4/17

SUNY Geneseo Interlibrary Loan(YGM)

ILLiad TN: 54967

Borrower: PMC

Lending String: *YGM,VXW,PSC,PBU,JNA

Patron: Borek, Helen

Journal Title: Cognitive science.

Volume: 1 **Issue:** 2
Month/Year: Apr 1977 **Pages:** 125-157

Article Author:

Article Title: Anderson, John R.; 'Induction of augmented transition networks.'

Imprint: Norwood, N.J., Ablex Pub. Corp.

ILL Number: 5751437

Call #: Journal Article

Location:

ARIEL
Charge
Maxcost: recip/$35.00IFM

Shipping Address:
Hunt Library ILL
Carnegie Mellon University
Frew Street
Pittsburgh, PA 15213-3890

Fax: IFM articles
Ariel: 128.2.20.246

COGNITIVE SCIENCE, **1**, 125–157 (1977)

Induction of Augmented Transition Networks*

JOHN R. ANDERSON

Yale University

LAS is a program that acquires augmented transition network (ATN) grammars. It requires as data sentences of the language and semantic network representatives of their meaning. In acquiring the ATN grammars, it induces the word classes of the language, the rules of formation for sentences, and the rules mapping sentences onto meaning. The induced ATN grammar can be used both for sentence generation and sentence comprehension. Critical to the performance of the program are assumptions that it makes about the relation between sentence structure and surface structure (the graph deformation condition), about when word classes may be formed and when ATN networks may be merged, and about the structure of noun phrases. These assumptions seem to be good heuristics which are largely true for natural languages although they would not be true for many nonnatural languages. Provided these assumptions are satisfied LAS seems capable of learning any context-free language.

It has occasionally been suggested that a promising way to develop language understanding systems would be by means of a learning program that would become competent in the language through experience. There are two motivations for the acquisition approach to the development of computer models for language processing. First, it might seem more efficient to leave to a program the analysis of the knowledge underlying language use and the programming of this knowledge. Current hand-programmed language systems (e.g., Schank, 1975; Winograd, 1972; Woods, Kaplan, & Nash-Webber, 1972) deal with rather modest domains in somewhat limited ways and yet represent considerable investment of programming time. In contrast, the average human is able to learn within 20 years a language that has a large vocabulary and permits of many different structures. He is able to comprehend utterances in sophisticated manners and is able to use language in a wide variety of purposes. The natural temptation is to think that a computer learning program would be able to match this learning accomplishment and perhaps in a much shorter time than a human. The second argument for the learning approach derives from the observation that current language programs lack an ability to adapt their behavior to changing cir-

*This research was supported by grants GB-40298 from NSF and grant MH26383 from NIMH. Correspondence concerning the paper should be addressed to the author at the Department of Psychology, Yale University, New Haven, Conn. 06520.

cumstances. The real life language processing tasks facing a human are constantly changing and the way he adapts to this is by learning the demands of the new situations and accordingly adjusting his language processing mechanisms. For instance, a student taking a class in set theory must learn new vocabulary, new syntactic constructions, and new ways of expressing his ideas (e.g., proofs). A language learning program has the promise of being able to be as adaptive in its language use as is a human.

I do not know whether to endorse or deny these arguments for language learning. The principal difficulty with the language learning approach, of course, is designing a program that can actually acquire a language. In this paper I report some of my efforts to develop a language learning program. This system is somewhat unique relative to other efforts (e.g., Feldman, 1970; Hamburger & Wexler, 1975; Horning, 1969; Klein, 1973; Siklóssy, 1972) in that it lays emphasis on the requirement that the output of the learning approach be something that can be used for both language comprehension and generation, rather than an abstract characterization of the language or a characterization that can only be used for comprehension or only for generation. Like the other language learning attempts, its accomplishments are quite limited. Certainly, it is nowhere near the goal of being able to induce a general language processing system. Still it seems a significant enough step to be worth reporting.

The program is dubbed LAS, an acronym for Language Acquisition System. Its principal motivation is not to provide a route to computer language processing. Rather it is an attempt to develop a psychological model of human language processing. I have reported on aspects of this model from a psychological point of view in a number of earlier papers (Anderson, 1974, 1975, 1977). Here I will focus primarily on the program and discuss only cursorily its relation to current psychological research on language acquisition.

LAS AS A LANGUAGE USER

The LAS program is written in Michigan LISP (Hafner & Wilcox, 1974). The program accepts as input strings of words, which it treats as sentences, and scene descriptions encoded as associative networks. The associative networks used to encode the scene are slight variants of the HAM propositional representation (see Anderson & Bower, 1973). This network representation is somewhat similar to those of Carbonell and Collins (1973), Norman, Rumelhart, and the LNR Research Group (1975), Quillian (1969), Schank (1975), and Simmons (1973). LAS obeys commands to speak, understand, and learn. Central to LAS is an augmented transition network (ATN) grammar similar to that of Woods (1970, 1973). In response to the command, *Listen*, LAS evokes the program UNDERSTAND. The input to UNDERSTAND is a sentence. LAS uses the information in the network grammar to parse the sentence and obtain a representation of the sentence's meaning (encoded as a HAM propositional network). In response to the command, *Speak*, LAS evokes the program SPEAK. SPEAK receives a to-be-spoken HAM conceptualization and uses the information in the network grammar to generate a sentence to describe the conceptualization. Note that LAS uses the same ATN formalism both to speak and to understand. The third part of the program is LEARNMORE which induces these ATN grammars. LEARNMORE takes as its inputs a sentence, a HAM representation of the meaning of the sentence, and an indication of the main proposition of the sentence. The outputs of the LEARNMORE program are changes to the ATN grammar.

The program models the process of learning to speak from the pairings of sentences and pictures. The HAM conceptualizations given to LEARNMORE are taken to represent the output of a picture parsing routine. Having once acquired an ATN grammar from these picture–sentence pairings, LAS can generate sentences to describe other pictures via SPEAK and derive descriptions of the picture situations corresponding to sentences via UNDERSTAND. This program ignores the acquisition of nondeclarative, procedural aspects of language such as the processing of questions. The handling of procedural aspects of language is just now being tackled in my simulation work.

The HAM Memory System

LAS uses a version of the HAM memory system (see Anderson & Bower, 1973) called HAM.2 which provides LAS with two essential features. First it provides a representational formalism. This is used for representing the semantic interpretations output by the understanding program, the semantic intentions that are the input to the language generation program, and semantic and syntactic information in long-term memory that is used to guide a parse. Second, HAM.2 also contains a memory searching algorithm, MATCH1, which is used to evaluate various parsing conditions. For instance, the UNDERSTAND program requires that certain features be true of a word for a parsing rule to apply. These are checked by the MATCH1 process. The same MATCH1 process is used by the generation program to determine whether the action associated with a parsing rule creates part of the to-be-spoken structure. This MATCH1 process is a variant of the MATCH process described in Anderson and Bower (1973, Chaps. 9 & 12) and its details will not be discussed here.

However, it would be helpful to describe here the representational formalisms used by HAM.2. Figure 1 illustrates how the information in the sentence *The man who robbed the bank had a bloody nose* would be represented within the HAM.2 network formalisms. There is a distinct node in the memory structure for each object referenced in the sentence—a node *X* for the man, a node *Y* for the bank, and a node *Z* for the nose. There are three propositions asserted about *X*—that *X* is a man, that *X* robbed *Y*, and that *X* has *Z*. Of *Y* it is also asserted that *Y* is a bank. Of *Z* it is also asserted that *Z* is bloody, and that *Z* is a nose. Each proposition is represented by a distinct tree structure. Each tree structure

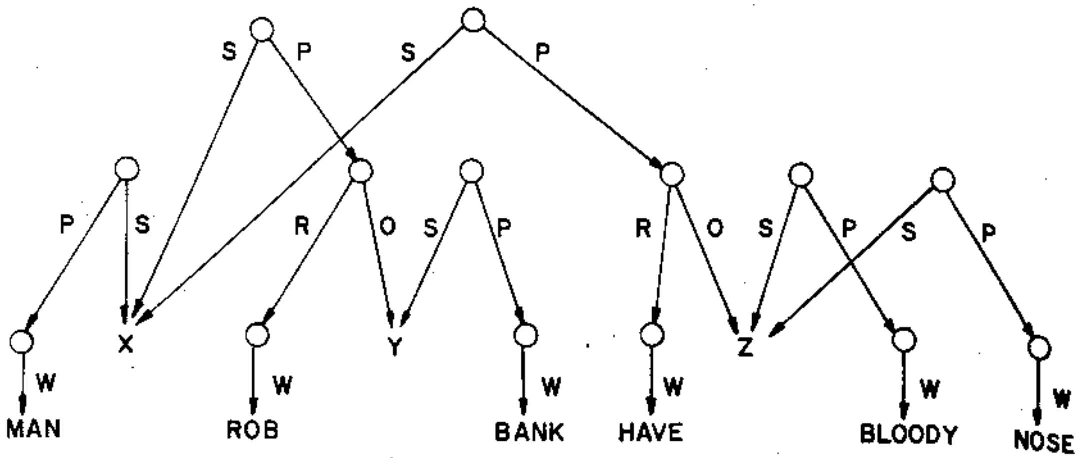


FIG. 1 An example of a propositional network representation in HAM.2.

consists of a root proposition node connected by an *S* link to a subject node and by a *P* link to a predicate node. The predicate nodes can be decomposed into an *R* link pointing to a relation node and into a *O* link pointing to an object node. The semantics of these representations are to be interpreted in terms of simple set theoretic notions. The subject is a subset of the predicate. Thus, the individual *X* is a subset of the men, the people that robbed *Y*, and the people that have *Z*. One other point needs emphasizing about this representation. There is a distinction made between words and the concepts which they reference. The words are connected to their corresponding concepts by links labeled *W*.

There are a number of motivations for the associative network representation. Anderson and Bower (1973) have combined this representation with a number of assumptions about the psychological processes that use them. Predictions derived from the Anderson and Bower model usually turn out to be true of human cognitive performances. However, many of the specific details of HAM have never been empirically tested. Also, there are some predictions derived from HAM that can be shown to be false (see Anderson, 1976). The principal feature that recommends associative network representations as a computer formalism has to do with the facility with which they can be searched. Another advantage of this representation is particularly relevant to the LAS project. This concerns the modularity of the representation. Each proposition is coded as a network structure that can be accessed and used, independent of other propositions.

So far, I have shown how the HAM.2 representation encodes the episodic information that might be the input to SPEAK and the output to UNDERSTAND. It is also used to encode the semantic and syntactic information required by the parsing system.

Augmented Transition Network Grammars

To illustrate LAS's ATN formalisms consider the grammar defined by the rewrite rules in Table 1. This grammar describes a two-dimensional world of geometric shapes that differ in color and size and spatial relation. Figure 2

TABLE 1
A Test Grammar

Grammar 2

S	→	NP is ADJ
	→	NP is RA NP
NP	→	(the,a) NP1
NP1	→	SHAPE
	→	SHAPE CLAUSE
	→	ADJ NP1
CLAUSE	→	that is ADJ
	→	that is RA NP
SHAPE	→	square, circle, etc.
ADJ	→	red, big, blue, etc.
RA	→	above, right-of

Example sentence
The red square which is small is above the circle which is right-of the triangle.

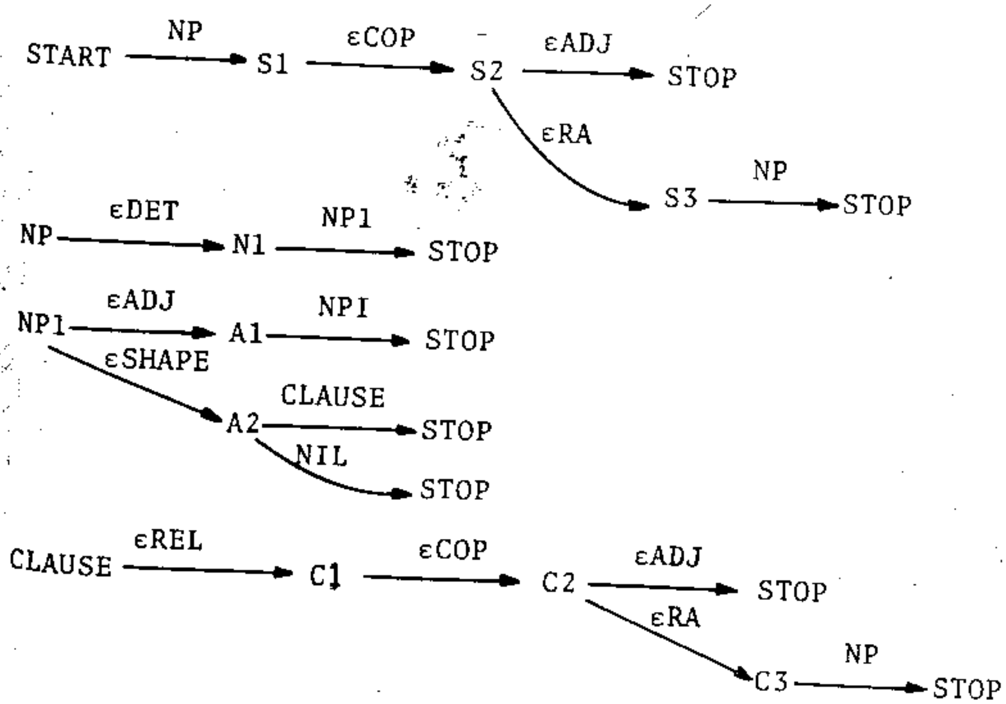


FIG. 2 The augmented transition networks encoding the grammar defined in Table 1.

illustrates the parsing networks for this grammar. There are a few conventions that need to be known to facilitate reading these networks. When a label like NP is alone on an arc it indicates that a successful push is required to that network. When the label is prefixed by an ϵ (e.g., ϵRA), this indicates that the next word must be in the word class referred to by the label (i.e., RA). If a NIL labels the arc, this means that the arc can be traversed without anything being processed about the sentence. These are the three types of conditions that can appear on ATN arcs. The actions that are placed on ATN arcs involve the construction of associative network structures to represent the sentence's meaning. For instance, the action associated with the NP arc linking START and S1 is to connect the node which is the referent of the noun phrase by an *S* link to a proposition node (see Fig. 1).

Such network grammars are modular in two senses. First, they are relatively independent of each other. Second, they are independent of the SPEAK and UNDERSTAND programs that use them. This modularity greatly simplifies LAS's task of induction. LAS only induces the network grammars; the interpretative SPEAK and UNDERSTAND programs represent innate linguistic competences for interpreting the networks. Finally, the networks themselves are very simple with limited conditions and actions. Thus, LAS need consider only a small range of possibilities in inducing a network. The network formalism gains its expressive power by the embedding of networks. Because of network modularity, the induction task does not increase with the complexity of embedding.

The same network is used by the SPEAK program for sentence generation as by the UNDERSTAND program for sentence comprehension. In comprehension the conditions on the arcs serve as tests of the sentence. If these tests are successfully met by the sentence, the actions associated with the arcs are executed, creating associative network structures. In generation, LAS works from a network structure tagged as to-be-spoken. The actions on the arcs serve as tests of the semantic structure. If the tagged semantic structure corresponds to the semantic structure that would be created by these actions, that arc path is taken. The information on the condition is used to decide what word or phrase will be generated. Since the same ATN can be used both to generate and understand, LAS has only to induce one set of grammatical rules to do both tasks. Thus the LAS program makes the prediction that acquisitions of the ability to understand and to generate go hand in hand. This use of ATN networks is different from that exemplified in the work of Simmons (1973). He had two network grammars—one for production and one for generation—but a single interpreter.

A more detailed description of how LAS uses ATNs for generation and comprehension can be obtained from Anderson (1974). They differ in a number of ways from ATNs as manifest in the work of Woods (1970). LAS's ATNs provide for direct mappings between semantic networks and sentences. Previous examples of ATNs had been concerned with mapping between sentences and deep structures or other "syntactic" objects. While this use within LAS is somewhat novel, it does not require any major new principles. Another difference between LAS's ATNs and Woods' is that the power of ATNs in LAS is somewhat reduced. They do not have the potential to compute arbitrary conditions and actions as is the case in the general ATN conception. The jettisoning of this ATN feature was both motivated by the belief that humans could not have such arbitrary computational powers and by the desire to keep ATN structure simple to facilitate induction.

It is of interest to consider the class of languages LAS can generate and the class of languages it can parse. Since the SPEAK program maps semantic structures onto sentences, the complexity of the language which LAS can generate will depend on the complexity of the class of semantic structures it is generating from. If these semantic structures have context-sensitive aspects, then the generated language can likewise have context-sensitive aspects. With respect to LAS as an accepter of languages, it will accept precisely the context-free languages. This is because, unlike Woods' (1970) system, actions on arcs cannot influence the results of conditions on arcs, and therefore they play no role in determining whether a string is accepted or not. The ATN maps these sentences onto semantic networks. One can think of some semantic interpreter of these networks being called upon to provide context-sensitive recognition powers. That is, the semantic interpreter might reject some of the output of the UNDERSTAND process as semantically ill-formed.

In any case I would want to argue that a context-free grammar (with appropriate semantic constraints) describes a large portion of any natural language. For instance, in English the only context-sensitive aspect of the syntax seems to be the *respectively* transformation. Certainly all the language spoken and understood by young children can be described by a context-free grammar. There may be some aspects more parsimoniously represented by a context-sensitive grammar, but parsimony of representation is a very different matter than requiring a context-sensitive grammar. Moreover, with respect to language learning it is my opinion that parsimony favors a context-free grammar. This is because the relevant object to which to apply the parsimony measure is the learning program and not the grammar it outputs. It seems that a much more complex learning program would be necessitated if we required that it output parsimonious-appearing grammars. It is also the case that these parsimonious-appearing grammars may not be particularly efficient as computational mediums for language comprehension and generation.

STRUCTURE OF THE LEARNING SITUATION

In describing a language learning program it is important to specify exactly what that program can learn. I will do this for LAS by first describing the nature of the learning problem that is posed. This will specify the sense in which LAS can be said to learn a language. Then, after describing LAS's learning mechanisms and an example learning history, I will attempt to define the class of languages which the program can learn.

Use of Semantic Information

LEARNMORE takes as its basic input *pairs* consisting of sentences and representations of their meaning. The source of the meaning representations can be considered to be pictures or other referents paired with the sentence. The ATN grammar that it induces provides it with a *map* that enables it to go back and forth between sentence and meaning. LAS is like a number of recent theories (Hamburger & Wexler, 1975; Klein, 1973; Siklóssy, 1972) in its attempt to achieve a semantic characterization of the target language. This contrasts with much of the earlier work (e.g., Feldman, 1970; Gold, 1967) and discussion of language acquisition where the attempt was to induce a grammar that would specify the syntactically well-formed strings in the language. The input to the language learning program under this syntactic approach consists of strings of words and indication of whether these strings are grammatical. It can be shown (Anderson, 1976) that it is *not* intrinsically easier to acquire a semantic characterization than a syntactic characterization of the language. However, it does seem that humans do find the semantic task easier for natural language. For instance, consider a series of experiments performed by Moeser and Bregman (1972, 1973). They contrasted the learning of artificial (but natural-like) languages under two conditions. In the no-referent condition their subjects only saw well-formed strings of the language. In the referent condition they saw well-formed strings plus pictures of the semantic referents of these strings. In either case, the criterion test was for the subject to be able to detect which strings of the language were well-formed—without the aid of any referent pictures. After 3,000 training trials in one experiment subjects in the no-referent condition were almost at chance in the criterion test, whereas subjects in the referent condition were essentially perfect. In addition to the fact that it appears easier to learn under the semantics approach than the syntax approach, there is the obvious fact that a semantic characterization of the language is more useful—both from the point of view of a human learner and from the point of view of developing a language processing program.

An interesting psychological question is how language learners emerge with an ability to make judgments about the syntactic well-formedness of a sentence when they are learning how to map sentences onto meanings and not onto judgments of syntactic well-formedness. One possibility is that the learner will judge as ungrammatical those strings for which his semantic procedures fail to compute *semantic referents*. Within LAS's framework this would mean that it would judge as ungrammatical those sentences which its ATN grammar cannot map into HAM meaning representations. Of course, in LAS these are the sentences which cannot be parsed through the ATN network.

Availability of Requisite Concepts

A basic prerequisite of language learning for LAS is that it already have the concepts that are referenced in the sentences from which it is to learn. LAS does not have any mechanisms for concept induction. This means that the user of the LAS program must provide it with the requisite concepts before induction begins. This could obviously be a considerable burden in applying the program to a realistically large semantic domain. However, it should be pointed out that all other learning programs that use the semantics approach also assume cognitive predevelopment.

It is becoming an accepted fact about human language acquisition that conceptual development is a prerequisite to grammar induction (e.g., Slobin, 1973). What seems to determine the timing of the acquisition of many grammatical structures such as pluralization is acquisition of the concept which these constructions signal. Much of the research that passes under the title of child language acquisition might better be described as studied of conceptual development (e.g., Clark, 1973, 1975; Nelson, 1974). The case of child language acquisition is quite complex because conceptual development and grammar acquisition are intertwined. In contrast, in second language acquisition it seems reasonable to assume that the language learner enters the learning situation with most of the concepts that will be signaled in the to-be-learned language. This is more like the LAS situation which requires a prespecification of the concepts. For this reason LAS is more naturally thought of as a model of a second language learner immersing himself in another language community and learning from examples.

Lexicalization Is Somewhat Complete

LAS, as currently developed, is a model of the acquisition of the grammar that relates strings of words to network representations of their meanings. It is not a model of how the meanings of individual words are acquired. Rather, it is assumed that the words are already attached to their meanings before grammar induction begins. In terms of a HAM representation like Fig. 1 this means that the *W* links already exist. Of course, like the assumption that conceptual development is complete, this requirement that lexicalization be complete is not absolute. There is no reason why LAS, having once learned the language, cannot pick up the meaning of some words from context just as humans can. What is required for grammar induction is just that lexicalization be complete for a substantial subset of the language.

The motivation for this assumption is psychological in that there is evidence (see Anderson, 1977) that the process of lexicalization is distinct from grammar acquisition. In point of fact, it is relatively trivial to write a computer program that will learn word meanings as well as grammar. This simply requires that the program store with each word the set of concepts that were in the semantic referent on each occasion that the word was used in a sentence. By intersection of a number of sets formed for a word on different occasions it is possible to identify the concept corresponding to the word. Because this process of lexicalization is so trivial computationally its omission does not constitute a serious weakness of the current LAS program.

HEURISTICS FOR LANGUAGE LEARNING

It can be shown (Anderson, 1976; Gold, 1967) that there does not exist any algorithm that can guarantee "sufficiently rapid" learning for most members of large, formally-defined classes of languages (e.g., all finite state languages). By "sufficiently rapid" I mean successful learning within some fixed time bound. However, it is possible to propose procedures that will produce sufficiently rapid learning of special language subclasses. Such procedures I refer to as *heuristics* because their success depends on their being given an appropriate language to learn. If given an inappropriate language these heuristics would take astronomically long to learn or completely fail to do so. I would argue that the languages which may be learned by these heuristics are the natural languages and those which are not learnable are not natural languages. This is similar to Chomsky's (1965) proposal that the language learner must contain constraints (universals of languages) on the possible form of a language. LAS's ability to rapidly learn certain language subsets depends critically on a number of heuristic procedures. I will describe these critical assumptions about language before describing the program in an actual induction situation.

The Graph Deformation Condition

ATNs are constructed such that there is a network for every phrase in an immediate constituent analysis of a sentence. Therefore, it is critical to be able to identify the phrase structure of a sentence in order to specify the hierarchy of ATN networks that must process the sentence. LAS has a program, BRACKET, which takes a sentence and a representation of the sentence's meaning and outputs a bracketing of the sentence which indicates its surface structure. The functioning of BRACKET is possible because it assumes a constraint between the surface structure of the sentence and the graph structure of the sentence's network representation. I have called this constraint the *graph deformation condition*. This constraint is illustrated in Fig. 3.

Figure 3a gives the HAM network structure for the meaning of the sentence *The girl hit the boy who liked the cake*. In Fig. 3b we have the graph structure of Fig. 3a deformed to provide a surface structure for the content words in the sentence. The structure in Fig. 3b is a graph deformation of the structure of Fig. 3a in that while the spatial locations of the nodes have been rearranged, the nodes still maintain their interconnections. That is, *girl* is still connected to node *A* which is still connected to node *B* and so on. Note that the graph deformation in

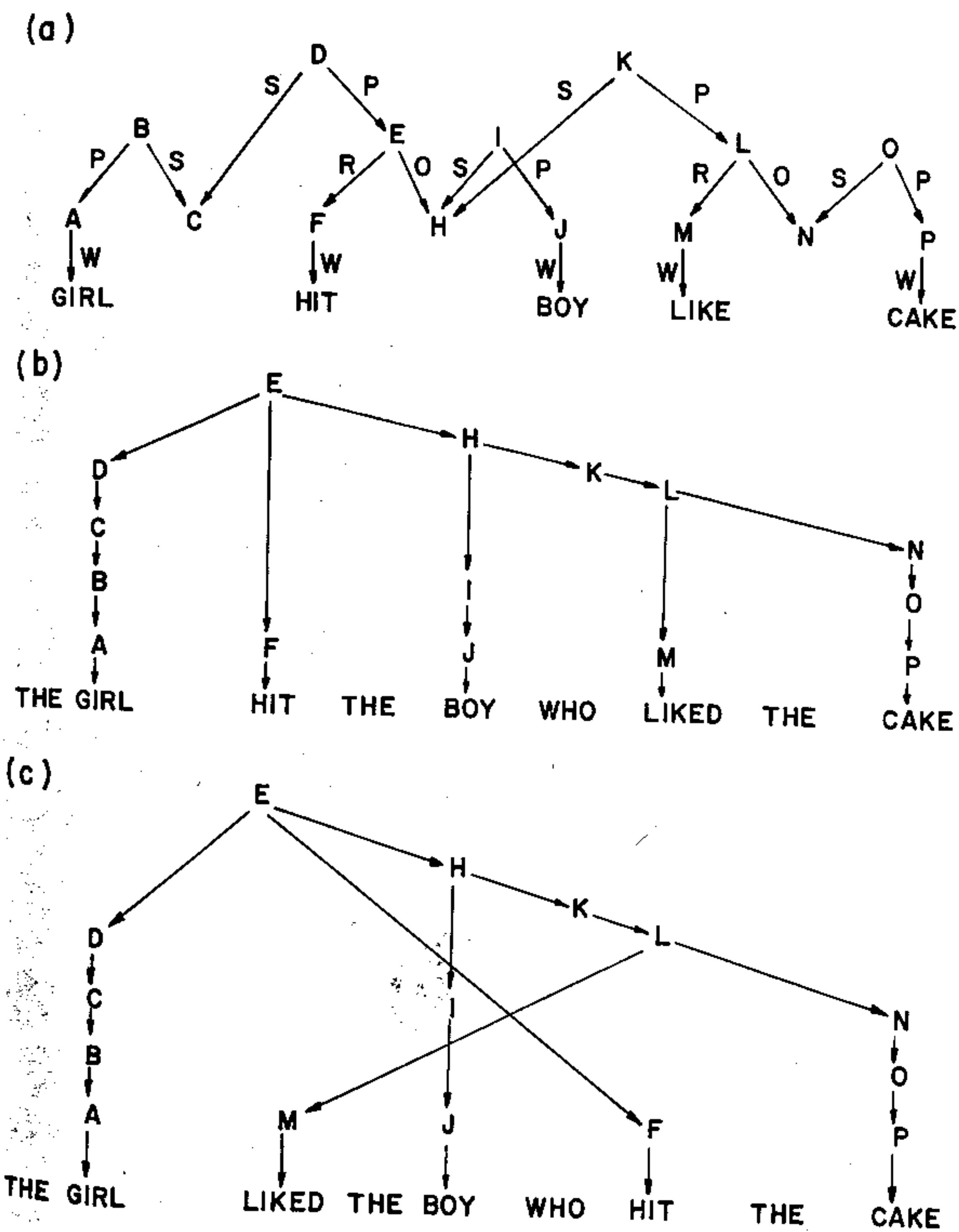


FIG. 3 The HAM structure in (a) can be deformed to provide a surface structure for the content words of the sentence in (b) but not for the sentence in (c).

Fig. 3b does capture some of the surface structure of the sentence. For instance, *girl, hit,* and *boy* are organized together under one unit and *liked* and *cake* are organized together as a modifier of *boy*. The structure in Fig. 3b does not specify how non-meaning-bearing morphemes like *the* and *who* fit into the surface structure. This is an issue to which we will return shortly.

The claim is that the surface structure interconnecting the content words of the sentence can always be represented as a graph deformation of the underlying semantic structure. This implies that certain word orders will be unacceptable ways to express certain semantic intentions. As Fig. 3c illustrates, there is no graph deformation of the semantic structure in Fig. 3a which will provide a surface structure for the sentence in Fig. 3c. No matter how this is attempted some branches must cross. A surface structure is, by definition, a tree structure without crossing branches.

BRACKET's Computations

If the graph deformation condition is satisfied for a sentence, BRACKET can identify the surface structure interconnecting the content words. The program is called BRACKET because it indicates the levels of surface structure by levels of bracketing. To appreciate informally the task performed by BRACKET consider Fig. 4. Here we have represented the information provided to BRACKET. This information is a picture semantic referent (actually a HAM network encoding of the picture) and a sentence in an unknown grammar describing this picture. Note that the words of the sentence are English. This is an attempt to recreate for the reader the situation facing BRACKET. That is, BRACKET knows the meaning of the words but not the grammar. Can you, the reader, guess a bracketing for the string in Fig. 4 that will reflect its surface structure? In this case, I think the bracketing is pretty obvious. Below I describe the nature of the computation performed by BRACKET to produce this bracketing.

Figure 5a shows how LAS would represent this picture. There are three objects in the picture, represented by the memory nodes, *I, K,* and *R*. Of *I* it is

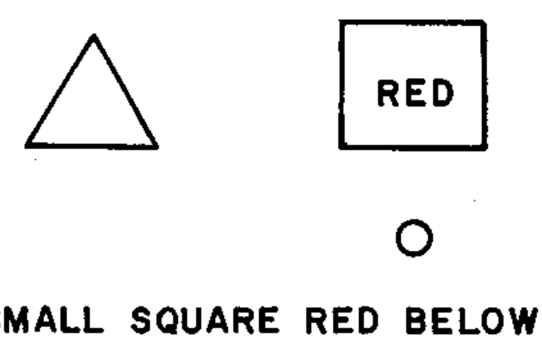


FIG. 4 BRACKET receives as input an encoding of this picture and the string of words.

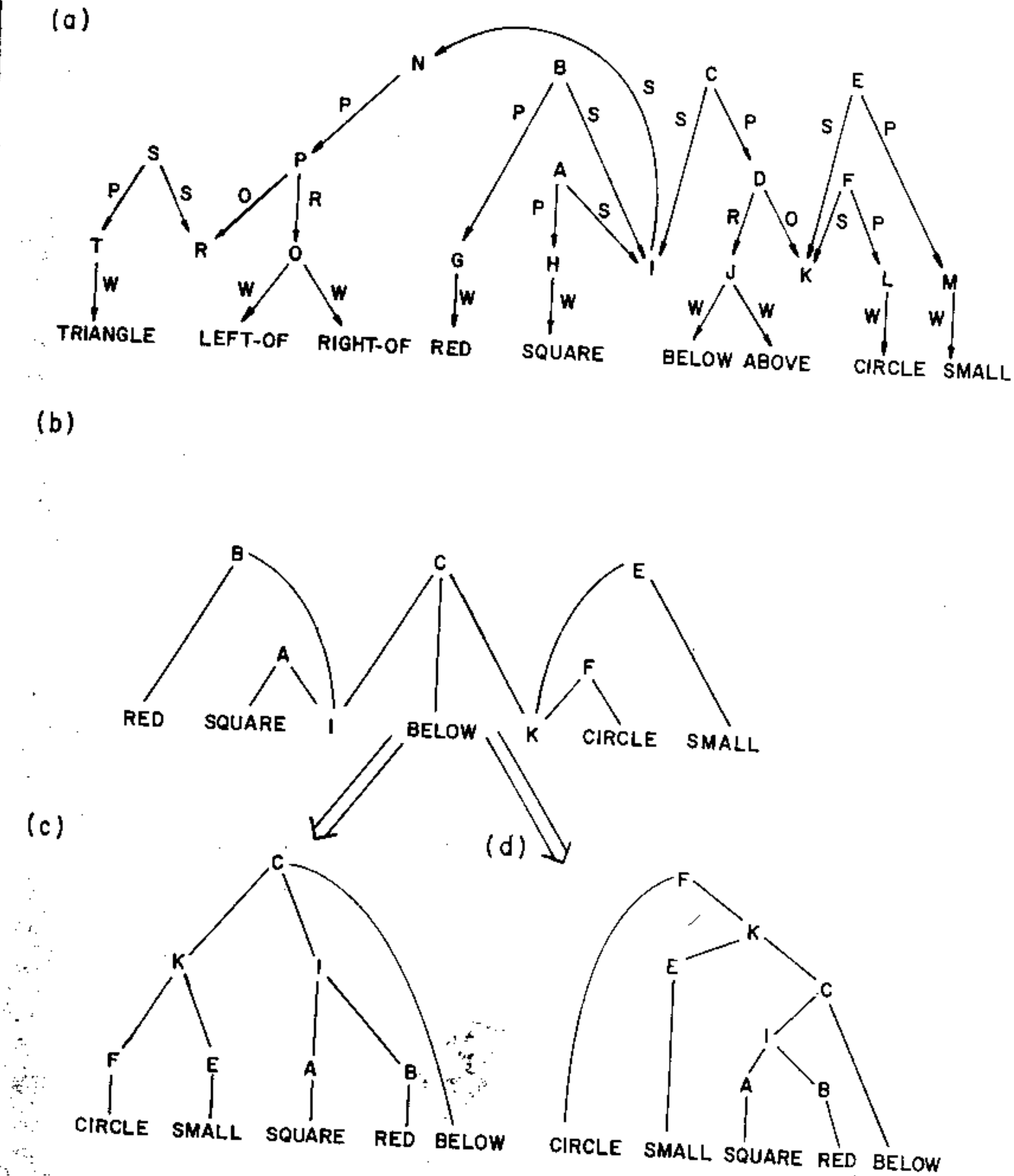


FIG. 5 The prototype structure in (b) is derived from the HAM structure in (a). From this prototype structure two surface structures, (c) and (d), can be imposed on the same string of words. The difference between (c) and (d) concerns whether Proposition C or F is the main proposition.

asserted that it is red, a square, above K, and right-of R. Of K it is asserted that it is small, below I, and a circle. Of R it is asserted that it is left-of I and a triangle.

Note that the relational terms J and O are both connected to two words. This reflects an assumption that will be important in understanding the forthcoming induction history: LAS has a single meaning corresponding to a symmetric relational term such as *above* and *below*. LAS will represent the picture in Fig. 4 to itself the same way regardless of whether *above* or *below* was used in the sentence. From the point of view of LAS the difference between these two sentences is purely syntactic. LAS will learn from examples that *above* takes the logical subject first in a sentence whereas *below* takes the logical object first. This means that the program will learn to represent sentences with *above* and *below* identically. Thus, LAS's learning program conveys upon the representational system an invariance under paraphrase which many (e.g., Anderson & Bower, 1973; Norman, Rumelhart, & LNR Research Group, 1975; Schank, 1972) have thought to be a characteristic of human memory.

From this semantic representation, BRACKET computes an intermediate structure which is much simpler than the semantic structure but which preserves enough distinctions to permit the surface structure of the sentence to be calculated. I have called this intermediate structure the *prototype structure*. It is calculated by comparisons between the semantic referent and the sentence. These comparisons determine what distinctions in the semantic referent are needed. Only these will be preserved in the prototype structure. The only nodes in the semantic structure that are needed are those representing (i) the proposition nodes (A, B, C, E, and F), (ii) the individual nodes (I and K), and (iii) the words in the sentence (red, square, below, circle, small). Figure 5b gives the prototype structure obtained by deleting all other nodes except these and by only representing the linkage between these critical nodes. Note that, although *above* is part of the HAM structure, it is deleted in the prototype structure. Rather, *below* is the relation term used in the sentence. In addition, the structure encoding the proposition *I is right-of the triangle* is deleted from the prototype. This was not mentioned in the to-be-bracketed sentence. This serves to illustrate an important product of the calculation of prototype structure. The calculation can disambiguate those aspects of a complex referent that are relevant to the sentence at hand. It will frequently be the case that a semantic referent will contain much information irrelevant to the sentence.

Having the prototype structure, LAS attempts to find some graph deformation of it that will provide a tree structure connecting the content words of the sentence. Figure 5c indicates one such graph deformation of the prototype sentence if the main proposition is C. If the main proposition is specified, there is always only one graph deformation of the prototype structure that will yield a surface structure for the sentence. Note that all the links in Fig. 5b are maintained but have been spatially rearranged to provide a tree structure for the sentence. Note that the prototype structure is not specific with respect to which links are above, and which are right-of, which others. Although the prototype structure in Fig. 5b is set forth in a particular spatial array the choice is arbitrary. In contrast, the surface structure in Fig. 5c does specify the spatial relations of links. From Fig. 5c we may derive a bracketing of the sentence indicating its surface structure—((circle small) (square red) below). The details of BRACKET's computations here are unnecessary. Suffice it to say that BRACKET retrieves the graph structure uniquely specified by the requirement that (i) it be a graph deformation of the prototype structure, (ii) it provide a surface structure for the content words of the sentence string, and (iii) its top node be the main proposition.

BRACKET needs to know more than just the prototype structure to infer the surface structure of the sentence. As shown by Figs. 5c and 5d, the same string of words can have the same prototype structure deformed into more than a single surface structure. The difference between Fig. 5c and 5d reflects a decision about which proposition is principal and which is subordinate. The structure in Fig. 5d has F as the main proposition and might be translated into English as *Circular is the small thing that is below the red square*. Therefore, BRACKET also needs information as to what the main proposition is to be able to unambiguously retrieve the surface structure of the sentence. The assumption that BRACKET is given the main proposition amounts, psychologically, to the claim that the teacher can direct the learner's attention to what is being asserted in the sentence. Thus, in Fig. 5c, the teacher would direct the learner to the picture of a red square above a small circle. He would have to assume both that the learner properly conceptualized the picture and also that the learner realized that the aboveness relation was what was being asserted of the picture.

The assumption that the learner can be told what is the main proposition seems a bit strong. It is important to inquire, therefore, what the performance of the program would be like if it were not given information about the main proposition. The first thing to note is that the program could generally make a good guess as to what the main proposition is. For instance, of the five propositions in Fig. 5, only two—C and F—could be main propositions given the ordering of the words in the main sentence.[1] Second, C seems clearly to be the more natural choice because it is the more central proposition. Usually, a few heuristics would suffice to identify the correct main proposition. Moreover, even if the incorrect main proposition is occasionally chosen, this will not do enormous harm to the network grammar induced. This will just introduce an additional possibility in the network and not alter other parsing possibilities. This possibility will not be ungrammatical. Its "defect" will be detected only in that the speech of LAS will occasionally violate pragmatics about how to express presupposed versus asserted information. In conclusion, while the assumption about the availability of main proposition information is convenient, it is marginal to the successful performance of the LAS program.

[1] As the reader may verify, there is no surface structure for the word order in Fig. 5 that is a graph deformation of Fig. 5b and has A, B, or E as a main proposition. Any attempt to make A, B, or E the top node in a graph structure, while preserving the linkage in Fig. 5b, results in crossing of links which violates the requirement of a surface structure. Note that the acceptable propositions C and F are on a path in Fig. 5b connecting the first word *circle* and the last word *below*. In the general case, more than two proposition nodes can be on the path connecting first and last content words. Only proposition nodes on this path can serve as main propositions in the surface structure.

The Details of BRACKET's Output

So far, for purposes of exposition I have simplified the specification of BRACKET's output. Also, the example in Fig. 5 was particularly simple because there were no non-meaning-bearing words. Consider how BRACKET would handle the sentence *The man who robbed the bank has a bloody nose*, given as semantic referent the HAM structure in Fig. 1. (It is left as an exercise for the reader to derive the sentence's prototype structure.) BRACKET would provide the following bracketing:

((The () man (who robbed (the () bank ()))) had (a (bloody) nose ()))

The embedding of parentheses reflects the levels of the surface structure. The highest level of bracketing involves three elements (The () man (who robbed (the () bank ()))), had, and (a (bloody) nose ())). These correspond to the three elements in the main proposition *X have Z* in Fig. 1. The organization that BRACKET imposes on noun phrases will be discussed shortly. BRACKET knows a phrase like *The man who robbed the bank* is a noun phrase because the words in this expression are connected to a node, *X*, in the semantic structure which represents an object. It can tell *X* is an object rather than a relation because of its position in the graph structure. The first noun phrase in this example contains a relative clause, *who robbed the bank*. All embedded clauses are organized in a similar manner as the main clause—that is, with one element in the bracketing for each element in the proposition expressed by the clause. In this case, the relative clause expresses the proposition *X rob Y*. Therefore, its level bracketing (who robbed (the () bank())) contains one element to express *rob* and a noun phrase, (the () bank ()), to express *Y*. It does not contain an element for *X*, as *X* is already expressed in the higher level of bracketing in which the relative clause is embedded.

Note that BRACKET induces a correspondence between each level of bracketing and a single proposition. That is, each level of bracketing expresses one proposition from the HAM network, and will be processed by a single ATN network. Thus, the modularity of HAM propositions is directly contributing to the modularity of the induced ATN networks.

The insertion of nonfunction words into the bracketing is a troublesome problem because there are no semantic features to indicate where they belong. Consider the first word *The* in the example sentence above. It could have been placed in the top level of bracketing or in the subexpression containing *man*. Currently, all the function words to the left of a content word are placed at the same level as the content word. The bracketing is closed immediately after this content word. Therefore, *is* is not placed in the noun-phrase bracketing. This heuristic seems to work more often than not. However, there clearly are cases where it will not work. Consider the sentence *The boy who Jane spoke to was deaf*. The current BRACKET program would return this as ((The () boy (who Jane spoke)) to was deaf). That is, it would not identify *to* as in the relative clause. Similarly, non-meaning-bearing suffixes like gender would not be retrieved as part of the noun by this heuristic. However, there may be a clue to make bracketing appropriate in these cases. There tends to be a pause after morphemes like *to*. Perhaps such pause structures could be called upon to help the BRACKET program decide how to insert the non-meaning-bearing morphemes into the bracketing.

It is also interesting to note that young children when initially learning a language, seem not to pay attention to non-meaning-bearing morphemes (function words) and do not generate these in their speech. Thus, young children manage to avoid the problem of deciding to what constituents non-meaning-bearing morphemes belong.

The output of the BRACKET program is used to dictate the embedding of ATN networks in the grammar. For instance, consider the above bracketing of the network. One ATN network will be built to process the elements at each level of bracketing. For instance, a START network will be built to process the sentence at the top level of bracketing. The first element in the highest level of the bracketing is (The () man (who robbed (the () bank ()))). An arc will be built to process this element in the START network. Because it is a bracketed subexpression a subnetwork will be built to process the noun phrase. The arc in the START network will contain a push to this noun phrase subnetwork. The START network will also contain an arc to process the single word *has*. It will finally contain an arc with a push to a subnetwork to process the subexpression (a (bloody) nose ()).

Discontinuous Elements

There is a class of sentences found in natural language which systematically violate the graph deformation condition. These are sentences with discontinuous elements. Figure 6 illustrates the clearest example of this in English—the *respectively* sentence. Figure 6a shows the HAM semantic structure for the sentence *John and Bill borrowed and returned, respectively, the lawnmower*. Figure 6b shows that there is no way to deform this semantic structure to achieve a surface structure for the sentence. Discontinuous elements are rare in English. Some of the few other discontinuous elements, like *up* in *John called the man up*, do not strictly violate the graph deformation condition because they are not meaning bearing. However, in other languages with freer word order it is possible to find more instances of content words dislocated. Apparently, Latin is a good example of this. For instance, in Latin there is a possible construction that would be reflected by the English word order: *The girl who the boys best saw ran away* where *best*, occurring within the relative clause, modifies *girl*, the subject of the main clause. LAS cannot learn any part of a natural language that involves such discontinuous elements. Fortunately such constructions, while clearly present, are not dominant even in languages like Latin. As a psychologist, I would want to claim that they are not the sort of constructions that are easy to comprehend or that are easily acquired. This certainly seems the case for the *respectively* trans-

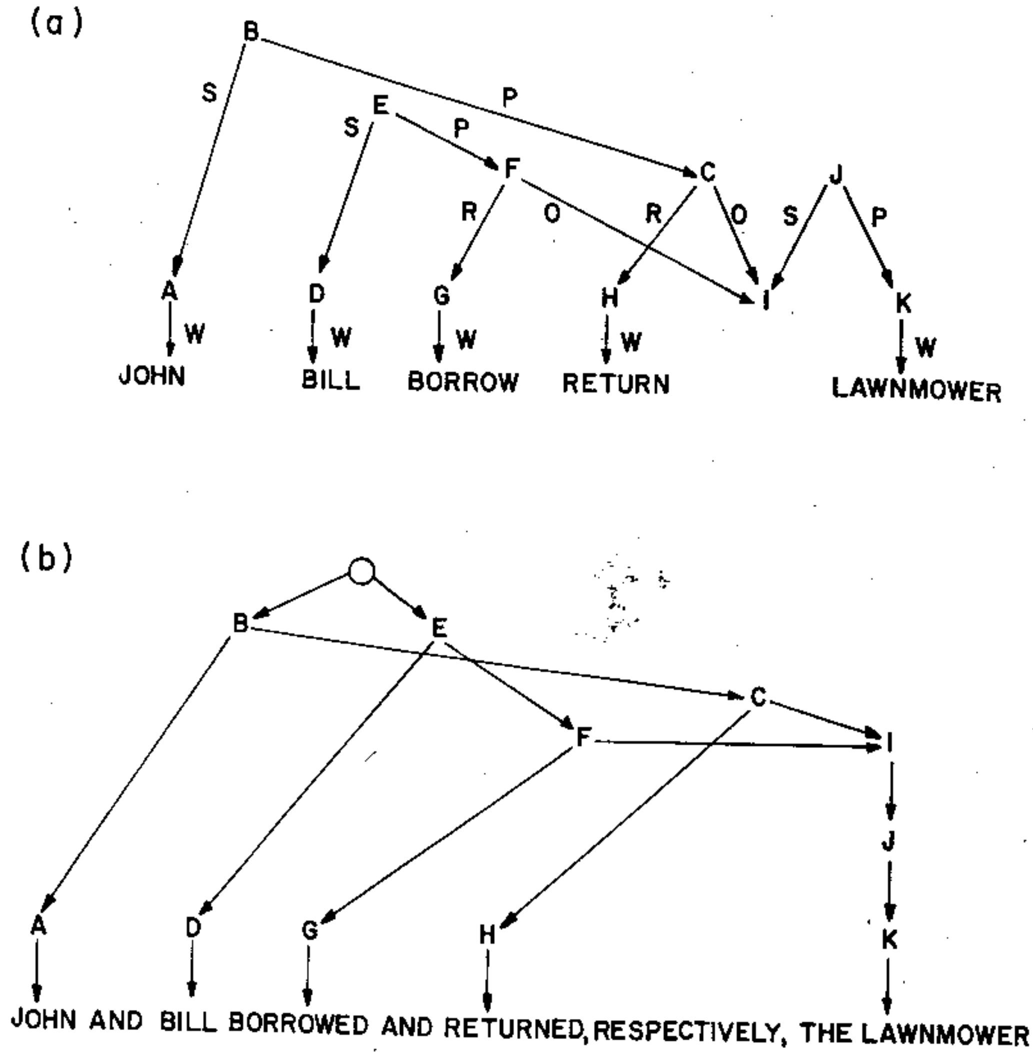


FIG. 6 There is no deformation of the HAM semantic structure in (a) that will provide a surface structure for the sentence in (b).

formation in English. While additional learning mechanisms must be brought to bear to learn discontinuous elements, the LAS mechanism will go a long way toward learning a natural language. Moreover, if it is shown that discontinuous elements are hard to learn, this would be a significant confirmation of LAS's reliance on the graph-deformation condition.

Note that BRACKET itself does not do any learning. Its function is to preprocess the sentence string into a form more appropriate for language induction. It requires as input a string, a semantic referent, and an indication of main proposition. Also it requires a specification of the word–concept connections that are encoded in the semantic network. It has knowledge of the graph-deformation condition as a relation between word order and meaning structure. This knowledge of the graph-deformation condition is embodied in the computation of BRACKET. As argued earlier (p. 133), providing the word–concept connections is a trivial matter; however, providing knowledge of the graph deformation is a very significant category of advance knowledge. A claim of the LAS program is that acquisition of natural languages (but not all languages) is greatly facilitated by use of this advance knowledge. A psychological claim would be that the graph-deformation condition serves as an innate universal of the variety postulated by Chomsky.

Assumptions about Noun Phrase Structure

As the earlier bracketing illustrates, LAS has built into it a number of assumptions about the bracketing of noun phrases. First, it assumes that all languages will have noun phrase syntactic constructions that serve the semantic function of referring to objects. Second, it assumes that noun phrases in all languages will obey an abstract structure indicated by the following rewrite rules:

NP → morphemes (MOD) *noun* morphemes (MOD)
MOD → *proposition* (MOD)

The obligatory elements in these rewrite rules are italicized. These rules indicate that noun phrases consist, optionally, of some initial non-meaning-bearing morphemes, followed by an optional embedded list of prepositional modifiers, followed by an *obligatory* noun, followed by optional postpositional morphemes, followed by an optional embedded list of postpositional modifiers. The rewrite rule for MOD indicates that modifiers consist of the expression of some proposition modifying the topic, plus an optional right-embedding of another MOD. This information about noun phrase structure is incorporated into BRACKET and is reflected by the embedding it imposes on the noun phrase.

These principles for structuring noun phrases might not seem to have any implications for the structure of language. However they do, in that they assert that there is a noun class of words from which it is obligatory to select a member for every noun phrase. Logically, there need not be this obligatory word class. One could imagine a language in which one could refer to a soft red pillow by any subset of these three terms, including *the soft, the red, the soft red,* as well as *the pillow, the soft pillow, the red pillow,* and *the soft red pillow*. However, all languages seem to have an obligatory noun class for referring to objects. The items in this obligatory class tend to be the functionally significant terms for classifying objects. For instance, little can be predicted about an object from the fact that it is soft or that it is red, but much follows from the fact that it is a pillow.

What serves as a noun is not hard and fast, but will change with context. Thus, while *square* is an adjective when referring to picture frames, it becomes a noun in a geometry class. Similarly, while *red* is usually an adjective it can serve as a noun in Las Vegas.

Note that the noun phrase grammar is built around this obligatory noun. Morphemes and modifiers which occur before the noun do not occur after the noun or vice versa. For this reason identifying the noun becomes the key to unlocking the structure of noun phrases. LAS is given information as to what the functionally significant classifications are in its environment—that is, what concepts will serve as nouns. This is supposed to reflect the outcome of cognitive predevelopment which we do not pretend to model. These cognitive prerequisites are critical because with this information about the noun class, LAS can appropriately structure its noun phrase grammar. This is another contribution of semantics to language acquisiton.

Some colleagues have claimed that providing BRACKET with this much information about noun phrase structure is a form of "cheating"—that the program should learn this information. If one's goal is to produce a program that can learn natural languages and if natural languages all have this structure, then this criticism is clearly not valid. Rather one should feel compelled to use any universals of natural language to improve the performance of the program. On the other hand, if one's goal is to produce an accurate psychological model, the issue is not so clear. Whether one wants to incorporate this knowledge into the program depends on whether one wants to endorse the claim that language learners come to a learning situation with this knowledge about noun phrase structure.

Expansion of Word Classes within a Network

LAS has a procedure for expanding the members of word classes on an ATN arc in a way that serves to permit quite powerful generalizations. I will illustrate this procedure with a particularly simple example. Suppose LAS was given the sentence, *John kicked Mary,* along with a HAM network representation of its meaning. Assuming that the three words were all bracketed together, LAS would construct the following network:

$$\text{START} \xrightarrow{\epsilon N1} \text{S1} \xrightarrow{\epsilon V1} \text{S2} \xrightarrow{\epsilon N2} \text{STOP}$$

where N1, V1, and N2 are word classes created by LAS that initially just contained *John, kicked,* and *Mary,* respectively. The syntactic conditions on these arcs are specifications that the words be in these word classes. The semantic actions associated with the three arcs will be to make the concept correspond to the first element subject, the second concept relation, and the third concept the object of the proposition. These actions are placed on the arcs as direct encodings of the role of *John, kicked,* and *Mary* in the semantic referent. Suppose the next sentence LAS encounters is *Fred amused Jane.* This cannot be parsed through the network because *Fred, amused,* and *Jane* are not in the word classes. However, LAS could parse this sentence if the word classes N1, V1, and N2 were expanded to include these terms. This is what in fact LAS will do. Note that this is a powerful principle for generalization. In this example LAS generalizes from the acceptability of two sentences to a grammar that will process eight (either of two words in each of the three positions—$2^3 = 8$).

LAS will parse an expression via an existing path through an ATN network, by just expanding the word classes on the path, if one condition is satisfied. This condition is that the semantic actions associated with the elements in the expression are identical to the sequence of semantic actions associated with the network arcs. In the above example this condition is satisfied: the concept corresponding to the first word *Fred* is made subject, the concept corresponding to the second word *amused* is made relation, and the concept corresponding to the third word *Mary* is made object. This condition on expanding word classes serves to avoid many overgeneralizations that would otherwise occur. For instance, this prevents the word classes in the above network from being expanded to incorporate the three-word sequence *Alice ran quickly,* because *ran* is not a relation and *quickly* is not an object. Another path through the network would have to be built to incorporate this possibility. LAS can tell that *ran* is not a relation in this sentence by inspecting the network referent that comes with this sentence. In that referent, the concept connected to *ran* would be the predicate of the referent, not the relational term.

Note that information about syntactic word class is something that LAS learns. Word classes are created whenever an arc is built for processing a word. These word classes are expanded when a new expression is merged into the network paths that contain these word classes.

Even with semantic editing some overgeneralization does occur in the formation of word classes. Such overgeneralizations are particularly likely to occur in highly inflected languages. Consider the noun phrase network that LAS would construct after hearing the two nominative Latin constructions—agricola longa (a tall farmer) and legatus bonus (a good lieutenant):

$$\mathrm{NP} \xrightarrow{\epsilon \mathrm{N1}} \mathrm{N1} \xrightarrow{\epsilon \mathrm{A1}} \mathrm{STOP}$$

where N1 would contain *agricola* and *legatus* and A1 would contain *longa* and *bonus*. This grammar would generate the noun phrases *agricola bonus* and *legatus longa* which are both incorrect. The noun *agricola* (farmer) is feminine and requires a feminine adjective inflection (i.e., bona). Similarly, *legatus* (Lieutenant) is masculine and requires a masculine inflection (i.e., longus). Clearly, there can be no semantic basis for avoiding this overgeneralization. From the point of view of a psychological evaluation of LAS it is comforting to note that human language learners also fall prey to such morphemic overgeneralizations (see Slobin, 1971).

Since the program overgeneralizes it must be given mechanisms that will enable it to recover from the overgeneralizations once they have occurred. These are relatively simple to form (see Klein, 1973) if the learner is given explicit negative information about his mistakes. However, there is psychological evidence (see Braine, 1971; Brown, 1973) that human language learners get little negative feedback and make little use of what negative feedback they get. Therefore, LAS has not been given such error recovery mechanisms. Development of some psychologically plausible mechanisms for error recovery remains a future goal for the program.

Merging of Networks

Consider the network grammars that LAS would construct to parse SVO sentences like:

The big girl hit the boy.
The dog chases the young cat.
etc.

The START network for the grammar would have the following form:

$$\text{START} \xrightarrow{\text{NP1}} \text{S1} \xrightarrow{\epsilon\text{V1}} \text{S2} \xrightarrow{\text{NP2}} \text{STOP}$$

Note that there is a push to a NP1 network to parse subject noun phrases and a push to a NP2 network to parse object noun phrases. One would like LAS to realize that NP1 and NP2 are really instances of the same network. LAS is constantly checking to see whether phrases that it is parsing by one network could be parsed by another network. If it finds a phrase that can be parsed by two networks, it will use this fact as an indication that the two networks might be capable of merging into a single network. LAS will inspect the amount of overlap between the two networks. If there is sufficient overlap it will merge the networks. LAS derives much of its power because of its principles for merging ATN networks together. This is how it can discover recursive rules—when it discovers that one network can call itself. We will see a number of examples of network merging in the next section.

These principles for merging networks could conceivably lead LAS to overgeneralize in learning a language. Suppose it were the case that some grammatical construction could be processed equally well by either of two ATN grammars. For instance, suppose a particular noun phrase could be parsed by either subject or object noun phrase grammars. This would be a stimulus for merging of the two grammars. However, it might not be the case that all the constructions permitted by one network were permitted by the other. For instance, not all the noun phrase constructions legal in subject position might be legal in object position. Then the grammar will have overgeneralized in merging the two networks.

CASE HISTORY OF LANGUAGE INDUCTION

In the preceding section a number of principles have been identified for language induction. I would now like to illustrate how they will work in combination to induce a grammar. We will observe the program as it induces the subset of

TABLE 2
The English Subset to be Learned

		Grammar
S	→	NP PRED
NP	→	DET (ADJP) Shape (CLAUSE)
ADJP	→	(Size) (Color)
PRED	→	is ADJ
	→	is Relation NP
CLAUSE	→	which PRED
ADJ	→	Size
	→	Color
DET	→	a, the
SHAPE	→	square, circle
RELATION	→	above, below, left-of, right-of
SIZE	→	large, small
COLOR	→	red, blue

Sentences studied

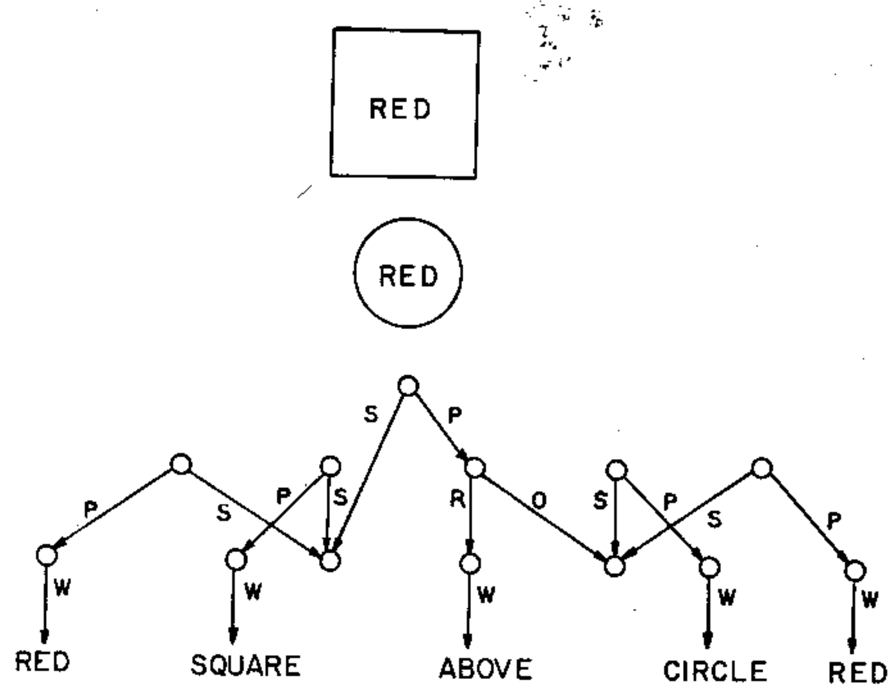
1. The red square is above the red circle.
2. The square is below the circle.
3. A large blue square is left-of the small red square.
4. A small square is right-of a large square.
5. The square which is above the red circle is red.
6. The circle which is red is small.
7. The circle which is right-of the circle is blue.
8. The circle which is blue is large.
9. The square is above the circle which is left-of the blue circle.
10. The blue square is right-of the square which is below the circle.
11. The circle which is small is right-of the circle which is large.

English defined in Table 2. This table describes a rather circumscribed semantic domain. This is a two-dimensional world of geometric objects which vary in the properties of size and color and which may bear various spatial relations one to another. LAS has learned a number of natural and artificial languages, but all have concerned this specific semantic domain. I think it is important to have a well-defined subset of language to learn. It is impossible to take as one's task the learning of an entire natural language. However, one can set as a goal the learning of a subset of a natural language adequate to completely describe a circumscribed semantic domain. The problem with some of the other language learning efforts (e.g., Klein, 1973; Siklóssy, 1972) is that they have taken on the learning of ill-defined chunks of the language. They present a history of the program learning a sequence of sentences, making some generalizations and then the program quits. It is very difficult on the basis of such histories to assess what aspects of the language the program can handle, let alone what aspects it cannot. Hamburger and Wexler (1975) have also made this criticism.

LAS was presented with the 11 sentences given at the bottom of Table 2 in that order. I will go through these sentences one by one and discuss how LAS evolves an augmented transition network grammar to parse these sentences.

Sentence 1

Figure 7 illustrates LAS's processing of the first sentence. LAS is presented with the sentence *The red square is above the red circle,* along with a picture of a red square above a red circle which is analyzed into the HAM structure shown in



((THE (RED) SQUARE ()) IS ABOVE (THE (RED) CIRCLE ()))

START $\xrightarrow{\text{NP}}$ S1 $\xrightarrow{\varepsilon\text{COP1}}$ S2 $\xrightarrow{\varepsilon\text{RA1}}$ S3 $\xrightarrow{\text{NPX}}$ STOP

NP $\xrightarrow{\varepsilon\text{DET}}$ N1 $\xrightarrow{\text{ADJP}}$ N2 $\xrightarrow{\varepsilon\text{NOUN}}$ N3 $\xrightarrow{\text{CLAUSE}}$ STOP

NPX $\xrightarrow{\varepsilon\text{DETX}}$ N4 $\xrightarrow{\text{ADJP}}$ N5 $\xrightarrow{\varepsilon\text{NOUNX}}$ N6 $\xrightarrow{\text{CLAUSEX}}$ STOP

ADJP $\xrightarrow{\varepsilon\text{COLOR}}$ STOP

COLOR = RED	RA1 = ABOVE	NOUNX = CIRCLE	DETX = THE
COP1 = IS	NOUN = SQUARE	DET = THE	

FIG. 7 Upon receiving the semantic referent at the top paired with the sentence, LAS constructed the ATN illustrated at the bottom.

Fig. 7. (Actually, the program is presented with the HAM structure directly.) Comparing the sentence to the HAM structure, the BRACKET program produces the bracketing of the sentence illustrated in Fig. 7. The LEARNMORE program will build a level of networks to reflect every level of bracketing in the sentence. The START network, also illustrated in Fig. 7, was set up to encode the top level of the sentence. The first expression in the sentence is a bracketed subexpression, and therefore the first arc in the START network consists of a push to a NP network to parse the subexpression.[2] On this arc is stored the information that the referent of NP serves the semantic role of subject of the sentence.

Note that the semantic category *subject* comes as a direct encoding of the information in the network referent. That is, the node referred to by the noun phrase is connected to the main proposition by an *S* link. The semantic referent in Fig. 7 is thought of as being the direct output of perceptual processes. Thus, LAS embodies the claim that the semantic categories which are used in language are directly derived from the categories of perception.

The next item in the main level of bracketing is the word *is*. A word class, COP1, is set up to hold this item. On the arc a condition that the word be a member of the COP1 word class is placed. There is no semantic action put on this arc. LAS determines that there is no action associated with this word class because *is* is a word not connected to any concept in memory. The third item in the bracketing is *above*. A word class RA1 is set up to contain this word. On the arc is put a condition that determines if the word is in the RA1 word class, and a semantic action that builds the meaning of this word as the relation in the main proposition. The fourth and final item is a bracketed subexpression. A push to network NPX is put on this arc to parse this bracketed subexpression. The semantic action put on the arc makes the referent of NPX the object of the main proposition.

The network NP is set up to parse the first bracketed subexpression (THE (RED) SQUARE()). For the first item, *the*, a word class DET is set up. The condition on the first arc is that the word be out of the DET word class. There is no semantic action associated with this arc. On the second arc a push is made to the ADJP network to handle the bracketed prepositional modifier. A word class, NOUN, is set up to handle the next item, *square*. This word class is made the condition of the third arc and the semantic action is to predicate of the topic of the NP network that is is a *square*. Note that the last expression in the noun phrase subexpression is a bracketing of the null element. BRACKET automatically imposes the bracketing for postpositional modifiers even when there are none. Thus, a final arc is built with an optional push to a CLAUSE network to parse

[2]Actually, the program did not generate labels like NP in building up the network. Rather, it generated nonsense labels. However, I have taken the liberty of replacing the program's nonsense labels by labels I thought were more mnemonic.

postpositional modifiers. The CLAUSE network will not be built until the fifth sentence that contains the first relative clause.

The NPX network for the object position is built up much in the manner of the NP network. Note that LAS has built up two redundant networks for noun phrases. However, it has no way to know this yet. LAS has only placed *square* in the NOUN word class which occurs in NP and only *circle* in the NOUNX which appears in the NPX wordclass. It has no basis for assuming yet that these two word classes will turn out to have the same members. It may be that words that appear in the subject position take a different morphological inflection than words that appear in the object position. LAS will only decide that NOUN and NOUNX are identical when it has expanded these word classes to the point where they have common members. At this point it will also decide to merge NP and NPX networks.

Note that both the NP and the NPX networks push to the same adjective phrase network, ADJP. A single ADJP network is used because, in building NPX, LAS detected that the ADJP network which it had built for NP could parse the expression (RED) which occurred in the expression that NPX was built to parse. The reason why just one adjective phrase network was built, but two noun phrase networks, is that the same word, *red,* was used in both adjective phrases, but two different words *square* and *circle* in the noun phrases. LAS can guess in the case of the adjectives, but not yet in the case of nouns, that they take the same inflections in subject and object position.

Figure 7 illustrates all the network structure and word class information built up after the first setnence. This would be adequate for the program to comprehend that sentence or for the SPEAK program to generate it. However, the grammar, after this first sentence, can handle virtually nothing else. This is not surprising since one sentence offers little basis for comparison and generalization. The one generalization contained in the grammar of Figure 7 is that the prepositional modifiers are optional. Thus, it would successfully parse *The square is above the circle.*

Sentence 2

Figure 8 summarizes the processing of the next four sentences. The second sentence, after comparison with its semantic referent, was returned in the bracketed form shown in Fig. 8a. This sentence involves use of the relation *below.* Recall that *above* and *below* are attached to the same concept in memory (e.g., see Fig. 5a). The difference between the two is whether the subject or object of the relation comes first. In the case of *below* the object comes first. LAS learns this fact about *below* versus *above* simply by inspecting the order of the noun phrases in the sentence and comparing this with the semantic referent. That is, the noun phrase which it determines as describing the object (by inspecting the arc labels in the semantic network) occurs first in this sentence using *below.* To parse this sentence LAS needs a path through the START network to handle

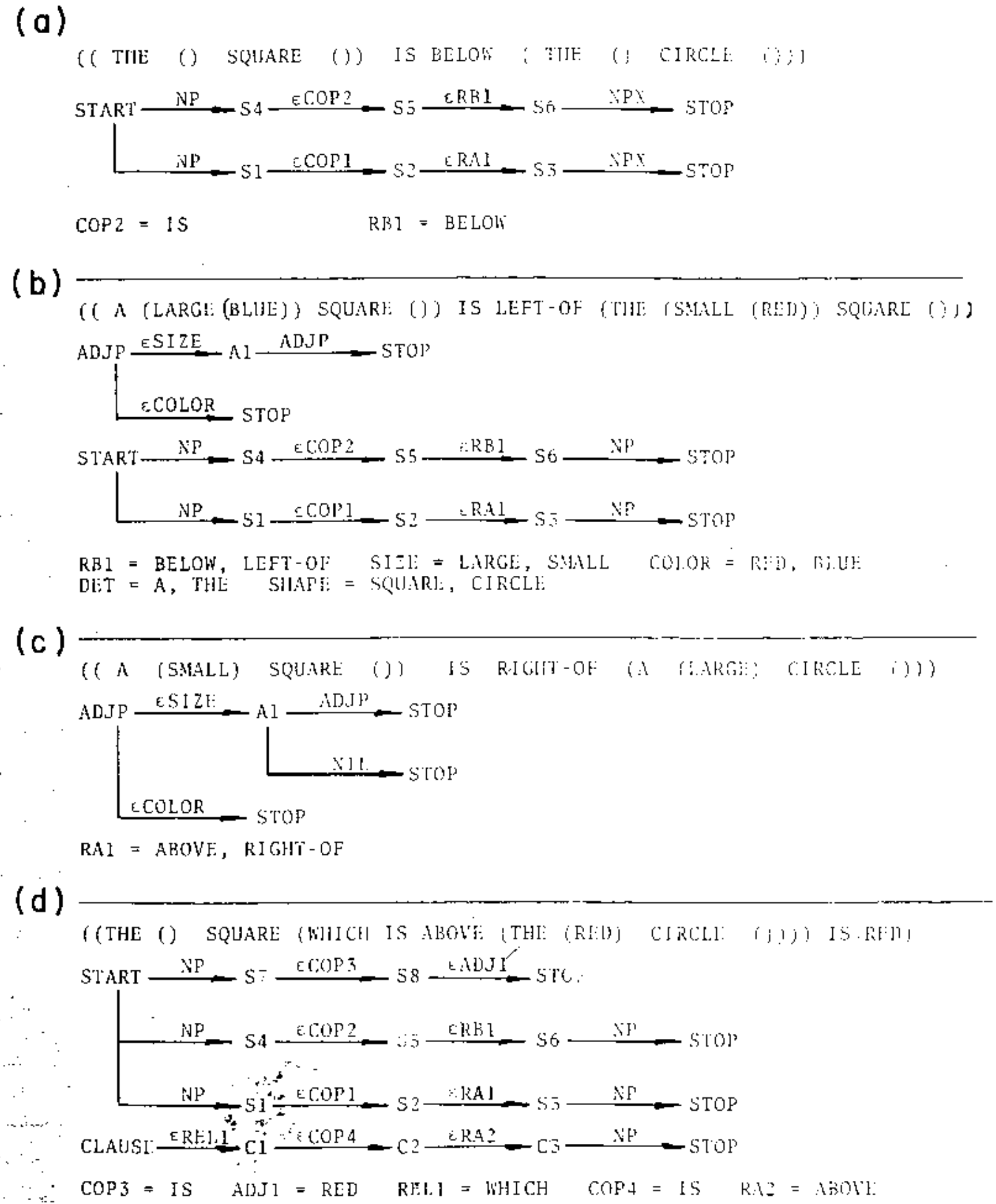


FIG. 8 Parts (a) through (d) illustrate the changes to the ATN as a consequence of the processing of Sentences (2) through (5) from Table 2.

object noun phrase first and subject noun phrase second. This is the opposite of the path built through the network for Sentence 1. A second path, illustrated in Fig. 8a, is built through the START network to accommodate this possibility. Note that the first arc in the path involves a push to the NP network set up to handle the first sentence. The old NP network is referenced, rather than a new one built, because that network can already parse the expression (The () square ()). The NOUN word class in NP contains *square*. Note that NP was chosen and not NPX because NOUNX in NPX does not contain *square*. The second arc in the new START network path references a word class, COP2, that contains *is*. The word class on the third arc, RB1, is set up to hold *below*. Finally, the fourth arc contains a push to a network, NPX. Network NPX was set up for sentence 1. It is referenced in the new path through the network, because LAS has determined that NPX will handle the second noun phrase. The NOUNX word class in NPX contains the word *circle*.

Sentence 3

Figure 8b illustrates some significant aspects of the processing of the third sentence. This sentence involves use of the relational term, *left-of*, which assigns the first noun phrase to the semantic role of object—just as does *below*. Note that the top level of the bracketed sentence consists of (a) a bracketed subexpression serving the semantic role of object; (b) a non-meaning-bearing morpheme; (c) a word indicating relation; and (d) a bracketed subexpression serving the semantic role of subject. This is just the sequence of items on the upper path of the START network. Therefore, according to the principles articulated earlier for induction of word classes (p. 144), it attempts to parse this sentence by the path already existing through the network. This requires that it expand the RB1 word class to include *left-of*.

The first noun phrase, (A (large blue)) square ()), can be parsed by the existing NP network (see Fig. 7), except that the DET word class (first arc of NP) must be expanded to include *a*. This noun phrase requires a push to the ADJP network to parse (large (blue)). This cannot be parsed by the existing ADJP network (see Fig. 7). As indicated in Fig. 8b, a second path is built through the ADJP network. The first arc on this path references the word class SIZE and parses *large*. The second arc contains a push to another network, ADJPX, to parse *blue*. As we will see momentarily, ADJPX is repalced by ADJP.

The second noun phrase should be parsed by NPX (see Fig. 7). However, it cannot do this without enlarging the NOUNX word class to include *square*. In contrast, the NP network will already successfully parse a noun phrase with *square*. This state of affairs is a stimulus for LAS to attempt to merge the NP and NPX networks. This it does, replacing NPX wherever it occurs in the grammar by NP. Another outcome of this merger is that the SHAPE word class is expanded to contain *circle* (from word class NOUNX in network NPX) as well as *square*. LAS has made a significant generalization here—namely, that the grammar that will handle first position noun phrases will also handle second position noun phrases.

The subexpression (small (red)) in the second noun phrase is to be handled by the ADJP noun phrase. The upper path through the ADJP network will handle this expression except the SIZE word class must be expanded to include *small*. The ADJP network will push to the ADJPX network (set up in parsing the first noun phrase of this sentence) to parse (red). This will require expanding the word class in the ADJPX network which so far only includes *blue*. In contrast, there is a path through the ADJP network that will parse this expression with no changes. This is the stimulus to merge the ADJPX and ADJP networks. Thus, as can be seen in Fig. 8b, the ADJP network involves a push to itself. Another consequence of the merging is that the COLOR word class is expanded to include *blue* as well as *red*.

Sentence 4

The effects of processing the fourth sentence are shown in Fig. 8c. This sentence involves the relational term, *right-of*, which takes subject noun phrase first. This can be handled by the lower path through the START network by expanding the RA1 word class to include *right-of*. Note that both noun phrases in this sentence contain adjectives of size. The first arc in the upper path through the ADJP network can parse these size adjectives, but that path expects a bracketed subexpression following the size terms. Therefore, a NIL arc is added to the ADJP network in Fig. 8c to allow size adjectives without subsequent color adjectives.

It is worth emphasizing how much generalization has occurred in formation of the grammar after just four sentences. LAS has generalized a grammar that will handle 5184 sentences. Such generalizations are clearly required if LAS is going to go from a finite corpus to a grammar that covers many more sentences than it studied. Of course, just how rapid the generalizations are will depend on the exact sentences presented. These sentences were chosen to provide rather rapid generalizations.

Sentence 5

The processing of the fifth sentence is illustrated in Fig. 8d. The highest level of bracketing of this sentence consists of a bracketed subexpression, a non-meaning-bearing morpheme, and an adjective. This is a new type of top-level structure. Therefore, an additional path is introduced through the START network. It is determined that the NP network can parse the first bracketed subexpression. Therefore, a push is made to the NP network on the first arc in this new path.

This noun phrase contains a relative clause—(which is above (the (red) circle ())). This is the first time there has been a nonnull expression to parse in the postpositional CLAUSE. Figure 8d shows the path built through the CLAUSE network to accommodate this possibility. Note that in the CLAUSE network a push has been made to NP to parse (the (red) circle ()). Thus, we have the first recursive structure in the network with NP calling CLAUSE which calls NP. On the basis of one right embedding, LAS has made the assumption that infinitely many right-embeddings are possible. As a consequence the grammar has been generalized to the point where it will handle an infinite number of sentences.

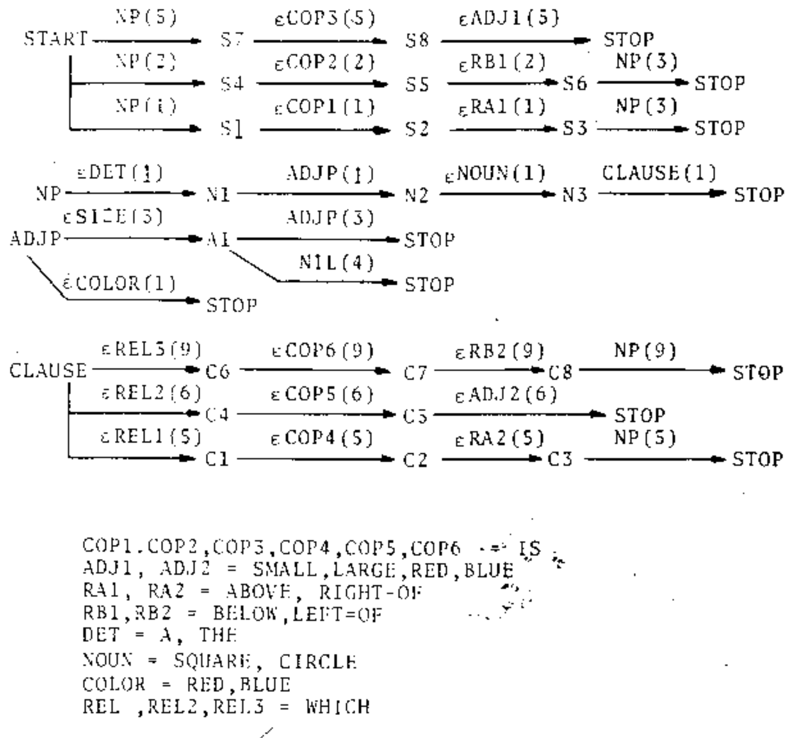


FIG. 9 The network induced by LAS II after studying the 11 English sentences in Table 2.

Sentences 6–11

The remaining sentences cause further additions and generalizations of the variety that have been discussed with respect to the first five sentences. Figure 9 shows the final network grammar induced, a network sufficient to handle all the sentences that can be generated by the grammar in Table 2. The arcs in Fig. 9 are labeled with the number of the sentence that first caused them tø be created.

In the Introduction it was asserted that a goal was to have a program that induced a grammar in a form that could be used for comprehension and generation. A number of tests of the grammar in Fig. 9 have been performed on this score. The grammar in Fig. 9 has been used to generate paraphrases. Also a French ATN grammar was learned by similar means. The two grammars were used to translate back and forth between the two languages. For more detail about these paraphrase and translation tests, see Anderson (1977).

SUMMARY EVALUATION OF LAS

The preceding example shows that LAS can learn portions of the grammar of a natural language. As mentioned above, LAS has learned a French subset as well as an English subset. By "learn a grammar" I mean that LAS induces three things: (a) the word classes used in the language; (b) a context-free grammar specifying the permissible sequences of word classes; and (c) a set of rules mapping between phrases in a sentence and propositions in the semantic referent. It embodies this knowledge in an ATN grammar that can be used both for sentence comprehension and sentence generation.

It remains to be defined what the class of languages is that LAS can learn. I think that it can, given the appropriate learning circumstances, learn any context-free language. However, this answer will prove to be less than satisfactory for two reasons. First, the learning program is sufficiently complex to make it impossible to provide anything like a formal proof of the conjecture. Second, this characterization of learning ability is purely syntactic, whereas we want some characterization that also takes semantics into account. That is, we would like to know what relations the program can learn between sentence and semantic referent.

My conjecture is that, given any context-free language, one could design a presentation sequence and semantics such that LAS could learn that language. I will describe the characteristics of the presentation sequence and semantics needed to achieve language learnability. The presentation sequence must, obviously, consist of sentences and their semantic referents. There must be no grammatical mistakes in this sequence, and the sequence must give examples of all the grammatical structures in the language. It would be easy enough to construct such a presentation sequence.

The semantics for the to-be-learned language would have to be constructed with care. There must be no syntactic dependencies which do not have semantic correlates. Otherwise, the overgeneralizations will occur that were discussed earlier (p. 145). The semantics associated with natural languages largely but not totally satisfies this requirement. Also the semantics must not associate the same interpretation with identical strings generated from distinct syntactic units (see discussion on page 146). Otherwise, LAS will incorrectly merge the grammars for these two syntactic units. Another requirement is that the semantics will have to satisfy the graph-deformation condition. The semantics must also be constructed so that there are no non-meaning-bearing morphemes which cannot be correctly placed in the bracketing by LAS's heuristics. Finally, LAS requires that noun phrases have a certain syntactic structure. This will be satisfied if the semantics identify as noun phrases only those objects that have the structure LAS expects of noun phrases.

I think it would be possible to construct, for any context-free language, a semantics that satisfies these requirements. As noted many times, these requirements are largely but not completely satisfied by the semantics associated with natural language. This is one way that LAS is an incorrect model of natural language learning—it assumes more of the semantics of natural language than they provide. Second, LAS is inadequate because there are a few aspects of natural language (such as the *respectively* construction) that cannot be captured with a context-free grammar. The weakness of LAS on both of these scores is sufficiently minor that I am of the opinion that LAS-like learning mechanisms, with the addition of some correcting procedures, could serve as the basis for language learning.

REFERENCES

Anderson, J. R. Language acquisition by computer and child. Tech. Rep. No. 55, Human Performance Center, University of Michigan, 1974.

Anderson, J. R. Computer simulation of a language acquisition system: A first report. In R. L. Solso (Ed.), *Information processing and cognition: The Loyola symposium*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1975.

Anderson, J. R. *Language, memory, and thought*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1976.

Anderson, J. R. Computer simulation of a language acquisition system: A second report. In D. LaBerge & S. J. Samuels (Eds.), *Perception and comprehension*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1977, in press.

Anderson, J. R., & Bower, G. H. *Human associative memory*. Washington, D.C.: Winston, 1973.

Braine, M. D. S. On two types of models of the internalization of grammars. In D. I. Slobin (Ed.), *The ontogenesis of grammar*. New York: Academic Press, 1971.

Brown, R. *A first language*. Cambridge, Mass.: Harvard University Press, 1973.

Carbonell, J. R., & Collins, A. M. Natural semantics in artificial intelligence. *Proceedings of the third international joint conference on artificial intelligence*, 1973.

Chomsky, N. *Aspects of the theory of syntax*. Cambridge, Mass.: MIT Press, 1965.

Clark, E. V. What's in a word? On the child's acquisition of semantics in his first language. In T. E. Moore (Ed.), *Cognitive development and the acquisition of language*. New York: Academic Press, 1973.

Clark, E. V. First language acquisition. In J. Martin & J. C. Marshall (Eds.), *Psycholinguistics series*. London: Paul Elek (Scientific Books), 1975.

Feldman, J. Some decidability results on grammatical inference and complexity. Stanford Artificial Intelligence Project Memo AIM-93.1, 1970.

Gold, E. M. Language identification in the limit. *Information and Control*, 1967, **10**, 447–474.

Hafner, C., & Wilcox, B. LISP MTS programmer's manual. Mental Health Research Communication 302 and Information Processing Paper 21, University of Michigan, 1974.

Hamburger, H., & Wexler, K. A mathematical theory of learning transformational grammar. *Journal of Mathematical Psychology*, 1975, **12**, 137–177.

Horning, J. J. A study of grammatical inference. Tech. Rept. No. CS 139, Computer Science Department, Stanford University, August 1969.

Klein, S. Automatic inference of semantic deep structure rules in generative semantic grammars. Tech. Rept. No. 180, Computer Sciences Department, University of Wisconsin, Madison, May 1973.

Moeser, S. D., & Bregman, A. S. The role of reference in the acquisition of a miniature artificial language. *Journal of Verbal Learning and Verbal Behavior*, 1972, **11**, 759–769.

Moeser, S. D., & Bregman, A. S. Imagery and language acquisition. *Journal of Verbal Learning and Verbal Behavior*, 1973, **12**, 91–98.

Nelson, K. Concept, word, and sentence: Interrelations in acquisition and development. *Psychological Review*, 1974, **81**, 267–285.

Norman, D. A., Rumelhart, D. E., & LNR Research Group. *Explorations in cognition*. San Francisco: Freeman, 1975.

Quillian, M. R. The teachable language comprehender. *Communications of the ACM*, 1969, **12**, 459–476.

Schank, R. C. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology,* 1972, **3,** 552–631.

Schank, R. C. *Conceptual information processing*. Amsterdam: North-Holland, 1975.

Siklóssy, L. Natural language learning by computer. In H. A. Simon & L. Siklóssy. *Representation and meaning: Experiments with information processing systems*. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.

Simmons, R. F. Semantic networks: Their computation and use for understanding English sentences. In R. C. Schank & K. M. Colby (Eds.), *Computer models of thought and language*. San Francisco: Freeman, 1973.

Slobin, D. I. The ontogenesis of grammar. New York: Academic Press, 1971.

Slobin, D. I. Cognitive prerequisites for the development of grammar. In C. A. Ferguson & D. I. Slobin (Eds.), *Studies of child language development*. New York: Holt, Rinehart, & Winston, 1973, Pp. 175–208.

Winograd, T. Understanding natural language. *Cognitive Psychology,* 1972, **3,** 1–191.

Woods, W. A. Transition network grammars for natural language analysis. *Communications of the ACM,* 1970, **13,** 591–606.

Woods, W. A. An experimental parsing system for transition network grammars. In R. Rustin (Ed.), *Natural language processing*. Englewood Cliffs, New Jersey: Prentice-Hall, 1973.

Woods, W. A., Kaplan, R. M., & Nash-Webber, B. The lunar sciences natural language information system: final report. BBN Rept. No. 2378, Bolt, Beranek, and Newman, 1972.