American Journal of Computational Linguistics *Microfiche 25*

NATURAL LANGUAGE AS A SPECIAL CASE OF PROGRAMMING LANGUAGES

Geoffrey Sampson

Department of Linguistics
University of Lancaster
Lancaster LA1-4YT
England

SUMMARY

I offer a tentative answer to a question posed by Leo Apostel: 'what type of automata would produce and use structures such as natural languages [possess]'?

Noam Chomsky has pointed out that natural languages share certain common structural characteristics, and he argues that these linguistic universals have implications for our understanding of human mental processes. In my _The Form of Language_ (1975), I suggest that we should develop a model of human mental machinery by designing an abstract automaton which accepts programs having the range of structures universally found in the semantic analyses of sentences of natural languages. This article makes concrete proposals about such an automaton.

An automaton is defined by specifying a set of states, a set of acceptable programs, an input function mapping pairs of program and prior state into new states, and a successor-state relation which permits the automaton to move spontaneously from one state to another, either deterministically or non-deterministically. In an automata-theoretic model of human mental processes, sentences will play the part of programs, automaton-states will correspond to structures of knowledge or belief, the input function will specify how a person's belief-structure is altered by the sentences he hears or reads, and the successor-state relation will correspond to the rules of inference by which one derives new beliefs from the beliefs one already has.

A computer is a physical realization of an automaton; but an automaton modelling the behaviour of users of natural language will certainly be very different from the automata which

and concentrate on semantic features which appear to be constant for all natural languages.)

The model I propose turns out, as an unexpected bonus, to offer satisfying solutions to a number of controversial points of philosophical logic. On the other hand, it remains to be seen whether the model can successfully be extended beyond the subset of natural language it now covers; I conclude by listing some unsolved problems.

TABLE OF CONTENTS

1. This article proposes a tentative answer to a question posed by Apostel (1971: 22): 'what type of automata would produce and use structures such as natural languages [possess]'?[1] Chomsky has pointed out that natural languages show common structural characteristics: each natural language is derived transformationally from a context-free phrase-structure language.[2] Chomsky (e.g. 1968, and cf. Lenneberg 1967) argues that this shows that we have innate psychological machinery for processing language. I have suggested (Sampson 1972a, 1975a: ch. 8) that a fruitful way to construct a theory of such psychological machinery will be to view the relation between sentence and hearer as analogous to that between computer program and computer. Here I wish to offer some concrete proposals about the psychological machinery involved in the comprehension of natural language, based on comparing the structure of natural language with that of actual computer programming languages in practical use.[3]

[1]I insert 'possess', since I prefer to speak of languages _having_ structures rather than _being_ structures. I discuss Apostel's own comments on this question elsewhere (Sampson 1974).

[2]I show elsewhere (Sampson 1973b) that this is an empirical hypothesis, despite the findings of Peters & Ritchie and others that any recursively enumerable language can be generated by some transformational grammar.

[3]The theory to be presented here is somewhat comparable with that of Winograd (1972), although constructed independently. By comparison with Winograd I am less interested in the practical problems of communicating with an automaton in idiomatic, 'surface-structure' English, and more interested in what characteristics of the human language-processing automaton are suggested by those features of English which appear to be universal.

2. It is usual to distinguish the terms *automaton* and *computer*: an automaton is a mathematical abstraction of a certain kind, while a computer is a physical object designed to embody the properties of a particular automaton (cf. Putnam [1960] 1961: 147), as an ink line on a sheet of graph paper is designed to embody the properties of a continuous function; thus e.g. a computer but not an automaton, may break down, as a graph, but not a function, may be smudged. Naturally, though, the only automata for which there exist corresponding computers are automata which it is both possible and useful to realize physically; so the class of computers represents a rather narrow subset of the class of automata as defined below. We shall sometimes speak of 'computers' meaning 'automata of the class to which actual computers correspond'; category mistakes need not bother us if we are alert to their dangers.

We may define an automaton $\mathcal{A}$ as a quadruple ($\mathcal{S}$, $\mathcal{L}$, *Int*, *Suc*), in which $\mathcal{S}$ is a (finite or infinite) set of *states*, $\mathcal{L}$ is a (finite or infinite) language (i.e. set of strings of symbols), *Int* is a partial function from $\mathcal{S} \times \mathcal{L}$ (the Cartesian product of $\mathcal{S}$ with $\mathcal{L}$) into $\mathcal{S}$ (the *input function*), and *Suc* is a relation on $\mathcal{S}$, i.e. a subset of $\mathcal{S} \times \mathcal{S}$ (the *successor-state relation*). $\mathcal{L}$ is called the *machine language* of $\mathcal{A}$; a member of $\mathcal{L}$ is a *program*.

We treat the flow of time as a succession of discrete instants (corresponding to cycles of actual computers). Between any adjacent pair of instants, the automaton is in some state $S \in \mathcal{S}$. At any given instant, a program may be input. If the automaton is in S and $L \in \mathcal{L}$ is input, the automaton moves to the state *Int*(S, L); if (S, L) $\notin$ dom(*Int*), we say that L is *undefined* for S (and no change of state occurs). If no program is input, the automaton moves to some state S'

such that S Suc S', provided there is such a state S'. (Otherwise, no change of state occurs, and S is called a stopping state.) If Suc is a (partial) function (i.e. if for each S there is at most one state S' such that S Suc S'), the automaton is deterministic.

An ordinary digital computer is a deterministic automaton whose states are realized as different distributions of electrical charge (representing the digits 0 and 1) over the ferrite cores in a store together with a set of working registers and an address counter. The number of states of such an automaton is finite but very large: a simple computer with a store containing 4096 words of 16 bits together with a single working register would have on the order of 5×10^{19736} states. The programs of the machine language of such an automaton will consist of sequences of machine words not exceeding the size of the store, and thus the machine language will again be finite. The input of such a program containing, say, n words will cause the automaton to load these words into the first n places in it store, replacing the current contents, and to set the address counter to 1. The successor-state function is determined by the number in the address-counter together with the code translating machine words into instructions; whenever the counter contains the number i the automaton changes its state by executing the instruction in the ith place in store and incrementing the counter by one. A proper subset of the automaton's states are stopping states: whenever the storage word indicated by the address counter is not the code of any instruction, the machine stops.

For any state S of a deterministic automaton, we may use the term succession of S for the sequence of states the automaton will pass through under the control of its successor-state function, beginning with $\underline{S}$ and ending (if the succession is finite) at a stopping-state. A computer is arranged so that, on entering certain states, it performs certain output actions (e.g. it prints a symbolic representation of part of its internal state onto paper). The art of programming such a computer consists of finding an input program which moves the computer into a state, the succession of which causes the computer to perform actions constituting a solution to the programmer's problem, while being finite and as short as possible.

3. A natural language, such as English, is specified syntactically and semantically by defining a set $\mathcal{L}$ constituting the sentences of the language together with a subset $\vdash$ of $\mathcal{L}^* \times \mathcal{L}$ (where '$\mathcal{L}^*$' denotes the power set of $\mathcal{L}$), such that $\{\underline{L}_1, \underline{L}_2, \ldots, \underline{L}_{\underline{n}}\} \vdash \underline{L}_0$ iff $\underline{L}_0$ is implied by the premisses $\underline{L}_1, \underline{L}_2, \ldots, \underline{L}_{\underline{n}}$ ($\underline{n} \geqslant 0$, $\underline{L}_{\underline{i}} \in \mathcal{L}$ whenever $0 \leqslant \underline{i} \leqslant \underline{n}$). (In the limiting case, the null set $\emptyset \vdash \underline{L}_0$ iff $\underline{L}_0$ is analytic.) In practice, the infinitely numerous members of $\mathcal{L}$ are generated by a finite set of context-free phrase-structure rules, together with syntactic transformations which operate on the structures defined by those rules. The infinitely numerous members of $\vdash$ will be defined by a specification of a relation between the sentences of $\mathcal{L}$ and a set of arrays of symbols called 'semantic representations' of those sentences, together with a finite set of rules of inference, similar to those of extant formal logics, which permit the construction of a derivation containing the semantic representation of $\underline{L}_0$ as conclusion and the semantic representations of $\underline{L}_1, \underline{L}_2, \ldots, \underline{L}_{\underline{n}}$ as premisses just when $\{\underline{L}_1, \underline{L}_2, \ldots, \underline{L}_{\underline{n}}\} \vdash \underline{L}_0$.[4] The 'generative semantic-

[4] 'The' semantic representation(s), on the assumption that

ists' have argued that the relation between sentences and their semantic representations is defined by the transformational rules revealed by independently-motivated syntactic analysis (e.g. Postal [1970] 1971: 252f.); although this hypothesis is certainly not altogether correct (see e.g. Partee 1971), it seems likely that the semantic representation of a sentence is some simple function of its syntactic deep structure and its surface structure. The rules of inference for natural languages will no doubt exhibit the 'structure-dependence' characteristic of syntactic transformations, as do the rules of inference of formal logics, cf. Sampson (1975a: 163-7, forthcoming) (thus, the term '$\underline{X} \supset \underline{Y}$' in the standard rule of modus ponens, i.e. '$\{\underline{X} \supset \underline{Y}, \underline{X}\} \rightarrow \underline{Y}$', is a structural description not of all formulae containing an instance of '$\supset$' but only of those in which '$\supset$' is an immediate constituent of the whole formula). For discussion of the philosophical problems involved in this way of describing natural-language semantic analysis, including problems relating to the analytic/synthetic distinction, cf. Sampson (1970, 1973a, 1975a: ch. 7).

4. It is tempting to view the mind of a speaker of e.g. English as an automaton in the defined sense, with the sentences of English as the programs of its machine language, and the rules of inference of English determining the successor-state relation. In other words, some component of the mind of an English-speaker would be a device capable of entering any

---

the relation between sentences and semantic representations is a function. In practice it will not be (ambiguous sentences will have more than one semantic representation), so 'the' should read 'one of the ... (respectively)'.

one of a (perhaps infinitely) large number of discrete states; hearing (or reading) a sentence would move this device from one state to another in accordance with definite rules; and other rules related to the rules of inference of English would govern how it passes through different states when not immediately reacting to speech (i.e. when the owner of the mind is thinking).

Although the analogy is tempting, extant computers and their machine languages are not promising as sources for a theory of the relation between human minds and natural languages. The machine language sketched above is not at all reminiscent of natural languages. The latter typically contain infinitely many sentences, only the simplest of which are used in practice; the machine language of §2 contains an enormous but finite number of programs, and the programs which are useful in practice (those which compute important functions) are not typically 'simple' in any obvious sense.

Fortunately, the machine languages of the various extant computers are not the only artificial programming languages in use. Partly for the very reason that machine languages are so different from natural languages, most programs are written not in machine languages but in so-called 'high-level' programming languages, such as FORTRAN, SNOBOL, APL, PL/1 (to name a few among many). We may think of a computer $\mathcal{A}_{\underline{M}}$ supplied with a compiler program for some high-level language $\mathcal{L}_{\underline{H}}$ as simulating the workings of a very different computer, say $\mathcal{A}_{\underline{H}}$. No such computer as $\mathcal{A}_{\underline{H}}$ actually exists: high-level languages are not typically the machine languages of any physical computers, and there are undoubtedly sound engineering reasons for this. But the abstract automaton $\mathcal{A}_{\underline{H}}$ may be described just as precisely as the automaton $\mathcal{A}_{\underline{M}}$ which underlies the real computer. One who programs an $\mathcal{A}_H$ 'system' (i.e. conjunction of computer with $\mathcal{L}_H$-compiler) commonly thinks of the machine he is dealing with as having the properties of $\mathcal{A}_H$, and may be quite unaware of the properties of the machine $\mathcal{A}_M$ with which he is in fact interacting.

High-level languages, and the abstract automata whose 'machine languages' they are, differ from one another in more interesting ways than do real computers and their machine languages; and furthermore (not surprisingly, since high-level languages are designed to be easily usable by human programmers) they are much more comparable with human languages than are real machine languages. (Typically, a high-level programming language is a context-free phrase-structure language, for instance.) I shall suggest that the relationship between high-level languages and their corresponding automata gives us much better clues about human mental machinery than does that between real computers and machine languages.

5. Let me first give an example of a high-level language: I shall choose the language APL (see e.g. Iverson 1962, Pakin 1968). APL is interesting for our purposes because it is _particularly_ high-level: i.e. it is related more distantly to machine languages of real computers, and more closely to human languages, than many other high-level languages. It is a _real-time_ rather than batch-processing language, which means that it is designed to be used in such a way that the result of inputting a program will normally be crucially dependent on the prior state of the system (in a batch-processing language, programs are designed to be unaffected by those remains of the prior state which survive their input): this is appropriate for an analogy with human language, since presum-

ably the effect on a person of hearing a sentence depends in general on his prior system of knowledge and belief.[5]

The complete language APL includes many features which are irrelevant to our analogy. For instance, there is a large amount of apparatus for making and breaking contact with the system, and the like; we shall ignore this, just as we shall ignore the fact that in human speech the effect of an utterance on a person depends among other things on whether the person is awake.[6] Also, APL provides what amounts to a method of using the language to alter itself by adding new vocabulary; to discuss this would again complicate the issues we are interested in.[7] We shall assume that programmer and system are permanently in contact with one another, and we shall restrict our attention to a subset of APL to be defined below: rather than resorting to a subscript to distinguish the restricted language from APL in its full complexity, we shall understand 'APL' to mean the subset of APL under consideration.

[5]The practising computer user may find my definition of the real-time/batch-processing distinction idiosyncratic; the difference I describe is the only one relevant for our present purposes, but it is far from the most salient difference in practice.

[6]In APL terms, we ignore all system instructions, i.e. words beginning with). Note that we use wavy underlining (corresponding to bold type in print) to quote symbols or sequences of symbols from an object-language, whether this is an artificial language such as APL or a natural language such as English.

[7]In APL terms, we ignore the definition mode and the use of characters ∇∆:→.

6. We begin by defining the set $\mathcal{S}_{APL}$ of states of $\mathcal{A}_{APL}$. First, we recursively define a set of *APL-properties*:

any positive or negative real number is a *numeric APL-property* of dimension ∅;

any character on the APL keyboard (i.e. any of a finite set of characters whose identity does not concern us) is a *literal APL-property* of dimension ∅;

for any integer *n* and integer-string *D*, any *n*-length string over the set of numeric (literal) APL-properties of dimension *D* is a numeric (literal) APL-property of dimension *D^n*.[8]

For any finite string *D* over the integers, any numeric or literal APL-property of dimension *D* is an APL-property, and nothing else is such. Clearly, there are infinitely many APL-properties. The length of the dimension of an APL-property is the *rank* of that APL-property. Thus, a number is a rank-0 numeric APL-property; a four-letter word, e.g. *LOVE*, is a rank-1 literal APL-property; a 2 by 3 matrix of numbers,

e.g. $\begin{matrix} 3.9 & 2 & 12 \\ 0 & -4.6 & 937 \end{matrix}$, is a rank-2 numeric APL-property, etc.

An *APL-identifier* is any rank-1 literal string beginning

[8] The symbol '^' stands for concatenation (a dimension is always an integer-string). Concatenation is a function from sets of strings, so we should strictly write '*D*^*N*' (where *N* is the length-one string over {*n*}) rather than '*D*^*n*'. For any set *S* and any integer *n* ≥ 0, we use the terms '*n-tuple* of elements of *S*' and '*length-n string* over *S*' interchangeably for any function from the segment {1, 2, ..., *n*} of the natural numbers into *S*; note that the null set ∅ is therefore the length-0 string over any set.

with an alphabetic character: there are therefore infinitely many APL-identifiers. We define Ident as the set including all APL-identifiers together with an entity, assumed to be distinct from all the APL-identifiers, denoted by the symbol □. An APL-object is a pairing of any member of Ident with an APL-property; we call the first member of an APL-object the identifier of the object and the second member the property of the object.

An APL-state is a finite set of APL-objects in which no distinct objects bear the same identifier. (We may thus think of an APL-state as a function from a finite subset of Ident into the set of APL-properties.) We write '$\mathcal{S}_{APL}$' for the set of all APL-states: clearly, $\mathcal{S}_{APL}$ is infinite.

We now define the language $\mathcal{L}_{APL}$ of $\mathcal{A}_{APL}$. $\mathcal{L}_{APL}$ is generated by the context-free grammar on p. 16. The initial symbol of that grammar is asst (for 'assignment'). Since capital letters occur among the terminal symbols of $\mathcal{L}_{APL}$, we use miniscules as non-terminals; terminal symbols of $\mathcal{L}_{APL}$ are wavy-underlined (cf. note 6, p. 13), whether they are letters or other characters.

asst → { id | ⎕ } ← dscr

dscr → {
- id
- numname
- litname
- deic
- mf dscr
- dscr df dscr
- dscr tf dscr dscr
- asst

}

id → [any APL-identifier]

numname → [any number or string of numbers, denoting the corresponding rank-0 or rank-1 numeric APL-property][9]

litname → [any character or string of characters between inverted commas, denoting the corresponding rank-0 or rank-1 literal APL-property][10]

mf → [any of a large finite set of symbols or symbol-strings denoting partial monadic functions on the set of APL-properties][11]

df → [any of a large finite set of symbol(-string)s denoting partial dyadic functions on the set of APL-properties]

tf → [any of a finite set of symbol(-string)s denoting partial triadic functions on the set of APL-properties][12]

deic → [any of a small finite set of symbol-strings denoting total monadic functions from the set of possible programming-acts into the set of APL-properties][13]

[9]In practice one cannot write a length-0 string, and one cannot distinguish a length-1 string from a rank-0 property; but

The sentences of $\mathcal{L}_{APL}$ are the strings defined by the above grammar, disambiguated by the use of round brackets (with association to the right where not indicated by bracketing). The sequence of symbols ⎕← may optionally be deleted when initial in a sentence.[14] Clearly there are infinitely many sentences in $\mathcal{L}_{APL}$. A sentence of $\mathcal{L}_{APL}$ is an APL-program.

We now go on to specify the function Int_{APL} from $\mathcal{S}_{APL} \times \mathcal{L}_{APL}$ into $\mathcal{S}_{APL}$ which specifies the change of APL-state brought about by a given APL-program.

To determine the new state arrived at from an arbitrary

I ignore these practical complications for the sake of simplicity.

[10] I ignore complications relating to strings containing the inverted comma character.

[11] Some of these functions, and their names, are common to all 'dialects' of APL: e.g. !, which denotes the function taking integers into their factorials, strings of integers into the corresponding strings of factorials, etc., and which is undefined e.g. for literal APL-properties. The facility of 'user-definition' (cf. note 7) permits a programmer to alter APL by adding new functions.

[12] APL contains no triadic functions other than user-defined ones.

[13] E.g. I20 denotes the function taking any programming act into a string of integers representing the time of day at which it occurs.

[14] There are a number of syntactic complications, akin to syntactic transformations in natural languages, concerning a dyadic function called index, which is denoted by square brackets; we ignore these complications, and shall not consider 'index' apart from the other dyadic functions.

current state on input of an arbitrary program, we consider the phrase-marker of which that program is the terminal string. Beginning at the leaves and working towards the root, and evaluating the rightmost node whenever there is a choice, we associate each dscr node with an APL-property as its denotation and each asst node with a change to be made to the current APL-state: the new APL-state is the one that results from the old state by making all the changes associated with the various asst nodes in the order mentioned, terminating with the change associated with the root asst node (which may of course be the only one). If at any point a dscr node cannot be assigned a denotation (e.g. because it is realized as an APL-identifier which is not the identifier of any object in the current state), the state-changes already made (if any) are the total changes achieved by that program.

The rules for evaluating nodes are as follows:

A dscr node realized as an identifier denotes the APL-property, if any, paired with that identifier in the current state;

a dscr node rewritten as numname or litname denotes the APL-property denoted by its 'exponent' (i.e. the terminal material it dominates);

a dscr node rewritten as deic denotes the APL-property given by applying the function denoted by its exponent to the current programming-act;

when a dscr node d_0 dominates an mf node d_1 followed by a dscr node d_2, if d_1 denotes some monadic function f and d_2 denotes an APL-property p then d_0 denotes the APL-property $f(p)$, provided $p \in \mathrm{dom}(f)$; the extension to dscr nodes rewritten dscr df dscr and dscr tf dscr dscr is obvious.

An asst node dominating a member i of Ident, followed by $\underset{\sim}{\leftarrow}$, followed by a dscr node denoting an APL-property p, adds a new APL-object (i, p) to the current state, destroying any object with the identifier i which may already exist in the current state; if i is $\underset{\sim}{\square}$, a representation of p is printed out;[15]

a dscr node immediately dominating an asst node which has created an APL-object with the property p denotes p.

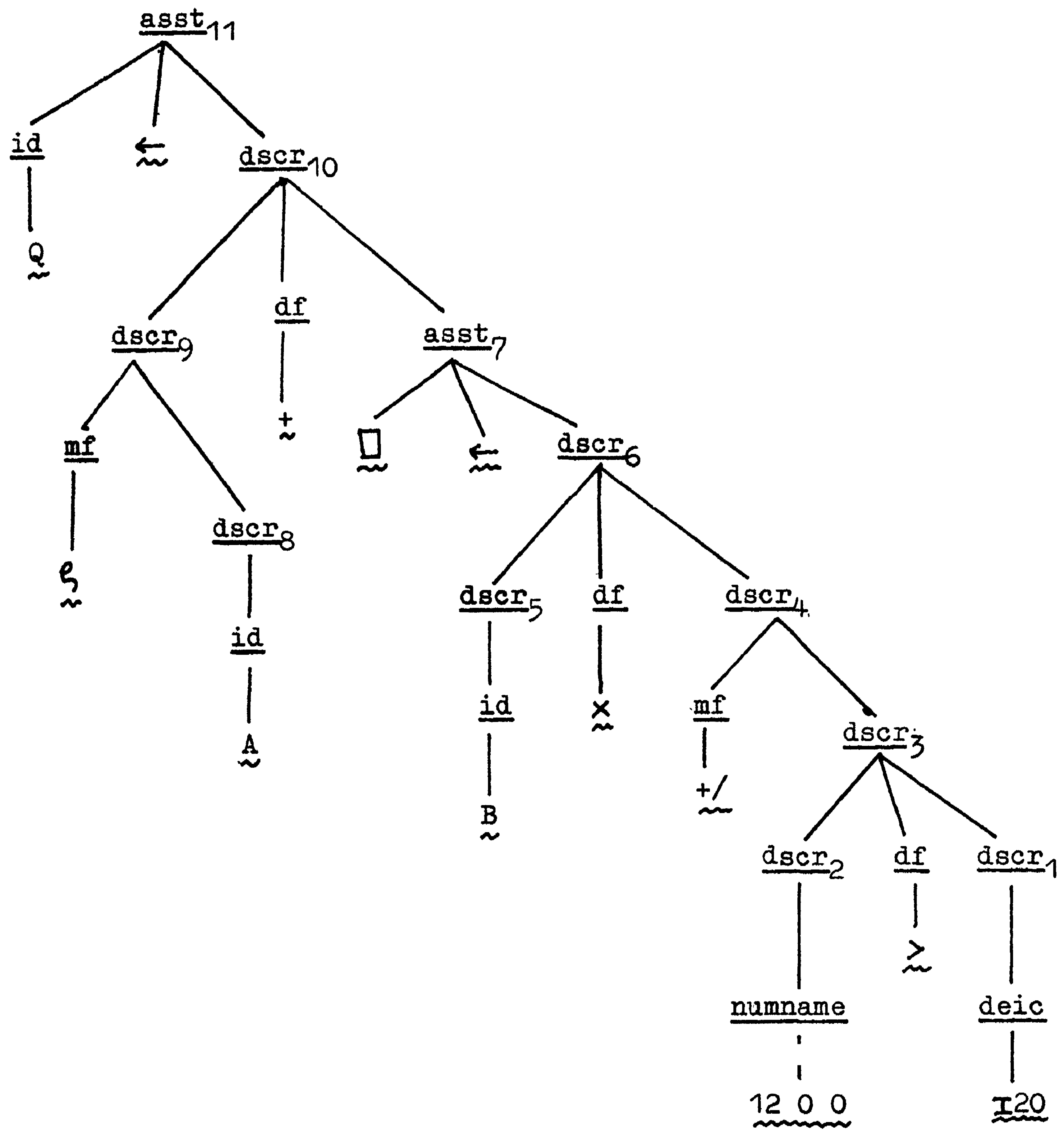
As a (trivial) example of the operation of these rules, consider the program

$$\underset{\sim}{Q \leftarrow (\rho A) + \square \leftarrow B \times +/\ 12\ 0\ 0 > \iota 20}$$

input to $\mathcal{A}_{APL}$ in state $\{(\underset{\sim}{A}, 1\ 1\ 1), (\underset{\sim}{B}, 10)\}$. This program has the constituent structure which appears in Fig. 1 on the next page (in which dscr and asst nodes are numbered in the order they are to be evaluated).

Suppose the program is input in the morning, say at 11.30 a.m. Then $dscr_1$ will denote the string 11 30 0. The function $\underset{\sim}{>}$ takes (12 0 0, 11 30 0) into 1 0 0, which becomes the denotation of $dscr_3$ -- in fact $dscr_3$ will denote 1 0 0 whenever the program is input in the morning and 0 0 0 whenever it is input in the afternoon (when the hour integer will be 13 or more). The monadic function $\underset{\sim}{+/}$ adds the numbers in a string, so if $dscr_3$ denotes 1 0 0 then $dscr_4$ denotes 1. $Dscr_5$ denotes 10 (identified by $\underset{\sim}{B}$), so $dscr_6$ also denotes 10. Accordingly,

[15]In the full version of APL, $\underset{\sim}{\square}$ can occur as a rewrite of dscr, in which case dscr is assigned an APL-property input by the programmer at the time dscr is evaluated by the system. We ignore this, since it interferes with the analogy with natural language. In the full version it is also possible to output symbol-strings which do not represent individual APL-properties; again we ignore this.


asst11
id
Q
dscr10
dscr9
df
+
asst7
mf
dscr8
id
A
dscr6
dscr5
df
×
dscr4
id
B
mf
+/
dscr3
dscr2
df
>
dscr1
numname
12 0 0
deic
ι20

(1)

$\underline{asst}_7$ adds an object ($\underset{\sim}{\Box}$, 10) to the current state and prints out $\underset{\sim}{10}$. The monadic function ρ gives the dimension of any APL-property, and $\underline{dscr}_8$ denotes 1 $\tilde{1}$ 1, so $\underline{dscr}_9$ denotes 3, and hence $\underline{dscr}_{10}$ denotes 13. Finally, $\underline{asst}_{11}$ adds an object ($\underset{\sim}{Q}$, 13) to the current state.

In other words, if the program is input in the morning and the prior state is as quoted, the final state will be {($\underset{\sim}{A}$, 1 1 1), ($\underset{\sim}{B}$, 10), ($\underset{\sim}{\Box}$, 10), ($\underset{\sim}{Q}$, 13)}, whereas if it were input in the afternoon to $\mathcal{A}_{APL}$ in the same prior state the resulting state would be {($\underset{\sim}{A}$, 1 1 1), ($\underset{\sim}{B}$, 10), ($\underset{\sim}{\Box}$, 0), ($\underset{\sim}{Q}$, 3)}.

To define $\mathcal{A}_{APL}$ fully it remains only to specify the relation $\underline{Suc}_{APL}$ on $\mathcal{S}_{APL}$ which controls the changes-of-state $\mathcal{A}_{APL}$ undergoes spontaneously. $\underline{Suc}_{APL}$ is the empty relation: every APL-state is a stopping state. A programmer working in APL has no wish for the system to take any actions beyond those specified by his programs: by defining monadic, dyadic, or triadic functions of any complexity he wishes, he can get the answers to his questions simply by carrying out the state-changes specified in his program. (In the machine language of a genuine computer, on the other hand, the state-changes brought about by programs are of no intrinsic interest, and the input of a program is of value only in that it brings the computer to a state from which it proceeds spontaneously to perform actions useful to its programmer.)

7. It may seem contradictory to say that a real digital computer, which will have only finitely many states and possible programs, can be made to simulate an automaton such as $\mathcal{A}_{APL}$ which has infinitely many states and programs. And, of course, in practice the simulation is not perfect. Although

an APL-state may contain any number of objects, for any APL computer/compiler system there will be a finite limit on the number of objects in a state; although any real number may be an APL-property, in a practical APL system real numbers are approximated to a finite tolerance. The situation is quite analogous to the case of natural language, where the individual's 'performance' is an imperfect realization of an ideal 'competence', in one sense of that distinction; just as in linguistics, so in the case of high-level programming languages it is normal to give a description of the ideal system separately from a statement of the limitations on the realization of that system in practice, which will differ from one person to another in the natural language case, from one computer/compiler pair to another in the programming language case.

Other high-level programming languages differ from APL not only in terms of their sentences but in terms of the nature of the states on which their sentences act. Thus in states of e.g. SNOBOL, all objects are character strings; in PL/1, objects include not only arrays of the APL kind but also trees, trees of arrays, arrays of trees, etc. Space does not permit a survey of the differences between high-level languages with respect to the nature of their states.

8. At this point we are ready to begin to answer Apostel's question, about what sort of automata natural languages are appropriate programming languages for. Any answer to such a novel question must obviously be very speculative; but the ideas that follow seem plausible enough to be worth consideration. We do not know with any certainty even what the semantic representations or syntactic deep structures of our sent-

ences are; but we have seen that there is good reason to think the two may be similar, and we can make more or less detailed conjectures about their form. In my exposition I shall make various assumptions about semantic representations, some of which have already been made for independent reasons by other scholars. Insofar as my theory depends on these assumptions, a refutation of the latter refutes the theory -- this is one of the respects in which my theory is falsifiable, i.e. scientific.

I shall present my theory in a relatively informal, intuitive way to begin with, and formalize it more carefully later.

What we are looking for is a specification of a set $\mathcal{S}_{Eng}$ of states, which we can interpret as states of some subpart of the mind of an English-speaker, such that semantic representations of English sentences are rather natural devices for moving this part of the mind from one of its states to another. It will be convenient to have some name for that part of a human's total psychological make-up which is described by specifying $\mathcal{S}_{Eng}$. In earlier, unpublished work I have called this the <u>topicon</u> (coined on the analogy of 'lexicon'), since I envisage it as containing a set of entities corresponding to the objects of which its owner is aware, and to which he can therefore take a definite description to refer. $\mathcal{S}_{Eng}$, then, is to be a set of possible topicon-states. The sets of topicon-states available to speakers of natural languages other than English will differ from $\mathcal{S}_{Eng}$ (cf. 17 below), but not in respect of the properties on which this paper will concentrate.

Note that a topicon-state is certainly not to be equated with a 'state of mind' or 'psychological state': a topicon is claimed to be only one small part of a human's mental machinery, and there will be many ways in which the latter can vary --

e.g. the human may be happy or sad, asleep or awake -- without implying any difference in topicon-state.

Just as an APL-state contains a set of APL-objects with properties drawn from a fixed class, so a topicon-state will contain a set of objects I shall call referents.[16] Suppose some person P knows of the existence of a red car C; then P's topicon will include a referent c corresponding to C. The referent c will be P's 'Idea' of C, in Geach's terms (1957). The possible properties for referents will be determined by the vocabulary of P's language, in this case English: each lexical item of English will correspond to a referent-property. I shall use Geach's operator '§( )' (1957: 52) to form names of referent-properties from lexical items: if P knows that C is a red car then c will have the properties §(red) and §(car). (An element of a mental state cannot be red, but it can be §(red).)

P's topicon will include not only referents representing physical objects but referents for any entities of which P is aware and which he can take definite descriptions (referential NPs) to denote: there will be referents for characters in fiction, for abstractions like the centre of this circle, etc. etc. But, at any given time, P's topicon will contain only a finite number of referents. Given enough time, of course, there is no limit to the number of objects whose existence P could deduce or imagine; and I shall suggest that for P to

[16]Here and below, rather than coining neologisms I use terms having established usages in philosophy and logic in senses which clash with their normal use; in such cases I use the term only in the sense I define.

deduce or imagine the existence of some entity B is for P's topicon to acquire a new referent representing B. But deduction and imagination take time: in a finite amount of time P's topicon will have acquired only finitely many referents.

9. Consider sentences (2a) and (2b) addressed to P (and let us simplify things initially by supposing that P does not previously know of any red cars -- we shall consider the more general case in §10):

(2a) *I bought a red car yesterday.*
(2b) *I sold the red car today.*

The NP *a red car* in (2a) will create a referent with the properties §(*red*) and §(*car*) in P's topicon. On the other hand, when he hears (2b) the NP *the red car* will pick out the referent which *a red car* has already created (in order to act on it in ways which will be discussed later). In other words, the distinction between *the red car* and *a red car* is quite parallel to the distinction between dscr and asst constituents, respectively, in APL: the former selects an object from the current state, the latter adds an object to the current state. Let us call natural-language expressions which act in the former way 'identifying expressions' (IEs), and expressions which act in the latter way 'establishing expressions' (EEs). Clearly the IE/EE distinction is related to the traditional distinction between 'definite' and 'indefinite' NPs. However, I do not claim that all definite and indefinite NPs count as IEs or EEs respectively. Consider, for instance, the de dicto / de re (or opaque reference / transparent reference) ambiguity exhibited by NPs in 'intensional contexts' (see e.g. Quine 1960: §30;

Hintikka 1973), e.g.:

(3) *I am looking for an elephant.*

In the de dicto sense of *an elephant*, (3) does not imply that there are any elephants, although in the de re sense it implies that there is at least one. Only in the de re sense is *an elephant* in (3) an EE in my terms, though syntactically *an elephant* is clearly an 'indefinite NP' in both cases. On the other hand, in sentences containing quantifiers, definite NPs may not always be IEs: thus, in (4) the definite NP *the dustjacket* does not denote a particular object (and therefore, perhaps, does not pick out a particular one of the hearer's referents). just as the indefinite NP *a book* does not seem to establish the existence of a single book:

(4) *Whenever I buy a book, I remove the dustjacket.*

I shall develop my theory with a view to handling the subset of English which excludes quantification and intensional contexts, and in which 'definite' and 'indefinite' NPs do coincide with IEs and EEs respectively. Later I shall consider some of the aspects of English which my theory does not handle successfully as it stands.

Even in the subset of English considered here, not all definite NPs will in fact refer to referents already in the hearer's topicon. For instance, a child may come home from school at the beginning of a new term and say to his mother:

(5) *I saw the new teacher today.*

It may be that the mother does not know that there is a new teacher, i.e. it may be that her topicon contains no referent for which the phrase *the new teacher* is appropriate; but in practice she is likely to work out from her child's sentence that there is a new teacher, and to understand that the child saw him. In other words, in this case the phrase *the new teacher* acts as an EE to create a new referent in the mother's topicon. However, it seems plausible to say that this is in some sense not the central use of a phrase such as *the new teacher*; it would be more appropriate, if the mother does not know about the new teacher, for the child to say something like:

(6) *There is a new teacher at school and I saw him today.*

in which the referent denoting the teacher is first introduced by an EE and only then re-identified by an IE. Notice that the mother may react to (5) by saying something like:

(7) *What new teacher? I didn't know there was one.*

which would not be a possible reaction to a sentence using the EE *a new teacher*.

What happens when the mother successfully acquires a new referent in response to (5), I suggest, is that she imagines some circumstance in which *the new teacher* would succeed in picking out a referent in her topicon -- for instance, if there were a new teacher at her child's school -- and, in imagining these circumstances, creates the referent; after which the sentence operates on her topicon in the normal way. The APL system does not work like this: if one inputs the sentence *A ← B* to an APL-state lacking an object named *B*, the system prints out a message pointing out one's error but does not change state. It is natural enough, though, that human linguistic behaviour shows more initiative than the behaviour of artificial automata. A programmer has complete control over the automaton he programs, and it is easier to require the programmer to get his programs right than to equip the automaton with routines to guess what the programmer means by defective sentences. A human speaker, on the other hand, has no way of knowing exactly what state his hearer's topicon is in, so it is all to the good if the hearer can compensate in simple cases for defects in the speaker's sentences.

Since I shall frequently be speaking of the relations between linguistic expressions, topicon-referents, and the entities in the outside world which the linguistic expressions denote, let me lay down some terminological conventions. I shall use denotation for the relation between an IE and the thing which a hearer takes that IE to correspond to; my theory asserts that denotation is a composition of two relations, a relation of reference between linguistic expressions and topicon referents, and a relation of representation between topicon-referents and things. Thus, if the phrase *your car* said to *P* now picks out a referent r_1 in *P*'s topicon, and if *P* owns exactly one car *C*, then your car refers to r_1 and denotes *C*, and r_1 represents *C*. We say that r_1 is the referent of, and *C* the denotatum of, *your car* (on this occasion).

Notice that an IE may refer, without denoting: if *P* has read Crime and Punishment, then his topicon will contain two referents, say r_2 and r_3, such that *Raskolnikov* refers to r_2 and *Alëna Ivanovna* refers to r_3, even though neither of these NPs denotes anything (and, correspondingly, r_2 and r_3 will both have the property §(*fictional*)). Furthermore, identity of the denotata of two IEs does not imply identity of their referents. Thus, if P knows that I have exactly one brother and that he is the new doctor, then the IEs the new doctor and Sampson's brother will refer to the same referent in P's topicon, and hence also denote the same man (the details of reference by means of the genitive construction are discussed in §12 below); but if P knows that I have one brother and that there is a new doctor, but does not realize that they are the same man, then the two IEs will refer to different referents in P's topicon, even though each of these referents will in fact represent the same man.

10. So far we have assumed that, when a hearer hears an IE such as the red car, his topicon contains only one referent with the properties §(red) and §(car). Clearly this will not in general (or even usually) be so: when one hears an IE, it will often be the case that one knows of a number of objects fitting the description.[17] One who hears the red car will take the phrase to refer to one among the various red cars of which he is aware which is in some way closer than the others to the

[17] If Russell's theory of descriptions (Russell [1905] 1949: 105; Whitehead & Russell 1927: 30) were an accurate semantic description of English (which Russell did not, of course, claim it to be -- cf. his [1957] 1969: 335-7), then most English sentences uttered in practice would be simply false because they contain IEs asserting the uniqueness of objects fitting descriptions which in fact are multiply instantiated. Philosophers who have discussed reference have treated it as a simple relation between expressions and things, rather than as the composite relation for which I argue; but they have succeeded in this only by devoting undue attention to NPs, such as Socrates or the author of 'Waverley', which perhaps have only one possible denotatum, but which seem to be quite rare in practice.

focus of his attention. This will translate into our theory as the notion that the referents in a topicon are arrayed in some kind of space, one point of which constitutes the focus of attention at any given time. The nature of this space, and the factors which determine the position of the referents and focus of attention in it, will be considered in §16 below; for the moment, let us simply assume that the notion can be made precise. Then we can say that any IE consisting of the word *the* followed by a series $\underline{w}_1\ \underline{w}_2\ \ldots\ \underline{w}_n$ of adjectives and noun will refer to the nearest referent to the hearer's focus of attention having all the properties $\S(\underline{w}_1)$, $\S(\underline{w}_2)$, ..., and $\S(\underline{w}_n)$. Thus, *the car* will refer to the nearest §(*car*) referent to the focus, while *the red car* will refer to the nearest referent to the hearer's focus which is both §(*red*) and §(*car*).[18] One would expect that the nearest referent of all to the focus in any topicon-state should be referred to as *the*; in English a syntactic rule replaces *the* as a complete NP by *he*, *she*, or *it*.

11. In APL, objects can be referred to by their identifiers. The obvious candidates as natural-language equivalents of identifiers are proper names. However, although some logicians have discussed proper names, under the label singular terms, as if

[18] I can offer no explanation of the syntactic distinction between nouns and adjectives, which serves no obvious semantic function; however, since the distinction appears to be universal in natural languages, my account of English semantic representations will incorporate it. (The solution to this puzzle may have to do with the fact that some adjectives are 'syncategorematic' in a way which nouns never are: a 'good actor' is not necessarily good though he is necessarily an actor.)

they are the equivalent of APL identifiers (for a summary of the alternative views, see Cheng 1968), English proper names in fact do not behave in this way. In APL, a state in which two distinct objects bear the same identifier is simply not a well-formed state. In English, on the other hand, locutions like:

(8) *Do you mean our Charles or your Charles?*

(9) *The London in England is bigger than the London in Ontario.*

occur frequently enough: although many proper nouns apply only to one referent in an average topicon, many apply to more than one. Superficially, proper nouns seem syntactically distinct from common nouns in that IEs containing proper nouns lack *the*: *the car*, but not **the London*. However, Sloat (1969) has argued convincingly that in deep syntactic structure proper nouns are preceded by *the*, and that proper and common nouns are syntactically quite parallel in the base. We shall take it that proper nouns correspond to properties for referents in just the same way as common nouns: *London* refers to the nearest §(*London*) referent to the hearer's focus, as *the car* refers to the nearest §(*car*) referent. (The problems of how the pairs of IEs in (8) and (9) succeed in referring to distinct referents will be answered in §12 and §15 respectively.) Clearly there is a distinction between names and common nouns in that the applicability of a name to an object is more 'arbitrary' than that of a common noun. But this distinction is gradient rather than all-or-none; e.g. a schoolboy's nickname, such as *Fatty*, will be intermediate in arbitrariness (a boy called *Fatty* will probably be fat, but not all fat boys will be called

Fatty).

12. A somewhat more complicated situation arises in connection with IEs involving genitive constructions. The 'basic' sense of the genitive is commonly taken to be possession, as in *John's car*; however, the genitive often represents other relationships, as in *John's father*, *John's country*, *John's God*, *the origin of the problem*, *the density of the liquid*, etc. etc. Even in a case where the genitive NP denotes a person and the head NP denotes an inanimate object, such as *John's car*, although on many occasions of use the NP will be paraphrasable as *the car which John owns*, the same NP will surely be used equally frequently in other situations in which the appropriate paraphrases would be *the car which we saw narrowly miss running John down*, *the car which John keeps saying he'd like to buy if he only had the money*, or other expressions of purely idiosyncratic and ephemeral relationships between the denotatum of the genitive NP and that of the head NP.

The device of the topicon space permits a neat account of this situation. In an NP of the form <u>A</u>*'s* <u>B</u> or *the* <u>B</u> *of* <u>A</u> (e.g. *John's car*, *the roof of the house*), <u>A</u> will as usual pick out the nearest referent to the hearer's focus (say $\underline{r}_a$) having the properties corresponding to the lexical items of $\bar{\underline{A}}$, while the NP as a whole will pick out the <u>nearest referent to</u> $\underline{r}_a$ having the properties corresponding to the lexical items of $\bar{\underline{B}}$. Thus in the case of *John's car*, *John* will pick out the nearest referent to the hearer's focus having the property §(*John*) and, if that referent is $\underline{r}_1$, *John's car* will pick out the nearest referent to $\underline{r}_1$ having the property §(*car*). The latter referent need not be the nearest §(*car*) referent to the hearer's focus; if it is not, *the car* and *John's car* will have different reference for him. As we shall see when we discuss the organization of the topicon-space in §16, ownership is only one of the factors that may cause a §(man) referent to be close to a particular §(car) referent in a topicon.

13. Certain English words, known as deictics or token-reflexives, correspond to the terms labelled deic in APL: these include I, you, now, here, etc.[19] Deictics, like other IEs, pick out referents of the hearer's topicon; but their referents depend on characteristics of the speech act in which they are used, and are independent of the arrangement or properties of referents in the hearer's topicon. For this reason, deictics never occur as the head of a genitive construction, and there are no phrases like *the greengrocer's you (with 's as genitive rather than short for is); you will refer to the same referent on a given occasion (namely the referent representing the addressee of the speech act -- the owner of the topicon, unless he is overhearing words addressed to someone else) whatever other referents are in the vicinity, so it would be otiose to modify a deictic with a genitive NP.

14. So far I have discussed only referents corresponding to noun-phrases in syntax, and representing individuals in the outside world. However, some referents will represent what would more normally be called 'facts' or 'events' than 'individuals'. Ordinary predicate logic distinguishes sharply between individuals on the one hand, and facts or events on the other: the former are translated into singular terms, the latter into

[19]Linguists do not usually include the first and second person pronouns among the 'deictics', but logically they are of the same category.

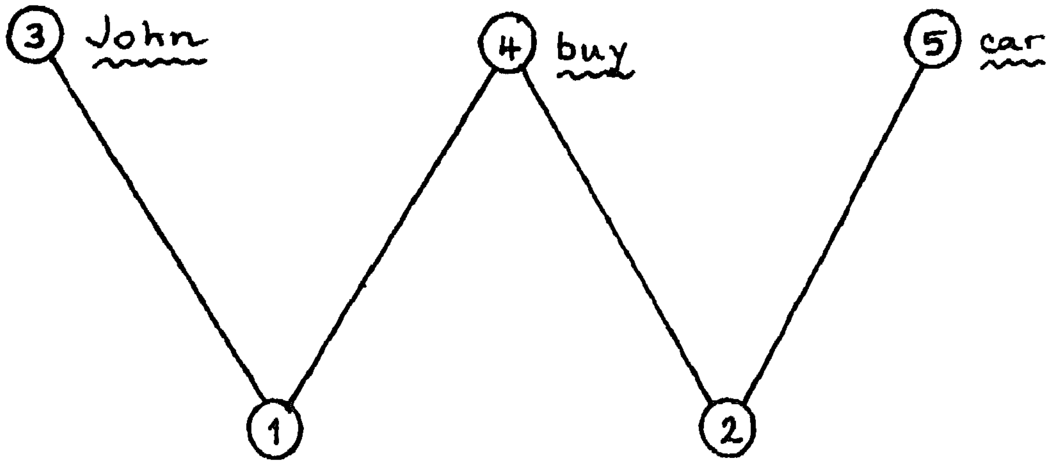
arrays of predicate followed by arguments, and the syntax of the predicate calculus does not permit one to occur in place of the other. However, in English, if e.g. *John bought the car* has the semantic representation '$\underline{f}(\underline{a}, \underline{b})$' (where $\underline{f}$ is the predicate *buy* and $\underline{a}$ and $\underline{b}$ are singular terms standing for John and the car), then presumably the semantic representation of:

(10) *It surprised Mary that John bought the car.*

will have to have '$\underline{f}(\underline{a}, \underline{b})$' as one of the arguments of the predicate *surprise* -- (10) will have to be represented as something like '$g(\underline{f}(\underline{a}, \underline{b}), \underline{c})$', where g is *surprise* and $\underline{c}$ stands for Mary (tense is discussed later in this section).[20] If

[20] Rosenbaum (1967) has shown that in deep syntax, before the application of a transformation called 'Extraposition', (10) has the normal subject-verb-object structure with *that John bought the car* as subject. The need to permit propositions as arguments of predicates is discussed by Leech (1969: 25-6). In the 'semantic representations' given here, I arrange the predicate-symbol to the left of all the arguments, in order to clarify the comparison with standard logical notation. It is by a quite arbitrary choice, however, that formal logic writes '$\underline{f}(\underline{a}, \underline{b})$' rather than '$\underline{a}\ \underline{f}\ \underline{b}$', and when I define $\mathcal{L}_{Eng}$ below I shall adopt the ordering which more closely reflects the surface structure of English. (It is a moot point within linguistics whether the deep structures of English sentences have the ordering subject-verb-object or verb-subject-object.)

facts, as well as things, may be denoted by suitable linguistic expressions, then we may suppose that a topicon contains referents representing facts (_propositional referents_) as well as referents representing things (_individual referents_). We will suppose further that the referents in a topicon are linked in a graph structure in which propositional referents dominate _n_-tuples of (propositional or individual) referents, corresponding to the arguments of the respective propositions. Consider e.g. one who knows that someone called John bought a car: his topicon will contain a structure of the following form:



(11)

In (11) nodes stand for referents, which I shall call '$\underline{r}_1$' '$\underline{r}_2$', etc. (N.B. I shall always use '_r_' for nodes of topicon-states, as opposed to '_d_' for nodes of phrase-markers of sentences). The lowest-level referents are unlabelled, while the higher-level referents are each labelled with an English word.

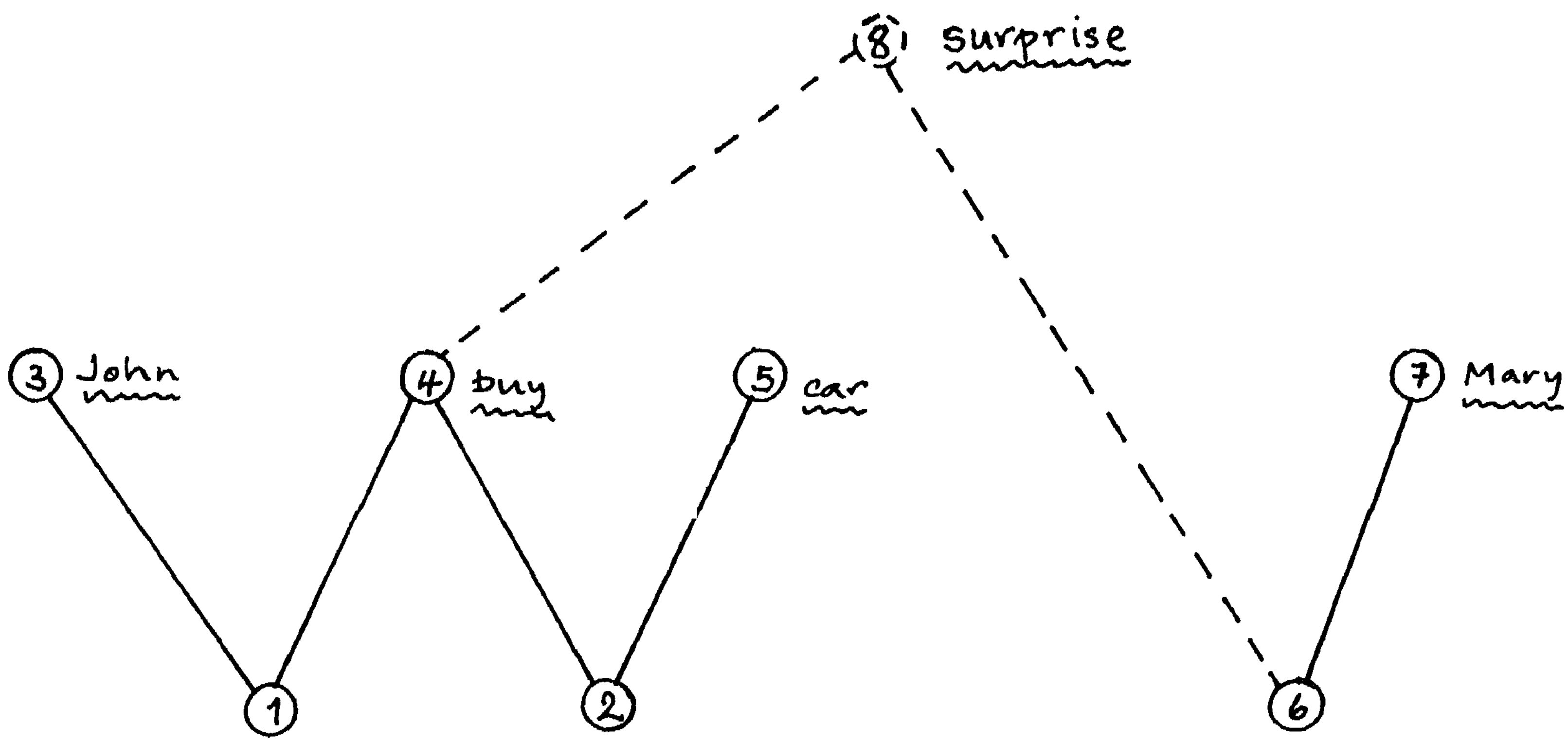
The referents r_1 and r_2 represent respectively John and the car, while r_4 represents the fact that John bought the car. The referent r_5 represents the fact that the thing represented by r_2 is a car, while r_3 represents the fact that the thing represented by r_1 is a John ('is called "John"'. as we usually say in the case of proper names). To say that a referent, say r_2, has the property §(car), is to say that there is some referent, in this case r_5, which dominates the 1-tuple r_2 and which is labelled car.

A sentence acts as an EE for the establishing of propositional referents, as an indefinite NP such as a car is an EE for establishing individual referents. Thus, suppose P's topicon contains the structure of (11), together with a §(Mary) referent, say r_6 (that is, r_6 is dominated by a propositional referent r_7 labelled Mary): then P's hearing (or reading) the sentence (10), i.e. It surprised Mary that John bought the car, will create in P's topicon a new propositional referent, say r_8, labelled surprise and dominating the 2-tuple (r_4, r_6): P's new topicon-state will contain the structure shown as (12) on the next page.

In (12), broken lines show the new structure created by sentence (10). Notice that propositional referents, like individual referents, may be referred to by pronouns; if r_4 is close enough to P's focus of attention, the same effect will be achieved by the sentence:

(13) It surprised Mary.

The number of referents dominated by a given referent in a topicon will correlate with the label of the latter referent. An unlabelled referent will be an individual referent and will



(12)

dominate nothing; a referent labelled with an *n*-adic predicate will dominate an *n*-tuple of referents. Thus a referent labelled with a noun will dominate one referent; a referent labelled with a verb taking subject, direct object, and indirect object will dominate a 3-tuple of referents; and so on. In natural languages the distinctions between the different arguments of a verb are shown sometimes by ordering, sometimes by prepositions (*to John*) or case endings (*Johanni*), etc.

I assume that some individual referents represent points of time, and that one of the arguments of most verbs in natural languages is the time at which the action in question occurred. Where a verb in the preterite occurs with no phrase overtly denoting a point of time, I take it that the nearest time referent to the hearer's current focus of attention becomes the respective argument of the new propositional referent: one would not normally say e.g. John bought the car unless the hearer can be expected to know what occasion one is speaking about. In other words, preterite tense picks out the nearest §(time) referent as he picks out the nearest §(male) referent. McCawley (1971) has argued that preterite tense and pronouns have a common syntactic origin, a finding which renders my semantic approach all the more appealing.[21]

Although I assume time arguments for verbs, to avoid clutter I shall not include them on diagrams.

That clauses may be used to refer to propositional referents either as IEs or as EEs, without the distinction being marked syntactically. Sentence (10) (It surprised Mary that John bought the car) is equally appropriate whether or not the hearer already knows that John bought the car. Thus, if $\underline{P}$'s topicon contains the structure of (11) (p. 35), the phrase that John bought the car in (10) will pick out r_4 and create the extra structure of (12); but if $\underline{P}$'s topicon lacks r_4, then the same phrase will create a referent labelled buy and dominating r_1 and r_2 (and a time referent) before the rest of the

[21]A verb in the perfect, as in John has bought the car, will act as an EE for a time referent, as a verb in the preterite acts as an IE. These remarks might, however, have to be modified to handle American usage: one of the characteristics of American English is that it permits the preterite in circumstances where the perfect would be obligatory in British usage.

sentence creates a node labelled *surprise* dominating this new node, $\underline{r}_6$, and a time referent. The absence of syntactic distinction between phrases establishing propositional referents and phrases identifying them can readily be explained. Either John bought the car at the time in question or he did not; there will never be two referents both labelled *buy* and dominating the same 3-tuple of individual referents, so if P's topicon contains (11) and he hears the phrase *that John bought the car* then he knows that this must refer to $\underline{r}_4$ rather than calling for the creation of a new propositional referent.[22] If there were no distinction between *the car* and *a car*, on the other hand, he would have no way of knowing, on hearing the NP *car*, whether $\underline{r}_2$ or some new §(*car*) referent was intended. I take it that languages lacking definite and indefinite articles mark the IE/EE distinction for individual referents by some other syntactic devices.

15. The graph structure into which an individual referent enters can be used to pick out that referent by means of relative clauses. Thus if *the car* refers to $\underline{r}_2$, then the IE:

(14) *the man who bought the car*

will refer to the nearest referent, say $\underline{r}_x$, to the hearer's focus such that $\underline{r}_x$ has the property §(*man*) and such that some

[22] It is convenient to speak of a topicon's owner as 'knowing' facts about his topicon, just as it is convenient to anthropomorphize a computer program and speak of it 'knowing' various facts; these locutions are, of course, literally nonsense, but they could easily be replaced by longer paraphrases which did not commit category mistakes.

referent labelled buy dominates a 3-tuple including r_x, r_2, and the nearest time referent. If P knows that the denotatum of r_1 is a man (i.e. if his topicon includes, in addition to the structure diagrammed in (12) (p. 37), a referent labelled man and dominating r_1), and if there are no tense problems, then (14), the man who bought the car, will refer to r_1.

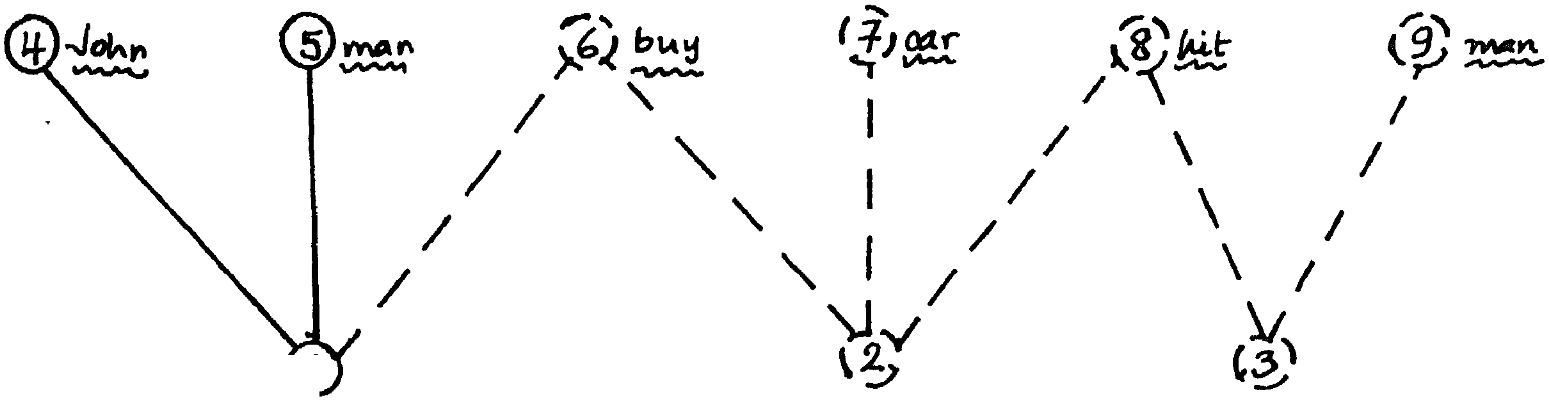
However, note the distinction between restrictive and appositive relative clauses (cf. Bach 1968). A restrictive relative clause, e.g. who bought the car in (14), is part of an IE: it gives a property of the target referent. An appositive relative clause, on the other hand, as in (15), acts as an EE:

(15) The man, who bought the car, is old.

In (15), the man acts as a complete IE; when (15) is input to a topicon, the man will pick out the nearest §(man) referent to the focus (say r_m), and then the appositive relative will create a new referent labelled buy and dominating r_m and the referent of the car, before the main clause creates a referent labelled old dominating r_m. The function of appositive relative clauses in natural languages is thus quite comparable to that of embedded asst clauses in APL.

16. The principle that each sentence received by a hearer creates a new referent in the hearer's topicon suggests a natural way of reconstructing within the theory the notion of a 'focus of attention', which varies with the topics being discussed: we may define the focus of attention as the most recently-created referent at any given time. The graph structure associated with propositional referents offers a way of formal-

izing the notion of distance between referents in the topicon: we may define the distance between any two referents as the minimum number of edges (i.e. lines which link nodes) that must be traversed to get from one referent to the other. Thus, consider the sequence of sentences: (i) *John bought a car.* (ii) *The car hit a man.* (iii) He called *the police.* Assume the hearer's topicon already contains a referent, say r_1, with the properties $\S$(*John*) and $\S$(*man*). After hearing sentences (i) and (ii) but before (iii) the hearer's topicon will include the structure of (16), with the focus at r_8 (the referent created by (ii)):



(16)

(16) contains two referents to which *he* could refer, namely r_1 and r_3; r_3 is one edge from the focus and r_1 is three edges away. Therefore the theory predicts that *he* in (iii) will be taken to refer to r_3 rather than r_1, and this prediction seems

correct: *he* in (iii) will be taken to denote the man who has been hit, rather than John. (Notice that this cannot be predicted from the **situation** described: when a driver hits a pedestrian, the driver is as likely as the pedestrian to call the police.)

17. We may now define the automaton which I claim to represent the mind of a speaker of English. The grammar of the subset of English we are analysing is as follows:

$$\mathrm{S} \rightarrow \left\{ \begin{array}{llll} \mathrm{NP} & \mathrm{Pr}^1 & & \\ \mathrm{NP} & \mathrm{Pr}^2 & \mathrm{NP} & \\ \mathrm{NP} & \mathrm{Pr}^3 & \mathrm{NP} & \mathrm{NP} \\ \vdots & & & \end{array} \right\}$$

$$\mathrm{NP} \rightarrow \left\{ \begin{array}{ll} (\mathrm{NP}) & \mathrm{S} \\ \mathrm{IE} & \\ \textit{wh} & \\ \mathrm{Noun} & \end{array} \right\}$$

$$\mathrm{IE} \rightarrow \left\{ \begin{array}{l} \mathrm{Deic} \\ \textit{the}\ \mathrm{NP}\ (\textit{of}\ \mathrm{IE}) \\ \textit{he},\ \textit{it} \end{array} \right\}$$

$$\begin{array}{lll} \mathrm{Deic} & \rightarrow & \textit{I},\ \textit{you},\ \textit{now},\ \ldots \\ \mathrm{Noun} & \rightarrow & \textit{Mary},\ \textit{man},\ \ldots \\ \mathrm{Pr}^1 & \rightarrow & \textit{red},\ \textit{real},\ \ldots \\ \mathrm{Pr}^2 & \rightarrow & \textit{love},\ \textit{know},\ \ldots \\ \vdots & & \end{array} \qquad (17)$$

The finite set of predicates of English, together with the phonetic shapes of particles such as *the* and *of*, will be specific to the English language. I would hypothesize that in other respects (17) generates the semantic representations of sentences in any natural language, though the rules which relate the phrase-markers generated by (17) to the corresponding surface forms will vary from language to language.

Some of the latter rules which operate in English will be obvious. Thus, subordinate clauses (non-root 'S' constituents) have *that* prefixed to them (replacing *the* in case the latter appears); nouns not preceded by *the* are supplied with *a/an*; '*the* NP *of* IE' may become 'IE'*s* NP' in some cases; *wh* is realized as *who*, *which*, or *that* and is fronted. and *he* is realized as *she* in certain circumstances;[23] adjectives have the verb *be* supplied, or are moved in front of their noun with the relative pronoun *wh* deleted; clauses outside an IE are given commas to mark them as appositive rather than restrictive relative clauses; etc. I shall not attempt to render explicit every detail of the relation between my semantic representations and superficial structures of English sentences.

We may define the set $\mathcal{S}_{Eng}$ of states of the automaton $\mathcal{A}_{Eng}$ as follows. Suppose $\underline{Pred}$ is the finite set of English predicates, i.e. the set $\underline{Pr}^1 \cup \underline{Pr}^2 \cup \ldots$ in (17). Then a pair ($\underline{M}$, $\underline{Foc}$) in which (i) $\underline{M}$ is a semiforest[24] over $\underline{Pred}$ such that

[23]I treat the distinction between *he* and *she* as determined rather than as needing to be marked in the semantic representation: in the standard use of English pronouns (leaving out of account the special rules operating under contrastive stress), *he* is appropriate only if the intended referent is the nearest individual referent of all to the hearer's focus, not merely the nearest of the §(*male*) referents.

[24]I use <u>semiforest</u> as a generalization of the notion <u>tree</u>:

each node immediately dominating a length-$\underline{n}$ string is labelled with an $\underline{n}$-adic predicate and each leaf is unlabelled, and (ii) Foc is a root of $\underline{M}$, is a member of $\mathcal{S}_{Eng}$.

The function $\underline{Int}_{Eng}$ which determines how a sentence of this grammar moves a topicon from one state to another is specified by rules which associate subsets of the referents of the current topicon state with nodes in the structural description of the sentence; as in the APL case, certain nodes cause additions to the current state. We shall write 'Ref' for the partial function, specified by these rules, from nodes of the sentence into subsets of the topicon-referents; in the case where a constituent refers (in our technical sense) to a referent, Ref will take the node dominating the constituent into the unit set containing that referent.

The rules determining $\underline{Int}_{Eng}$ are as follows: (see next page)

---

a semiforest is allowed to have more than one root, and nodes are allowed to branch upwards as well as downwards. A semiforest over a vocabulary $\underline{V}$ is a triple ($\underline{D}$, δ, α) where $\underline{D}$ is a set of nodes, δ is a partial function of immediate dominance from $\underline{D}$ into strings over $\underline{D}$ such that every node is dominated (not necessarily immediately -- dominate is the ancestral of 'immediately dominate') by at least one root (i.e. undominated node), and α is a partial function of labelling from $\underline{D}$ into $\underline{V}$. Nodes outside the domain of δ are leaves, or terminal nodes. Note that, by defining the range of δ as containing strings, I have built left-to-right ordering into my definition; semiforests as defined here are 'stringsemiforests' rather than 'setsemiforests' in the sense of Sampson (forthcoming).

(R1) Whenever two nodes *d*, *d'* of the phrase-marker are such that *d* immediately dominates the length-1 string *d'*, if *Ref*(*d'*) is defined then *Ref*(*d*) = *Ref*(*d'*).

(R2) If IE immediately dominates *the* NP, then *Ref*(IE) is the unit set containing the nearest member of *Ref*(NP) to the current focus.[25] ('Nearest' as defined on p. 41.)

(R3) If IE immediately dominates *the* NP *of* IE' then *Ref*(IE) is the unit set containing the nearest member of *Ref*(NP) to the sole member of *Ref*(IE').

(R4) *Ref*(*he*) is the unit set containing the nearest individual referent to the current focus not having the property §(*inanimate*); *Ref*(*it*) is the unit set containing the nearest (propositional or individual) referent to the current focus not having the property §(*human*).

(R5) If the speech-act being analysed is *A*, then *Ref*(*I*) is the unit set containing the referent representing the performer of *A*, *Ref*(*now*) is the unit set containing the referent representing the time at which *A* occurs, etc.

The remaining rules depend on whether or not a given node is dominated (not necessarily immediately) by an IE node.

(R6) If an S not dominated by IE immediately dominates $NP_1 \; Pr^n \; NP_2 \ldots NP_n$ ($n \geqslant 1$), where Pr^n is realized as *P*, then (see next page):

[25]Strictly, R2 should read: 'If a node *d* labelled IE immediately dominates the length-2 string *d' d"* in which *d'* is labelled *the* ... (etc.)': the abbreviations used here should be self-explanatory. (Cf. also the prime on 'IE' in rule R3, used to distinguish two nodes each labelled IE.)

(i) if NP_0 is the left sister of S and some NP_i ($1 \leq i \leq n$) is realized as **wh**, then *Ref*(NP_i) = *Ref*(NP_0);

(ii) if there is a referent r_0 labelled **P** and immediately dominating $r_1\ r_2 \ldots r_n$, where $\{r_1\}$ = *Ref*(NP_1), $\{r_2\}$ = *Ref*(NP_2), ..., and $\{r_n\}$ = *Ref*(NP_n), then *Ref*(S) = $\{r_0\}$;

(iii) if no such referent as r_0 in (ii) exists, then it is created, and *Ref*(S) = $\{r_0\}$.

(R7) If an S dominated by an IE node immediately dominates $NP_1\ Pr^n\ NP_2 \ldots NP_n$ ($n \geq 1$), where Pr^n is realized as **P**, and if, for some i ($1 \leq i \leq n$), NP_i is realized as **wh**, then *Ref*(S) is the set of all referents r_x such that, for some referent r_0 labelled **P** and immediately dominating some string of referents $r_1\ r_2 \ldots r_n$, $r_j \in$ *Ref*(NP_j) if $1 \leq j \leq n$ and $j \neq i$ and $r_j = r_x$ if $j = i$.

(R8) If NP immediately dominates NP' S, then *Ref*(NP) = *Ref*(NP') if NP is not dominated by an IE node; otherwise *Ref*(NP) = *Ref*(NP') ∩ *Ref*(S).

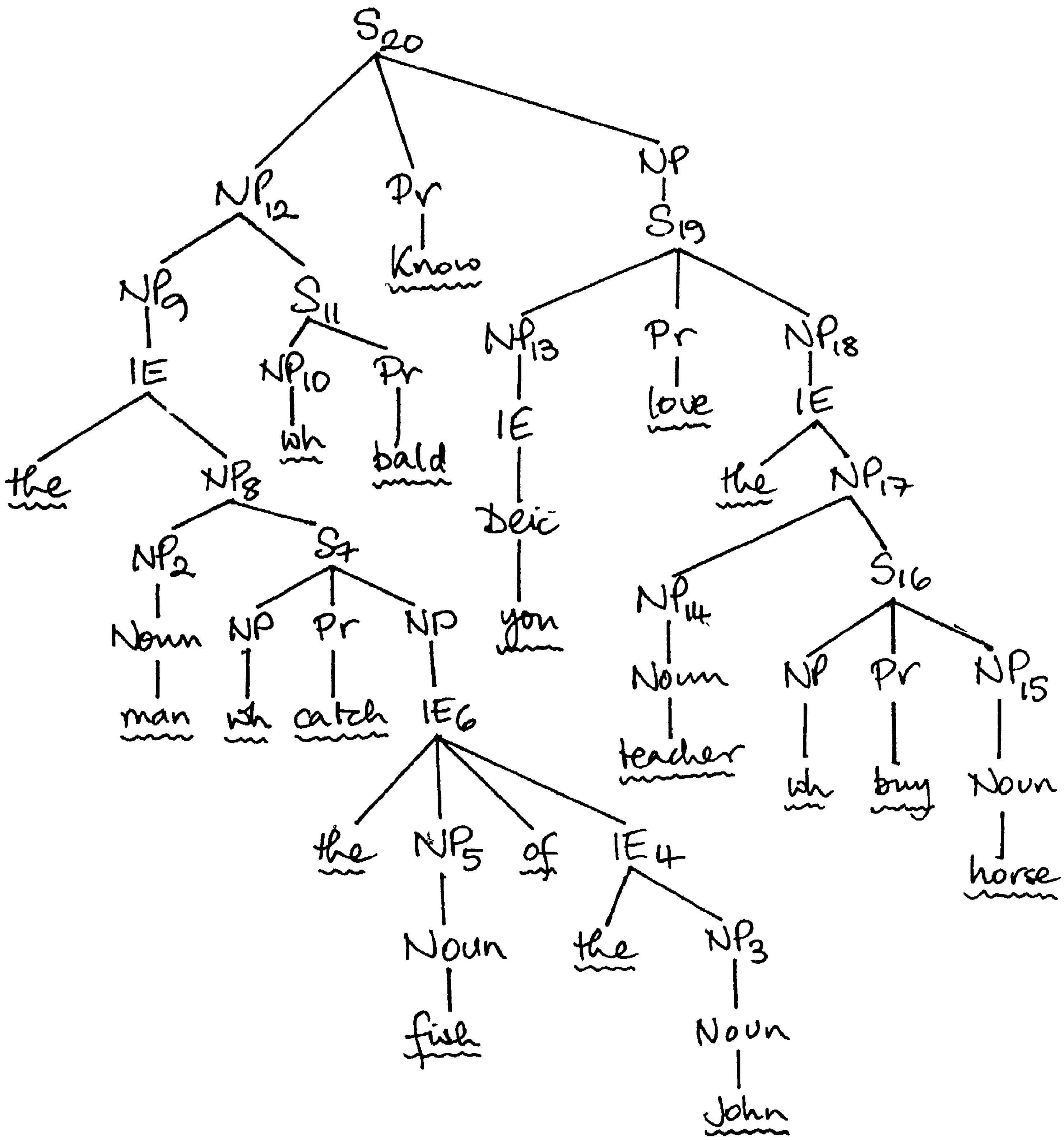
(R9) If a Noun node realized as **N** is not dominated by an IE node then Noun creates an individual referent r and a referent labelled **N** immediately dominating r, and *Ref*(Noun) = $\{r\}$; otherwise *Ref*(Noun) is the set of all §(**N**) referents in the topicon.

(R10) Whenever a new referent is created it becomes the current focus.

To illustrate the operation of $\textit{Int}_{Eng}$, I shall consider a sample sentence and a sample topicon-state. The sentence is:

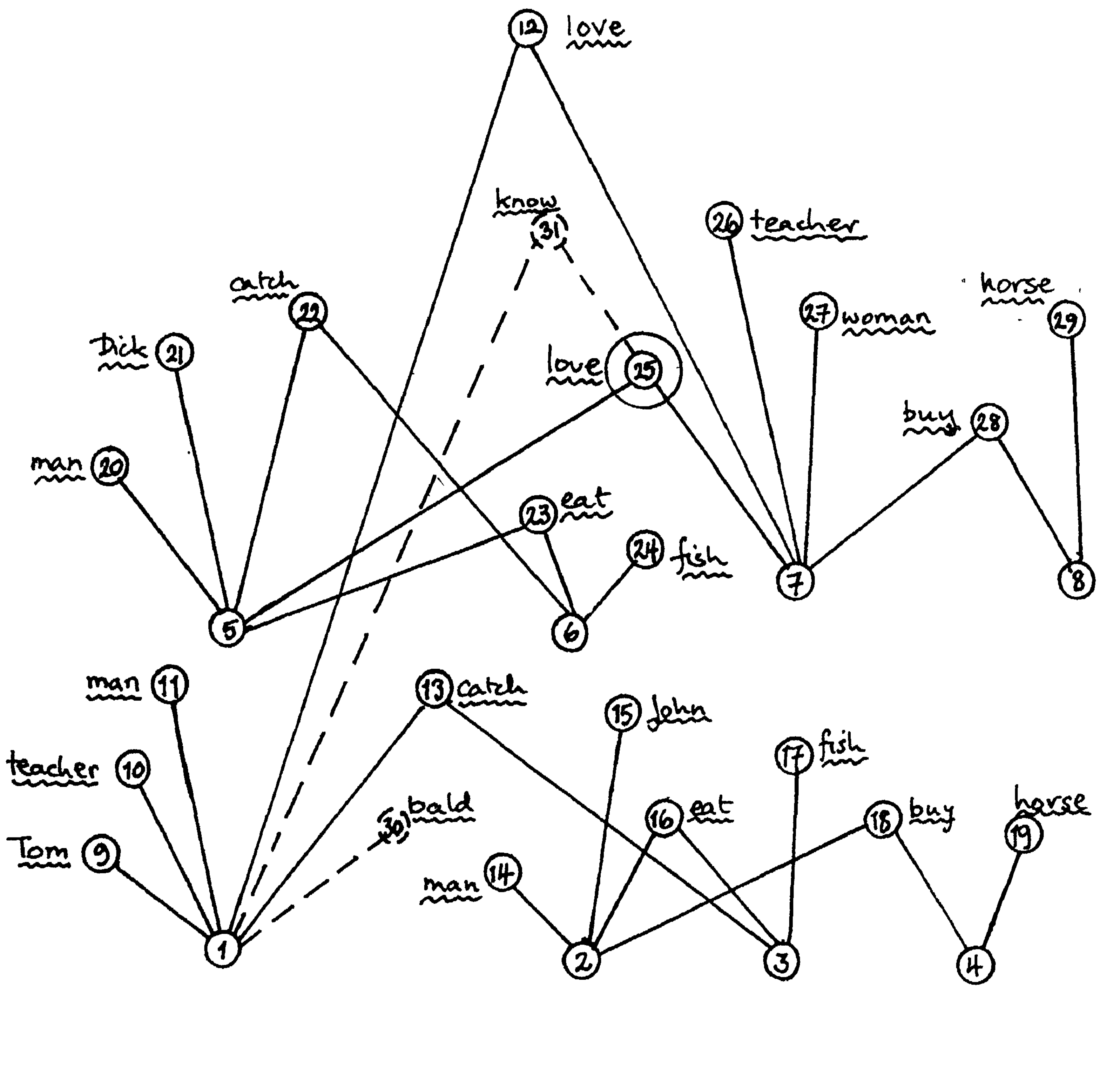
(18) **The man who caught John's fish, who was bald, knew that you love the teacher who bought a horse.**

Sentence (18) has the following semantic representation (as usual I ignore tense for simplicity). I omit the superscripts from 'Pr' nodes, since they are obvious, and I subscript certain nodes for later discussion.



(19)

The topicon state to which (19) is input is assumed to be as in (20) below (without the material drawn in dotted lines), with the current focus at $\underline{r}_{25}$ (indicated by concentric circles):



(20)

The owner of the topicon diagrammed in (20), to whom (19) is addressed, is represented in (20) by r_5: he is a man called Dick who has caught and eaten a fish, and who loves the denotatum of r_7, who is a woman teacher who has bought a horse. The denotatum of r_2 is a man called John who has also bought a horse and has eaten a fish which was caught by the denotatum of r_1, a man teacher called Tom, who loves the same woman as Dick.

We now use rules R1-R10 to interpret the nodes of (19), beginning with the leftmost interpretable leaf (since the material on the left of (19) is what is heard first).

$Noun_1$ is dominated by IE, so by R9 Ref($Noun_1$) = $\{r_1, r_2, r_5\}$; hence by R1 Ref(NP_2) is also $\{r_1, r_2, r_5\}$. Similarly Ref(NP_3) is $\{r_2\}$, so, trivially, by R2 Ref(IE_4) is $\{r_2\}$. Ref(NP_5) is $\{r_3, r_6\}$; by R3, since r_3 is two edges from r_2 while r_6 is eight edges from r_2, Ref(IE_6) is $\{r_3\}$ (*John's fish* denotes the fish that John ate, on this occasion). By R7, Ref(S_7) = $\{r_1\}$ (only Tom caught the denotatum of r_3); so, by R8, Ref(NP_8) = $\{r_1, r_2, r_5\} \cap \{r_1\}$, i.e. $\{r_1\}$, and by R2 and R1 Ref(NP_9) is also $\{r_1\}$. By R6i, Ref(NP_{10}) = $\{r_1\}$. S_{11} is not dominated by an IE node, so by R6iii r_{30} is created with the label *bald*, dominating r_1, and by R10 the focus shifts to r_{30}. By R8, Ref(NP_{12}) = Ref(NP_9) = $\{r_1\}$.

You is a deictic which always refers to the referent representing the addressee, so Ref(NP_{13}) = $\{r_5\}$. Ref(NP_{14}) = $\{r_1, r_7\}$. Ref(NP_{15}) = $\{r_4, r_8\}$; both r_{18} and r_{29} dominate pairs of referents the second member of which belongs to Ref(NP_{14}), so, by R7, Ref(S_{16}) = $\{r_2, r_7\}$, and by R8 Ref(NP_{17}) (and hence Ref(NP_{18})) = $\{r_7\}$. By R6ii, Ref(S_{19}) = $\{r_{25}\}$. Finally, by R6iii, S_{20} creates a referent r_{31} labelled *know* and dominating (r_1, r_{25}), and the new focus is at r_{31}.

Notice that, were it not for the appositive clause *who was bald*, the phrase *that you loved the young teacher* in (18) would be redundant. The initial focus was at r_{25} (the previous sentence had been *You love the young teacher*, say); the sentence *The man who caught John's fish knew it* would serve as well as *The man who caught John's fish knew that you love the young teacher* or *... that you love her* to create the referent r_{31}. However, the appositive clause *who was bald* shifts the focus to r_{30}, so my theory predicts that *it* in the sentence *The man who caught John's fish, who was bald, knew it* will be taken to refer to r_{30} rather than to r_{25} -- i.e. to denote the fact of his baldness rather than that of Dick's loving the teacher. Intuitively this prediction seems correct.

The relation *Suc*$_{Eng}$, which determines which possible next states $\mathcal{A}_{Eng}$ can move to from any given state independently of input, will correspond to the rules of inference in the semantic description of English. Thus, suppose there is a rule 'x *fish* & y *catch* $x \rightarrow x$ *die*' in English (i.e. suppose it is part of the meaning of the words *fish*, *die*, and *catch* that a fish dies if it is caught); then the topicon of (20) will be liable at any time to acquire a referent labelled *die* and dominating r_3 or r_6, since each of these have the property δ(*fish*) and occur as second argument of a referent labelled *catch*.

Clearly, $\mathcal{A}_{Eng}$ will be a non-deterministic automaton: the single rule of inference mentioned permits two alternative successor states for (20). Anyone with experience of constructing deductions in formal logic knows that there are typically a large (though finite) number of ways of continuing a given derivation; similarly, the rules of inference for a natural language will no doubt permit many possible successor-states for any given state. If the process of moving through states under the control of the successor-state relation is to be the reconstruction within the topicon theory of the pretheoretical notion of thinking, this characteristic seems desirable: we do not feel that human thought flows along deterministic channels.[26]

18. Although the effects of most changes of state in the cases of the machine-language discussed in §2 and of APL were confined to the automata themselves, in both cases certain state-changes were associated with action by the automaton on its environment. Thus, whenever an APL-state acquired an object named ⎕, a representation of the property of that object was printed by the system on an output sheet of paper. We may imagine that action is linked to thought in this way also in the human case. Suppose some referent r_0 in a topicon represents the person who owns that topicon; then it might be that whenever, during a sequence of state-changes controlled by the successor-state relation, the topicon acquires a referent labelled assert and dominating r_0 in subject position and some propositional referent r_1 in object position, the owner of the topicon utters a sentence which asserts the proposition represented by r_1. And, supposing r_2 represents some person, say John, if a hit referent is created dominating (r_0, r_2) then

[26] I do not intend this paragraph to imply any position on the determinism/free-will issue. If determinism is correct, then there will presumably be laws deciding which out of the various successor states permitted by the rules of inference of its language a given topicon actually moves into at a given time. Such laws lie outside the scope of this article.

the topicon-owner hits John.[27]

19. There are two obvious problems connected with the notion that the referents in a topicon, which are supposed to correspond to the entities of which the topicon-owner is aware and the propositions he believes, are created by input sentences. The first problem is that no allowance is made for the possibility that speakers are not believed. Thus, if the topicon-owner hears John, the denotatum of r_2, say *I bought a car yesterday*, then according to the rules I have laid down his topicon acquires a §(*car*) referent representing John's new car. But in practice, obviously the topicon-owner may choose to disbelieve John; what happens to his topicon in this case?

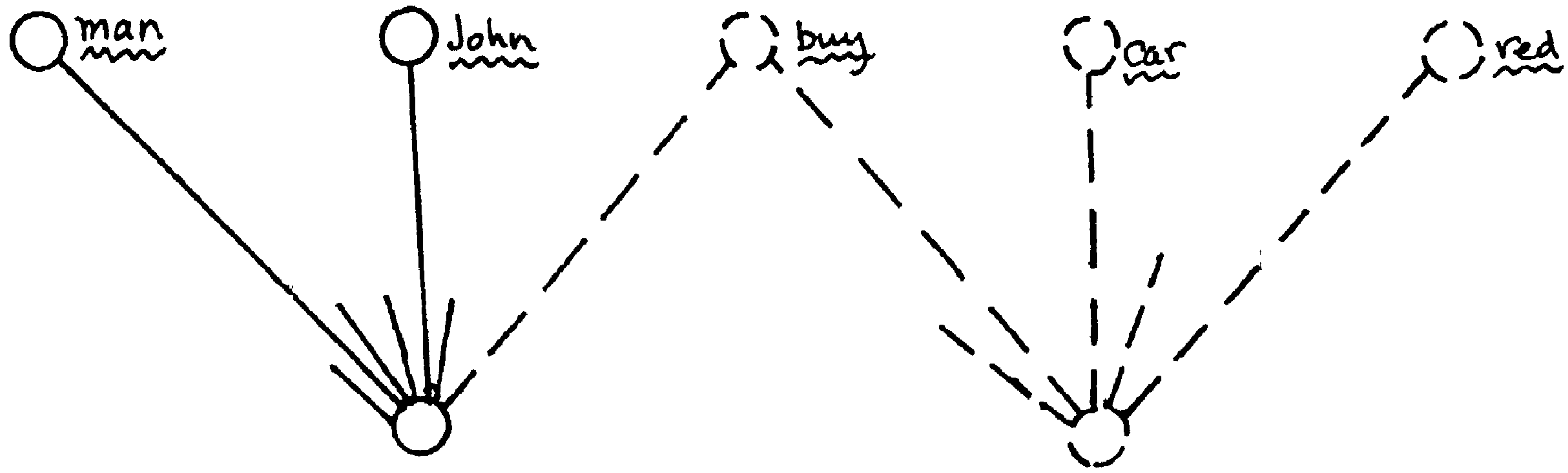
The second problem is that it is simply untrue that a person acquires beliefs about the existence of entities and the truth of propositions only by being told about them. I may come to believe that there exists a red car either because John tells me that he has bought a red car and I believe him, or because I see the red car; similarly, I may come to believe that John bought the red car either because he tells me so or because I watched the transaction take place. The car may subsequently be denoted by the phrase *the red car*, and the proposition about it by the clause *that John bought the red car*, irrespective of whether the referents representing the car and the proposition were created in response to speech or observation.

[27]These remarks may sound as if I am treating humans as mindless robots -- 'automata' in the pejorative, deterministic sense -- but quite the reverse: remember that the referents whose creation correlates with the topicon-owner's actions are brought into being by the process we have identified with thinking. There is nothing disrespectful to our species in suggesting that our actions are controlled by our thought.

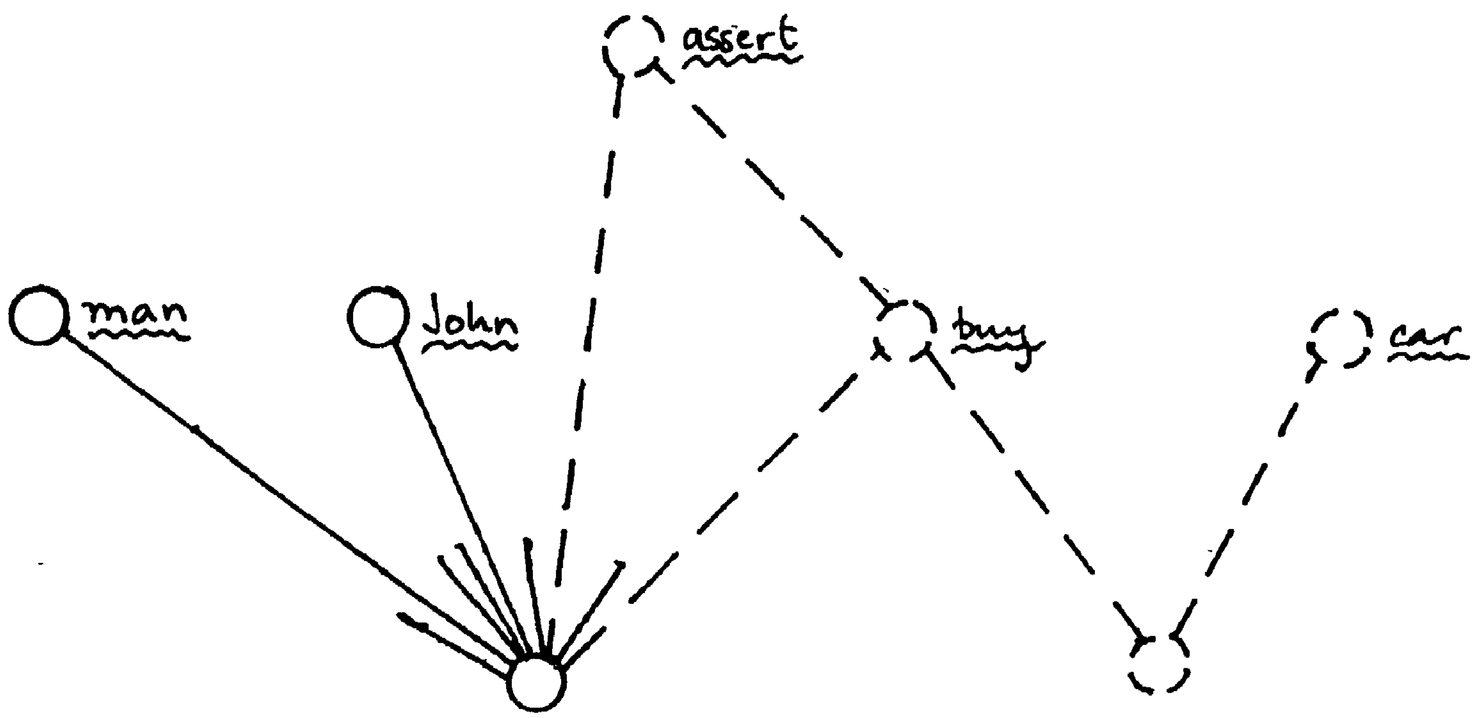
The answers to these problems are related. I suggest that the sight of John buying a car is the kind of input to a person that has the effect on his topicon which I have so far attributed to the hearing of the sentence *John bought a car* (or *John bought the car*, if the car is one of which the topicon owner is already aware): in other words, this sight creates a referent labelled *buy* and dominating referents representing John and the car. On the other hand, hearing, say, Mary uttering the words *John bought a car*, or hearing John say *I bought a car*, has a more complex effect than I have been suggesting: it creates a node labelled *assert* dominating the referent representing Mary (or John) together with a new *buy* referent as already mentioned.[28]

I diagram the two cases in (21) and (22), on the next page. The part of the diagram in solid lines is the same in each case, and represents part of the hearer's topicon before the change of state. In (21) the dotted lines represent the effect on the topicon of seeing John buy a car; in this case, since the topicon owner sees the car, we may assume that he adds some further facts about it (such as that it is red) to his topicon. In (22) the dotted lines show the result instead of hearing John say *I bought a car*. In this case, the referent representing the car will be dominated just by the *car* node and the *buy* node, since the hearer has no independent information about it.

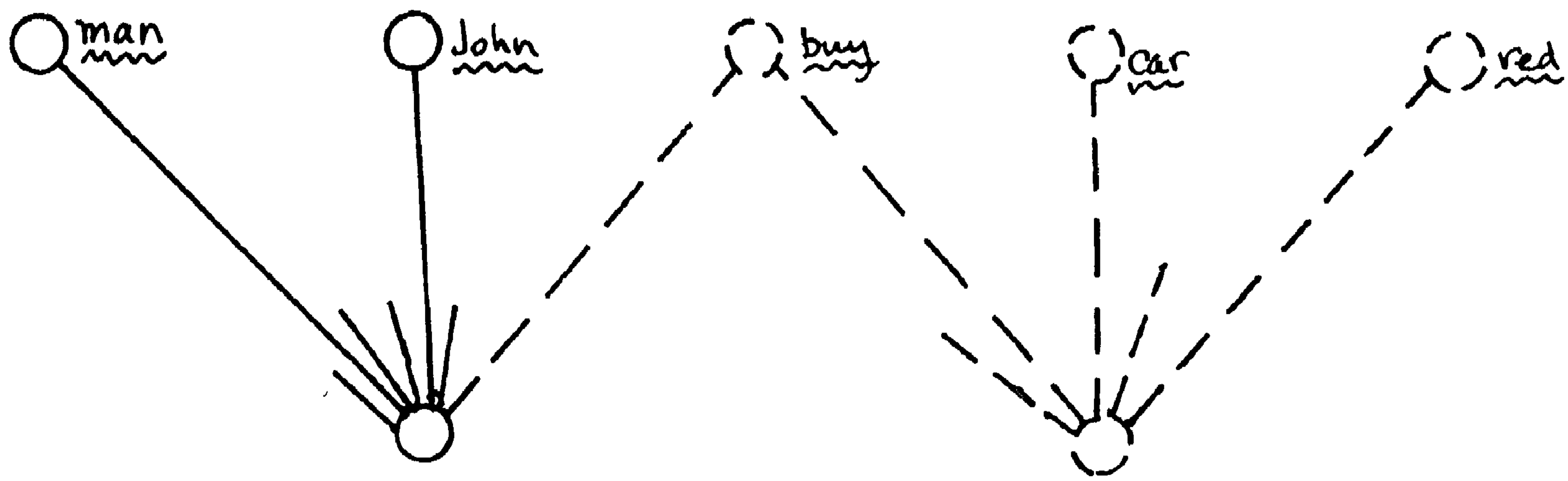
[28]Ross (1970) and others have claimed that there is actually syntactic evidence that *John bought a car* has a deep structure something like *I assert that John bought a car*. Ross's arguments are attacked by Fraser (1970), Anderson (1970), Matthews (1972). My theory is intended to be independent of Ross's claim, although the latter, if accepted, would possibly make my theory seem more plausible.



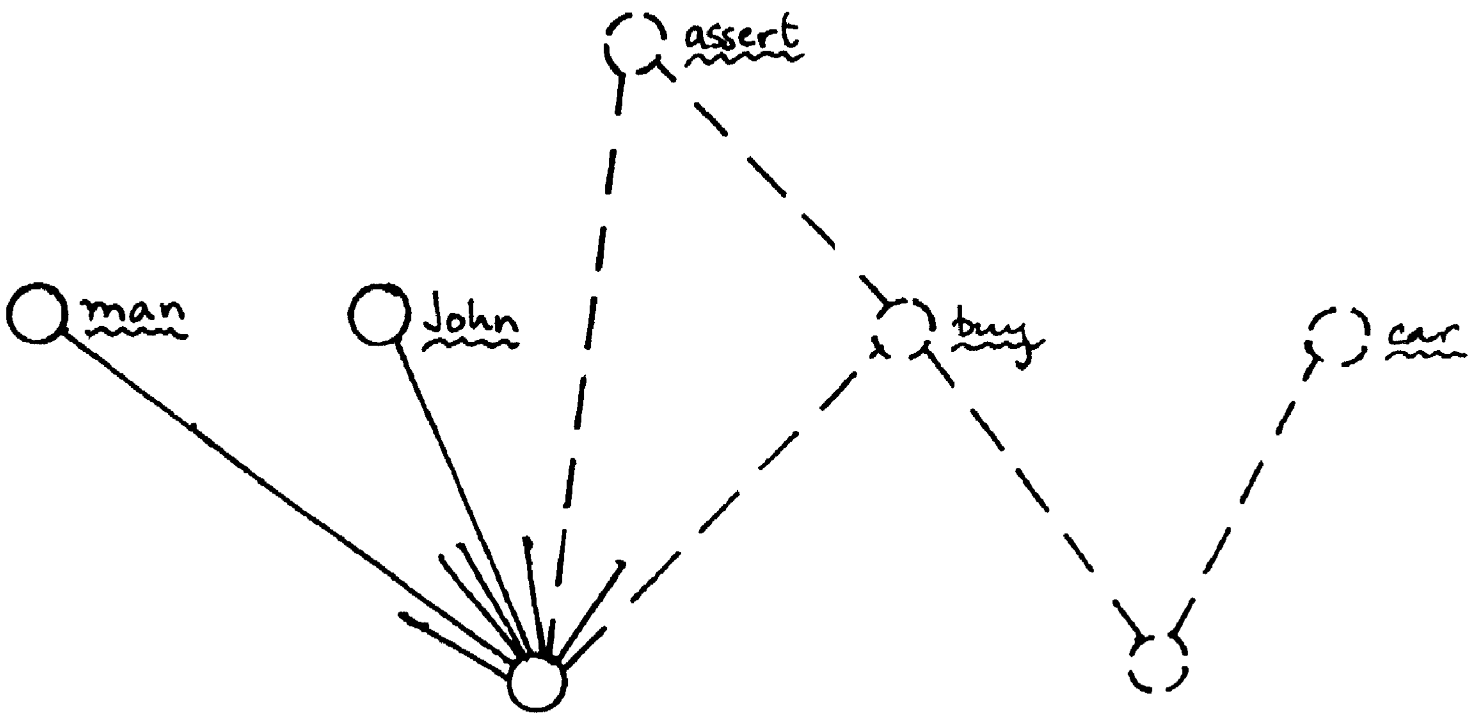
(21)



(22)



(21)



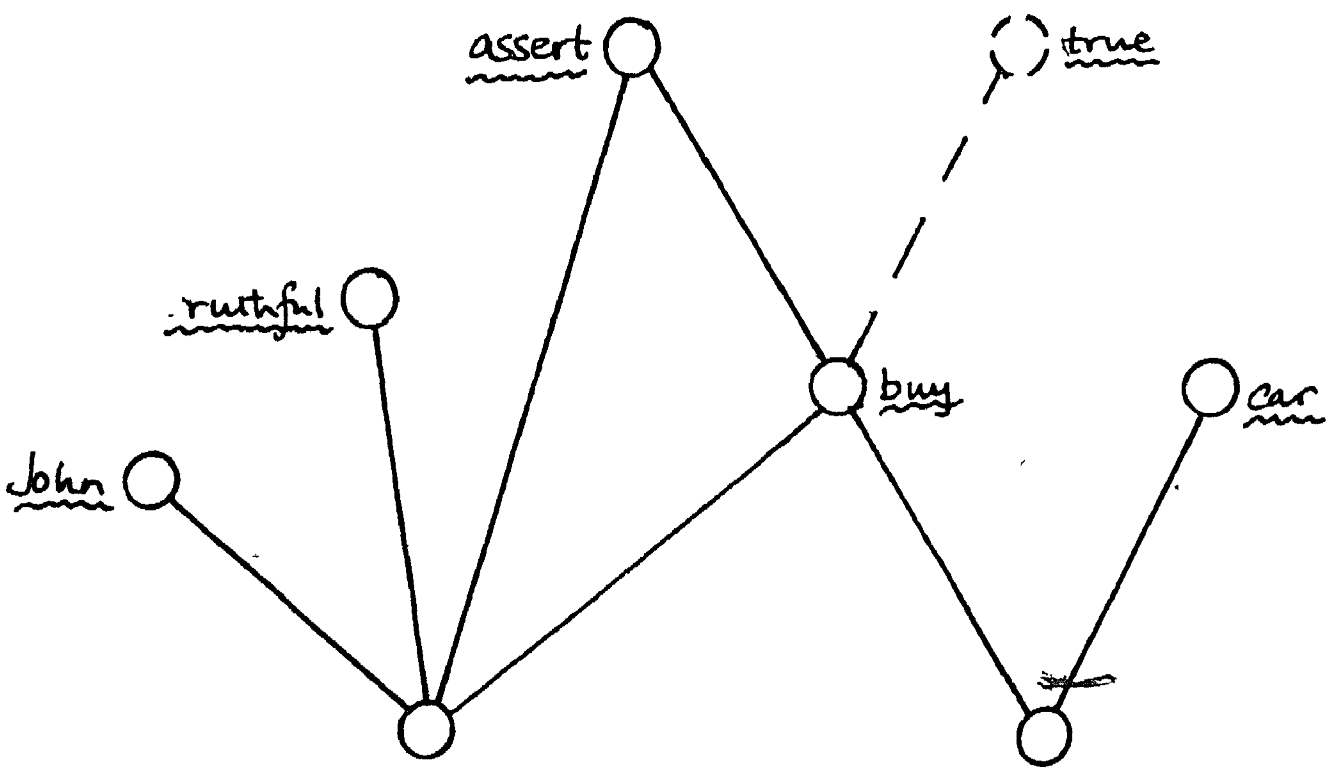
(22)

Notice that, if John tells you he has bought a car, you may well doubt that he has bought a car but you are not free in the same way to doubt that John has asserted that he has bought a car. You may, of course, doubt the latter also -- 'Did he really say the words I thought I heard him say?', 'Can I be sure it was really John speaking?' -- but this is to doubt the accuracy of one's observations, as one may doubt whether John bought a car after watching him buy it, rather than doubting the truth of what is said to one.

Clearly there are enormous problems about how observations via the senses of a complex and continuous environment result in topicon changes corresponding to the input of a discrete sentence: why should my view of John handing over a cheque on the car-dealer's forecourt change my topicon in the way which corresponds to the sentence John bought a car, rather than any of the (surely) infinitely many other propositions which could be corroborated on the evidence of my current visual, auditory, etc. inputs? However, these problems are in no sense created by the topicon theory: these are already familiar problems in psychology and in the philosophy of science. (Cf. e.g. Hanson 1958, Gregory 1970.) Some process of deriving discrete propositional beliefs from continuous sensory input must occur, if observation is to be relevant to propositional knowledge at all. Since this process is known to exist independently of my theory, and since I can make no contribution to understanding it, I shall not consider it further.

Once we agree to treat simple declarative sentences as creating propositional referents labelled assert, there is no special difficulty in handling sentences performing other illocutionary acts; e.g. Shut the door! will establish a command referent dominating referents representing speaker, hearer, and the proposition that the hearer will shut the door.

Rules of inference may permit referents representing facts about the world to be created on the basis of referents representing facts about assertions. Suppose, for instance, that there is a rule of inference which we might state as '*x* *assert* *y*, *x* *truthful* → *y* *true*'; then a topicon including referents representing the fact that John is truthful and the fact that John asserted that he bought a car will be able to move to a state in which the representation of the proposition asserted by John is §(*true*), as shown:



(23)

Similarly, one can imagine that there might be rules of inference taking a topicon from the state created by the reception of *Shut the door!* to a state which causes the topicon-owner to shut the door. However, here we come close to the point at which my theory in its present state breaks down; I defer discussing this until §22.

20. According to the theory I have sketched, English as a programming language is not dissimilar to APL, SNOBOL, etc. It resembles the latter in that its states consist of arrays of objects drawn from a specified class (although the precise structure of the arrays is different as between English and the artificial programming languages, as it is between the latter themselves), and in that the structural descriptions of its sentences include a subclass of nodes which pick out objects from the current state and another subclass which add new objects to the current state. English differs from APL, SNOBOL, etc., in lacking identifiers, and in using the property of distance between objects in a state in order to identify objects.

My theory is certainly inadequate to account for many quite elementary facts about English and other natural languages. It may be that its deficiencies are too great for the theory to merit consideration. However, I would argue that it is worth according my theory the temporary immunity from falsification to which Lakatos (1970: 179) suggests new research programmes are entitled, in case anyone can suggest modifications which preserve its good points while removing its defects.

21. Before discussing the objections to it, let me mention a number of points to which my theory offers satisfactory solutions.

In the first place, the theory is attractive simply because it offers an answer (even if the answer eventually turns out to be wrong) to the question why humans should spend so much of their time exchanging the abstract structures called 'sentences': unlike cultivating the ground or building houses, the utility of this occupation is not immediately apparent to the observer (Sampson 1972a, 1975a:133-6). In my theory, the exchange of sentences, like direct observation of the environment, helps humans build up a complex but finite 'map' or 'model' of the world, a model which can be described in quite concrete terms and which controls the human's actions in ways which, again, in principle should be quite explicitly definable.

The notion 'model' is of course a central one in the most influential current view of what language is for -- the view which explicates natural-language semantics in terms of 'model theory'. But the 'possible worlds' of model theory, unlike the topicons of my theory, are infinitely complex entities which can hardly be taken to represent characteristics of finite human minds. Furthermore, in the model-theoretic approach to natural language, the point about a true sentence is that it denotes the Fregean truth-value True (see e.g. Suppes 1973); but if we think of the act of uttering a true sentence as the act of denoting the True, then it is quite unclear why people should utter sentences (let alone why they should utter one true sentence rather than another).[29] [Footnote on p.59.]

In my theory, to utter a particular true sentence to a hearer is to make a particular change to his mental model of the world which gives the hearer more premisses from which to predict the consequences of his actions; thus, the more true sentences a person hears, the more rational his actions can be.

My theory has some more specific points in its favour. It explicates neatly some syntactic/semantic distinctions which seem rather pervasive in natural language but which have resisted other attempts at explication: the definite/indefinite distinction in noun-phrases, the restrictive/appositive distinction in relative clauses. Also it neatly explains the genitive construction. Accounts of the genitive which treat it in terms of possession (e.g. Suppes 1973: 382-3) are simply unfaithful to the facts; it seems that <u>any</u> relation between the denotatum of the head NP and that of the genitive NP in a genitive phrase can be used to understand such a phrase, but this makes sense only if, for a given hearer, there are a well-defined, limited set of relations between denotata -- as my theory asserts. My theory shows how it can be that definite descriptions succeed in referring even though, contrary to Russell's theory of descriptions, the properties they mention are typically not uniquely instantiated -- and, more remarkably, in the case of pronouns no properties of the denotatum are specified at all.

My theory is also satisfying in its treatment of presuppositions. Although the fact that sentences typically embody presuppositions has by now received much discussion in linguistics, it has not been clear how the distinction between assert-

---

[29] For objections to model theory as a means of explicating natural-language semantics and pragmatics, cf. Sampson (1974, 1975b), Potts (1975), Jardine & Jardine (1975).

ions and presuppositions should be represented in terms of syntactic or semantic descriptions. One proposal (cf. Fillmore 1969, Lakoff 1969, Horn 1970) is that the semantic description of a sentence should be a pair of objects, one element representing the proposition asserted and the other the proposition presupposed. This proposal is problematical, first because it seems arbitrary -- why should a semantic description of a sentence consist of a pair of propositions rather than one proposition or a 5-tuple of propositions? -- and, more seriously, because it is not clear that there is in general just one or even any fixed number of propositions presupposed by a sentence, as there is just one proposition asserted by a sentence. Thus, the sentence:

(24) *The car which John bought is red.*

presupposes that *John bought a car*, but also presumably that *there is someone called John*; *John's car* perhaps presupposes that John bought a car, but perhaps alludes to the fact that John was almost run down by a car, etc. etc. On my theory, failure of presupposition occurs when the input sentence is undefined for the current topicon-state. (24) will fail if there is no triple r_1, r_2, r_3 of referents in the current state such that r_1 is §(*John*), r_2 is §(*car*), r_3 is labelled *buy*, and r_3 immediately dominates (r_1, r_2). To say that (24) presupposes *that John bought a car* corresponds to the fact that if the latter phrase does not pick out any current referent by the rules R1-R10 which define the function $\underline{Int}_{Eng}$, then the sentence (24) will fail to create a node labelled *assert* -- i.e. will fail to make an assertion. Presupposition-failure is quite akin to the case in APL when a <u>dscr</u> node is realized as an identifier belonging to no current object, or as a function together with a set of arguments falling outside the domain of that function; in the APL case, higher asst nodes will fail to create corresponding APL-objects, as the sentence (24) fails to create either a referent labelled *red* or one labelled *assert* in a topicon lacking r_{1-3}.

I have argued elsewhere (1972b) that the reason why the Liar paradox does not render English inconsistent is that, as a matter of observable fact, a definite description in a natural language is never taken by naïve native speakers to refer to a proposition asserted by the sentence in which that definite description occurs, whether or not paradox would result if it were. This immediately raises the question why natural languages should have such a convenient property. My theory explains this simply: in natural languages, as in APL, interpretation of nodes takes place not simultaneously but sequentially, from the bottom upwards. At the time the referent of the NP *what I am now saying* is to be located, the referent to be created by the sentence [*what I am now saying*] *is false* cannot yet have been brought into existence, so the possibility that the two might be identical does not arise.

The theory also explains the puzzling fact that

(25) *Scott is identical to the author of 'Waverley'.*

can be a useful thing to say, while

(26) *Scott is identical to Scott.*

can hardly be so (Russell 1905: 108; cf. e.g. Linsky 1967: 26). (26) will pick out the same referent, say r_1, twice, and create a referent labelled identical and dominating (r_1, r_1); but we may assume that a rule of inference of English states that anything is identical to itself, i.e. that a referent labelled identical may always be created dominating (r_x, r_x) for any referent r_x. Therefore the input sentence achieves nothing that the successor-state relation could not have achieved independently of any input. In the case of (25), however, if the hearer does not know that Scott is the author of 'Waverley', then the two NPs will pick out different referents r_1, r_2 in his topicon and will create an identical node dominating (r_1, r_2); clearly no English rule of inference will do this.[30]

The composite nature of the denotation relation incorporated within my theory copes neatly with the fact that natural languages use exactly the same syntactic devices for discussing characters in fiction, and the like, as for discussing real entities. Anyone who has read Crime and Punishment will understand the sentences:

(27) Raskolnikov killed Alena Ivanovna.
(28) Alëna Ivanovna killed Raskolnikov.

and will agree that the former is true and the latter false.

[30]Strictly speaking, (25) will create an assert node dominating the referent representing the utterer of (25) and the identical node mentioned. We may assume that one of the rules defining Suc_{Eng} lays down that when two distinct nodes r, r' are dominated by an identical node dominated by true (i.e. when the hearer comes to believe that the denotata of two phrases are identical) a new state may be formed in which r and r' are replaced by a single referent connected with all the referents to which either r or r' were linked.

Yet, in the case of formulae of the predicate calculus such as *f*(*a*, *b*), *f*(*b*, *a*), if *a* or *b* lack denotation then the formulae as wholes seem to be either both false or both meaningless, but not interestingly different. Reichenbach (1947: §49) has offered a logic which includes representations of sentences about fictional entities, but in his system the symbol-arrays corresponding to NPs having fictional referents are quite different in kind from those corresponding to NPs having real referents. There is no trace of such a distinction in the syntax of natural languages. In my theory, the NPs *Raskolnikov* and *Richard Nixon* work in exactly the same way as each other -- they each pick out one of the referents in the hearer's topicon -- so it is natural that the NPs are syntactically parallel. The fact that the referent of *Raskolnikov* will have the property §(*fictional*) while that of *Richard Nixon* has the property §(*real*) is no more reason to distinguish sentences (27) and (28) from sentences about Richard Nixon and Spiro Agnew syntactically than is the fact that the referent of *Raskolnikov* has the property §(*Russian*) while that of *Nixon* has the property §(*American*).[31]

[31]The topicon theory thus seems to make some sense of the ontological views of Meinong (1913) and the early Russell (cf. Linsky 1967: 2-3). Meinong was troubled by the truth of e.g.

(i) *Pegasus does not exist.*

since, if Pegasus really does not exist, there appears to be nothing which (i) can be about, and thus (i) cannot make a true statement. Meinong therefore suggested (382-3, 491) that, although it was true of only some definite descriptions that their denotata actually existed, the denotatum of any definite description had quasi-existence, and this was enough for an entity to serve as the subject of a statement. In our terms, to 'denote a quasi-existent object' is to refer to a referent; to 'denote an existent object' is to refer to a referent having a denotatum.

Finally, my theory suggests why there are three categories of Austinian 'speech acts'. Austin (1962) distinguished (not consistently, admittedly) between locutionary acts (speaking), illocutionary acts (doing something, e.g. giving an order, in speaking), and perlocutionary acts (achieving some effect, e.g. causing the hearer to perform an action, through speaking). (A number of current commentators on Austin would not agree with my presentation of his distinctions; however, I believe my discussion is faithful to Austin's own views in much of How to Do Things with Words.) Why should there be just three categories of speech act, rather than two or four? Some scholars have suggested that the three-category analysis is incorrect; but I would support it. Consider the various consequences of speaking. At the first level, sound is produced; the production of this sound is a locutionary act. If the sound is a well-formed sentence of English which is defined by the input function for the hearer's topicon state, then that sentence produces a specific effect on the hearer's topicon: the production of this effect is an illocutionary act of the type defined by the label of the topmost new referent. Thus, if the sentence adds to the topicon a referent labelled assert, the illocutionary act is one of assertion; the 'misfiring' of an illocutionary act, as when a sentence syntactically in declarative form fails to make an assertion because one of its definite descriptions fails to refer, corresponds to failure to create an assert referent in the hearer's topicon. (We have seen that, when a subordinate node cannot be evaluated, processing of the phrase-marker stops.) The new topicon state may lead to other topicon states and, perhaps, to actions on the hearer's part, via the successor-state relation: the production of such effects may be identified with Austin's perlocutionary acts. My theory predicts that the illocutionary force of a given sentence should be well-defined and drawn from a finite class of illocutionary types (corresponding to the possible labels of sentential phrase-marker roots), while (since Eng is non-deterministic) the perlocutionary effects may be many and various; this seems to accord with Austin's discussion.[32]

22. Having presented my theory and discussed the respects in which it seems successful, I must now discuss its many inadequacies. Some aspects of English have been omitted from the present account simply for the sake of brevity; I believe there is no difficulty of principle in expanding my account to handle e.g. plurality, co-ordination with *and*, prepositions, adverbs, modality, and most subordinate clauses. But a number of English constructions present greater problems. These include, for instance, negation and universal quantification.[33] For negation, one might think of treating *not* as a monadic predicate whose argument is a proposition, so that, e.g., *John did not buy a car* would have the same effect on a hearer's topicon as *John bought a car*, followed by the creation of a

[32] In the framework of my theory, the locutionary/illocutionary distinction becomes rather parallel to the distinction between *seeing* and *seeing as* which exercised Wittgenstein and other philosophers. *I see* the duck-rabbit picture (cf. Hanson 1958) if light reflected from that picture stimulates my optic nerve; I see it *as* a duck, if this stimulation leads to the creation of a §(*duck*) referent in my topicon.

[33] The fact that these two constructions should both be problematic is no coincidence. We can handle sentences whose translations into predicate calculus involve *existential* quantification, e.g. '$(\exists x)(c(x) \,\&\, b(j, x))$' for *John bought a car*; '$\sim\exists x\sim$' is interchangeable with '$\forall x$', so, if we could handle negation, we should be able to handle universal quantification.

not referent dominating the *buy* referent just created. But then it would make sense to speak of *the car which John didn't buy*, whereas in its commoner sense *John did not buy a car* does not imply the existence of any particular unbought car. Again, one might think of interpreting e.g. *All girls love John* as creating a *love* referent dominating each pair (r_1, r_2) of referents in the hearer's topicon such that r_1 is §(*girl*) and r_2 is the referent representing John. But this would be quite inadequate: the sentence is about, not the particular girls the hearer knows of when he hears it uttered, but all girls whatsoever. A related point is that the theory does not handle the generic sense of definite NPs, as in *The elephant is a noble beast.*[34] Other difficulties are with yes/no questions, with the distinction between 'factive' and 'action' interpretations of complement clauses (*Mary's dancing was unexpected* v. *Mary's dancing was graceful*), with truth-functional connectives such as *if*, *or*, with conjunctions such as *but* v. *and*, *although* v. *because* whose appropriateness depends on a given proposition constituting evidence for or against the truth of another, and with comparative and superlative constructions.[35]

[34]One approach to these problems might involve introducing referents representing 'universals' (in the logical sense), so that for an individual referent to have the property §(*girl*) or §(*elephant*) would be for that referent to be one of the unordered set of individual referents dominated by the referent representing the universal *girl* or *elephant* (in which case the notion of 'labelling' propositional referents might be dropped). Then the propositional referent created by *All girls love John* would link the referent representing the universal *girl* with the individual referent for John. It remains to be seen whether an adequate solution can be produced along these lines.

[35]I am not sure whether the opaque/transparent reference distinction belongs on this list. I am inclined to explain the

Another deficiency of the present theory is that there are phrase-markers generated by (17) whose effects are not specified by R1-10, e.g. phrase-markers in which NP is rewritten wh S. I hope that an account of the unexplained constructions in the above list may turn out to involve uses for the phrase-markers which are not handled by R1-10, but I have no idea whether this will be so.

23. I am not at present clear how to adapt my theory to account for these constructions, and, since they include some very basic ones, my discussion of the nature of the automaton underlying a hearer's linguistic abilities may be worthless. However, although my theory may be rejected, it would seem that there must be some adequate theory of the human comprehension of language in terms of automata whose states are of finite complexity. An account of natural-language semantics in terms of infinitely large sets of, in general, infinitely complex possible worlds cannot be the whole truth about how finite human beings understand language. I hope, therefore, that the inadequacies of the above account may spur others to improve on my work.

---

two senses of e.g. John is looking for the dean by saying that the hearer's topicon will contain referents representing referents in John's topicon (as well as referents representing objects in the outside world), and that while, in the transparent sense, the dean picks out one of the ordinary referents, in the opaque sense it picks out one of the referents representing John's referents. But clearly this needs to be spelled out more fully than I have done. A D. Phil. thesis currently being prepared by Ephraim Borowski of Hertford College, Oxford, incorporates some promising lines of attack on a number of these problems.

References

Anderson, S.R. (1970) 'On the linguistic status of the performative/constative distinction'. Mathematical linguistics and automatic translation, Report no. NSF-26, Computation Laboratory of Harvard University.

Apostel, L. (1971) 'Further remarks on the pragmatics of natural languages'. Bar-Hillel (1971: 1-34).

Austin, J.L. (1962) How to Do Things with Words. Harvard U.P.

Bach, E. (1968) 'Nouns and noun phrases'. E. Bach & R.T. Harms, eds., Universals in Linguistic Theory, Holt-Rinehart-Winston, 90-122.

Bar-Hillel, Y., ed. (1971) Pragmatics of Natural Languages. Reidel.

Cheng Chung-ying (1968) 'Eliminability of singular terms reconsidered'. Foundations of Lg. 4.282-95.

Chomsky, A.N. (1968) Language and Mind. Harcourt, Brace, & World.

Fillmore, C.J. (1969) 'Verbs of judging: an exercise in semantic description'. Papers in Ling. 1.91-117.

Fillmore, C.J. & D.T. Langendoen, eds. (1971) Studies in Linguistic Semantics. Holt-Rinehart-Winston.

Fraser, B. (1970) 'A reply to "On declarative sentences"'. Mathematical linguistics and automatic translation, Report no. NSF-24, Computation Laboratory of Harvard University.

Geach, P. (1957) Mental Acts. Routledge & Kegan Paul.

Gregory, R.L. (1970) The Intelligent Eye. Weidenfeld & Nicolson.

Hanson, N.R. (1958) Patterns of Discovery. Cambridge U.P.

Hintikka, K.J.J. (1973) 'Grammar and logic: some borderline problems'. Hintikka, Moravcsik, & Suppes (1973: 197-214).

Hintikka, K.J.J., J.M.E. Moravcsik, & P. Suppes, eds. (1973) Approaches to Natural Language. Reidel.

Horn, L.R. (1970) 'Ain't it hard (anymore)'. Papers from the Sixth Regional Meeting, Chicago Linguistic Society. 318-27.

Iverson, K.E. (1962) A Programming Language. Wiley.

Jacobs, R.A. & P.S. Rosenbaum, eds. (1970) Readings in English Transformational Grammar. Ginn.

Jardine, N. & C.J. (1973) 'Model theoretic semantics and natural language'. Keenan (1975).

Keenan, E., ed. (1975) Formal Semantics of Natural Language. Cambridge U.P.

Lakatos, I. (1970) 'Falsification and the methodology of scientific research programmes'. I. Lakatos & A. Musgrave, eds., Criticism and the Growth of Knowledge, Cambridge U.P., 91-195.

Lakoff, G. (1969) 'Presuppositions and relative grammaticality' (quoted in A.N. Chomsky, 'Some empirical issues in the theory of transformational grammar', S. Peters, ed., Goals of Linguistic Theory, Prentice-Hall, 1972, p. 64).

Leech, G.N. (1969) Towards a Semantic Description of English. Longmans.

Lenneberg, E.H. (1967) Biological Foundations of Language. Wiley.

Linsky, L. (1967) Referrring. Routledge & Kegan Paul.

McCawley, J.D. (1971) 'Tense and time reference in English'. Fillmore & Langendoen (1971: 97-113).

Matthews, P.H. (1972) Review of Jacobs & Rosenbaum (1970). Journal of Linguistics 8.125-36.

Meinong, A. (1913) Abhandlungen zur Erkenntnistheorie und

Gegenstandstheorie. (Gesammelte Abhandlungen, vol. 2.) Barth (Leipzig).

Pakin, Sandra (1968) APL\360 Reference Manual. Science Research Associates.

Partee, Barbara H. (1971) 'On the requirement that transformations preserve meaning'. Fillmore & Langendoen (1971: 1-21).

Postal, P.M. (1970) 'On the surface verb "remind"'. Linguistic Inquiry 1.37-120; reprinted in Fillmore & Langendoen (1971: 180-270).

Potts, T.C. (1973) 'Model theory and linguistics'. Keenan (1975).

Putnam, H. (1960) 'Minds and machines'. S. Hook, ed., Dimensions of Mind, Collier Books ed. 1961, 138-64.

Quine, W. van O. (1960) Word and Object. MIT Press.

Reichenbach, H. (1947) Elements of Symbolic Logic. Macmillan.

Rosenbaum, P.S. (1967) The Grammar of English Predicate Complement Constructions. MIT Press.

Ross, J.R. (1970) 'On declarative sentences'. Jacobs & Rosenbaum (1970: 222-72).

Russell, B. (1905) 'On denoting'. Mind n.s. 14.479-93; reprinted in H. Feigl & W. Sellars, eds., Readings in Philosophical Analysis, Appleton-Century-Crofts, 1949, 103-15.

Russell, B. (1957) 'Mr Strawson on referring'. Mind n.s. 66.385-9; reprinted in T.M. Olshewsky, ed., Problems in the Philosophy of Language, Holt-Rinehart-Winston, 1969, 333-8.

Sampson, G.R. (1970) 'The reality of linguistic decoding'. Journal of Philosophy 67.961-9.

Sampson, G.R. (1972a) 'Can language be explained functionally?' Synthese 23.477-86.

Sampson, G.R. (1972b) 'Natural language and the paradox of the Liar'. _Semiotica_ 5.305-23.

Sampson, G.R. (1973a) 'The concept "semantic representation"' _Semiotica_ 7.97-134.

Sampson, G.R. (1973b) 'The irrelevance of transformational omnipotence'. _Journal of Linguistics_ 9.299-302.

Sampson, G.R. (1974) Review of Bar-Hillel (1971). _Philosophia_ 4.385-91.

Sampson, G.R. (1975a) _The Form of Language_. Weidenfeld & Nicolson.

Sampson, G.R. (1975b) 'Thoughts on the recent marriage of philosophy and linguistics' (Review article on Hintikka, Moravcsik, & Suppes (1973)). _Foundations of Language_ 12.537-60.

Sampson, G.R. (forthcoming) 'An empirical hypothesis about natural semantics'.

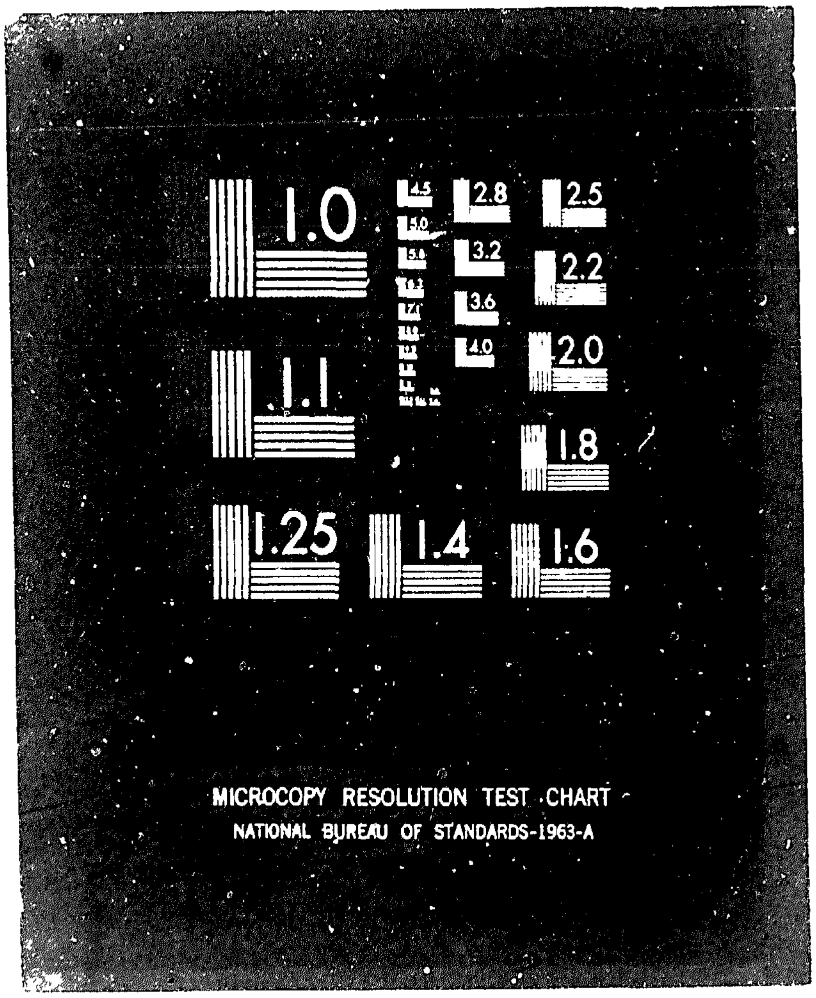
Sloat, C. (1969) 'Proper nouns in English'. _Language_ 45. 26-30.

Suppes, P. (1973) 'Semantics of context-free fragments of natural languages'. Hintikka, Moravcsik, & Suppes (1973: 370-94).

Whitehead, A.N., & B. Russell (1927) _Principia Mathematica_. Cambridge U.P.

Winograd, T. (1972) 'Understanding natural language'. _Cognitive Psychology_ 3.1-191.

END


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A