
  .ISKNOWN;
  NIL->PRESUPL;
  1->ISAD;
END;


FUNCTION RELEASE;
  0->ISAD;
  HOLDPRSP->PRESUPL;
  SPEAKER->ADDRESSEE;
  HOLDSAID->SPEAKER;
  HOLDBLF->PERSBLF;
  HOLDNUMS->NUMS;
  HOLDFCN->FUNCN;
  HOLDFC->FUNC;
END;


FUNCTION NEG;
  VARS Q;
  POWER+1->POWER;
  INTOF(2↑POWER)->Q;
  CHECKATT+Q->CHECKATT;
END;


FUNCTION PASTUTT;
  IF MEM(ARGUSED,G-SAGT) THEN TRUE
  ELSE FALSE CLOSE;
END;
FFNLIST<>[%"PASTUTT"%]->FFNLIST;


FUNCTION NPASTUTT;
  IF .PASTUTT THEN FALSE
  ELSE TRUE CLOSE;
END;
FFNLIST<>[%"NPASTUTT"%]->FFNLIST;


FUNCTION DUMMY;
END;
FFNLIST<>[%"DUMMY"%]->FFNLIST;
```

```
FUNCTION VALIDBLF;
   0->POWER;
   CHECKATT - 1 ->CHECKATT;
   IF NEGFLAG THEN .NEG CLOSE;
END;
PFNLIST<>[%"VALIDBLF"%]->PFNLIST;



FUNCTION NVALIDBF;
   VARS Q;
   .VALIDBLF;
   IF NEGFLAG THEN INTOF(2↑POWER)->Q;
                   CHECKATT-Q->CHECKATT
   ELSE .NEG CLOSE;
END;
PFNLIST<>[%"NVALIDBF"%]->PFNLIST;



FUNCTION VALID;
   IF MFLAG = 1 AND MEMBER(VALIDBLF,SAVEPRSP) THEN GOTO A
   ELSEIF MFLAG = 2 AND MEMBER(VALIDBLF,PRESUPL) THEN GOTO A
   CLOSE;
   POPVAL([%"VALIDBLF",":","GOON",";"%]);
   IF MFLAG = 1 THEN SAVEPRSP.APPEND->SAVEPRSP
   ELSE PRESUPL.APPEND->PRESUPL
   CLOSE;
A:.VALIDBLF;
END;
PFNLIST<>[%"VALID"%]->PFNLIST;



FUNCTION NVALID;
   IF MFLAG = 1 AND MEMBER(NVALIDBF,SAVEPRSP) THEN GOTO A
   ELSEIF MFLAG = 2 AND MEMBER(NVALIDBF,PRESUPL) THEN GOTO A
   CLOSE;
   POPVAL([%"NVALIDBF",":","GOON",";"%]);
   IF MFLAG = 1 THEN SAVEPRSP.APPEND->SAVEPRSP
   ELSE PRESUPL.APPEND->PRESUPL
   CLOSE;
A:.NVALIDBF;
END;
PFNLIST<>[%"NVALID"%]->PFNLIST;
```

```
FUNCTION GOOD;
   2->POWER;
   CHECKATT - 4 ->CHECKATT;
   IF NEGFLAG THEN .NEG CLOSE;
END;
PENLIST<>[%"GOOD"%]->PENLIST;


FUNCTION BAD;
   VARS Q;
   .GOOD;
   IF NEGFLAG THEN INTOF(2↑POWER)->Q;
                       CHECKATT-Q->CHECKATT
   ELSE .NEG CLOSE;
END;
PENLIST<>[%"BAD"%]->PENLIST;


FUNCTION CONFDENT;
   4->POWER;
   CHECKATT - 16 -> CHECKATT;
END;
PENLIST<>[%"CONFDENT"%]->PENLIST;


FUNCTION NCONF;
   .CONFDENT;
   .NEG;
END;
PENLIST<>[%"NCONF"%]->PENLIST;


FUNCTION FUTACT;
   6->POWER;
   CHECKATT - 64 -> CHECKATT;
   IF NEGFLAG THEN .NEG CLOSE;
END;
PENLIST<>[%"FUTACT"%]->PENLIST;


FUNCTION NFUTACT;
   VARS Q;
   .FUTACT;
   IF NEGFLAG THEN INTOF(2↑POWER)->Q;
                       CHECKATT-Q->CHECKATT
   ELSE .NEG CLOSE;
END;
PENLIST<>[%"NFUTACT"%]->PENLIST;
```

```
FUNCTION WANTS;
  8->POWER;
  CHECKATT-256->CHECKATT;
  IF NEGFLAG THEN .NEG CLOSE;
END;
PFNLIST<>[%"WANTS"%]->PFNLIST;


FUNCTION NWANTS;
  VARS Q;
  .WANTS;
  IF NEGFLAG THEN INTOF(2↑POWER)->Q;
                  CHECKATT-Q->CHECKATT
  ELSE .NEG CLOSE;
END;
PFNLIST<>[%"NWANTS"%]->PFNLIST;


FUNCTION RSPONSBL;
  IF MFLAG = 1 AND MEMBER(VALIDBLF,SAVEPRSP) THEN GOTO A
  ELSEIF MFLAG = 2 AND MEMBER(VALIDBLF,PRESUPL) THEN GOTO A
  CLOSE;
  POPVAL([%"VALIDBLF",":","GOON",";"%]);
  IF MFLAG = 1 THEN SAVEPRSP.APPEND->SAVEPRSP
  ELSE PRESUPL.APPEND->PRESUPL
  CLOSE;
A:10->POWER;
  CHECKATT-1024->CHECKATT;
  IF NEGFLAG THEN .NEG CLOSE;
END;
PFNLIST<>[%"RSPONSBL"%]->PFNLIST;


FUNCTION NRSPNSBL;
  VARS Q;
  IF MFLAG = 1 AND MEMBER(NVALIDBF,SAVEPRSP) THEN GOTO A
  ELSEIF MFLAG = 2 AND MEMBER(NVALIDBF,PRESUPL) THEN GOTO A
  CLOSE;
  POPVAL([%"NVALIDBF",":","GOON",";"%]);
  IF MFLAG = 1 THEN SAVEPRSP.APPEND->SAVEPRSP
  ELSE PRESUPL.APPEND->PRESUPL
  CLOSE;
A:.RSPONSBL;
  IF NEGFLAG THEN INTOF(2↑POWER)->Q;
                  CHECKATT-Q->CHECKATT
  ELSE .NEG CLOSE;
END;
PFNLIST<>[%"NRSPNSBL"%]->PFNLIST;
```

```
FUNCTION VALIDBFH;
   .KEEP;
   IF ISAD = 0 THEN EXIT;
   VALIDBLF;
   PRESUPL.APPEND->PRESUPL;
   .ISHAPPY;
   .RELEASE;
END;
PFNLIST<>[%"VALIDBFH"%]->PFNLIST;
```

```
FUNCTION NVALDBFH;
   .KEEP;
   IF ISAD = 0 THEN EXIT;
   NVALIDBF;
   PRESUPL.APPEND->PRESUPL;
   .ISHAPPY;
   .RELEASE;
END;
PFNLIST<>[%"NVALDBFH"%]->PFNLIST;
```

```
FUNCTION WANTH;
   .KEEP;
   IF ISAD = 0 THEN EXIT;
   WANTS;
   PRESUPL.APPEND->PRESUPL;
   .ISHAPPY;
   .RELEASE;
END;
PFNLIST<>[%"WANTH"%]->PFNLIST;
```

```
FUNCTION NWANTH;
   .KEEP;
   IF ISAD = 0 THEN EXIT;
   NWANTS;
   PRESUPL.APPEND->PRESUPL;
   .ISHAPPY;
   .RELEASE;
END;
PFNLIST<>[%"NWANTH"%]->PFNLIST;
```

```
[VALIDBLF ARGTHREE]->VALIDBLF.FNPROPS;
[NVALIDBF ARGTHREE]->NVALIDBF.FNPROPS;
[VALID ARGONE]->VALID.FNPROPS;
[NVALID ARGONE]->NVALID.FNPROPS;
[GOOD ARG]->GOOD.FNPROPS;
[BAD ARG]->BAD.FNPROPS;
[CONFDENT ARGTWO]->CONFDENT.FNPROPS;
[NCONF ARGTWO]->NCONF.FNPROPS;
[FUTACT ARG]->FUTACT.FNPROPS;
[NFUTACT ARG]->NFUTACT.FNPROPS;
[WANTS ARGTWO]->WANTS.FNPROPS;
[NWANTS ARGTWO]->NWANTS.FNPROPS;
[RSPONSBL ARGFOUR]->RSPONSBL.FNPROPS;
[NRSPNSBL ARGFOUR]->NRSPNSBL.FNPROPS;
[VALIDBFH ARGTWO]->VALIDBFH.FNPROPS;
[NVALDBFH ARGTWO]->NVALDBFH.FNPROPS;
[WANTH ARGTWO]->WANTH.FNPROPS;
[NWANTH ARGTWO]->NWANTH.FNPROPS;
[PASTUTT ARG]->PASTUTT.FNPROPS;
[NPASTUTT ARG]->NPASTUTT.FNPROPS;
[DUMMY ARG]->DUMMY.FNPROPS;


FUNCTION TEST;
   VARS L M;
   .REMEMBER;
   .WRITE;
A:.PARSE;
   IF S.HD = "END" THEN GOTO F
   ELSEIF FINISH = "FINISH" THEN GOTO F
   ELSEIF FINISH = "AGAIN" THEN "CARRYON"->FINISH;
                                      GOTO A
   ELSEIF VB = "IMPLIES" THEN .RETAIN;
                                   GOTO A
                          CLOSE;
   0->MFLAG;
   .ISHAPPY;
   .RETAIN;
   SPEAKER->LASTSAID;
   GOTO A;
F:.MEMORY;
   IF .ISJOHN THEN POPVAL([%".","FILEIN",";","GOON",";"%]);
                   POPVAL([%".","CHECKBLF",";","GOON",";"%]);
                   POPVAL([%".","FILEOUT",";","GOON",";"%])
   CLOSE;
   NL(2);
   TEXT('THE ACTUAL NUMBER OF COMPUTER WORDS ');
   TEXT('USED ON THIS RUN WAS:-');
   NL(2);
   PR(COREUSED);
   NL(3);
   .KILL;
END;


COMMENT     A CALL OF   .TEST   WILL START THE PROGRAM;
```

SAMPLE RUN OF PROGRAM 1

```
ASTON UNIVERSITY G3 7.7Z  8AUG74
09.45.50- LOGIN MOPJNDF,:SSP0428
TYPE PASSWORD-
STARTED :SSP0428,MOPJNDF, 8AUG74  09.46.20 TYPE:MOP
09.46.22- PIP




PIP IS NOW READY FOR USE

DO YOU REQUIRE INFORMATION?

- YES


ONLY TYPE WHEN INVITED

IF YOU USE A VERB THAT PIP DOES NOT KNOW AND YOU REQUIRE HELP
   THEN PLEASE TYPE THE WORD "HELP" WITHOUT INVERTED COMMAS

REMEMBER THAT THE VERB "IMPLIES" ONLY REFERS TO PAST UTTERANCES
   IF YOU WISH THE IMPLICATION TO HOLD LATER THEN YOU MUST
      RETYPE THE SENTENCE WITH "IMPLIES" IN IT

WHEN YOU WISH TO FINISH THE SESSION, PLEASE TYPE "END."
   (WITHOUT INVERTED COMMAS) FOR YOUR SENTENCE

PIP ALREADY KNOWS THE FOLLOWING VERBS:-

 SWEAR       THAT
 INTEND       TO
 STATE       THAT
 BELIEVE       THAT
 ASSERT      THAT
 CONFIRM       THAT
 IMPLIES
 KNOW       THAT
 ACCUSE       OF
 PRAISE      FOR
 QUESTION       WHETHER
 PROMISE       TO
 WANT      TO
 PRESUME       THAT
 DENY      THAT
 AFFIRM       THAT
 BLAME       FOR
```

CONFESS	THAT
REPRIMAN	FOR
WISH	FOR
PLEDGE	
ASK	FOR
IMAGINE	THAT
GUESS	THAT
REQUEST	THAT
DOUBT	WHETHER
ADMIT	THAT
CONGRATU	ON
INDICT	FOR
PARDON	FOR
THINK	THAT
LIKE	
DISLIKE	
PUNISH	FOR
DREAD	THAT
LOVE	
HATE	
SUPPOSE	THAT
WONDER	WHETHER
GRANT	
SUGGEST	THAT
DECLARE	THAT
HOLD	THAT
MAINTAIN	THAT
SURMISE	THAT
REFUSE	TO
SUSPECT	THAT
PROCLAIM	THAT
VOW	THAT
PROTEST	THAT
CONTEND	THAT
CLAIM	THAT
DISBELIE	THAT
REJECT	
REFUTE	
WANT	
ACQUIT	OF
REPROACH	FOR
REPROVE	FOR
DISAPPRO	OF
APPROVE	OF
CONDEMN	FOR
ADVOCATE	THAT
COMPLIME	ON

NOW FOR THE FIRST SENTENCE

← JOHN SAID "I KNOW THAT INFLATION IS EVIL".

O.K.

← "INFLATION MUST BE STOPPED".

PIP DOES NOT KNOW THIS VERB

DO YOU WISH TO EXPLAIN THIS VERB?
← NO

O.K.
PLEASE TYPE THE NEXT SENTENCE

← "I ASSERT THAT INFLATION MUST BE STOPPED".

O.K.

← "INFLATION IS EVIL IMPLIES INFLATION MUST NOT BE STOPPED".

FOR JOHN THERE WAS 1 INCONSISTENCY

← ANNE SAID "I SUPPOSE THAT INFLATION MUST BE STOPPED".

O.K.

← "I STATE THAT MONEY IS THE ROOT OF EVIL" SAID BOB.

O.K.

← MONEY IS THEREFORE THE ROOT OF INFLATION.

THIS SENTENCE IS UNGRAMMATICAL

DO YOU WISH TO CARRY ON WITH THE TEXT?

← YES

O.K.
PLEASE TYPE THE NEXT SENTENCE

← "MONEY IS THE ROOT OF EVIL IMPLIES
← MONEY IS THE ROOT OF INFLATION".

O.K.

← JANE SAID "I QUESTION WHETHER
← MONEY IS THE ROOT OF INFLATION".

O.K.

← "I BELIEVE THAT INFLATION IS CAUSED BY COMPETITION".

O.K.

←

```
← "I MAINTAIN THAT COMPETITION IS NECESSARY" HENRY SAID.

O.K.

← "I HATE COMPETITION SAID ALICE.


THIS SENTENCE IS UNGRAMMATICAL

DO YOU WISH TO CARRY ON WITH THE TEXT?

← BURP
PARDON?
DO YOU WISH TO CARRY ON WITH THE TEXT?

← YES

O.K.
PLEASE TYPE THE NEXT SENTENCE

← "I HATE COMPETITION" SAID ALICE.

O.K.

← JOHN SAID "I DENY THAT COMPETITION IS NECESSARY".

O.K.

← "I ALSO BELIEVE THAT COMPETITION IS NOT NECESSARY"
← SAID HENRY.

THE MEANING OF  BELIEVE  IS:-
      VALIDBLF ( NOT [COMPETIT IS NECESSAR] )
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF  HENRY

THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE

← ALICE SAID "I APPROVE OF COMPETITION".

THE MEANING OF  APPROVE  IS:-
      GOOD [COMPETIT]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF  ALICE

THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE

← "COMPETITION IMPLIES COMPETITION IS NECESSARY".

O.K.

← "I ADVOCATE THAT COMPETITION IS NECESSARY".

THE MEANING OF  ADVOCATE  IS:-
      GOOD [COMPETIT IS NECESSAR]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF  ALICE

A PRESUPPOSITION OF  ADVOCATE  IS:-
      WANTS [COMPETIT IS NECESSAR]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF  ALICE

THERE WERE 2 INCONSISTENCIES IN THAT UTTERANCE

←
```

```
← GEORGE SAID "I COMPLIMENT PIP ON ITS ASTUTENESS".

O.K.

← "I SUGGEST THAT WE CHANGE THE SUBJECT" SAID HARRY.

O.K.

← "I DISAPPROVE OF HUNTING FOXES" SAID ROY.

O.K.

← JOHN SAID "I DESPISE PEOPLE FOR GOING FOXHUNTING".

PIP DOES NOT KNOW THIS VERB

DO YOU WISH TO EXPLAIN THIS VERB?
← YES

WHAT IS THE VERB IN THIS SENTENCE?
← DESPISE

WHAT IS THE DELIMITER IN THIS SENTENCE?
← FOR

WHAT IS THE MEANING OF DESPISE
← BAD

PLEASE LIST THE PRESUPPOSITIONS INHERENT IN  DESPISE
PLEASE FINISH THE LIST WITH A "."

← HELP

THE FOLLOWING FUNCTIONS REFER TO FACTS:-

 PASTUTT      NPASTUTT      VALID      NVALID

THE FOLLOWING FUNCTIONS REFER TO THE
COGNITIVE STRUCTURES OF INDIVIDUALS:-

 VALIDBLF      NVALIDBF      GOOD      BAD      CONFDENT

 NCONF      FUTACT      NFUTACT      WANTS      NWANTS

 RSPONSBL      NRSPNSBL      VALIDBFH      NVALDBFH      WANTH

 NWANTH

YOU MAY NOW LIST THE PRESUPPOSITIONS
← RSPONSBL.

O.K.

←
```

```
← BARBARA SAID "I RATHER LIKE HUNTING FOXES".
TOO MANY PEOPLE
DO YOU WISH TO CARRY ON WITH THE TEXT?

← YES

O.K.
PLEASE TYPE THE NEXT SENTENCE

← "I LIKE HUNTING FOXES" SAID HARRY.

O.K.

← JOHN SAID "I DENY THAT INFLATION MUST BE STOPPED".

THE MEANING OF  DENY  IS:-
      NVALIDBF [INFLATIO MUST BE STOPPED]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF  JOHN

THERE WAS 1 INCONSISTENCY IN THAT UTTERANCE

← HARRY SAID "I DISLIKE HUNTING FOXES".

THE MEANING OF  DISLIKE  IS:-
      NWANTS [HUNTING FOXES]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF  HARRY

A PRESUPPOSITION OF  DISLIKE  IS:-
      BAD [HUNTING FOXES]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF  HARRY

THERE WERE 2 INCONSISTENCIES IN THAT UTTERANCF

← END.


THE ACTUAL NUMBER OF COMPUTER WORDS USED ON THIS RUN WAS:-

 12786


10.30.25← LOGOUT
MAXIMUM ONLINE BS USED 26 KWORDS
10.30.41 1.29 FINISHED :SSP0428,MOPJNDF : 3 LISTFILES
JOB COST=      90P:UNITS LEFT   19049P
```

APPENDIX B

CONTROL STRUCTURE FOR PROGRAM 2

- CHAT
  - REMEMBER
    - SETVERB
    - SETTAG
  - THOUGHT
  - PARSE (see page 178)
  - ISHAPPY (see page 179)
  - RETAIN
    - MEM
  - ANSWER
    - CLARIFY
      - FINDNEWV
        - ANYRESP
          - HASARG
        - COMPARE
      - DISPLAY
      - INCLUDES
    - INCLUDES
  - REPLY
    - WHATPRSP
    - ISHAPPY (see page 179)
    - CLARIFY
      - FINDNEWV
        - ANYRESP
          - HASARG
        - COMPARE
      - DISPLAY
      - INCLUDES
  - MEMORY
  - FILEOUT

ALPHABETICAL LIST OF FUNCTIONS USED IN PROGRAM 2

This list does not contain those functions used in Program 1

ANSWER	replies to a question.
ANYRESP	tries to find DOERBLF.
CHAT	directs the whole program.
CLARIFY	constructs the sentence using NEWV.
COMPARE	returns TRUE, iff ISHAPPY produces zero MISTAKES for a putative NEWV.
DISPLAY	tells the user of inconsistencies if no sentence can be constructed from the verbs that PIP knows to explain its attitudes.
FINDNEWV	tries to find a verb to express PIP's attitudes to the arguments.
HASARG	returns TRUE, iff PIP has an attitude to some argument in the sentence.
INCLUDES	returns TRUE, iff one list is part of one other list.
REPLY	replies when PIP agrees with, or has no attitude to, ARGUSED; else calls CLARIFY.
SETTAG	sets up the data structure for TAG.
THOUGHT	outputs PIP's attitudes.

ALPHABETICAL LIST OF GLOBAL VARIABLES USED IN PROGRAM 2

This list does not contain those variables used in Program 1

CFLAG	logical variable, <u>TRUE</u> iff PIP is trying to find a NEWV.
DOERBLF	contains PIP's idea of DOER.
HASBLF	logical variable, <u>TRUE</u> iff PIP has some attitude to the arguments of the input sentence.
INITTAG	constructs a strip of tags from a variable.
NEWV	contains the name of the verb used by PIP when generating its own utterances; it is set to zero if no verb can be found.
OFFENDER	a list of those presuppositions indicating attitudes to the input utterance where PIP disagrees with the user.
PRESUPM	a list containing the presuppositions of the putative NEWV.
QMARK	logical variable, <u>TRUE</u> iff the input is a question.
SUBSCRTAG	a function for referring to the subscripts of a strip of tags.
TAG	an array of tags.
TEMPARGT	used by <u>FINDNEWV</u> to hold the original ARGTWO.
TESTATT	used by <u>ANYRESP</u> to test PIP's attitude to <u>RSPONSBL</u> or <u>NRSPNSBL</u>.

LISTING OF PROGRAM 2

This listing contains only the differences from Program 1

```
COMMENT      THIS PAGE SETS UP ALL VARIABLES;



VARS VERBS PRESUP DELIMIT PERFORM DESIVERB CONSVERB;
VARS S SPEAKER SAID VB NUM PRESUPL FUNC TAG HASBLF TESTATT;
VARS OFFENDER PRESUPM CONSWLD DESTWLD NAME BELIEFS TEMPARGT;
VARS SPEAKERL NUMS ARG ADDRESSEE PERSBLF DOER NEWV PIPTHINK;
VARS FINISH TEXT GESAGT QMARK SUBSCRTAG INITTAG DOERBLF;
VARS FUNCN PENLIST FENLIST FACT NEGFLAG ARGUSED CFLAG;
VARS CHECKATT ARGATT ISAD MEANING MFLAG SAVEPRSP ARGFOUR;
VARS FFLAG MASK POWER CHECKA CHECKB MISTAKES ARGONE ARGTWO;
VARS HOLDNUMS HOLDBLF HOLDSAID HOLDPRSP HOLDFC HOLDFCN;
SUBSCR(%INIT(100)%)->VERBS;
STRIPFNS("VERBTAG",3)->SUBSCRTAG->INITTAG;
SUBSCRTAG(%INITTAG(100)%)->TAG;
0->MISTAKES;
0->ISAD;
0->CFLAG;
NIL->PENLIST;
NIL->FENLIST;
NIL->GESAGT;
NIL->DOERBLF;
"CARRYON"->FINISH;
RECORDFNS("VERB",[0 0 0 0])->PRESUP->MEANING->DELIMIT->PERFORM
                              ->DESTVERB->CONSVERB;
RECORDFNS("WORLD",[0 0])->BELIEFS->NAME->DESTWLD->CONSWLD;
SUBSCR(%INIT(3)%)->SPEAKERL;
CONSWLD("REALITY",NIL)->SPEAKERL(1);
CONSWLD("PIP",NIL)->SPEAKERL(2);
PRSTRING(%CHAROUT%)->TEXT;


FUNCTION SETVERB N;
  VARS I;
  1->I;
A:IF I > N THEN RETURN
  ELSE CONSVERB(NIL,NIL,NIL,NIL)->VERBS(I);
       I+1->I;
       GOTO A
  CLOSE;
END;


FUNCTION SETTAG N;
  VARS I;
  1->I;
A:IF I > N THEN RETURN
  ELSE 0->TAG(I);
       I+1->I;
       GOTO A
  CLOSE;
END;
```

```
FUNCTION THOUGHT;
  COMPILE(POPMESS([CDIN PIPRLF]));
END;


FUNCTION READSENT;
  VARS X;
  NIL->S;
  0->QMARK;
A:.ITEMREAD->X;
  IF X = "*" THEN 1->QMARK;
                  RETURN
  ELSEIF X = "." THEN RETURN
  ELSE S<>[%X%]->S;
       GOTO A
  CLOSE;
END;


FUNCTION INCLUDES X Y;
  VARS XL YL I L;
  X->XL;
  Y->YL;
  IF XL.LENGTH > YL.LENGTH THEN FALSE EXIT;
  XL.LENGTH->I;
  IF I = 0 THEN TEXT('ERROR IN INCLUDES!);
                NL(1)
           CLOSE;
  [%YL.HD%]->L;
  YL.TL->YL;
A:I-1->I;
  IF I = 0 THEN
  ELSE L<>[%YL.HD%]->L;
       YL.TL->YL;
       GOTO A
  CLOSE;
B:IF EQUAL(XL,L) THEN TRUE EXIT;
  IF YL = "NIL" THEN FALSE EXIT;
  L.TL<>[%YL.HD%]->L;
  YL.TL->YL;
  GOTO B;
END;
```

```
FUNCTION PARSE;
  VARS J L Q;
  .READSENT;
  IF S.HD = "END" THEN EXIT;
  .FINDVERB;
  IF VB = "NIL" THEN GOTO H
  ELSEIF VB = "IMPLIES" THEN .IMPLIES;
                              NL(1)
                        EXIT;
  S->L;
A:DEST(L)->L->J;
  IF J = "YOU" THEN IF QMARK THEN "PIP"->SPEAKER;
                                  "JOHN"->ADDRESSEE;
                                  2->NUMS
                  ELSE NL(1);
            TEXT('I CAN MAKE UP MY OWN MIND THANK YOU!);
                        "AGAIN"->FINISH;
                        NL(2)
                  EXIT
  ELSEIF J = "I" THEN "JOHN"->SPEAKER;
                      "PIP"->ADDRESSEE;
                      3->NUMS
  ELSEIF L.HD = "NIL" THEN NL(2);
            TEXT('PLEASE TALK ABOUT YOU OR ME!);
                          NL(1);
                          GOTO K
  ELSE GOTO A
  CLOSE;
  .ISNEG;
  .DEPRO;
  .FINDALL;
  IF VB = "NIL" THEN GOTO H CLOSE;
  ARGTWO->TEMPARGT;
  .WHATPRSP;
  RETURN;
R;TEXT('DO YOU WISH TO CARRY ON WITH THE TEXT?!);
  NL(2);
  .ITEMREAD->Q;
  IF Q = "YES" THEN GOTO Z
  ELSEIF Q = "NO" THEN TEXT('O.K.!);
                       NL(1);
            TEXT(' THANK YOU FOR THE SESSION, GOODBYE.!);
                       NL(1);
                       "FINISH"->FINISH;
                       RETURN
  ELSE TEXT('PARDON?!);
       NL(1);
       GOTO R;
  CLOSE;
H:.DEFVB;
  IF FINISH = "FINISH" THEN GOTO Z
  ELSEIF FINISH = "SORRY" THEN GOTO K CLOSE;
  GOTO A;
Z:NL(1);
  TEXT('O,K,!);
  NL(1);
K:TEXT('PLEASE TYPE THE NEXT SENTENCE!);
  NL(2);
  "AGAIN"->FINISH;
END;
```

```
FUNCTION BINARY;
  VARS Y Z JLOOP;
  IF PFLAG = "NIL" THEN GOTO A
  ELSEIF PFLAG THEN RETURN
  ELSE GOTO J CLOSE;
A:0->JLOOP;
  0->POWER;
  3->MASK;
  1->Z;
  IF DOERBLF.NULL THEN DOER->Y
  ELSE DOERBLF->Y CLOSE;
B:LOGAND(CHECKATT,MASK)->CHECKA;
  IF CHECKA = Z THEN GOTO C CLOSE;
  LOGAND(ARGATT,MASK)->CHECKB;
  IF CHECKB = Z THEN
                IF CFLAG THEN 1->MISTAKES EXIT;
                IF CHECKATT < 1596101 THEN ARGATT-Z->ARGATT
                                      ELSE ARGATT+Z->ARGATT
                CLOSE;
                GOTO H
  ELSEIF CHECKA = CHECKB THEN GOTO H
  CLOSE;
  IF POWER = 10 AND NOT(EQUAL(Y,ARGUSED.TL.HD))
                THEN GOTO E
  CLOSE;
J:MISTAKES+1->MISTAKES;
  IF CFLAG THEN EXIT;
  1->JLOOP;
  FUNC.FNPROPS.TL.HD::OFFENDER->OFFENDER;
  IF OFFENDER.HD = "PASTUTT" THEN OFFENDER.TL->OFFENDER;
                                 MISTAKES-1->MISTAKES
  ELSEIF OFFENDER.HD = "NPASTUTT" THEN OFFENDER.TL->OFFENDER;
                                      MISTAKES-1->MISTAKES
                                 CLOSE;
  IF NUMS = 2 OR QMARK THEN EXIT;
  NL(1);
  IF MFLAG = 1 THEN TEXT('THE MEANING OF ');
                    GOTO D
               CLOSE;
  TEXT('A PRESUPPOSITION OF ');
D:PR(VB);
  TEXT('  IS:-');
  NL(1);
  SP(5);
  IF ISAD THEN PR(HOLDFCN)
  ELSE PR(FUNCN)
  CLOSE;


COMMENT     THIS FUNCTION CONTINUES;
```

```
  IF PFLAG = 0 THEN 16.CHAROUT;
                    26.CHAROUT;
                    24.CHAROUT;
                    PR(ARGUSED);
                    16.CHAROUT;
                    25.CHAROUT;
                    MISTAKES-1->MISTAKES
  ELSEIF POWER = 10 THEN
          16.CHAROUT;
          24.CHAROUT;
          PRLIST(ARGFOUR.HD);
          16.CHAROUT;
          28.CHAROUT;
          IF NEGFLAG THEN TEXT(' NOT!) CLOSE;
          PRLIST(ARGFOUR.TL.HD);
          16.CHAROUT;
          25.CHAROUT
  ELSE IF NEGFLAG THEN 16.CHAROUT;
                       24.CHAROUT;
                       TEXT(' NOT!)
       CLOSE;
       PR(ARGUSED.TL);
       IF NEGFLAG THEN 16.CHAROUT;
                       25.CHAROUT
       CLOSE
  CLOSE;
  NL(1);
  IF MEMBER(FUNCN,FFNLIST) THEN
          TEXT('THIS IS INCONSISTENT WITH A PREVIOUS FACT!)
  ELSE TEXT('THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE!);
       TEXT(' OF YOURS,!);
       PR(SPEAKERL(3).NAME)
  CLOSE;
  NL(1);
  IF PFLAG = "NIL" THEN GOTO H
  ELSE EXIT;
H:IF JLOOP THEN
  ELSEIF DOEF = "NIL" THEN
  ELSEIF POWER = 10 AND NOT(EQUAL(V,ARGUSED.TL.HD))
                THEN GOTO J
  CLOSE;
E:IF NOT(NUMS = 3) OR OMARK THEN EXIT;
  ARGATT->ARGUSED.HD;
  IF MEMBER(FUNCN,PFNLIST) THEN IF PERSBLF.HD = "NIL" THEN
                                          ARGUSED->PERSBLF
                                ELSE
                                ARGUSED::PERSBLF->PERSBLF
                                CLOSE;
                                .MEMBLF
  ELSE IF FACT.HD = "NIL" THEN ARGUSED->FACT
       ELSE ARGUSED::FACT->FACT
       CLOSE;
       FACT->SPEAKERL(1).BELIEFS
  CLOSE;
  RETURN;
C:POWER+2->POWER;
  INTOF(2↑POWER)->Z;
  MASK * 4 ->MASK;
  GOTO B;
END;
```

```
FUNCTION ISHAPPY;
  VARS X Y Z I J HOLDZ;
  0->J;
  IF MFLAG = 2 THEN GOTO A
  ELSEIF MFLAG = 1 THEN GOTO C
  ELSE .MEAN
  CLOSE;
A:IF PRESUPL.HD = "NIL" THEN EXIT;
C:NIL->PFLAG;
  1398101->CHECKATT;
  .WHATBLF;
  SPEAKERL(1).BELIEFS->FACT;
  DEST(PRESUPL)->PRESUPL->FUNC;
  IF FUNC = "NIL" THEN "THERE"->FUNCN
  ELSE FUNC.FNPROPS.HD->FUNCN
  CLOSE;
  IF MEMBER(FUNCN,PFNLIST) THEN PERSBLF->Y;
  ELSEIF MEMBER(FUNCN,FFNLIST) THEN FACT->Y
  ELSE PR(FUNCN);
       TEXT(' WOULD APPEAR TO BE AN INVALID PRESUPPOSITION!);
       NL(1);
       GOTO A
  CLOSE;
  FUNC.FNPROPS.TL.TL.HD->I;
  IF EQUAL(ARGONE,ARGTWO) THEN 1->J CLOSE;
  IF I = "ARG" THEN ARG->X
  ELSEIF I = "ARGONE" THEN ARGONE->X
  ELSEIF I = "ARGTWO" THEN ARGTWO->X
  ELSEIF I = "ARGTHREE" THEN IF J = 0 THEN ARGONE->X;
                                              1->J;
                                          APPEND(FUNC,PRESUPL)->PRESUPL
                                     ELSE ARGTWO->X;
                                          0->J
                                     CLOSE
  ELSEIF I = "ARGFOUR" THEN ARGFOUR.REDUCE->X
  ELSE TEXT('PROGRAMMING ERROR!);
       NL(2)
  EXIT;
  IF X.HD = "NIL" THEN GOTO A CLOSE;
  IF FUNCN = "PASTUTT" OR FUNCN = "NPASTUTT" THEN X->ARGUSED;
                                                  .FUNC->PFLAG;
                                                  .BINARY;
                                                  GOTO A

  CLOSE;


COMMENT      THIS FUNCTION CONTINUES;
```

```
F:IF Y.HD = "NIL" THEN GOTO E CLOSE;
  DEST(Y)->Y->Z;
  IF Z.ISINTEGER THEN Z::Y->Z CLOSE;
  IF Z.ATOM THEN GOTO E CLOSE;
  IF NOT(I = "ARGFOUR") THEN GOTO B
  ELSEIF Z.HD = 1397077 THEN
  ELSEIF Z.HD = 1399125 THEN
  ELSE GOTO F
  CLOSE;
  Z->HOLDZ;
  Z.HD::(REDUCE(Z.TL))->Z;
B:IF NOT(EQUAL(Z.TL,X)) THEN GOTO F
  ELSEIF NOT(MEMBER(FUNCN,FENLIST)) THEN 1->HASBLF CLOSE;
  IF I = "ARGFOUR" THEN HOLDZ->Z CLOSE;
  IF NOT(NUMS = 3) OR QMARK THEN
  ELSEIF MEMBER(FUNCN,PENLIST) THEN DELETE(Z,PERSBLF)->PERSBLF
  ELSE DELETE(Z,FACT)->FACT
  CLOSE;
  Z->ARGUSED;
  GOTO G;
E:IF CFLAG THEN 1->MISTAKES EXIT;
  IF NUMS = 2 OR QMARK THEN GOTO A CLOSE;
  IF I = "ARGFOUR" THEN ARGFOUR->X CLOSE;
  1398101::X->ARGUSED;
G:.FUNC;
  FUNCN DESTWORD;
  .ERASE;
  ->I;
  .EMPSTACK;
  IF I = 40 THEN EXIT;
  ARGUSED.HD->ARGATT;
  .BINARY;
  GOTO A;
END;
```

```
FUNCTION HASARG LIST;
  VARS Y Z;
  LIST->Y;
  Y.REDUCE.HD->Y;
  IF EQUAL(ARG,Y) THEN TRUE ELSE FALSE CLOSE;
END;



FUNCTION ANYRESP;
  VARS L X AAA BBB;
  NIL->AAA;
  DOER->BBB;
  .WHATBLF;
  PERSBLF->L;
A:IF L.HD = "NIL" THEN FALSE EXIT;
  DEST(L)->L->X;
  IF X.ISINTEGER THEN X::L->X CLOSE;
  IF X.ATOM THEN GOTO A CLOSE;
  IF X.HD = TESTATT AND HASARG(X.TL) THEN GOTO B CLOSE;
  GOTO A;
B:X.TL->X;
  X.HD->DOERBLF;
  IF EQUAL(DOER,DOERBLF) THEN GOTO A CLOSE;
  DOERBLF->ARGFOUR.HD;
C:IF BBB.HD = "NIL" THEN DOERBLF<>AAA->ARGTWO
  ELSE DELETE(BBB.HD,ARGTWO).REV->AAA;
       BBB.TL->BBB;
       GOTO C
  CLOSE;
  TRUE;
END;



FUNCTION COMPARE;
  VARS HOLDPRSP;
  PRESUPL->HOLDPRSP;
  PRESUPM->PRESUPL;
  1->CFLAG;
  0->MFLAG;
  0->MISTAKES;
  .ISHAPPY;
  0->CFLAG;
  HOLDPRSP->PRESUPL;
  IF MISTAKES = 0 THEN TRUE
                  ELSE FALSE
  CLOSE;
END;
```

```
FUNCTION FINDNEWV;
  VARS I K L M N SL HOLDNUM;
  0->N;
  NIL->L;
  NUM->HOLDNUM;
  OFFENDER->K;
A:1->I;
  IF L = "RSPONSBL" THEN DOER->ARGFOUR.HD;
                         TEMPARGT->ARGTWO
  ELSEIF L = "NRSPNSBL" THEN DOER->ARGFOUR.HD;
                             TEMPARGT->ARGTWO
  CLOSE;
  DEST(K)->K->L;
  IF L = "NRSPNSBL" THEN 1397077->TESTATT;
                         IF .ANYRESP THEN "RSPONSBL"->L
                         ELSE GOTO A CLOSE
  ELSEIF L = "RSPONSBL" THEN 1399125->TESTATT;
                             IF .ANYRESP THEN "NRSPNSBL"->L
                             ELSE GOTO A CLOSE
  CLOSE;
B:IF VERBS(I).MEANING = "NIL" THEN GOTO C CLOSE;
  IF VERBS(I).MEANING.FNPROPS.HD = L THEN GOTO D CLOSE;
  I+1->I;
  IF I =< 100 THEN GOTO B CLOSE;
C:IF K.HD = "NIL" THEN N->NEWV;
          IF NOT(N = 0) THEN TAG(N)+1->TAG(N) CLOSE;
                   HOLDNUM->NUM
               EXIT;
  GOTO A;
D:VERBS(I).PRESUP->PRESUPM;
  I->NUM;
  IF .COMPARE.NOT THEN IF I < 100 THEN I+1->I;
                                       GOTO B
                                  ELSE GOTO C
                       CLOSE
  CLOSE;
  IF TAG(I) = 0 THEN 1->TAG(I);
                     I->NEWV;
                     HOLDNUM->NUM;
                     RETURN
  ELSEIF I = 100 THEN IF TAG(100)<TAG(N) THEN
                                  TAG(100)+1->TAG(100);
                                           100->NEWV;
                                           HOLDNUM->NUM;
                                           RETURN
                                      ELSE GOTO C
                      CLOSE
  ELSEIF N = 0 THEN I->N;
                    I+1->I;
                    GOTO B
  ELSE IF TAG(I)<TAG(N) THEN I->N;
                             I+1->I;
                             GOTO B
       ELSE I+1->I;
            GOTO B
       CLOSE
  CLOSE;
END;
```

```
FUNCTION DISPLAY;
   VARS L;
   1->CFLAG;
   2->MFLAG;
   0->NEGFLAG;
   0->MISTAKES;
   [%VALIDBLE%]->PRESUPL;
   .ISHAPPY;
   0->CFLAG;
   IF NOT(MISTAKES = 0) THEN GOTO A CLOSE;
   TEXT('ON THE CONTRARY, !);
   IF NUMS = 2 THEN TEXT('I !)
               ELSE TEXT('YOU !)
   CLOSE;
   TEXT('DO BELIEVE THAT!);
   PRLIST(ARGTWO);
   NL(2);
   TAG(4)+1->TAG(4);
   RETURN;
A:IF QMARK THEN TEXT('NO, THERE !)
   ELSE TEXT('THERE !) CLOSE;
   IF MISTAKES = 1 THEN TEXT('IS 1 PLACE !)
   ELSE TEXT('ARE!);
        PR(MISTAKES);
        TEXT(' PLACES !)
   CLOSE;
   TEXT('WHERE !);
   IF NUMS = 2 THEN TEXT('I DISAGREE.!)
               ELSE TEXT('YOU ARE INCONSISTENT.!)
   CLOSE;
   NL(1);
   SP(2);
   IF NUMS = 2 THEN TEXT('MY !)
               ELSE TEXT('YOUR !)
   CLOSE;
   TEXT('COGNITIVE STRUCTURE IS THE FOLLOWING:-!);
   NL(1);
D:IF OFFENDER.HD = "NIL" THEN NL(1) EXIT;
   DEST(OFFENDER)->OFFENDER->L;
   POPVAL([%L,":","GOON",";"%])->L;
   SP(5);
   L.FNPROPS.TL.HD.PR;
   16.CHAROUT;
   24.CHAROUT;
   L.FNPROPS.TL.TL.HD->L;
   IF L = "ARGFOUR" THEN "ARG"->L
   ELSEIF L = "ARGTHREE" THEN "ARGTWO"->L CLOSE;
   POPVAL([%L,":","GOON",";"%]);
   .PRLIST;
   16.CHAROUT;
   25.CHAROUT;
   NL(1);
   GOTO D;
END;
```

```
FUNCTION CLARIFY;
   VARS L SL;
   .FINDNEWV;
   IF NEWV = 0 THEN IF NEGFLAG.NOT THEN GOTO E
                        ELSE .DISPLAY
                        EXIT
   CLOSE;
   S->L;
   NL(1);
   IF QMARK THEN TEXT('NO, !)
   ELSEIF NUMS = 2 THEN TEXT('FOR MY PART, !) CLOSE;
   IF NUMS = 2 THEN TEXT('I!)
   ELSE TEXT('YOU!)
   CLOSE;
   IF L.HD = "JOHN" OR L.HD = "PIP" OR L.HD = "DO" THEN
   ELSE PR(L.HD) CLOSE;
   VERBS(NEWV).MEANING.FNPROPS.TL.TL.HD->SL;
   IF SL = "ARGFOUR" THEN "ARG"->SL
   ELSEIF SL = "ARGTHREE" THEN "ARGTWO"->SL CLOSE;
   POPVAL([%SL,";","GOON",";"%])->SL;
   PR(VERBS(NEWV).PERFORM);
   IF DOERBLE.NULL THEN IF DOER.NULL THEN
                           ELSEIF INCLUDES(DOER,SL) THEN
                           ELSE PRLIST(DOER)
                           CLOSE
   ELSEIF INCLUDES(DOERBLE,SL) THEN
   ELSE PRLIST(DOERBLE)
   CLOSE;
   IF DOERBLE.NULL.NOT THEN .RETAIN CLOSE;
   IF VERBS(NEWV).DELIMIT = "NOTHING" THEN
   ELSE PR(VERBS(NEWV).DELIMIT) CLOSE;
   PRLIST(SL);
   NL(2);
   RETURN;
E:1->CFLAG;
   2->MFLAG;
   0->MISTAKES;
   [%NVALIDBF%]->PRESUPL;
   .ISHAPPY;
   0->CFLAG;
   IF NOT(MISTAKES = 0) THEN 1->NEGFLAG;
                             0->MISTAKES;
                             .ANSWER
                        EXIT;
   NL(1);
   IF NUMS = 2 THEN TEXT('I !)
             ELSE TEXT('YOU !)
   CLOSE;
   TEXT('DO NOT BELIEVE THAT!);
   PRLIST(ARGTWO);
   NL(2);
   TAG(4)+1->TAG(4);
END;
```

```
FUNCTION ANSWER;
   VARS I L M;
   IF MISTAKES > 0 THEN .CLARIFY EXIT;
   S->L;
   NL(1);
X:IF QMARK.NOT THEN GOTO Y
   ELSEIF NEGFLAG THEN TEXT('NO, !)
   ELSE TEXT('YES, !)
   CLOSE;
Y:IF NUMS = 2 THEN TEXT('I!)
   ELSE TEXT('YOU!)
   CLOSE;
   IF NEGFLAG.NOT THEN GOTO P CLOSE;
   IF L.HD = "JOHN" OR L.HD = "PIP" THEN TEXT(' DO!)
   ELSE PR(L.HD) CLOSE;
   TEXT(' NOT!);
   GOTO R;
P:IF L.HD = "JOHN" OR L.HD = "PIP" OR L.HD = "DO" THEN GOTO R
   ELSE PR(L.HD) CLOSE;
R:PR(VB);
   VERBS(NUM).DELIMIT->I;
   IF I = "NOTHING" THEN VB->I CLOSE;
A:DEST(L)->L->M;
   IF M = "NIL" THEN TEXT('PROGRAM ERROR!);
                     NL(2)
                  EXIT;
   IF NOT(M = I) THEN GOTO A CLOSE;
   IF DOER NULL THEN
   ELSEIF INCLUDES(DOER,L) THEN
   ELSE PRLIST(DOER)
   CLOSE;
   IF NOT(I = VB) THEN PR(I) CLOSE;
   PRLIST(L);
   NL(2);
END;
```

```
FUNCTION REPLY;
  VARS L;
  2->NUMS;
  0->HASBLF;
  0->MFLAG;
  .WHATPRSP;
  NL(1);
  .ISHAPPY;
  VERBS(NUM).MEANING.FNPROPS.HD->L;
  IF HASBLF AND MISTAKES = 0 THEN GOTO H
  ELSEIF HASBLF THEN GOTO K
  CLOSE;
F:IF L = "GOOD" THEN TEXT('GOOD FOR ');
                     PRLIST(DOER);
                     NL(2);
                     RETURN
  ELSEIF L = "BAD" THEN TEXT('THAT IS WRONG OF ');
                        PRLIST(DOER);
                        NL(2);
                        RETURN
  ELSEIF L = "RSPONSEL" THEN GOTO G
  ELSE TEXT('THAT IS INTERESTING');
       NL(2)
  EXIT;
G:.WHATPRSP;
  IF MEMBER(GOOD,PRESUPL) THEN "GOOD"->L
  ELSEIF MEMBER(BAD,PRESUPL) THEN "BAD"->L
  ELSE "NO"->L
  CLOSE;
  GOTO F;
H:IF L = "WANTS" THEN TEXT('I WANT THAT TOO.');
                      NL(2);
                      RETURN
  ELSE TEXT('I AGREE. ');
       GOTO F
  CLOSE;
K:TEXT('I DO NOT AGREE. ');
  .CLARIFY;
END;
```

```
[VALIDBLF NVALIDBF ARGTHREE]->VALIDBLF.FNPROPS;
[NVALIDBF VALIDBLF ARGTHREE]->NVALIDBF.FNPROPS;
[VALID NVALID ARGONE]->VALID.FNPROPS;
[NVALID VALID ARGONE]->NVALID.FNPROPS;
[GOOD BAD ARG]->GOOD.FNPROPS;
[BAD GOOD ARG]->BAD.FNPROPS;
[CONFDENT NCONF ARGTWO]->CONFDENT.FNPROPS;
[NCONF CONFDENT ARGTWO]->NCONF.FNPROPS;
[FUTACT NFUTACT ARG]->FUTACT.FNPROPS;
[NFUTACT FUTACT ARG]->NFUTACT.FNPROPS;
[WANTS NWANTS ARGTWO]->WANTS.FNPROPS;
[NWANTS WANTS ARGTWO]->NWANTS.FNPROPS;
[RSPONSBL NRSPNSBL ARGFOUR]->RSPONSBL.FNPROPS;
[NRSPNSBL RSPONSBL ARGFOUR]->NRSPNSBL.FNPROPS;
[VALIDBFH NVALDBFH ARGTWO]->VALIDBFH.FNPROPS;
[NVALDBFH VALIDBFH ARGTWO]->NVALDBFH.FNPROPS;
[WANTH NWANTH ARGTWO]->WANTH.FNPROPS;
[NWANTH WANTH ARGTWO]->NWANTH.FNPROPS;
[PASTUTT NPASTUTT ARG]->PASTUTT.FNPROPS;
[NPASTUTT PASTUTT ARG]->NPASTUTT.FNPROPS;
[DUMMY DUMMY ARG]->DUMMY.FNPROPS;
```

```
FUNCTION CHAT;
  VARS Q;
  .REMEMBER;
  POPVAL([%".","THOUGHT",";","GOON",";"%]);
  PIPTHINK->SPEAKERL(2).BELIEFS;
  NL(6);
  TEXT('WHO ARE YOU?!);
  NL(2);
  .ITEMREAD->Q;
  CONSWLD(Q,"NIL")->SPEAKERL(3);
  NL(1);
  TEXT('I AM READY NOW!);
  NL(2);
A:.PARSE;
  IF S.HD = "END" THEN GOTO B
  ELSEIF FINISH = "FINISH" THEN GOTO B
  ELSEIF FINISH = "AGAIN" THEN "CARRYON"->FINISH;
                                      GOTO A
  ELSEIF VB = "IMPLIES" THEN .RETAIN;
                                  GOTO A
                             CLOSE;
  0->MFLAG;
  0->MISTAKES;
  0->HASBLF;
  NIL->OFFENDER;
  .ISHAPPY;
  .RETAIN;
  IF QMARK AND HASBLF THEN .ANSWER
  ELSEIF QMARK AND NUMS = 2 THEN NL(1);
            TEXT('I DO NOT KNOW ANYTHING ABOUT THAT!);
     :                NL(2)
  ELSEIF QMARK THEN NL(1);
            TEXT('I DO NOT KNOW WHAT YOU THINK OF THAT!);
                      NL(2)
  ELSEIF MISTAKES = 0 THEN .REPLY
  ELSE 0->MISTAKES;
       NL(1)
  CLOSE;
  NIL->OOFRBLF;
  GOTO A;
B:.MEMORY;
  .FILEOUT;
  NL(2);
  TEXT('THE ACTUAL NUMBER OF COMPUTER WORDS!);
  TEXT(' USED ON THIS RUN WAS:-!);
  NL(2);
  PR(COREUSED);
  NL(3);
  .KILL;
END;


COMMENT     A CALL OF .CHAT WILL RUN THE PROGRAM;
```

In addition to the above listings, the following changes must be made to enable Program 2 to run properly:-

1. In <u>REMEMBER</u> (see page 191)

```
SETTAG(100);
```

must be included after line 6.

2. In <u>IMPLIES</u> (see page 205), in line 30, the number 11 must be replaced by the number 4.

3. In <u>KEEP</u> (see page 215), lines 8 - 16 must be relaced by

```
ADDRESSEE -> SPEAKER;
IF NUMS = 2 THEN 3 -> NUMS
ELSE 2 -> NUMS
CLOSE;
```

4. In <u>RELEASE</u> (see page 215)

```
IF NUMS = 2 THEN 3 -> NUMS
ELSE 2 -> NUMS
CLOSE;
```

must be inserted after line 4.

SAMPLE RUN OF PROGRAM 2

```
ASTON UNIVERSITY G3 7.7A  9SEP74
11.07.13← LOGIN MOPJNDF,:SSP0428
TYPE PASSWORD←
STARTED :SSP0428,MOPJNDF, 9SEP74  11.07.49 TYPE:MOP
11.08.03← PIPTALK




WHO ARE YOU?

← JOHN

I AM READY NOW

← I BELIEVE THAT I AM INNOCENT.

I DO NOT AGREE.
FOR MY PART, I QUESTION WHETHER JOHN IS INNOCENT

← DO YOU DOUBT WHETHER I AM INNOCENT*

I DO NOT BELIEVE THAT JOHN IS INNOCENT

← I WANT TO HONOUR THE QUEEN.

I WANT THAT TOO.

← I CREDIT CHARLES WITH HONOURING THE QUEEN.

I DO NOT AGREE.
I DO NOT CREDIT CHARLES WITH HONOURIN THE QUEEN

← DO YOU DISAPPROVE OF HONOURING THE QUEEN*

NO. I APPROVE OF HONOURIN THE QUEEN

← DO YOU BELIEVE THAT CHARLES DID NOT HONOUR THE QUEEN*

I DO NOT KNOW ANYTHING ABOUT THAT

← WOULD YOU NOT MAINTAIN THAT CHARLES KILLED THE POSTMAN*

NO. I WOULD NOT MAINTAIN THAT CHARLES KILLED THE POSTMAN

← YOU WOULD CLAIM THAT CHARLES KILLED THE POSTMAN*

NO. I DISBELIE THAT CHARLES KILLED THE POSTMAN

← DO YOU THEREFORE REFUSE TO ACQUIT CHARLES*

NO. I INTEND TO ACQUIT CHARLES
```

← I PROCLAIM THAT CASSIUS CLAY IS THE GREATEST.

THAT IS INTERESTING

← CASSIUS CLAY IS THE GREATEST IMPLIES
← MUHAMMED ALI IS THE GREATEST.

O.K.

← WOULD I DENY THAT MUHAMMED ALI IS THE GREATEST*

NO. YOU WOULD SWEAR THAT MUHAMMED ALI IS THE GREATEST

← I SUGGEST THAT I AM INNOCENT.

A PRESUPPOSITION OF SUGGEST IS:-
NPASTUTT *([JOHN IS INNOCENT])
THIS IS INCONSISTENT WITH A PREVIOUS FACT

← I IMAGINE THAT RATIONALISM IS CORRECT.

I AGREE. THAT IS INTERESTING

← WOULD YOU THEN STATE THAT MAN IS BORN WITH AN EMPTY BRAIN*

NO. I WOULD QUESTION WHETHER MAN IS BORN WITH AN EMPTY BRAIN

← I WANT SUPPORT FOR THE TORIES.

I DO NOT AGREE.
I DO NOT WANT SUPPORT FOR THE TORIES

← I DESPISE YOU FOR LISTENING TO ME.

THAT IS WRONG OF PIP

← DO I APPROVE OF LISTENING TO JOHN*

NO. YOU DISAPPRO OF LISTENIN TO JOHN

← WOULD I ACCUSE JOAN OF STARTING A WAR*

I DO NOT KNOW WHAT YOU THINK OF THAT

← I DO BLAME JOAN FOR STARTING A WAR.

THAT IS WRONG OF JOAN

← STARTING A WAR IMPLIES FIGHTING.

O.K.

←

← YOU INTEND TO SUPPORT THE LIBERAL PARTY.

I CAN MAKE UP MY OWN MIND THANK YOU

← YOU INTEND TO SUPPORT THE TORIES*

NO. I REFUSE TO SUPPORT THE TORIES

← YOU DO INTEND TO SUPPORT THE LIBERAL PARTY*

YES. I INTEND TO SUPPORT THE LIBERAL PARTY

← I APPROVE OF FIGHTING.

THE MEANING OF APPROVE IS:-
GOOD [FIGHTING]
THIS IS INCONSISTENT WITH A PREVIOUS ATTITUDE OF YOURS. JOHN

← WOULD YOU PRAISE FRED FOR HONOURING THE QUEEN*

YES. I WOULD PRAISE FRED FOR HONOURIN THE QUEEN

← I CONGRATULATE THE ASTON COMPUTER CENTRE
← ON LETTING ME COMPLETE THIS THESIS.

GOOD FOR THE ASTON COMPUTER CENTRE

← END.

THE ACTUAL NUMBER OF COMPUTER WORDS USED ON THIS RUN WAS:-

13650

11.47.30← LOGOUT
MAXIMUM ONLINE BS USED 34 KWORDS
11.47.40 2.04 FINISHED :SSP0428,MOPJNDF : 2 LISTFILES
JOB COST= 125P:UNITS LEFT 17165P