012408

HUMAN PERFORMANCE CENTER
DEPARTMENT OF PSYCHOLOGY
The University of Michigan, Ann Arbor

Language Acquisition by Computer and Child

JOHN R. ANDERSON



Technical Report No. 55

December 1974

THE UNIVERSITY OF MICHIGAN

COLLEGE OF LITERATURE, SCIENCE AND THE ARTS

DEPARTMENT OF PSYCHOLOGY

LANGUAGE ACQUISITION BY COMPUTER AND CHILD

John R. Anderson

HUMAN PERFORMANCE CENTER--TECHNICAL REPORT NO. 55

Preparation of this report was supported by National Science Foundation Grant GB-40298.

LANGUAGE ACQUISITION BY COMPUTER AND CHILD[1]

John Robert Anderson
Human Performance Center
330 Packard Road
University of Michigan
Ann Arbor, Michigan 48104

Contents

		Page
1.	Introduction	1
2.	The General Problem of Induction	2
3.	Formal Analysis of Language Acquisition	10
4.	Heuristics for Language Learning	27
5.	Data about Language Learning	45
6.	The HAM Memory System	66
7.	The LAS Language System	70
8.	The Program LAS.1	83
9.	The Program LAS.2	99

1. Introduction

It is generally conceded to be a rather significant fact about humans that they learn language. Since the early speculations of Solomonoff (1958, 1964) there have been a number of attempts to try to understand what sort of mechanisms might be responsible for language acquisition. Some, although not all of these attempts, have taken the form of computer simulation models. The point of this chapter is to review some of this work, assess its psychological relevance, and present some recent work of my own.

This chapter is divided into a number of sections. Section 2 will be concerned with establishing a formal framework in which to study the general problem of induction. Language acquisition can profitably be seen as a special case of the induction problem. The formal results we will establish in that section will be with us throughout the remainder of the chapter.

In Section 3, a formal analysis will be given of the special problem of language acquisition. In this section the syntactic and semantic approaches will be compared. The syntactic approach attempts to induce a characterization of the language which will permit us to judge which strings are grammatical and which are not. The semantic approach attempts to induce a procedure which will allow the system to translate from semantic referents to sentences that express the referents. The semantic approach is of principal interest in this chapter. In Section 4 I will review various semantics-based heuristics for language induction that have been proposed. Section 5 will attempt to review the psychological literature on language acquisition with the intent of assessing the psychological validity of these mechanisms.

The remainder of the chapter will present my work attempting to develop a viable computer simulation model of language acquisition. This computer model, LAS, uses a memory system called HAM (see Anderson & Bower, 1973). Therefore, Section 6 describes the essential aspects of this memory system. Section 7 describes the LAS system for speaking and comprehension. It uses a variant of Woods' (1970) network grammar. Sections 8 and 9 describe two versions of the induction program. The system is still in development. So this is by no means a final report.

2. The General Problem of Induction

The study of language acquisition is important for a number of reasons. There is a practical purpose in that such research might aid the development of competence in one's language. It is common to find remarks in the psycholinguistics literature (e.g., Chomsky & Miller, 1963) that full competence in a language comes to almost all humans despite their differing experiences and general intelligence. This would seem to deny that any practical benefits could arise from an understanding of language acquisition. However, competence in a language is not an all-or-none affair. Most members of our society suffer deficits in their language ability. The difficulty I have in writing this chapter is witness to the fact that a Ph.D. is no guarantee of perfection in the use of language. If we understood how language is acquired and organized, it might be possible to improve the usefulness of language as a tool for all of us.

The study of language acquisition also derives its significance from a point of view promoted in linguistics by Noam Chomsky. He often formulated problems of linguistic theory as questions about the construction of a language-acquisition device. In his view, the deepest problems of linguistics

were specifying what such a device would have to know in order to acquire a language. As Chomsky saw it, this would constitute specifying the defining (universal) features of natural language.

From my own point of view, the principle significance of language acquisition is that it is a paradigm case of the induction problem. The question of induction is central to my attempt to develop a general model for human intelligence. Other induction tasks studied by psychologists (e.g., concept formation, pattern formation, sequence learning, rule induction) are all much simpler (at least, in the laboratory experiments that study them) than language learning. The consequence is that the theories that evolve for these tasks tend to not be powerful enough to handle the induction problem in its full generality. An adequate set of induction mechanisms for language acquisition, however, will be much less vulnerable to such problems of logical adequacy.

In Section 2 and 3 I will be drawing heavily from the results and the spirit of analysis in two papers by Gold (1965, 1967). The formally trained reader is urged to consult these original papers. I think they are essential to anyone interested in the problem of induction. My presentation does not do full justice to Gold's work.

I have borrowed from him techniques for analyzing a formal framework in which all induction problems can be conceptualized. Language learning is just a special case within this framework. The basic problem of induction is to learn to respond to input with the desired output. In terms of language acquisition, this means learning to respond to a semantic intention with the sentence that communicates that intention. In terms of concept formation, this means appropriately labelling a stimulus array. In terms of sequence extrapolation, this means responding to an initial part of a sequence with

the correct ending. In terms of playing chess, this means responding with the correct move to a board position. In terms of riding a bike, this means responding with compensatory adjustments of weight to changes in speed and vertical orientation of the bike. In terms of being a scientist, it means constructing the correct hypothesis in response to a set of data.

This analysis of induction may seem vacuous, but it is all that is needed to make some important points. The analysis consists of three things: a learner, the desired input-output relation to be learned, and the learner's current hypothesis about that relation. In Gold's framework each of these objects is embodied by a formal machine. Consider Figure 1. Here we have an Induction Machine that is trying to build an Hypothesis Machine which will behave like a Target Machine. The Induction Machine is a formal embodiment of the learner; the Target Machine embodies the desired relation; and the Hypothesis Machine the current hypothesis of the learner. The Induction Machine can submit input to the Target Machine and observe the output. After each such experiment it produces a Hypothesis Machine. It should be emphasized that the input to the Induction Machine is the input-output pair of the Target Machine. The output of the Induction Machine is the Hypothesis Machine. The induction problem is considered solved when the Hypothesis Machine mimics the behavior of the Target Machine for all future input.

Insert Figure 1 about here

What does it mean to submit input to the Target Machine and observe the output? This assumes that our human learner can try out various responses to a particular stimulus until he finds the correct one. Once he has done

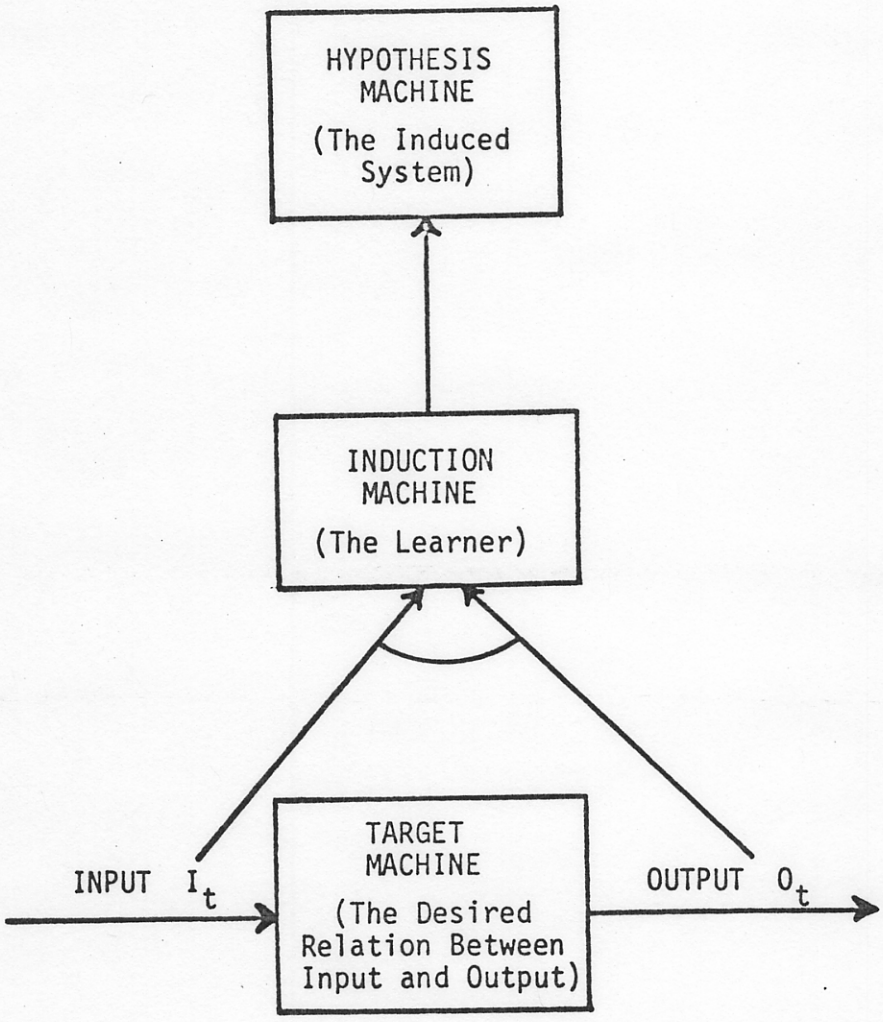


Fig. 1. Machine representation of the general induction problem.

this he has observed one input-output sample from the Target Machine.

The machine framework in Figure 1 is an attempt to abstract the logical essence of the induction problem. We speak of machines because we want to call on formal results from automata theory and related disciplines. It turns out that one can say some interesting things about what induction problems are tractible and what problems are not. These statements depend only on the logical character of the problem and not on the nature of the device that is performing the induction.

The Ahistorical Case

There are two possibilities about the relation between input and output. In one case, the output at time t depends only on the input at time t. In the other case, the output depends on the past history of inputs. The first, the ahistorical case, is much more tractible than the second, the historical case. I will focus on the ahistorical case because it seems more representative of the psychological situation (see Gold, 1965, for the historical case). The correct interpretation of a sentence does not depend on all of our past experience with the language. It might depend on the past few sentences, of course. However, as long as the response depends only on a fixed finite portion of the immediately preceeding inputs, the induction problem is equivalent to the ahistorical case. That fixed portion of the past input can be regarded as the input at time t.

The basic induction problem is for the Induction Machine to produce a Hypothesis Machine which is the same as the Target Machine. There is a sense in which this cannot be done for many interesting classes of machines. This is because two different machines can display the identical behavior.

But this clearly is not an interesting problem from the point of view of human induction. It should suffice that the Hypothesis Machine display the same behavior as the Target Machine. It is meaningless to require that it have the same form since the Target Machine was only a convenient fiction for formalizing the desired relation between input and output.

Identification in the Limit

We will say rather that the Target Machine is identified in the limit if, after some finite time, a Hypothesis Machine is constructed that mimics the Target Machine and the Induction Machine does not change the Hypothesis Machine thereafter. With this criterion for success at the induction problem, we can ask the question of what classes of machines are identifiable in the limit. By this it is meant the following: Suppose we know the machine comes from class C. Is there an effective algorithm for identifying in the limit which machine it is?

As an example, suppose the Target Machine can be any Turing Maching (TM). Can it be identified in the limit? The answer is No. The reason has to do with the Halting Problem for TMs. Since it is not possible to specify whether a TM will halt for a particular input it is not possible to decide whether the responses of a Hypothesis Machine matches the observed response of the Target Machine. Therefore, it is not possible to reject that Hypothesis Machine.

However, we can restrict a class of machines to those TMs for which the Halting Problem is solvable. These TMS happen to correspond to a well defined set of functions in the formal study of Arithmetic (see Minsky, 1967). These are the primitive recursive functions. So we will call these Primitive Recursive Turing Machine (PRTMs). It turns out that PRTMs can be identified in the limit.

The data for the induction problem is a sequence of observed inputs and outputs. It will be assumed that this sequence is not under the control of the Induction Machine. It will also be assumed that somewhere in the infinite sequence is to be found all possible input-output pairs. There may be repetitions in this sequence. Assuming this little about the information sequence it can be shown that PRTMs are identifiable in the limit.

The proof is fairly simple. Consider the following algorithm: There is an effective enumeration of all PRTMs. By this I mean it is possible to order all PRTMs to correspond to the positive integers in such a way that an algorithm can be specified for generating the nth PRTM. This enumeration gives an infinite sequence of PRTMs. For instance, the machines could be ordered by an alphabetic enumeration of the rules which define the machine. Despite the fact that this enumeration is infinite, each PRTM has some finite position in it.

The algorithm starts out considering the first PRTM as the Hypothesis and stays with it until it finds some input-output pair that is inconsistent with it. The algorithm then searches for the next PRTM in the ordering which is consistent with all the existing data. In general, it stays with a PRTM until inconsistent evidence is found and then proceeds to the next consistent machine in the enumeration. Since the information sequence contains all input-output pairs it will reject any incorrect PRTM after some finite time. Since the correct PRTM occupies some finite position in the ordering and since each incorrect PRTM preceeding it will be rejected after a finite amount of information, the correct PRTM will be uncovered after a finite time.

This is called the enumeration algorithm. If the reader doesn't like it, I don't blame him. It obviously does not correspond to a

psychologically adequate model. In addition, it is hopeless as a useful means of induction. The position of the correct PRTM in any obvious ordering of PRTMs is almost certainly so astronomical for most induction problems that it could not be computed by the fastest computer after centuries of computing.

However, there is a startling fact which can be proven about this enumeration algorithm. There is no other algorithm uniformly faster than it. To make sense of this last statement we need a definition of what constitutes a uniformly faster algorithm. Algorithm A is uniformly faster than Algorithm A' if (a) A will identify all target machines at least as fast as A' and (b) A will identify at least one target machine faster.

Suppose there was an algorithm, A, which was uniformly faster than the enumeration algorithm, E. Then there must be some machine, M, which it identifies faster. Let n be the trial at which A first guesses M. On that trial E must guess some other machine M_1. M_1 is consistent with the input-output so far observed. Suppose the target machine really were M_1 rather than M. Then E would identify M_1 by trial n. If A is to be uniformly faster than E it must have identified M_1 on an earlier trial n_1. We can look at E's guess on n_1. It will be M_2 and it could have been correct. A must have guessed M_2 on an earlier trial n_2. I don't think there is any need to continue. As the reader can see, eventually we are going to reason our way back to the first guess. By that time E must have made a guess before A and that guess could have been correct.

Computational Aspects of the Enumeration Algorithm

There are a number of aspects of the enumeration algorithm which make it undesirable. First, it must remember all past input-output pairs so that its

current guess is consistent with all past information. Such a memory is unreasonable psychologically and would also prove burdensome in a computer implementation.

Suppose the enumeration is considering machine M_i and its guess is disconfirmed by the next input-output pair. Then it may have to skip through many million machines before it comes to the next machine compatible with the input so far. Each of these machines would have to be tested with the existing information and rejected. So, while in the abstract sense it is just one step from Trial n to n+1, computationally that can be a very complex process.

It would be useful if the space of possible machines were so organized that the incorrect alternatives could be rejected without consideration. For instance, it might be able to organize the machines according to some sort of tree structure. On the basis of one piece of evidence, a whole branch of the tree might be rejected. This sort of structuring has been done with some success with simpler machines such as (finite state machines) (see Biermann, 1972 for an example). This will not reduce the number of trials to success, but will cut down on the amount of computation per trial.

Decreasing Number of Trials

It is important to cut down on the number of trials required before the correct machine is identified. There is basically only one way to do this. One must have the induction algorithm try first machines which are more likely. This means, of course, that less likely machines will be identified even later by this algorithm than an algorithm which does not use an ordering by plausibility. However, it is a price one should be willing to pay. The simple fact is that most PRTMs are not plausible candidates as Target Machines.

For instance, one can consider one of the many computer programs for generating random numbers. These are really not random and could be embodied by a PRTM. A human faced with identifying the principle used by such a PRTM would almost certainly end in failure. Therefore, we would not want to have to consider such a PRTM. This can be accomplished either by choosing an algorithm which would never guess that PRTM or by choosing an algorithm which would postpone it until more plausible hypotheses had been rejected. The first task amounts to restricting oneself to a subclass of PRTMs. The second amounts to using an enumeration of PRTMs ordered by plausibility.

It becomes an interesting question as to what defines the plausibility of a PRTM. One frequent suggestion is that it is done on the basis of simplicity. Some metric is computed measuring the number and complexity of the rules used by the machine. Machines are considered in order of their complexity. However, this does not seem the correct interpretation of plausibility. The random number generators are often very simple. Certainly, they are much simpler than a machine adequate to understand natural language. However, the random number generator is not considered by a human while he does eventually induce a language comprehension system.

3. Formal Analysis of Language Acquisition

The purpose of this section is to continue the formal analysis of induction of the preceeding section but to specialize it for the case of language acquisition. First, we will consider the results that characterize the syntactic approach. The goal in this approach is to induce an algorithm that will be able to identify the grammatical sentences of the language and separate these from ungrammatical strings. While this syntactic approach has

dubious psychological relevance it was the first formal approach and results from it can be transferred in part to a semantic approach to language acquisition. The goal of the semantic approach is to induce an algorithm which permits one to go from a sentence to a characterization of its meaning and also to go from meaning to sentence. This is clearly much closer to the psychological problem of language acquisition. A syntactic characterization of the language emerges as a by-product, only, of the map acquired between sentence and meaning. Ungrammatical sentences are those which do not properly map into an interpretation.

Syntactic Approach

An early, important paper in the field was by Gold (1967). He provided an explicit criterion for success in a language induction problem and proceeded to formally determine which learner-teacher interactions could achieve that criterion for which languages. The framework in which he places the problem is similar as the framework given in Figure 1. (Indeed I developed the approach in Section 2 as a generalization of Gold's approach to language induction.) In this framework, the learner can observe strings input to the Target Machine and see the output of that machine. Since the Target Machine only makes grammaticality judgments a binary output is required - 0 for ungrammatical and 1 for grammatical. The task of the inducer is to develop a Hypothesis Machine which mimics the behavior of the Target Machine. To do this is to induce a syntactic characterization of the target language.

The psychological equivalent of this abstract model is a learner who hears strings of words marked as sentence or non-sentence. It is easy to see how the human learner would receive the positive information about what is a sentence. Every time someone utters a sentence he can assume it is grammatical.

It is much less clear from where his information about non-sentences comes. We might consider him trying out non-sentences and being corrected. However, this does not jive with the facts of the matter (Brown, 1973; Braine, 1971). Language learners do not try out all combinations of words. Second, they are often not corrected when they utter a syntactically incorrect sentence. Third, they appear to make little use of what negative feedback they get. However, it needs to be emphasized that this framework assumes that the human learner gets both positive and negative feedback.

The results from the previous situation can be translated to this situation. Any language which can be recognized by a PRTM can be induced. This includes the class of context-sensitive languages and the sub-classes of context-free and finite-state languages. It does not include all transformational languages as some of these require a full-powered TM for recognition.

Also the results about learning time can be directly translated to the language acquisition problem. For any sequence of information about sentences and non-sentences there is no algorithm uniformly faster than the enumeration algorithm. Again it is necessary to make assumptions about what are likely languages and to construct algorithms which consider these first.

Positive Information Only

One might wonder what would happen if the inducer were only given positive examples of sentences in the language. This seems to be the information that a human receives. Gold shows that not even the class of finite state languages can be induced given such an information sequence. The proof is deceptively simple. Among the finite state languages are all languages of finite cardinality (i.e., with only finitely many sentences). Suppose the learner is given a sequence of positive instances consistent with an infinite cardinality finite state language. At every point in this information sequence

the learner will not know if the language is generated by one of the infinite number of finite cardinality languages which includes the sample of sentences seen so far or an infinite cardinality finite state grammar which includes the sample. Logically, it would be either. Negative information would get the learner out of this bind because it would allow him to reject any incorrect language after finite time.

Thus positive information only is useless. This result is disturbing because we know that the human learner does not have negative information available to him. We will shortly see that the solution to this dilemma falls out when we take into account the role of semantics.

There are a number of circumstances in which a syntax-based approach will yield identification of the target grammar without negative evidence. All require that stronger assumptions be made about the language learning situation. It is worthwhile to review these. So far we have considered only the possibility that the sequence of sentences are randomly presented to the informant. However, suppose there is a principle in the sequencing of sentences. Then Gold shows us that there is a sense in which the language can be learned. Suppose that some PRTM is deciding according to some algorithm what the nth sentence should be in the information sequence. Then we can transform the grammar induction problem with positive information into something tractable. Rather than trying to induce the grammar, try to induce the PRTM which maps the n into a choice of a sentence. Given the results of Section 2 this PRTM is identifiable. Then one has a semi-effective algorithm for recognizing a sentence of the language. Given a string s, proceed to compute the output of the induced PRTM for the integers. For each integer, the induced PRTM will generate a sentence s'. The recognition algorithm

checks if s' is the target string s. If it is, the algorithm stops and recognizes the sentence. If s is not in the language the algorithm will go on forever. Thus, it can recognize sentences but cannot identify non-sentences. For this reason the algorithm is referred to as semi-effective.

It is very important to understand why one cannot directly induce a recognition algorithm for the language from positive instances but can induce the PRTM that is enumerating the positive instances. In the first problem one is not presented with all possible inputs for the recognition algorithm. That is, one only sees grammatical strings not ungrammatical strings. In contrast, one does eventually see all the inputs for the enumerating PRTM. That is, one sees all the integers n where n is the position in the sequence. The induction problem of Figure 1 can only be solved when the inducer has eventual access to all possible inputs.

Horning (1969) provides a version of the positive-information-only case which has a solution. He achieves this by assigning probabilities to grammars in light of the observed sample of sentences. He assigns to each rule in a grammar a certain probability. The case he most considers is the one in which the rewrite rules involving a particular non-terminal are equi-probable. For instance, consider the following simple grammar and example derivation in it:

$S \rightarrow X\ S\ Y \quad Pr = 1/2$

$\rightarrow c \quad Pr = 1/2$

$X \rightarrow a\ X\ a \quad Pr = 1/2$

$\rightarrow b \quad Pr = 1/2$

$Y \rightarrow b\ Y\ b \quad Pr = 1/2$

$\rightarrow a \quad Pr = 1/2$

$S \rightarrow X\ S\ Y \rightarrow a\ X\ a\ S\ Y \rightarrow a\ b\ a\ S\ Y$

$\rightarrow a\ b\ a\ c\ Y \rightarrow a\ b\ a\ c\ a$

Each of the non-terminals (S, X, and Y) are involved in two rewrite rules. Therefore, each rule has probability 1/2. The reader may verify that the probability of this particular string, abaca, is 1/32.

Horning assumes that the information sequence being observed is being produced by a stochastic device which is generating sentences according to these probabilities. His goal is to select the most probable grammar in light of the evidence.

In deciding the most probable grammar, Horning has to select the grammar G_i which maximizes the following quantity: $P(G_i/S) = \frac{P(G_i)P(S/G_i)}{P(S)}$

where $P(G_i/S)$ is probability of the grammar G_i given the sample S, $P(G_i)$ is the prior probability of the grammar; $P(S/G_i)$ is the probability of the sample given G_i, and P(S) is the prior probability of the sample. This is of course, just Bayes Theorem. To maximize $P(G_i/S)$, a algorithm must maximize the combination of $P(G_i)$ and $P(S/G_i)$. The value of $P(S/G_i)$ can be computed from the probabilities assigned to individual rules in the grammar. It is less obvious how to measure $P(G_i)$. Horning introduces a "grammar-grammar" which generates grammars in a canonical form. By associating probabilities to the rewrite rules in the grammar-grammar it is possible to assign a probability to a grammar derived from it - just as it is possible to assign probabilities to sentences derived from a grammar.

Horning proves that it is not only possible to select a correct grammar in this situation, but also to select the most probable grammar of the equivalent correct grammars. The reason is that we again have a situation in which each possible input is assigned an output. That is, in this case each string of words is assigned a probability which is its relative frequency in the sample. The situation is tricky because these probabilities are

constantly being updated as new sentences come in. Nonetheless, with sufficient time the observed probabilities of sentences will come arbitrarily close to their true probabilities.

Horning also shows that in selecting the most probable grammar he is also selecting the simplest. Here simplicity is measured by the number of rules in the grammar and the number of applications of rules needed to derive sentences in the sample. Clearly, the more rules in the grammar, the lower the probability of deriving it from the grammar-grammar. The more applications of rules required to derive a sentence from the grammar, the lower the probability of the sentence in the grammar. Thus, Horning's procedures can be interpreted as trying to maximize the simplicity of a grammar which is a frequently stated linguistic goal. That probability and simplicity should be related is just what one would expect from information theory.

I think Horning's work represents the best of the syntax-based approaches. He shows how, by trying to optimize probability or simplicity, one can in principle induce any context-sensitive language from positive information only. He has also done some work (which I have not reviewed) on improving the efficiency of the induction algorithm over the case of pure enumeration. However, his methods are basically enumerative and, despite all his impressive advances, hopelessly inefficient.

There are two ways one might go to improve efficiency. One is to work with greatly restricted sub-classes of languages and so avoid the magnitude of the combinatiorial explosion he faces - even when he works with finite state grammars. This may be possible in some applications, but I think it is clear from the work in linguistics, that the syntactic structures of natural languages are very diverse and do not fall into a narrowly circumscribed class.

The other method is to make available other non-linguistic information which helps indicate the structure of the language.

For instance, one idea that has appeared in the syntax-based work, is that induction of context-free languages would be much easier if information was given as to the phrase structure of sentences. Both Pao (1969) and Crespi-Reghizzi (1970) have developed relatively efficient algorithms that made use of such surface structure information. In fact, for a restricted sub-class of languages (a special subset of operator-procedence languages) Crespi-Reghizzi was able to obtain language identification with only positive information. The problem with their work is that this information about phrase structure is provided in an ad hoc manner. The sentence is provided to the induction algorithm bracketed to indicate surface structure. This has the flavor of "cheating" and certainly is not the way things happen with respect to natural language induction. I will show in later sections how this surface structure information can be inferred by comparing the sentence to its semantic referent.

The Semantics Approach

The importance of semantics has been very forcefully brought home to psychologists by a pair of experiments by Moeser and Bregman (1972, 1973) on the induction of artificial languages. They compared language learning in the situation where their subjects only saw well-formed strings of the language versus the situation where they saw well-formed strings plus pictures of the semantic referent 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 aid of any referent pictures. After

3000 training trials subjects in the no-referent condition were at chance in the criterion test, whereas subjects in the referent condition were essentially perfect.

Results like these have left some believing that there must be magical powers associated with having a semantic referent. I would like to pursue a more sober and rigorous analysis of the contribution of semantics. To do this requires that we set up the formal framework in Figure 2. The situation depicted there is a Target PRTM which generates sentences (y_t) in response to semantic referents (x_t). The inducer has access to the semantic referents and the sentences. It produces as guesses $\underline{M_t}$ which are guesses as to the structure of the target PRTM. The $\underline{M_t}$ are also PRTMs which map semantic referents into sentences.

Insert Figure 2 about here

It is important to understand what are the psychological correlates of the objects in this formal framework. The Induction Machine is the language learner. He observes situations in his environment. His perceptions of these situations constitute the semantic referents. He hears sentences spoken to describe the semantic referents. It is the pair of these perceptions and sentences which are the basic data for the learner. The Target Machine formalizes the desired relationship. The learner's task is to construct a system which will produce appropriate sentences to semantic referents. The Hypothesis Machine $\underline{M_t}$ formalizes the current state of that induced system.

The semantic referents have an internal structure. In my applications I will regard this structure as being basically that of a HAM propositional

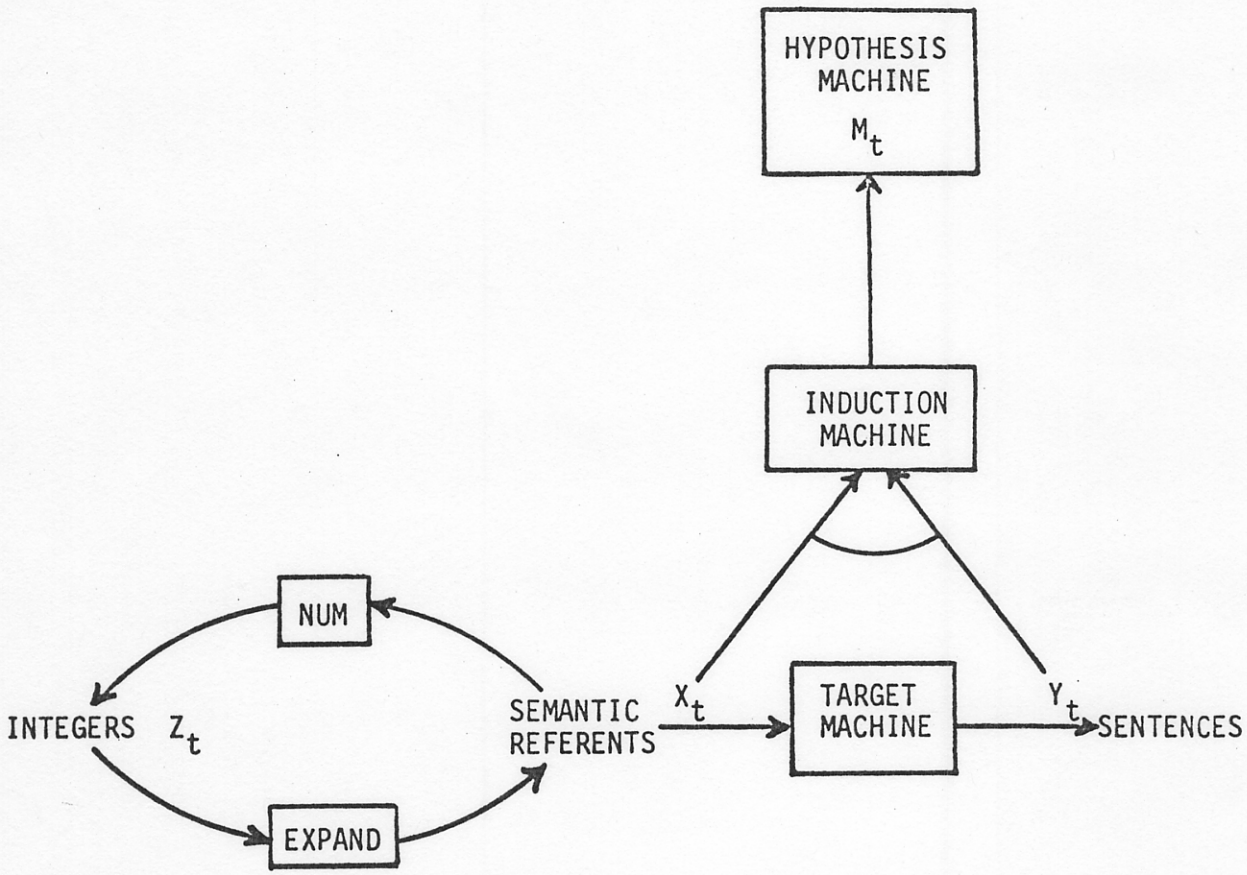


Fig. 2. Machine representation of the semantics approach to language induction.

network (Anderson & Bower, 1973). However, for purposes of this formal analysis it is not necessary to specify what the structure is. It is only assumed that there is a semantic grammar, SG, which generates these semantic structures. Also it is assumed that the inducer knows SG. It is also assumed that SG is sufficiently simple that there is an effective recognition procedure for it. For instance, SG might be a context-sensitive grammar.

Positive Information

It can be shown in this context that the inducer can learn the language given only positive information about what are sentences of the language - provided the learner receives an information sequence in which he will eventually hear the appropriate sentence for all semantic referents. Since the inducer knows SG he has a procedure for enumerating all the semantic structures. That is to say, he can generate each semantic referent. This means, in addition, that the inducer has a procedure for assigning integers to semantic referents. Each semantic referent gets as a number its position in the enumeration. There is a procedure NUM which assigns semantic referents to numbers and a procedure EXPAND which gives the semantic referent corresponding to a particular number. Both of these functions can be calculated by PRTMs if SG is context-sensitive.

The importance of NUM and EXPAND will become clear shortly. Note that the semantic referents are concatenations of symbols but that not all concatenations are necessarily acceptable. This follows from the fact that they have rules of formation. Consequently, the inducer will not see all possible semantic inputs; it will only see the well-formed ones. As we saw earlier, it cannot, in general, induce a target TM unless it has access to the responses for all inputs.

This is where NUM and EXPAND come in. They allow us to circumvent the problem. Rather than trying to identify $\underline{T}$, the inducer tries to identify M' = T (EXPAND) which denotes the machine which EXPANDs an integer into a semantic referent and then calls on $\underline{T}$ to produce a sentence for that referent. The following describes the behavior of the inducer. It sees a semantic referent $\underline{x}$ input to $\underline{T}$ and notes the output sentence $\underline{y}$. It computes NUM(x) = z which is the integer corresponding to $\underline{x}$. It then tries to induce the machine which maps $\underline{z}$ into $\underline{y}$. This machine M' = T (EXPAND) is a PRTM and so can be induced given the results in Section 2. Lets call the INDUCER's t-th guess as to the identity of T (EXPAND) $\underline{M_t}'$. From this guess it can form a guess $\underline{M_t} = \underline{M_t}'$ (NUM) which denotes machine $\underline{M_t}'$ applied to the integer corresponding to the semantic referent. Since M' will eventually be identified we know $\underline{M}$ will also be identified. M is the desired machine mapping referent into sentences.

Important in this demonstration is the fact that the grammar SG is known and EXPAND and NUM may be calculated. Without these it would not have been possible to create a tractible induction problem. The reader should understand that NUM and EXPAND are not meant to correspond to psychological processes. They only are brute-force formal means of showing that $\underline{M}$ can be identified if the grammar of the semantic referent is known.

So we have shown that the introduction of a semantic referent makes it possible to work with positive information only. One might suppose that we can also have an improvement over blind enumeration as an induction procedure when semantic information is available. However, it can be shown that there is no procedure uniformly faster than enumeration of all possible relations between the semantic referent and sentences. The proof of this claim was given in Section 2 when we discussed the general induction problem. The reader should also realize that with respect to computational efficiency, the

enumeration algorithm is as hopeless a means for identifying the relation between semantic referent and sentence as it was for learning to recognize well-formed sentences. In point of fact, the space of semantic functions is much richer than of recognition functions. Recognition functions associate with each sentence 0 or 1 while semantic function associate with each sentence one of an infinite number of semantic interpretations. So really the set of possibilities is larger when we consider semantic functions. Therefore, there is a sense in which it would take the enumeration algorithm even longer to find the correct semantic relationship than the correct syntactic relationship.

I find it difficult to appreciate what is being talked about when I mention the space of all possible semantic relations. I am realizing that there are very bizarre relations in that space. For instance, corresponding to every syntactic recognition function R_L for a language one can construct a semantic function, F_L. F_L operates as follows:

1. It takes a semantic referent x and calculates z = NUM(x) which is the integer giving the position of x in the enumeration induced by the semantic grammar, SG.

2. It takes z and converts it to w, which is a string of words in the language recognized by R_L. By a process known as Gödelization (see Yasuhara, 1971) it is always possible to uniquely convert between integer and strings of symbols.

3. It uses the recognition function R_L to detect whether w is a sentence in L. If it is it computes y = w# which denotes the string w concatenated with the special symbol #. Otherwise y = w.

4. The output of F_L is y.

Thus, corresponding to each distinct language L there will be a new, different semantic relation F_L. Of course, there are many other semantic relations. However, the point about the relations F_L is that they provide us with examples of some of the many perfectly absurd relations that exist! The reason why the enumeration algorithm for semantic relations is so slow is that it must consider the absurd candidates. The reason it must is because we have no precise notion of what it means for a relation to be absurd. However, such information must be implicitly encoded into the human induction system or else it would not succeed. Thus, it becomes a significant psychological question to identify what constitutes an absurd vs. a natural semantic relation.

This is the fundamental point about the success of language induction in the human case. It proceeds rapidly because relatively strong assumptions are being made about the relation between semantic referent and sentence. The problem is that no one has clearly formulated what these constraints are. One of the principle tasks that will occupy us throughout the remainder of this chapter is a partial specification of what these constraints are.

4. Heuristics for Language Learning

A frequently made distinction in computer science involves the difference between a heuristic and an algorithm. An algorithm is a computational procedure which is guaranteed, by proof, to provide a solution to a class of problems. A heuristic does not come with such formal guarantees, but does often work. Heuristics are often preferred over algorithms because they can be faster.

We have already seen that no algorithm will be uniformly faster than the enumeration algorithm. Therefore, it is not surprising that one should turn

to heuristics for learning natural languages. This section contains a discussion of a number of possible heuristics. No combination of these heuristics would be sufficient to learn an arbitrary language. However, the claim is that these heuristics will work with natural languages and will result in relatively rapid learning. If the claim is correct, then it is due to peculiarities of natural language. We will review in the next section the evidence that humans bring these heuristics to bear in learning languages. One might argue that humans have developed these heuristics because they are the ones that will work with natural languages. However, I think the matter goes deeper than that: Natural languages are the way they are because these are the sort of languages humans could learn, given their heuristics.

Enumerative vs. Constructive Approaches

In the last section we principally considered enumerative approaches to language identification. These procedures basically enumerate all the possible languages until they come across one which is consistent with the information being received about the language. Their great virtue is that they are guaranteed to find the target language if it is among the class of languages being enumerated. Their basic drawback is that they take so very long. Horning (1969) worked hard at developing heuristics which would make the process computationally quicker. However, he was only able to deal with the most trivial languages. At the end of his dissertation he conceded:

> Although the enumerative procedure . . . is formally optimal, its Achilles' heel is efficiency . . . the enumerative problem is immense; our implementation . . . can infer only grammars of extremely modest proportions within reasonable bounds on computation.
>
> (pp. 151-152)

The alternative is to use constructive methods for language learning. These methods start with an initial grammar of no rules and proceed to add and generalize rules so that they will account for incoming information. The method is called constructive because the grammar is built up rule by rule. It also tends to be a growth process in another sense. The number of strings it will accept generally increases as the learning progresses. The number of sentences acceptable has to grow faster than the number of sentences observed. This is because the goal of language learning is to develop a grammar which will accept an infinite number of sentences after observing only finitely many. The inevitable consequence of the necessary generalizations is the danger of error. The grammar might be over-generalized to accept ungrammatical sentences. This cannot be avoided by any constructive algorithm. The algorithm, in this regard, is at the mercy of the languages it is trying to learn. For any algorithm that is going to generalize one can construct a language for which these generalizations are all wrong.

It is interesting that we only regard a constructive program as making errors when it accepts an unacceptable sentence and not when it rejects an acceptable sentence. This is a consequence of the growth feature of constructive algorithm - that they naturally come to accept more and more sentences. So it is difficult for them to recover from overgeneralizations but it is in their nature to be able to immediately recover from under-generalizations.

The Need for Error Recovery

There are two ways to deal with the problem of over-generalizations in learning natural languages. One is to try to coin a set of learning heuristics

which will not lead to any errors for natural languages. Of course, there are languages for which these heuristics would fail miserably but one could claim that these are not natural languages. This seems to be the approach of Siklóssy (1971, 1972). However, this is not true to the psychological data. Children are notorious for their over-generalizations in language learning.

If one is going to permit the program to overgeneralize then one must specify procedures for recovery from these overgeneralizations. One could either delete the rules which led to the overgeneralization or modify them in some way. The strategy of modifying erroneous rules seems to have been the one evolved by Klein (1970, 1973). Klein modifies rules upon explicit negative feedback from an informant. This does not seem true to the psychological circumstance. It seems rather that more accurate versions of rules slowly evolve and replace the old rules.

I would like to discuss now some principles for rule construction, their generalizations, and the dangers of these principles. I will go through a Japanese example used by Klein (1973). This will serve to illustrate some of the principles of grammar construction. However, the progression through this example will be interrupted by long digressions into matters not dealt with in the Japanese example. Klein's approach is in many ways similar to that of Siklóssy. I have chosen to focus on Klein because I find him easier to understand.

1. Coining Initial Rules

The first sentence in Klein's sequence is:

1) Nara wa hanasi o kiita (Nara heard the story).

In general there is very little that can be done on the input of the first sentence except to encode it in a very particulate rule. So Klein proposes the rule:

S1 → nara wa hanasi o kiita

and stores its semantic interpretation. As we will see in Section 8 it is possible to structure the grammar for the initial sentences so as to facilitate the process of generalization with later sentences. But until one can compare a number of sentences it is pretty well impossible to make any generalizations.

2. Principal of Minimal Contrast

The second sentence in the Klein example is

2) Jon wa hanasi o kiita. (John heard the story.)

Comparing these two sentences, only one contrast is detected. This naturally leads to the conjecture that the two mismatching words (Jon and Nara) belong to a single form class and that they appear in otherwise syntactically identical sentences. Therefore, the following rules are constructed:

S1 → S2 wa hanasi o kiita

S2 → nara, jon

The third sentence is:

3) nara wa jon o kiita (Nara heard John.)

This sentence mismatches with rule S1 at but one point. Therefore, on the basis of this minimal contrast the grammar is again changed to merge jon and hanasi into the same form class:

S1 → S2 wa S3 o kiita
S2 → nara, jon
S3 → jon, hanasi

This generation of classes on the basis of minimal contrast between sentences and grammatical rules like S1. It has produced the first generalization. The grammar above would produce jon wa jon o kiita which it did not encounter. Whether this sentence is grammatical or not I do not know, but one can produce

English examples where the generalization is not valid. Consider the following three sentences:

They are cooking apples
Frenchmen are cooking apples
They are delicious apples

Using the same principle of minimal contrast, this would lead the grammar into accepting

Frenchmen are delicious apples

which is unacceptable. This overgeneralization might have been avoided if reference had been made to the meaning of the three sentences on which the generalization was based. Although they had similar surface syntax, they did not have identical meaning structure. This leads to another principle for grammar induction:

3. Principle of Semantics-Induced Equivalence of Syntax (PSIES)

Only merge rules or components of rules if they serve identical semantic functions. In this way, Klein would be able to generalize in the Japanese case but avoid the problem in the English case. However, it can still be shown that, even with this restriction, generalization by means of minimal contrast can lead to error. Consider the following example:

The boys danced
The girl danced
The girl dances

From these three sentences the induction routine would generalize to the acceptability of The boys dances which violates verb-number agreement. In the three source sentences the subject-verb structure has the same semantic interpretation. Therefore, there is no semantics-based way of avoiding the generalization.

There is no real protection against such overgeneralizations. It is comforting that they also occur in child acquisition of language. Faced with the fact that such overgeneralizations will occur it is necessary to provide the induction algorithm with principles by which to recover. It is not clear what Klein's recovery mechanisms are for this case. I would favor one in which the constant hearing of The boys dance would strengthen a rule that would eventually come to dominate the incorrect, overgeneral rule.

The Principle of Semantics-Induced Equivalence of Syntax (PSIES) can be used as more than a mere check on potential generalizations. It can be used to suggest generalizations. Suppose a learner saw the following two sentences:

The boy kicked a girl
A woman stabbed the man

Suppose, further that the learner already knew the meanings of the words boy, kicked, girl, woman, stabbed, and man. It is not unreasonable to suppose we learn individual word meanings before we learn the interpretations of sentences. Finally, suppose that the semantic structure of these sentences was indicated so the learner could see that they were of the form subject-verb-object. Then, it would not seem unreasonable for the learner to emerge with relatively strong generalizations from this example. He might posit the following grammar:

S → NP V NP
NP → ART N
N → boy, girl, woman, man
V → kicked, stabbed
ART → a, the

The reader may confirm that this grammar has generalized from the acceptability of just two sentences to the acceptability of 128. Note that these generalizations were not made on the basis of minimal contrast. Rather the learner just assumed that similar grammatical constructions which served the same semantic function were interchangeable.

4. Left Generalization

There is a type of generalization advocated by Klein and Kuppin (1970) which is a slight variant of generalization by minimal contrast. It is illustrated by the example below:

I like the boy
I like the girl
I like the girl who is tall

The natural generalization to make from these three sentences is that both boy and girl may occur in the same position and that either may be followed by a relative clause like who is tall.

However, once again this type of generalization can lead to overgeneralizations. Consider the following example

I like the songs
I like the poem
I like the poem which is short

From these three examples, one would generalize to the acceptability of I like the songs which is short.

5. Lexicalization

Klein uses minimal contrasts not just to make grammatical generalizations but also to make hypotheses about the meaning of a word. Suppose two sentences are identical except that where the first has word X, the second has Y. Also

suppose that the meaning representation for these two sentences is the same except that where the first representation has element A the second element has B. Then it is a reasonable hypothesis that word X means A and that word Y means B. Klein's program will make such conjectures. On the basis of the Japanese sentences 1-3 that have been presented, it makes the hypotheses jon = John, nara = Nara, and hanasi = story.

More generally, lexicalization can be accomplished by comparing a grammatical rule with a sentence. So far I have only been talking as if pairs of sentences were being compared. It is more efficient if grammatical rules are compared to incoming sentences. This avoids the need to retain in memory all past sentences. It also makes possible certain minimal contrasts not otherwise possible. Thus the principle for lexicalization can be stated as follows: If a sentence can be generated by a grammar except for one word and if the semantics accompanying that sentence match the semantics associated with the generation by the grammar except for one semantic element, then the meaning of the word is that one semantic element.

At first, this principle seems universally valid for natural language. However, even in this case one can concoct examples where it would fail. Suppose a language learner sees Bill hitting a girl with a stick and hears Bill hits a girl. Another time he sees Bill hitting a girl with his hand, and hears Bill hits the girl. Assume visual inputs to be what the learner regards as the meaning of the sentences. Then he would come to the conclusion a = stick and the = hand. The basic problem here is that the information in the semantic referent and in the sentence may not perfectly correspond. This surely is a realistic difficulty in natural language learning situations. A learner's guess as to the meaning of a speaker's sentence will not always perfectly correspond with what the speaker is conveying in the sentence.

If we assume that there were perfect correspondence between sentence and semantic referent, then the principle underlying lexicalization seems universally safe. This is a testimony to the morphemic structure of natural language. That is, there exist units of a sentence, namely the morphemes, which correspond to units of meaning. However, it would be easy to concoct artificial languages where this principle was not valid.

The principle used by Klein for lexicalization is an obvious one and has been used by others (e.g., Siklóssy). Despite the fact that it seems based on a universal truth of natural language, I do not think it is psychologically valid. First, it does seem unreasonable to suppose a perfectly accurate semantic interpretation. Second, it requires that the learner retain information about meaningless strings of words (either the strings themselves or a grammar that generates the meaningless strings) so that this information can be contrasted with later strings and minimal contrasts noted. I think it more reasonable to assume that children initially learn the meanings of words in contexts which are very simple. For instance, a child sees a book, it is pointed out, and the child hears book. After many such pairings he probably builds up the concept of book and associates it with the word. Probably all initial lexicalization is accomplished this way. Later words may be acquired in context of elaborate sentences but it is only when the child already knows most of the important words in the sentence and there are just one or two unknown words.

6. Category Merging

Note in the Japanese example that the subject category S2 contains jon and nara and the object category S3 contains hanasi and jon. Thus, the two

categories overlap in one word - jon. On this basis, a generalization is made by Klein that they are the same category which is called S2:

S1 → S2 wa S2 o kiita

S2 → nara, jon, hanasi

The underlying assumption is that words tend to participate in only one or a few grammatical categories. Therefore, if one sees a word in two places one can assume the same category occurs in the two places. This clearly is not universally valid. In this very example the generalization leads to the unacceptable sentence hanasi wa jon o kiita.

7. Questioning of the Informant

Klein's solution is to have his program generate the sentence and get feedback from the informant as to its unacceptability. With this negative information he then reformulates the grammar again:

S1 → S2 wa S3 o kiita

S2 → nara, jon

S3 → nara jon, hanasi

This questioning mode does not seem psychologically valid. Moreover, it cannot work generally. Whenever a generalization is made which results in adding an infinite number of sentences to the vocabulary one cannot check them all. It was fortunate that Klein's simulation chose to check the bad sentence and not the sentence:

jon wa nara o kiita

which the informant would have accepted. In any case, by just checking one sentence one cannot hope to assure the validity of a generalization.

I do not think that informant feedback was needed to correct this particular overgeneralization. Its semantic interpretation is unacceptable. Therefore,

such a grammar would never generate the sentence and in attempting to interpret such a sentence, a human speaker would probably not reject it as ungrammatical but rather as nonsense. This is one of the important features about having a semantic referent. Remember the task is to learn a mapping from semantic referents to sentences. Negative information is not necessarily required. The reason it is not is that it is not important what the mapping does with ill-formed semantic referents.

However, there are other situations where category merging will cause difficulties even when there are semantic safeguards. Consider the following three French sentences:

Le carre est grand
Le carre est bleu
Le grande carre est au-dessus du cercle

On the basis of the first two grand and bleu would be merged into a single category. On the basis of the third sentence this category would be inserted in front of carre. This would lead to the sentence Le bleu carre est au-dessus du cercle which is not acceptable because French requires bleu to follow the noun.

8. Learning Rules of Semantic Interpretation

The pairing of sentence with semantic referent not only allows one to learn the meaning of words but also the correspondence between grammatical structures in the sentence and rules of semantic interpretation. So by this point, Klein's program has learned to interpret the first noun as subject and second as object in an underlying meaning structure. The fourth sentence is:

4. Nara wa jon o sinzita (Nara believed John)

On the basis of minimal contrast it decides sinzita and kiita are also in a common category. It also assumes that this category refers to the verb of the sentence. So now the complete grammar has the structure:

S1 → S2 wa S3 o S4 S-V-O

S2 → jon, Nara

S3 → jon, nara, hanasi

S4 → kiita, sinzita

With S1 I have indicated its semantic interpretation.

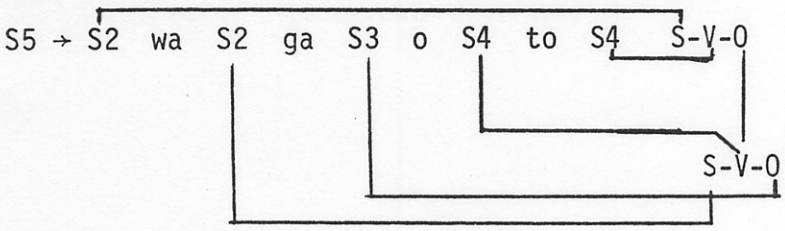
The next three sentences are

5. Anata wa nara o kiita (You heard Nara)

6. Nara wa jon o sinzita (Nara believed John)

7. Anata wa nara ga hanasi o kiita to sinzita (You believe that Nara heard the story)

Sentence (5) is parsed by the existing grammar except that Anata must be added to the S2 and S3 categories. Klein initially adds pronouns to all object categories and deletes them as evidence warrants. Sentence (6) is completely parsed by the existing grammar. Sentence (7) is destined to force major revisions in the grammar. At first there is just an analysis of the sentence into the known categories. This is accomplished by means of rule S5:



9. The Graph Deformation Condition

Note that the underlying structure that Klein wants to assign to semantic referent is: S_1-V_1-O_1

S_2-V_2-O_2

That is, in the structure S_2-V_2-O_2 is organized into a distinct substructure. Klein wants to assign a surface structure to the sentence that reflects the semantic structure. This can be better illustrated if we represent the phrase

structure of the sentence and the semantic referent as trees. This is done in Figure 3. We have illustrated there a phrase structure for the sentence (Part a) and a tree structure for the semantic referent (Part b). Note that the sentence's phrase structure is a graph deformation of the semantic referent. That is, all the connections among nodes have been preserved but there is a rearrangement of the location of the links relative to one another. The semantic structure specifies nothing about the non-meaning-bearing morphemes in the sentence (i.e., wa, ga, o, to). Therefore, these are left unbracketed in the sentence.

Insert Figure 3 about here

It frequently seems that the appropriate surface structure for a sentence is a graph deformation of its underlying semantic structure. This graph deformation condition is thoroughly discussed in Anderson (1975). I will provide a discussion in Section 8 of its implementation in LAS. While Klein does not explicitly state that this is what he is doing, he is basically taking advantage of the graph deformation condition to decide the surface structure for sentence (6). He posits the new grammar:

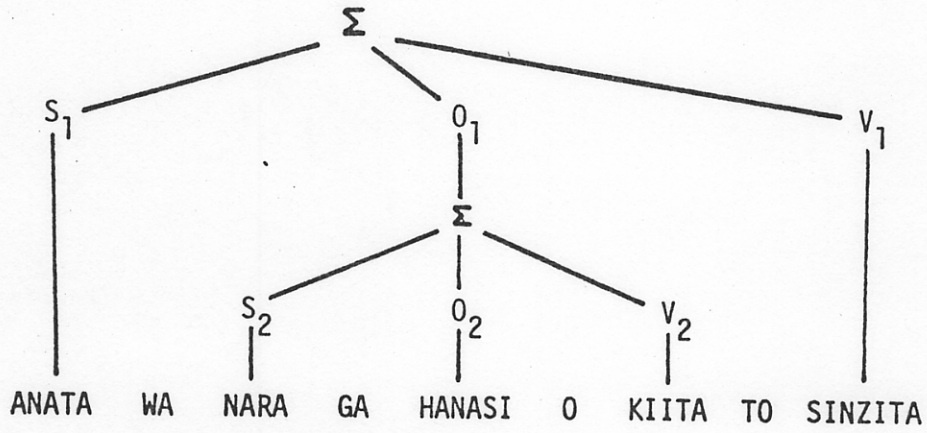
S5 → S2 wa S6 to S4 S-V-O

S6 → S2 ga S3 o S4 S-V-O

In this grammar, S6 is derived from S5 and it rewrites into a substructure. The substructure derived from S6 corresponds to the subtree structure in Figure 3.

Note that there are no generalizations, per se, in going from the previous single rule for sentence (6) to the pair above. However, it does alter the

(a)



(b)

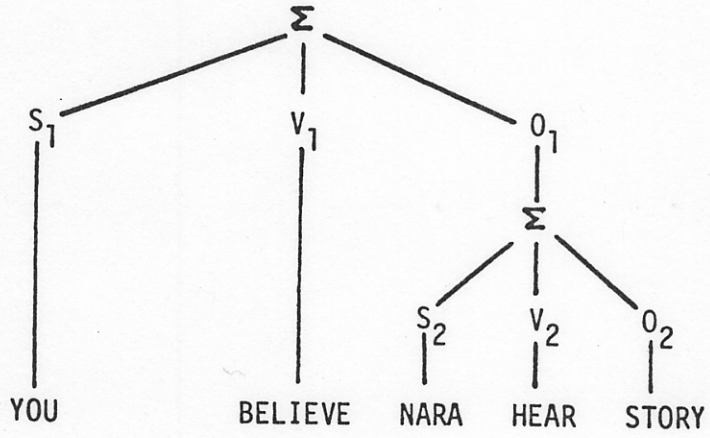


Fig. 3. (a) Surface structure of the content words in a Japanese sentence;
(b) Tree structure of concepts in the semantic referent.

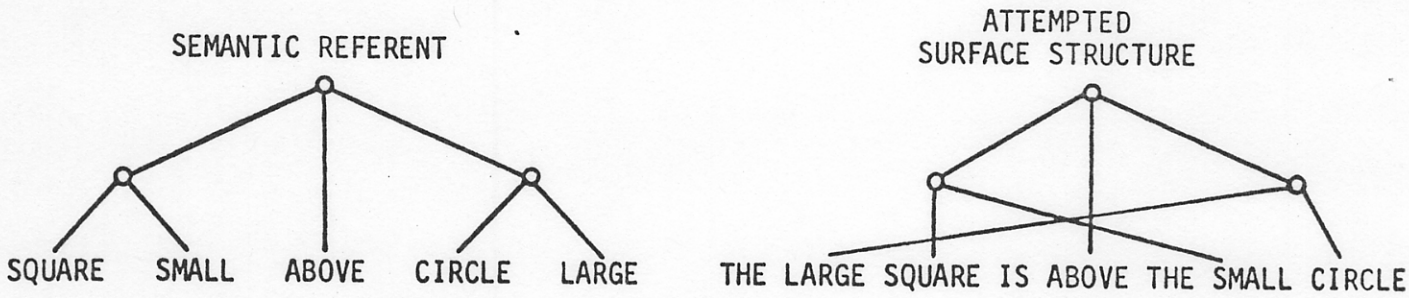
structure of the rules in the grammar. Consequently, it will drastically affect the course of future generalizations.

The graph deformation condition imposes strong constraints on what an acceptable sentence can be like. All acceptable sentences must have a phrase structure of the sort in Figure 3. That is, there is a tree structure in which no branches cross. The graph deformation condition asserts that this surface structure must be a graph deformation of an underlying semantically-based structure. Many strings of words could not be graph deformations of any reasonable semantic structure. Consider two examples in Figure 4. In Figure 4 we have a possible semantic referent for the English sentence The small square is above the large circle. This can very easily be used to impose an appropriate surface structure over the content words of this sentence. However, suppose some language chose to express this by the sentence, The large square is above the small circle, in which the object adjective preceeded the subject noun and vice versa. As illustrated in Figure 4, there is no way to use the graph structure of the semantic referent to get a surface structure for the sentence. No matter how it is done, some branches must cross. The graph deformation conditions claim that such sentences could not be used in natural languages to express the semantic referent.

Insert Figure 4 about here

However, Figure 4b shows the one example I have been able to find of English sentences which violate the graph deformation conditions. These are respectively sentences. In the semantic representation the pair John and runs and the pair Bill and walks are naturally put together in substructures. However, the respectively sentence orders these words in just the way that makes it impossible to extract a surface structure from the semantic referent.

(a)



(b)

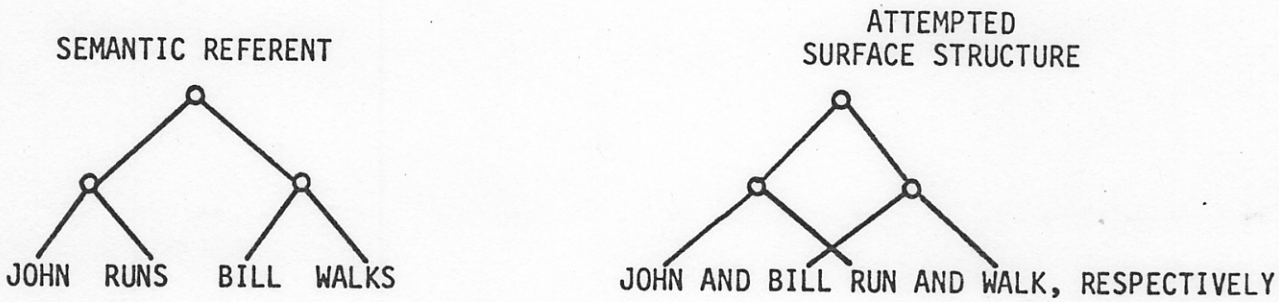


Fig. 4. Two violations of the graph deformation condition. Example (a) is fictitious and presumably represents a non-natural language. Example (b) comes from English.

Respectively sentences and other similar constructions like vice versa seem to be in a class apart from the rest of English. They are only acquired very late after the rest of the language is well-established. It is my suspicion that special control structures are set up which convert a spoken sentence of this class into another simpler sentence. For instance, upon hearing John and Bill run and walk, respectively these processes might convert it into John runs and Bill walks. Then the more primitive language devices operate on these sentences. Similarly, in speaking, the simpler sentence is first generated and then transformed into the more complex sentence. With such sentences the evidence for transformations seems overwhelming. However, the work of myself and others on mechanisms for language acquisition is focused on inducing grammars for simpler, pre-transformational sentences.

10. The Relational Structure of Language

The main assertion of a sentence is a relational structure consisting of one or more relational terms and one or more noun phrases. For instance, in the sentence The ball is red, The ball is a noun phrase and is red a relational term. In the sentence The quarterback threw the ball over the linebacker, The quarterback, the ball, and the linebacker are the noun phrases and threw and over the relational terms. Alternatively, one could analyze over the linebacker as an embedded relational structure consisting of one relation and one noun phrase.

Noun phrases serve to gather together information about an object in the semantic referent--e.g., The tall girl in my class who talks very loud. In principle, this information about the object could be scattered throughout the sentence, but because of the graph deformation condition this is not possible.

Relational structures occur at the level of the main assertion, as modifiers in noun phrases; as modifiers of other relations, and in the place of a noun phrase as in John believes Mary stole the book. Thus, a sentence is a sequence of embeddings of grammatical structures where each level of embedding either corresponds to a noun phrase or relational structure. Clearly, arbitrary languages need not have this structure. It is a remarkable fact about natural languages that they do. Relational structures are subject to the graph deformation condition just like noun phrases are. That is, one cannot intersperse elements from multiple relational structures. Thus, while one can say John believes that Mary stole the book one should not say John of the book believes that Mary stole.

Consider the following semantic referent. A man is holding a cane. He gives ten dollars to a florist. In exchange the florist gives him a dozen roses. The roses are for Mary. All this might be conveyed by the sentence: The man who is holding a cane bought with ten dollars a dozen roses for Mary from a florist. Consider the following unacceptable sentence: The man who gave the florist ten dollars bought with a cane a dozen roses for Mary. This sentence is unacceptable because the graph deformation condition does not let one insert in the middle of the relational structure for buy the object of hold. But this is arbitrary. There could be a verb which had as its relational structure something of the form (Agent)(verb) with (object being held by agent) (object being received) for (benefactor). What distinguishes this from the case structure of the acceptable sentence? - (agent) (verb) with (object being given up by agent) (object being received) for (benefactor) from (person receiving the money).

It seems the language is somewhat arbitrary in having some relational terms like buy with certain case structures but not other terms with other conceivable case structures. The solution taken in the work of Siklóssy and Klein is to assume as semantic primitives the appropriate relations and their case structures. However, this tack will not succeed because different languages have different relations with different case structure. Although there is considerable agreement on how to cut up the world relationally, there is by no means complete consensus across languages. The consequence is that the relational structure must be learned by a language learning program, not assumed. There are probably quite strong semantic constraints on what are acceptable candidates for relational structures. For instance, an acceptable structure is one like X opens Y by using Z which underlies John opened the door with a key. However, X opens Y while eating Z would lead to sentences like John opened the door with an ice-cream cone. Such a sentence seems that it might be unacceptable in any natural language. It seems that all the arguments of a relational term must be causally interconnected.

In any case, the fact that relational structure is not universally fixed means that the relational structure of language must be learned before the Graph Deformation Condition can be utilized to assign a surface structure to the sentence. In part, the graph deformation condition might be used to infer the relational structure. That is, an induction heuristic could be constructed which asked what the relational grouping of elements would have to be to derive a surface structure for the sentence consistent with the graph deformation condition.

There is another complication posed by the relational structure of language. The same underlying semantic referent can have more than one relational structure.

Thus, we can equivalently say John, who turned the key, opened the door or John opened the door with a key. This means that really three levels of representation are needed in the mapping from semantic referent to sentence. Intermediate between the sentence and the semantic referent there is a relational structure which has incorporated the relational structure of the sentence but which is otherwise unchanged from the semantic referent. Figure 5 shows what I intend (with the HAM memory structure as the semantic referent). Depending on the relations in the surface sentence one of two relational structures are derived. Then the graph deformation condition is applied to the derived relational structures to determine the surface structure.

Insert Figure 5 about here

In this system, the HAM structure corresponds to something like the abyssal structure of generative semantics and the relational structure to a deep structure. I generally refer to the relational structure as the prototype structure. Unlike Klein and Siklóssy's programs which use a prototype structure, my LAS system goes from the semantic referent. The prototype structure is never really computed, but is implicit in the induction process.

11. Generalization About Relational Structures

Relational structures and noun phrases each permit rather different sorts of generalizations. I will review first the principles of generalization suggested by Klein and Siklóssy for relational structures. These generalizations are made on the basis of similarity of order of constituents in two relational structures. For instance, rule S6 and S1 are highly similar

S1 → S2 wa S3 o S4

S6 → S2 ga S3 o S4

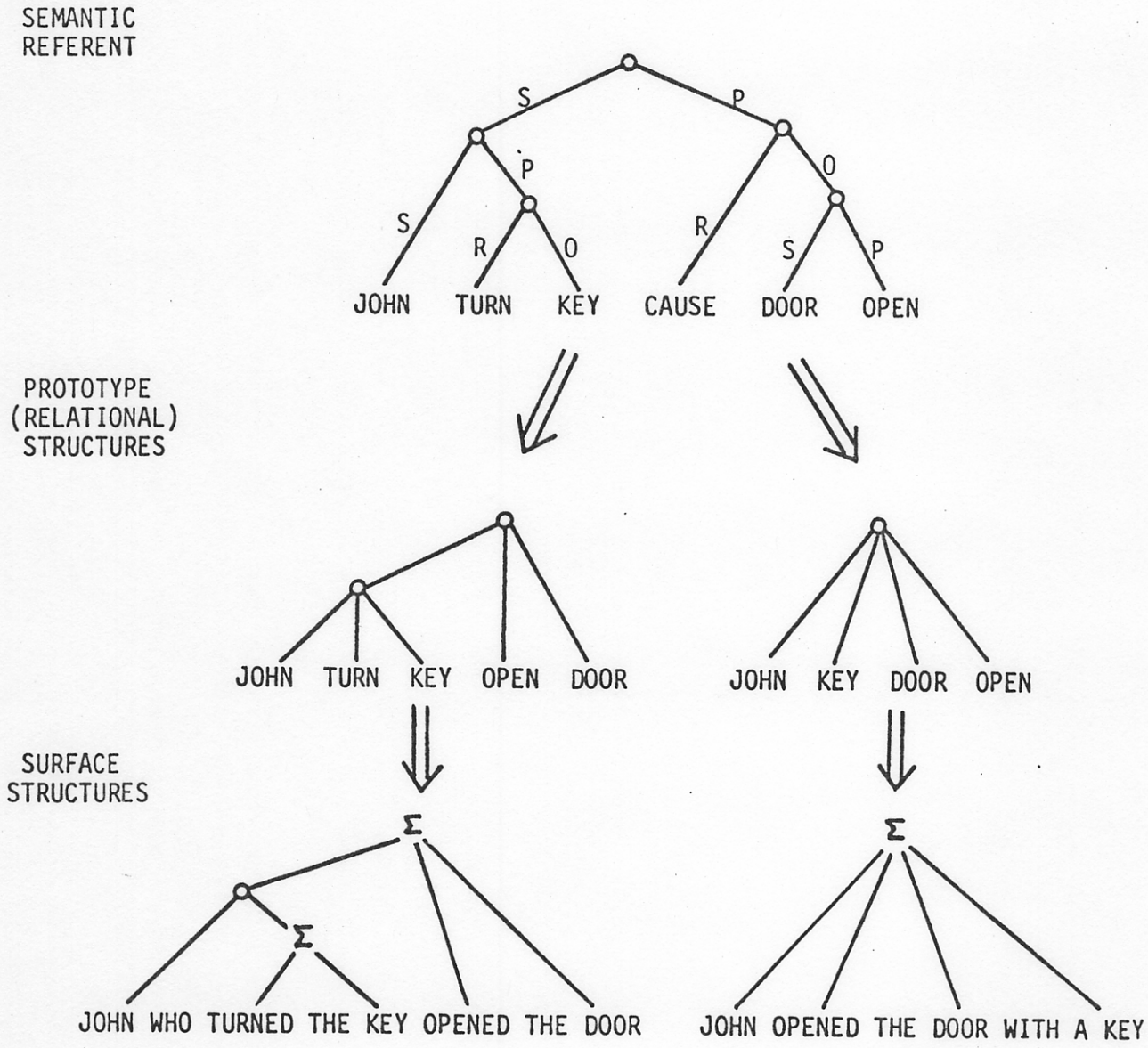


Fig. 5. Two surface structures are generated from the same semantic structure via two prototype structures.

Therefore, an attempt is made to replace reference to S6 in S5 by S1. Rule S5 now becomes:

$$S5 \rightarrow S2 \quad wa \quad S1^{T} \quad to \quad S4$$

The problem is that S1 has wa whereas S6 has ga. Therefore, the T superscript is added to S1 in the context of S5. This indicates the need for the following obligatory transformation:

$$S1^{T}: \quad (S2 \quad wa \quad S3 \quad o \quad S4) \rightarrow (S2 \quad ga \quad S3 \quad o \quad S4).$$

This is one role of transformations in Klein's system - namely, to edit out overgeneralizations. Thus, while the phrase structure rules like S1 and S5 evolve to create ever-more inclusive grammars (that is, permitting more sentences), the transformational rules are designed to make the grammar more restrictive. In this example, the transformation served to prevent what would have eventually become free wa-ga variation.

Exactly how one would like to formulate the matter can be debated, but it seems that generalizations of relational structure from one context to another are among the safest of generalizations. The fact that such generalizations are permissable is a significant fact about natural language.

12. Generalization About Noun Phrases

Noun phrases permit two sorts of generalization. First, with respect to a noun phrase occurring at a particular point in the grammar, one can generalize from the acceptability of some sequences for that noun-phrase unit to the acceptability of other sequences. For instance, having seen the red square and the green circle as sentence subjects, it might seem reasonable to generalize to the acceptability of the green square and the red circle as subjects. The second sort of generalization involves the merging of rules for a noun phrase occurring at one point in the grammar with rules for a noun phrase occurring at another point.

One has to be careful about forming generalizations within noun phrases. Klein's generalization on the basis of serial position will not do. Consider the following sequence of noun phrases:

The foam pillow
The large pillow
The red foam pillow
The comfortable large pillow

On the basis of the first two and the principle of minimal contrast, Klein would put foam and large into one word class. On the basis of the next two red and comfortable would be put in a second word class and this would be made an optional predecessor of the first. Then the program would accept The red large pillow which is not acceptable. The question of the ordering of adjectives in English has been studied extensively by Zendler (1968). He concludes that there is a relatively strict and complex ordering involving 19 classes of adjectives. Adjectives in a higher class must precede adjectives in a lower class if they both occur in a noun phrase. A phrase like (The red large pillow) can be acceptable, but only in special circumstances where it can be interpreted as the red one of the large pillows.

These adjective classes are not arbitrary but are correlated with semantic features. It seems that the more noun-like the adjective class is, the closer it can occur to the noun. So, for instance, adjectives that refer to substance like foam follow adjectives which refer to absolute properties like white, which follow adjectives like big which refer to relative properties, which follow adjectives like comfortable which refer to features of the object's use.

The problem with Klein's attack on this problem is that he is looking for grammatical principles of absolute position, whereas what is needed are principles of relative order. The way to deal with the problem of adjective

ordering is to build into the induction routine the bias to look for semantically-defined word classes and to learn the principles of ordering of these word classes. The noun-ness principle of ordering uncovered by Zendler for English may also be universal. If so, it could be built into induction routines.

Another interesting fact about noun phrases is that there is one class of words, nouns, which is obligatory whereas the adjective word classes are optional. The existence of such a noun class seems to be universal. There is no necessary reason why this should be so, but it does serve to create problems. Consider the following noun phrases:

The square
The square box
The red box

After seeing these Klein's principle of left generalization, would accept The red.

One might protest that red is an adjective and square a noun and the generalization could have been prevented on this basis. However, there seems no semantic basis for deciding that one is an adjective and the other a noun. There is nothing more inherently noun-like in the concept red than in the concept square.

Thus, the conclusion seems to be that the generalization within noun phrases should not be concerned with absolute position; rather they should be concerned with relative position. An induction program should emerge with a set of semantically-identified word classes, some of which are obligatory and some of which are optional. LAS should learn what is the ordering that the language places on these word classes.

Noun-phrases that occur at different points in the grammar have a lot in common. This is clearly the case in English where there is essentially just one grammar for noun phrases in all contexts. For languages inflected

for case, however, there can be inflectional differences between noun phrases depending on whether the phrase is subject, or object or instrument, etc. But even for these there is much in common between the different noun phrases.

As far as I know there are no differences in any natural language with respect to noun phrase word order dependent on position of that noun phrase within the sentence. In this regard, noun phrase structure seems as promising a place for generalization as relational structure.

General Conclusions

I make no pretense that these principles for language learning are exhaustive. They are just the ones I have found so far working on the problem and reviewing the work of others. However, the fact that these many exist is significant. It clearly reinforces the notion that natural languages are in no way a random selection from the set of all possible languages. This idea is not new. It was advanced by Chomsky (1965) to account for why children seem to have the success they do in language acquisition. However, Chomsky seems to have thought that there were purely syntactic constraints on the forms of natural language. The only possible role he saw for semantics was one of motivating the learner. As he wrote in 1962: ". . . it might be maintained, not without plausibility, that semantic information of some sort is essential even if the formalized grammar that is the output of the device does not contain statements of direct semantic nature. Here, care is necessary. It may well be that a child given only the input of nonsense elements would not come to learn the principles of sentence formation. This is not necessarily a relevant observation, however, even if true. It may only indicate that meaningfulness and semantic function provide the motivation for language learning, while playing no necessary part in its mechanism, which is what concerns us here."

It is now clear, however, that many of the constraints on language, and many of the potential generalizations are only possible when we make use of the language's semantics. It seems hard to specify any strong properties that directly constrain the syntactic form of a natural language. It seems these constraints only come indirectly by making reference to semantic information.

However, I suspect none of these principles are iron-clad. That is, there probably exist in some natural language exceptions to each. I have been able to point out exceptions in English for all but the last two. Certainly artificial languages could be learned that had slight exceptions to these two. How can language be learned, then, if there are no principles that an induction algorithm can count on with 100% confidence? The answer is that the exceptions to these principles must be few. If they were not language could not be learned. However, a successful induction algorithm will have to tolerate exceptions to the generalizations it is forming.

5. Data About Language Learning

This section should first begin with a disclaimer. There is no attempt here to provide a review on the literature concerning language acquisition. There already exist some very good sources for that purpose (Brown, 1973; Clark, 1975; McNeill, 1970; Slobin, 1973). What I have attempted to do is to select from that large and rapidly-growing literature that which is relevant to the task at hand. It will become quickly apparent that what is being modelled here is an abstraction and idealization of any real-life language learning situation. Therefore, I have taken enormous liberties in deciding what is relevant and what is not.

The principal source of data concerns child language learning. As we will shortly see, this is really not the most appropriate testing ground for studying

language acquisition of the sort I am interested in. My concern is with the learning of the connection between strings of words and their meanings. This assumes a number of things on the part of the learner. First, he has already isolated those acoustic objects that are words. Second, he has already a richly developed meaning system. Third, he understands what language is about - that it is a map from symbols to meaning. Fourth, he wants to acquire that language. None of these conditions are initially met by a child learning his first language. Many of the complexities we see in the development of child language reflects the gradual evolution of these pre-conditions for language acquisition.

Consider the matter of phonological development. There is considerable evidence that a child can early perceive and produce the phonetic distinctions required in natural language (Eimas, Siqueland, Jusczyk, & Vigorito, 1971; Grégoire, 1949; Jakobson, 1941; Moffitt, 1968, 1971). However, the child's task is to learn which distinctions in his language convey information. For example, a child learning English need not pay attention to aspiration with voiceless stops because the contrast, aspiration or nonaspiration, does not have a communicative function. However, the child cannot know this until he is well into language learning. The matter of word segmentation is another serious phonological problem. As anyone knows who has looked at a speech spectogram the speech signal does not naturally break itself up into morphemes. The child must first figure how to segment the signal before he can address himself to questions of the grammatical structure of the signal.

For analytical purposes it would be nice if the child would first complete all of his phonological learning and then get on with the task of grammar induction. However, well on into the third and fourth year, when the child

is understanding and producing relatively complex grammatical constructions, he is still picking up phonological distinctions.

On the semantic side of things, one would want the child to have evolved the necessary conceptual distinctions before he embarks upon grammar induction. However, we as scientists, are again not to be so fortunate. Indeed as many have now begun to argue (Clark, 1975; Nelson, 1974; Slobin, 1973) what seems to underlie the timing of the acquisition of many grammatical structures is the acquisition of the requisite conceptual knowledge. That is, most linguistic structures signal conceptual distinctions and the child does not develop these conceptual distinctions until he is well into the language learning process. For instance, Slobin (1973) argues that the reason plurality or past tense do not appear earlier in child language is that the child does not have these concepts. As soon as the child develops the concept some manifestation of it will appear in the language. For instance, a child may initially mark plurality by more - e.g., more shoe. Shortly thereafter he will acquire the adult grammatical mechanism - i.e., the suffix s. Slobin proposes that the gap between the appearance of the conceptual distinction and the use of the adult marking for that distinction is the true indicator of the grammatical complexity involved in making that distinction in that language.

It seems also clear that children initially do not really understand the purpose of language. Initially they use it largely to make simple requests of their enviornment. Early child speech and adult speech to young children is marked by a high frequency of questions and imperatives. Declarative sentences are low in frequency and negatives almost nonexistent. Children are notorious for not hearing sentences that do not deal with their immediate

sphere of needs and concerns. It seems clear that the child is slowly seduced by his environment into being a language user. Initially he is attracted by the immediate gains it brings him, but slowly he comes to appreciate its more abstract purposes.

Thus one cannot observe in children the unhindered course of grammatical induction. It is always being held back by the slow development of these pre requisites. However, one would be wrong to totally characterize the process as one of just dependency of the grammar on the phonological, conceptual, and sociolinguistic development of the child. All these processes are also obviously aided by grammatical development. A new phonological distinction is easier to detect when embedded in an otherwise comprehensible sentence. Many concepts evolve from hearing language used in context. Certainly, different grammatical structures must be one cue to the child of the different communicative functions of language. Thus, the whole course of language development is very interactive with different processes influencing each other. In my opinion, it tends to make child language acquisition a rather intractible scientific puzzle.

If one is going to be able to make any sense at all out of child language acquisition, he is going to have to deal with the data at a level of abstraction that is a cut above trying to account for the day-to-day changes in the child's language. This is what I have attempted to do in the next chapter. There will be developed a grammar induction model which assumes that these prerequisites are satisfied. It makes predictions about trends in the course of language acquisition. Our task will be to look for these trends in among the noise created by these other factors.

One might hope to also find a source of data in the second-language acquisition of adults. Here is a situation where the prerequisites to grammar

induction are much more closely satisfied. This would be particularly true if the second language had a similar phonological and conceptual structure as the first. However, there are two serious problems. First, second language learners have a natural tendency to use their first language as a crutch in the language learning situation. Thus, rather than learning to map between sentence and semantic referent, they learn to map between the sentences of one language and another. The second, related problem is that there is a critical initial period during which language can be learned much more successfully than in later years (see Lenneberg , 1967). Thus, one might wonder whether the same processes are being studied with older subjects as in the younger child. Whether this is really a critical problem depends on whether one is particularly interested in child language acquisition. I personally am not, but find very disturbing the possibility that there might be a basic change in the mechanisms of induction with age.

However, I do not think that conclusion is required by the existing data. The existing data is of two sorts. First, relearning that part of a language lost after brain damage is never complete after puberty whereas it is before. Second, another language can be learned easily and without accent before puberty but not after. Both of these facts are cast in terms that suggests that there is an all-or-none shift in cognitive functioning. However, there is no evidence whatsoever to indicate that this is really not a gradual shift in cognitive functioning. I think what is happening is that as one gets older it becomes impossible to acquire a language without making reference to existing linguistic mechanisms. It is probably not optimal to try to build a new language on the control structure set up for an old or damaged language. In aphasia cases some linguistic facility usually remains which the adult probably tries to build on. In second language learning, the problem arises because the

second language is built on the first. Adults probably cannot help this because their old language processes have become so automatized. Young children probably have more ability to inhibit the application of their language processes.

So, in conclusion, I think second language learning could prove a valuable source of data. However, one must deal with the fact that the second language is being developed in the background of a set of deeply-intrenched mechanisms for first language : These mechanisms inevitably influence and, it seems, hinder the course of second language development. Unfortunately, this potentially rich area for data has not been carefully researched. Therefore, the remainder of this section will largely consider first language acquisition in children, despite its interpretative problems.

Speech to Children

For a good period of time it was commonly held that children performed a truly remarkable feat. They heard from the adult community speech full of false starts, hesitations, slips of tongue, and generally made up of ungrammatical utterances (e.g., Chomsky, 1967; Bever, Fodor & Weksel, 1965). Despite this, the child managed to induce the correct grammar of the language. Certainly, none of the heuristics reviewed in the past section would be able to retrieve the correct grammar given such degenerate input. Given this belief about the input to children, one can hardly be surprised to find theories of language acquisition that bordered on mysticism.

However, the adult's speech to children is remarkably good. It is certainly much better than to other adults. Less than 2% is ungrammatical (Labov, 1970). It is much simpler with far fewer embeddings (Sachs, Brown, & Salerno, 1972; Drach, 1968). It is spoken much more slowly (Farwell, 1973;

Broen, 1972; Sachs, Brown, & Salerno). A smaller vocabulary is used (Farwell) and there is a greater tendency to stop at sentence boundaries (Broen). Thus, early speech is not only grammatical, it is very simple. Simplicity in early speech is not terribly critical for any of the induction heuristics we discussed. However, it does become critical when one realizes a child is not a modern computer and is subject to very severe information-processing limitations.

It has also been noted (e.g., Shipley, Smith, & Gleitman, 1969; Snow, 1972) that the speech of adults to children tends to be geared to the child's grammatical level and that the adult will adjust his speech as the child develops. Adults seem to be always at the next stage up from the child in complexity of grammar. Gelman and Shatz (1972) found that children even as young as four years of age modified their speech when talking to two year olds. Moreover, they simplified their speech more in the case of two year olds than three year olds.

One is naturally led to the hypothesis that the child is being paced by the adult. However, there are a number of sources that indicate that direct attempts to increase the complexity of the child's speech does not help him (Cazden, 1965). Cazden had adults expand the speech of children. For instance, the child might say Baby chair and the adult would restate, in proper English, what he thinks the child is saying Yes, the baby's sitting in the chair. This was not found more beneficial than other types of linguistic interaction. So, it is apparently not important that a child get feedback on the particular grammatical structures he is using. Rather, what he needs is just an enriching environment of linguistic input. This would seem to indicate that induction heuristics are not psychologically valid if they make use of feedback on productions.

Lexicalization

Lexicalization does not occur in child language the way it does in programs like Klein's. That is to say, children do not learn the meanings of words by hearing them in ~~complex~~ sentences paired with semantic referents. Rather much of the early lexicalization in child language occurs by hearing single words paired with referents. Or if the words occur in sentences they occur in frames like "There is a cat" which just have one content word (Ferguson, Peizer, & Weeks, 1973). Similarly, this is the way vocabulary is taught in school. Each word in the foreign language is paired with a definition in the mother language.

Of course, it does not follow from the fact that this is the training procedure that it is the only one or even the best. Let us consider a simple algorithm for word learning and see what it implies about learning words in isolation vs. sentence context. Suppose a learner receives a sentence with m words paired with a meaning representation that contained m meaning elements. Suppose further he has no idea of what the meaning might be of any of the words. Suppose he took a guess as to the meaning of each word, selecting at random from each of the m memory elements. Then he would have probability of $1/m$ of being correct on any particular word. With m elements he expects to learn $m \times 1/m = 1$ element on the trial. Thus, given this algorithm there is no effect of m, number of content elements, on expected learning when the subject does not know the meanings of any of the words.

Now consider the situation where he knows n_2 of the n_1 words in the vocabulary. Assume further, that on any trial, he is presented in a sentence with a random selection m of the total n_2 words. Suppose $n_2 > m$. What is the probability that he will know the meaning of all the words in a particular sentence? It can be shown to be $\frac{n_2!\ (n_1-m)!}{n_1!\ (n_2-m)!}$. This probability decreases

with increases in m. A trial, in which the subject knows the meaning of all the words, is completely non-informative. Thus, as m increases we increase the probability of an informative trial. On information trials the subject expects to learn one item, independent of the number of unknown words on that trial. So, this algorithm expects learning to be more rapid the larger m once the meanings of some words in the vocabulary are acquired. Thus, learning a word in isolation (m = 1) is the worst situation.

Of course, the problem with this algorithm is that it is not taking into account the information-processing cost of dealing with larger m. A series of experiments by Anderson and Paulson (in preparation) are relevant in this respect. We had subjects study objects which had three attributes like size, shape, and color. A subject might be shown a small red circle and a small blue circle and he asked to indicate which was the DAX. After making his decision he was given feedback as to the correct decision. Thus, he might be told that the DAX referred to the small red circle. Because the pair of objects only differed on a single attribute, the subject would know DAX meant red. So this would constitute one trial on the pairing DAX = red. This is a case where m = 1.

We looked at two distinct cases where m = 3. In one case, he would see the same two pair of referent objects as in the m = 1 case but he asked to indicate which was the DAX JIB GUR. Because the referent pair contrasted on only color, he could make a correct choice only if he knew that DAX, or JIB, or GUR meant red (whichever did--the words were randomly ordered). However, he got via feedback information that allowed him to know DAX, JIB, and JIR should be paired in some way with small, red, and circle. The other m = 3 case, involved a pair of referents that contrasted in all three dimensions. Thus, he might be asked to indicate whether DAX JIB GUR was a small red circle or a large green triangle. Thus,

this choice was easier than the other m = 3 case because the subject could make it on the basis of knowing DAX or JIB or GUR. However, the feedback gave him identical information as to the possible meanings of DAX, JIB, and GUR. We will call these three cases m = 1, m = 3, hard, and m = 3, easy.

The subjects were given 50 trials on this task, paced at a 40 second rate. The criterion test was number of words learned out of 9 after the experiment. Averaging over two slightly different experiments the lexicalization scores were:

m = 1	7.71 words
m = 3, hard	6.13 words
m = 3, easy	7.75 words

So subjects clearly cannot take advantage of the extra information in the m = 3 case. It is particularly interesting how bad they were in the m = 3, hard case. The difference between this and the m = 3, easy case is significant ($t_{66} = 2.20$; $p < .05$). This is despite the fact that the two training sequences gave identical information as to the meaning of the words. However, subjects had a harder decision to make in selecting the correct object referent for the description. This extra information-processing load apparently "seeped" into the lexicalization learning.

Telegraphic Speech

The clearest discrepancy between the behavior of a child and any induction program has to do with the telegraphic speech that the child produces. That is, children initially speak in two and three word utterances. To condense messages into such short utterances it appears that children have omitted most function words and subordinate constructions. Thus, they speak somewhat like one might write a telegraph - bye-bye Mummy, put gas in, no sit here, etc. There has been some question as to whether these samples are properly characterized as telegraphic

(see Brown, 1973) but there is no doubt that they are grossly ungrammatical in a way that would not be produced by the existing induction heuristics. These heuristics can generate ungrammatical sentences as a consequence of over-generalization, but over-generalization is clearly not the origin of the telegraphic quality of early child speech.

It is the case that students learning a language in a classroom do not display any marked tendency to telegraphic speech. However, observing myself in more free-learning situations I can report that at least one adult will revert to this speech style. These are situations where one is trying to communicate in a foreign language and cannot wait to determine the niceties of that language. One picks up a few words, some principles of orderings, throws the words together as best he can, and hopes the native speaker will figure out what is intended.

It may be that the child in his telegraphic speech is not even attempting to observe the grammar of the language. If so, we should see one of two things: Either word order would be completely random. Or, if not random, it would not bear any systematic relationship to word order in the particular language. Word order in early child speech might reflect the existence of a primitive ordering in pre-linguistic thought (see McNeill, 1975). In fact, there is some evidence that early child speech has a word order independent of language. Evidence for a universal simple pivot grammar, along the lines proposed by Braine (1963), has been found by a number of researchers (Brown & Bellugi, 1964; Brown & Fraser, 1963; McNeill, 1966, 1970; Miller & Ervin, 1964; Slobin, 1966). However, more recent work has challenged the apparent universality of pivot grammars (Blount, 1969; Bowerman, 1973; Kernan, 1969). Also, there is ample evidence that in later child speech the utterances maintain their telegraphic

quality but do begin to incorporate principals of the adult word order (Slobin, 1971).

Telegraphic Perception Hypothesis

The joint facts of telegraphic speech and the apparent partial incorporation of adult grammar cause real difficulties for the kinds of induction heuristics considered. Therefore, I have proposed what I have called the telegraphic perception hypothesis. Suppose the induction heuristics did not receive as input complete sentences but rather telegraphic sentences. Then naturally they would induce a telegraphic grammar. It seems quite reasonable to suppose that a child cannot hold in immediate memory the total sentence he has heard but rather a depleted version of that sentence. If so, then his induction algorithms would be receiving telegraphic sentences as their basic data.

Evidence for this hypothesis comes from studies of child imitation of adult speech. It is found that these imitations, while longer than the child's own productions, are also telegraphic in nature (e.g., Brown & Fraser, 1964). Blaisdell and Jensen (1970) found that children tend to repeat those words which are stressed and those words which occur in terminal positions. The semantically important words tend to be stressed in adult speech. Schols (1969, 1970) found that subjects tended to omit words that had unclear semantic roles or unknown meanings. What I find striking is that these are just the variables which control what I can repeat back after hearing a French sentence - a language I know quite imperfectly. Of course, the variables of serial position, perceptual isolation, and meaningfulness all have well established effects in verbal learning experiments on immediate memory.

One could reasonably propose, that as the grammar developed, it could be used to permit the child to encode and remember more of sentences. In this way, sentence perception and grammar induction could interact each feeding on developments in the other. Inducing a grammar from degenerate sentences poses an interesting problem. How is the system ever to abandon its old rules as it develops new rules built on more adequate input (i.e., less degenerate sentences)? Merely because new grammatical constructions have appeared it does not follow that old ones were wrong. Some mechanism is required for eliminating the old rules. The gradual disappearance of telegraphic constructions from the child's speech would seem to suggest that the new rules are gradually strengthened relative to the old rules.

This telegraphic perception hypothesis brings us face-to-face with two facts that are not often acknowledged in the discussion of heuristics for language acquisition. First, humans must process sentences in a left-to-right fashion. They have only one chance to look at any portion of the sentence. Unlike current language learning programs they do not have the luxury of being able to scan back and forth over the sentence looking for relevant information. The second, and related fact, is that humans are subject to severe constraints of memory. They can only hold a rather small amount of information in memory at a time. They also have virtually no memory for the exact sentences they have seen. Any information about the past sentences has been compressed into the current grammar.

All-or-None vs. Continuous Modification of the Grammar

An interesting question is whether humans acquire, modify, and delete rules in the all-or-none manner suggested by the induction heuristics in the preceeding

section. Brown (1973) documents the acquisition of various morphemic rules. Rules such as for forming past tense do not appear in an all-or-none fashion. Rather correct applications of the rule for a single child gradually increases in probability over a period of months. However, this does not entirely decide the issue in favor of continuous growth. There is nothing to prevent a grammar developing with multiple rules for past tense occurring in different contexts. That is, the rule might occur separately for the main clause, a clause modifying the subject, a clause modifying the object, etc. Also there might be multiple rules to derive the main clause. The apparent continuous growth in use of the past tense rule may reflect the fact that it is acquired at different times for these different linguistic contexts. Put this way, of course, it is difficult to empirically discriminate between all-or-none vs. continuous learning. I think the decision as to which description to use will turn out to depend on which is conceptually more tractable. At first blush, it would appear simpler to account for an apparently continuous change in behavior by a continuous process.

Over-Regularization and Over-Generalization

Section 4 noted that almost any reasonable induction heuristic would make generalizations for which there were some exceptions in some natural languages. Therefore, it is interesting to note that children's speech runs rampant with overgeneralization. This is particularly true in the case of morphemic rules. Consider Ervin's (1964) case history of foot. The child initially may use feet for both singular and plural. However, when he identifies -s plural morpheme, he will use feets or foots. Later he may learn of the special pluralization that occurs with pairs like box-boxes and house-houses and use footses. It is significant that the strongest cases of overgeneralization

should occur in morphemic development. What is distinctive about morphemic rules such as those for pluralization is that there are a number of alternate rules that signal the same semantic distinction. This is just the circumstances under which induction heuristics that observe the Principle of Semantics-Induced Equivalence of Syntax (PSIES) (see Section 4) will overgeneralize.

Evidence for the Importance of Semantics

I have argued that principles of generalization are only possible when reference is made to the syntactic referent of a sentence. There is very striking evidence for this claim in a pair of experiments reported by Moesser and Bregman (1972, 1973) on the learning of artificial languages. These were discussed in Section 3. Moesser and Bregman showed that adult subjects could not learn even relatively simple languages without the aid of a semantic referent.

It is assumed that a child learns language in a context where the meaning of sentences spoken to him is obvious. Clearly, this assumption is not always satisfied. However, there is evidence that children just "turn-off" if it is not possible to understand the meaning of the sentence. For instance, hearing children of deaf parents do not learn language by listening to radio or television. The speech spoken by a child and spoken to a child is largely concerned with the here and now. So it seems that it is not unreasonable to propose that the child does have access to the semantic interpretations of the sentences he attends to. Sections 8 and 9 will be modeling situations where this is most obviously true. These are situations where the learner is presented with a picture and a sentence describing it.

Evidence for the Graph-Deformation Condition

Recall that we proposed that there was a correspondence between the structure of the semantic referent and the surface structure of the sentence. This was called the Graph Deformation Condition (GDC). The claim was that connections among elements in the semantic prototype had to be preserved in the surface structure of the sentence, although they could be spatially rearranged. This claim imposed rather severe restrictions on the structure of the sentence. The GDC was important because it determined the structure of the grammar rules and hence influenced the course of future generalizations. In particular, it largely determined the formation of recursive rules. Identification of recursion has always been a difficult matter in grammar induction.

To test the psychological reality of the GDC I performed an experiment on artificial language learning. It utilized the following grammar:

S → NP PRED
NP → Shape (Size) (Pattern) (CLAUSE)
CLAUSE → te PRED
PRED → ADJ
→ NP Rel
Shape → square, circle, diamond, triangle
Size → large, small
Pattern → triped, dotted
Adj → red, broken
Rel → above, below, right-of, left-of

An example of a sentence in this language is Square striped te triange large te broken above circle dotted small right-of. The experiment compares four conditions of learning for this language:

1. No reference. Here subjects simply study strings of the language trying to infer their grammatical structure.

2. Bad semantics. Here a picture of the sentence's referent is presented along with the sentences. However, the relationship between the sentence's semantic referent and the surface structure violates the GDC. The adjective preceeding with the ith shape will modify the (n + 1 - i)th shape in the sentence (where n is the number of noun phrases). For example, the adjective associated with the first noun phrase (striped) modifies the last shape (circle). Similarly, the ith relation describes the relation between the (m + 1 - i)th pair of shapes (where m is the number of relations). So for instance the second relation right-of describes the relationship between the first pair of shapes square and triangle. The appropriate picture for the example sentence is given in Figure 6a.

Insert Figure 6 about here

---

3. Good semantics. Here the adjective in each noun phrase modifies the noun in that phrase. Relations relate the appropriate nouns in the surface structure. The appropriate picture for the example sentence in this case is given in Figure 6b.

4. Good semantics plus highlighting. The picture in this condition is the same as in (3) but the shapes in the main proposition are highlighted. The LAS program, to be presented in the Section 8 , predicts better learning when the main proposition is indicated. The picture for this condition is given in Figure 6c.

In some ways this experiment is like Moeser and Bregman's (1973). However, here English words are used so that the subjects do not need to induce the language's lexicalization as well as its grammar. This corresponds to the situation faced by LAS.1 (an early version of the LAS program).

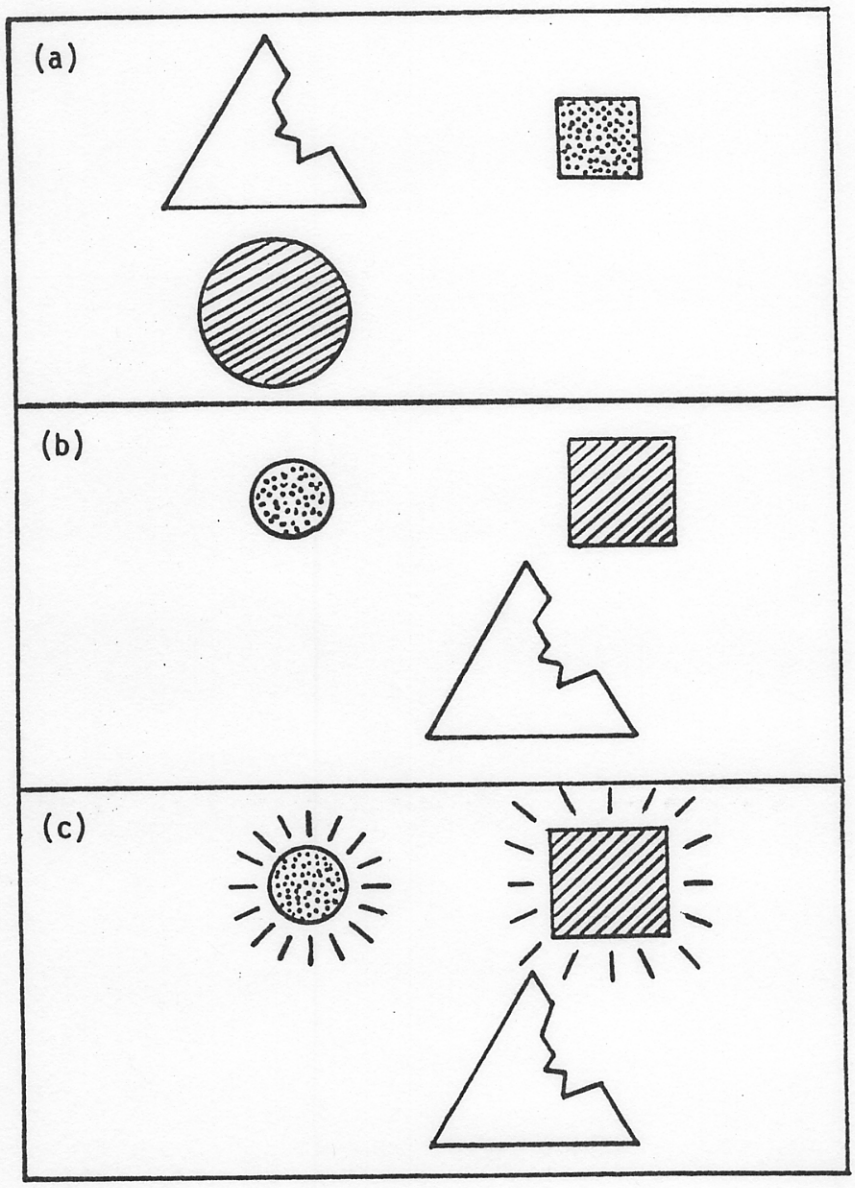


Fig. 6. Picture (a) was presented as a semantic referent in the bad semantics condition; (b) in the good semantics condition; and (c) in the good semantics plus highlighting condition.

Moeser and Bregman's language also differed from this language in that it only consisted of finitely many sentences. Essentially they contrasted (1) and (3) and found Condition (1) much worse. They did not have a condition like (2) where there was a semantics as elaborate as Condition (3), but where the relation between referent and sentence violated the graph-deformation condition. The GDC would predict no difference between Conditions (1) and (2) and predict that both would be much worse than (3).

The basic procedure in the experiment involved having all subjects pass through eight blocks of study-test. In each block the subject studied six sentences with the semantics appropriate to his condition (if any). The sentences were presented to the subjects on cards with pictures given below, depicting the appropriate semantic information. Subjects in the no-semantics condition had just the sentence printed on the card. Subjects were given 30 seconds to study each sentence. After studying the six sentences the subjects were given a test booklet which contained on separate pages six pairs of sentences without any picture referents. The subject's task was to indicate which sentence of each pair was grammatically well-formed for the language studied. Subjects were given 30 seconds to make their decision for each sentence pair. Subjects in all conditions studied and were tested with the same set of sentences. The only variation between conditions was the information that accompanied the sentences on the study trials. The study and test sentences were randomly generated within the constraint that they mention at least two objects and no more than four subjects.

The test pairs were of two sorts. There were pairs that tested for some minimal syntactic contrast. So a subject might have to choose between:

A. Square striped large triangle te red above
B. Square large striped triangle te red above

The second sort of test presented a correct sentence with some unrelated sentence that had a gross semantic defect. So a subject might see:

C. Circle large te triangle small below above
D. Square striped large triangle te red above

In this example C is wrong semantically because above requires two noun phrase arguments and only one is given (triangle is an argument for below). Subjects found the two types of tests to be of approximately equal difficulty. Therefore, I will present data pooled over the two test types.

Figure 7 provides a summary of the main results of the experiment. It is based on data collected from 12 subjects in each condition. In Figure 7 the data are classified by whether they came from the first or second half of the experiment, and by the condition. Plotted is percent correct choice on the test pairs. In all conditions subjects were able to pick up on some regularities and perform better than chance (50%). However, subjects were much worse in the bad semantics and no semantics conditions than in the two good semantics conditions. Also subjects in the bad and no semantics conditions showed little improvement from the first to second half of the experiment whereas subjects in the good semantics conditions showed considerable improvement.

Insert Figure 7 about here

Subjects with good semantics plus highlighting of main propositions are non-significantly worse than subjects without highlighting. The difference is completely due to the two subjects in the highlighting condition who performed

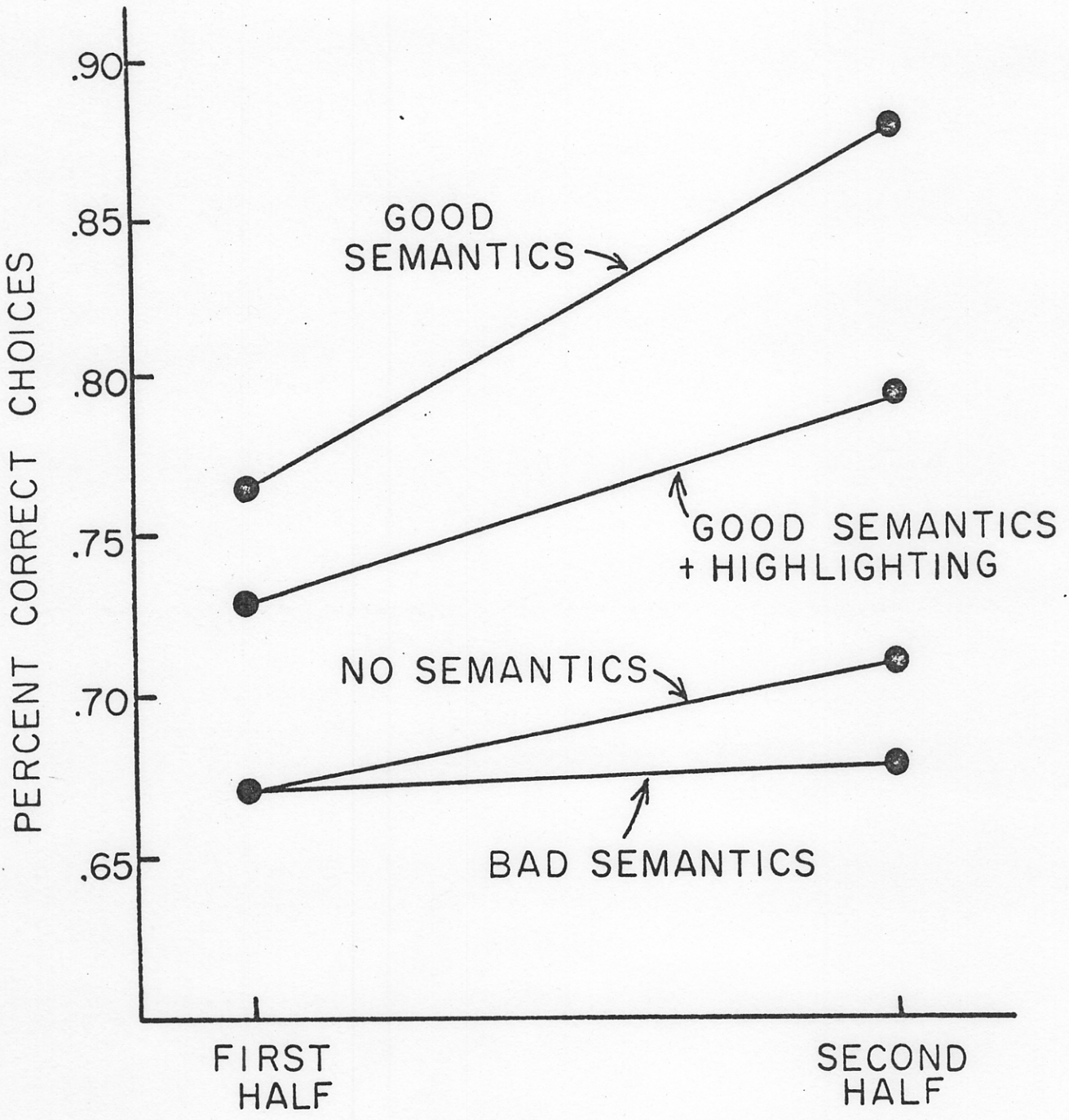


Fig. 7. Percent correct choice in the grammaticality task as a function of semantic condition and block of the experiment.

very poorly. It seems that these two subjects did not understand the intention of the highlighting information.

I think when the GDC is tested in the context of this experiment its truth is obvious. However, it is far from a trivial condition. As we will see in the next chapter it can make for very rapid acquisition of a language.

Comprehension vs. Production

An interesting question concerns the relation between the acquisition of the grammatical competence underlying production of sentences and the acquisition of the competence underlying their comprehension. In Section 7 I will propose that acquisition of a single grammar underlies both - that when we learn to understand a grammatical construction we simultaneously learn to produce it. This seems to conflict with the generally accepted wisdom that comprehension precedes production. However, there may be reasons why child production does not mirror comprehension other than that different grammatical competences underlie the two. The child may not yet have acquired the physical mastery to produce certain words. This clearly is the case, for instance, with Lenneberg's (1962) anarthric child who understood but was not able to speak. Also the child may have the potential to use a certain grammatical construction, but instead uses other preferred modes of production. The final possibility is that the child may be resorting to non-linguistic strategies in language understanding. Bever (1970) has presented evidence that young children do not understand passives, but can still act out passives when they are not reversible. It seems that the child can take advantage of the conceptual constraints between subject, verb, and object. The child's grammatical deficit only appears when asked to act out reversible passives. Similarly, Clark (1974) has shown that young children understand relational terms like in, on, and

under by resorting to heuristic strategies. It is clear that we also have the ability to understand speech without knowing the syntax. For instance, when Tarzan utters food boy eat, we know what he must mean. This is because we can take advantage of conceptual constraints among the words.

The study of Fraser, Bellugi and Brown (1963) is often cited as showing comprehension precedes production. They found children had a higher probability of understanding a sentence (as manifested by pointing to an appropriate picture) than of producing the sentence. However, there were difficulties of equating the measures of production and comprehension. Fernald (1970), using different scoring procedures, found no difference. Interestingly, Fraser et al. did find a strong correlation between which sentence forms could be understood and which could be produced. That is, sentence forms which were relatively easy to understand were relatively easy to produce. It is hard to understand this correlation except in terms of a common base for comprehension and production.

General Conclusions

I think the data reviewed in this section is largely consistent with the heuristics reviewed in Section 4 - with one notable exception. It does not seem that negative information is used to abandon or modify incorrect rules. Rather, it seems more adequate rules slowly replace the less adequate rules by some mechanisms of gradually strengthening. There is good reason for this. While the speech of adults to children is remarkably good relative to their speech to adults, it still contains ungrammatical sentences. The problem with the all-or-none induction heuristics reviewed earlier is that they would be completely turned off course by the occasional piece of mis-information.

Braine reports an interesting pair of studies in this regard. In both experiments subjects were trying to learn the grammar of an artificial language from syntactic information only. In the one experiment (Braine, 1971) they only saw what were purported to be positive instances of sentences. One group of subjects saw a small 7% ungrammatical sentences and a second group saw none. There was no difference in the learning of these two groups. In the other experiment Braine (1963) gave a group of subjects information as to what was a sentence of the language as well as to what was not. They did more poorly than a group of subjects given positive information only. Thus it seems clear that while the human unduction system can make little use of negative information, it can deal with the occasional mis-information.

6. The HAM Memory System

The remaining sections of this chapter are concerned with describing my work on computer simulation of language acquisition. This program for language acquisition is called LAS. LAS involves the integration of a *memory* network representation with a network parsing formalism like that of Woods (1970). This section describes the memory system. The next section will describe the parsing networks and the programs UNDERSTAND and SPEAK which *use* those networks for language understanding and generation, respectively. The remaining part of the program is LEARNMORE which induces the network grammars. There are two versions of the LEARNMORE program. The first one is part of the program LAS.1 and the second part of the program LAS.2. These two induction programs are described in Sections 8 and 9.

LAS uses a version of the HAM memory system (see Anderson & Bower, 1973) called HAM.2 HAM.2 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 are 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 understanding 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 one described in Anderson and Bower (1973; Ch. 9 & 12) and its details will not be discussed here.

However, it would be useful to describe here the representational formalisms used by HAM.2. Figure 8 illustrates how the information in the sentence A red square is above the circle would be represented with the HAM.2 network formalisms. There are four distinct propositions predicated about the two nodes X and Y: X is red, X is a square, X is above Y, and Y is a circle.

Insert Figure 8 about here

Each proposition is represented by a distinct tree structure. Each tree structure consists of a root proposition node connected by a S link to a subject node and by a P link to a predicate node. The predicate nodes can be decomposed into a 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 red things, the square things, and the things above Y. The individual Y is a subset of the circular things.

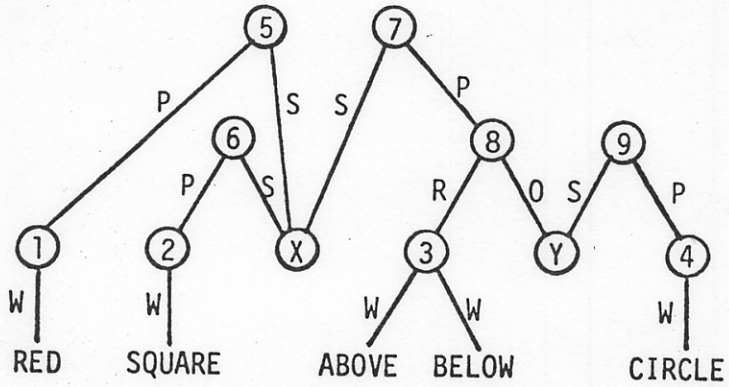


Fig. 8. An example of a propositional network representation in HAM.2.

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 ideas by links labelled W. Note also that the same idea is referenced by the two relation words above and below. As will be discussed in Section 8, such relation words are treated in LAS as having identical meaning, but simply differing in the order of their noun-phrase arguments.

Figure 8 illustrates all the network notation needed in the current implementation of LAS. There are a number of respects in which this representation is simpler than the old HAM representation. There are not the means for representing the situation (time + place) in which such a fact is true or for embedding one proposition with another. Thus, we cannot express in HAM.2 such sentences as Yesterday in my bedroom a red square above the circle or John believes that a red square is above the circle. Representations for such statements are not needed in the current LAS project because we are only concerned with representing information that can be conveyed by ostension (i.e., by pointing to physical situations). In ostension, the assumed time and place are here and now. Concepts like belief which require embedded propositions are too abstract for ostension. Should we later choose to extend LAS beyond its original ostensive referent domain, then complications will be required in the HAM.2 representations. But when starting out on a project it is preferrable to keep things as simple as possible.

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 turn out to be generally true of

human cognitive performances. (However, many of the specific details of HAM's representation have never been empirically tested.) 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 has to do with the modularity of the representation. Each proposition is coded as a network structure that can be accessed and used, independent of other structures.

So far, I have shown how the HAM.2 representation encodes the episodic information that is input to SPEAK and the output of UNDERSTAND. It can also be used to encode the semantic and syntactic information required by the parsing system. Figure 9 illustrates how HAM.2 would encode the fact that circle and square are both shapes, red and blue are both colors, circle and red belong to the word class *CA but square and blue belong to the word class *CB. Note the word class information is predicated of the words while the categorical information is predicated of the concepts attached to these words. The categorical information would be used if some syntactic rule only applied to shapes or only to colors. The word class information might be evoked if a language arbitrarily applied one syntactic rule to one word class and another rule to a different word class. Inflections are a common example of syntactic rules which apply to arbitrarily defined word classes.

Insert Figure 9 about here

HAM.2 has a small innate language of commands which cause various memory links to be built. The following four are all that are currently used:

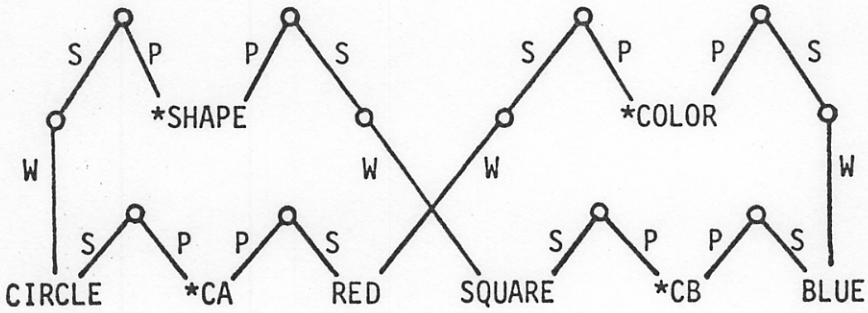


Fig. 9. An example of a HAM structure encoding both categorical and word-class information.

1. (Ideate X Y) - create a W link from word X to idea Y.
2. (Out-of X Y) - create a proposition node Z. From this root node create a S link to X and a P link to Y.
3. (Relatify X Y) - create an R link from X to Y.
4. (Objectify X Y) - create an O link from X to Y.

These commands will appear in LAS's parsing networks to create memory structures required in the conditions and actions. Often rather than memory nodes, variables (denoted X1, X2, etc.) will appear in these commands. If the variable has as its value a memory node, that node is used in the structure building. If the variable has no value, a memory node is created and assigned to it and that node is used in the memory operation.

To illustrate the use of these commands, the following is a listing of the commands that would create the structure in Figure 8.

```
(Ideate red 1)
(Ideate square 2)
(Ideate above 3)
(Ideate circle 4)
(Out-of X 1)
(Out-of X 2)
(Out-of X 8)
(Objectify 8 3)
(Relatify 8 Y)
(Out-of Y 4)
```

There are, of course, a lot of details about the HAM memory system not discussed. This is sufficient, however, to understand the use of HAM in the language system which is the concern of the remainder of this chapter.

7. The LAS Language System

The LAS language system is not very interesting as a serious attempt to model the complexities of language comprehension and production. It is clearly dwarfed by the very impressive programs of workers in artificial intelligence such as Winograd (1972) or Schank (1972). Its principle significance is that

it provides a coherent, if simplified, framework in which to conceptualize the language acquisition program.

The LAS program is written in Michigan LISP (Hafner & Wilcox, 1974). The program accepts as input lists of words, which it treats as sentences, and scene descriptions encoded in a variant of the HAM propositional language. It obeys commands to speak, understand, and learn. The logical structure of LAS is illustrated in Figure 10. Central to LAS is an augmented transition network grammar similar to that of Woods (1970). In response to the command,

---

Insert Figure 10 about here

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. In response to the command, Speak, LAS evokes the program SPEAK. SPEAK receives a picture encoding and uses the information in the network grammar to generate a sentence to describe the encoding. Note that LAS is using the same network formalism both to speak and understand. The third part of the program is LEARNMORE which induces these network grammars. Two versions of this program will be described in the next two sections. The purpose of this section is to describe the operation of SPEAK and UNDERSTAND.

The picture encodings mentioned above are expressed in the HAM network formalisms. Also encoded in the HAM memory formalism is long-term information about word classes, concepts, etc. The grammar LAS uses is also encoded in a network, but these networks are not to be confused with the HAM memory networks. They are entirely independent information formalisms.

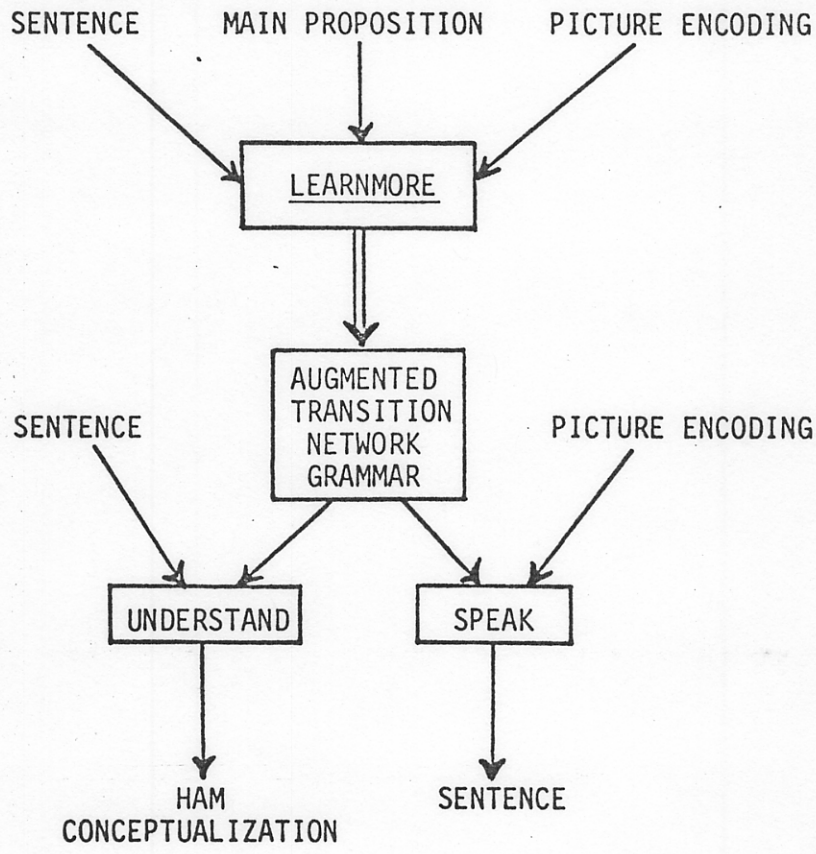


Fig. 10. A schematic representation giving the input and output of the major components of LAS--LEARNMORE, SPEAK, and UNDERSTAND.

Network Grammars

The basic idea behind a network grammar like Woods' is to have networks correspond to the phrases that would be identified in an immediate constituent analysis. So for instance, given the sentence The man who robbed the bank had a bloody nose one would analyze it into two noun phrases The man who robbed the bank and a bloody nose. The first noun phrase would be analyzed as having an embedded relative clause who robbed the bank. That embedded relative clause contains an embedded noun phrase. There would be different networks set up to analyze each of these different types of phrases - sentences, noun phrases, and relative clauses. Suppose the network analyzing a sentence encounters a noun phrase. Then it will call upon the noun phrase network to process the embedded noun phrase. In fact, one network can call itself, giving the network grammar the power of a context-free grammar. The call to one network by another is known as a push - a term I will use extensively. In Woods' formalism there was the power to perform arbitrary computations in analyzing a sentence. As we will see, this is not the case in the LAS system. To permit the LAS system such powers would be unrealistic psychologically.

To illustrate LAS's network formalisms I will present the grammar for a test language that has been used in the LAS project. It is 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 among each other. This has served as the domain for the language induction attempts.

Insert Table 1 about here

Figure 11 illustrates the parsing networks for this grammar. There are a few conventions that need to be known in reading these networks. When a

TABLE 1

A Test Grammar

GRAMMAR2

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

Example

The red square which is small is below
the circle which is right-of the triangle

label like NP is on an arc it indicates that a successful push is required to that network. When the label is prefixed by an ε (e.g., ε RA), 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.

Insert Figure 11 about here

Table 2 provides a formal specification of the information stored in LAS's network grammars. A node either has a number of arcs proceeding out of it (1a) or it is a stop node (1b). In speaking and understanding LAS will try to find some path through the network ending with a stop node. Each arc consists of some condition that must be true of the sentence for that arc to be used in parsing (understanding) the sentence. The second element is an action

---

Insert Table 2 about here

---

to be taken if the condition is met. This action will create a piece of HAM conceptual structure to correspond to the meaning conveyed by the sentence at that point. Finally, an arc includes specification of the next node to which control should transfer after performing the action. An action consists of zero or more HAM memory commands (rule 3). It will consist of zero actions when no meaning corresponds to the word parsed by that arc (e.g., the). A condition can also consist of zero or more memory commands (rule 4a). These specify properties that must be true of the incoming word. Alternatively, a condition may involve a push to an embedded network (rule 4b). For instance, suppose the structure in Figure 8 were to be spoken using GRAMMAR1. The START

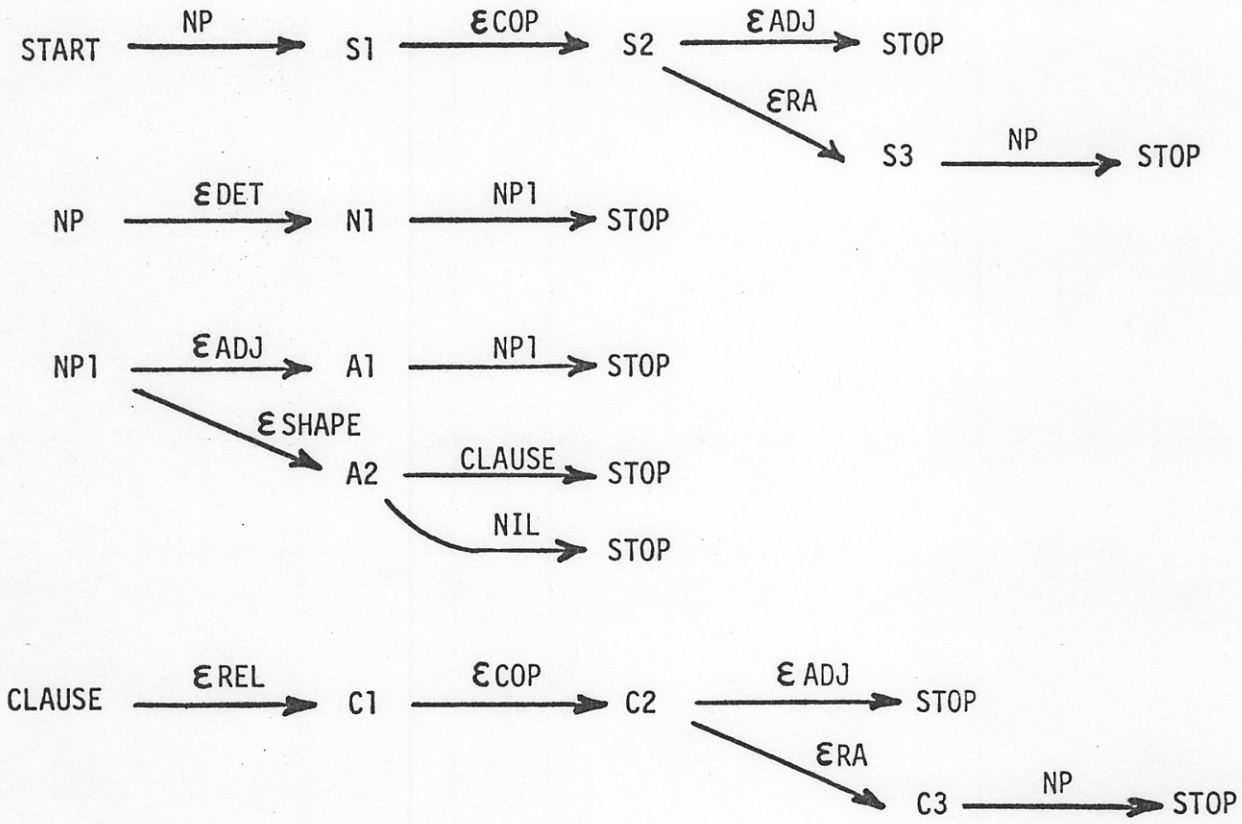


Fig. 11. The augmented transition networks encoding the grammar defined in Table 1.

TABLE 2

Formal Specification of the Network Grammar

NODE	→ node ARC*	(1a)
	→ stop	(1b)
ARC	→ (CONDITION ACTION NODE)	(2)
ACTION	→ (COMMAND*)	(3)
CONDITION	→ (COMMAND*)	(4a)
	→ (push VAR T NODE)	(4b)
COMMAND	→ (FUNCTION ARG ARG)	(5)
ARG	→ memory node	(6a)
	→ word	(6b)
	→ VAR	(6c)
FUNCTION	→ out-of, objectify, relatify, ideate	(7)
VAR	→ X1, X2, X3, X4, X5	

Footnote: The superscript (*) indicates the possibility of multiple occurrences of this symbol. Parentheses indicate the element is optional.

network would be called to realize the X is above Y proposition. The embedded NP network would be called to realize X is red and X is square propositions. In pushing to a network two things must be specified--NODE, which is the first node in the embedded transition network and VAR, which is the memory node at which the main and embedded propositions intersect. In the example the memory node is X. The three rules 6a, 6b, and 6c specify three types of arguments that memory commands can have. They can either directly refer to memory nodes, or refer to the current word in the sentence, or refer to variables which are bound to memory nodes in the course of parsing. Table 3 provides the LISP encoding of the network in Figure 11. It would be useful to examine one of the arcs in Table 3 and see how it is generated by the grammar in Table 2. Consider the first arc leading out of START which is encoded as ((PUSH X1 T NP) ((OUT-OF X1 X2)) S1). It is encoded, according to rule 2, as a condition, (PUSH X1 T NP); an action, ((OUT-OF X1 X2)); and a next network node, S1. The condition is a push and so obeys rule 4b. The symbol T in the push is a place-holder for control information that will be used by the UNDERSTAND program. The action obeys rule 3 and consists of a list of one memory command (OUT-OF X1 X2). That memory command obeys the syntax of rule 5.

Insert Table 3 about here

Relation Between Grammar Networks and Propositional Networks

Note that there tends to be a 1-1 correspondence between HAM propositions and LAS networks. That is, each network expresses just one proposition and calls one embedded network to express any other propositions. This correspondence is not quite perfect in Figure 11, but the grammars induced by LEARNMORE have necessarily a perfect correspondence.

TABLE 3

LISP Commands Creating Figure 11

```
(PUT  START  PATH
       (((PUSH  X1  T  NP) ((OUT-OF  X1  X2))  S2)))

(PUT  S1  PATH
       ((((OUT-OF  WORD  COP))      NIL     S3)))

(PUT  S2  PATH
       ((((OUT-OF  WORD  ADJ)  (IDEATE  WORD  X2))     NIL     STOP)
        (((OUT-OF  WORD  RA)  (IDEATE  WORD  X3))  ((RELATIFY  X2  X3))  S3)))

(PUT  S3  PATH
       ((((PUSH  X4  T  NP))  ((OBJECTIFY  X2  X4))  STOP)))

(PUT  NP  PATH
       ((((OUT-OF  WORD  DET))     NIL  N1)))

(PUT  N1  PATH
       (((PUSH  X1  T  NP1))     NIL     STOP)))

(PUT  NP1  PATH
       ((((OUT-OF  WORD  ADJ)  (IDEATE  WORD  X2))  ((OUT-OF  X1  X2))  A1)
        (((OUT-OF  WORD  SHAPE)  (IDEATE  WORD  X2))  ((OUT-OF  X1  X2))  A2)))

(PUT  A1  PATH
       (((PUSH  X1  T  NP1)     NIL     STOP)))

(PUT  A2  PATH
       (((PUSH  X1  T  CLAUSE))     NIL     STOP)
        ((  NIL                     NIL     STOP)))

(PUT  CLAUSE  PATH
       ((((OUT-OF  WORD  REL))     NIL     C1)))

(PUT  C1  PATH
       ((((OUT-OF  WORD  COP))     NIL     C2)))

(PUT  C2  PATH
       ((((OUT-OF  WORD  ADJ)  (IDEATE  WORD  X2))  ((OUT-OF  X1  X2))  STOP)
        (((OUT-OF  WORD  RA)  (IDEATE  WORD  X3))  ((OUT-OF  X1  X2)  (RELATIFY
                                                          X2  X3))  C3)))

(PUT  C3  PATH
       (((PUSH  X4  T  NP)  ((OBJECTIFY  X2  X4))  STOP)))
```

These grammar networks have a number of features to commend them. SPEAK and UNDERSTAND use the same network for sentence comprehension and generation. In this way, LAS has only to induce one set of grammatical rules to do both tasks. Such networks 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 SPEAK Program

The SPEAK program is simpler than the UNDERSTAND program because it does not require as elaborate a control mechanism for back-up. A flow-chart giving a gross and approximate diagram of its information control is given in Figure 12. SPEAK starts with a HAM network of propositions tagged as to-be-spoken and a topic of the sentence. The topic of the sentence will correspond to the first meaning-bearing element in the START network. SPEAK searches through its START network looking for some path that will express a to-be-spoken proposition attached to the topic and which expresses the topic as the first element. It determines whether a path accomplishes this by evaluating

---

Insert Figure 12 about here

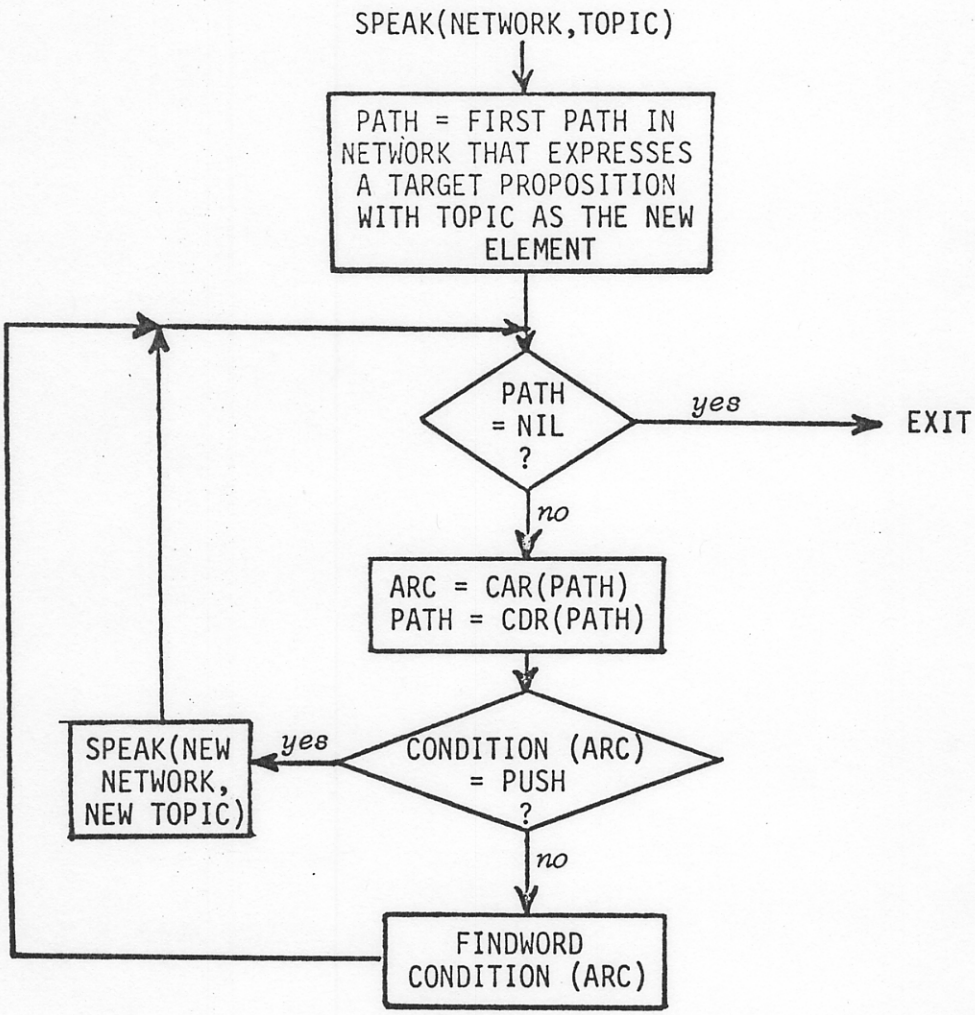


Fig. 12. A flowchart illustrating the high-level control structure of the SPEAK program.

the actions associated with a path and determining if they created a structure that appropriately matches the to-be-spoken structure. When it finds such a path it uses it for generation.

Generation is accomplished by evaluating the conditions along the path. If a condition involves a push to an embedded network, SPEAK is recursively called to speak some sub-phrase expressing a proposition attached to the main proposition. The argument for a recursive call of SPEAK are the embedded network and the node that connects the main proposition and the embedded proposition. This connecting node is what is common between the memory structure described by the embedded and embedding proposition . It is also the topic of the embedded network. In effect, the embedded network is elaborating on the semantic referent of that node.

If the condition does not involve a push it will contain a set of memory commands specifying that some features be true of a word. It will use these features to determine what the word is. The word so determined will be spoken. The subprogram FINDWORD is the one that uses a condition to retrieve a word.

As an example, consider how SPEAK would generate a sentence corresponding to the HAM structure in Figure 13 using the English-like grammar in Figure 11. Figure 13 contains a set of propositions about three objects denoted by the nodes G246, G195, and G182. Of node G246 it is asserted that it is a triangle, and that G195 is right of it. Of G195 it is also asserted that it is a square and that it is above G182. Of G182 it is also asserted that it is square, small, and red. Figure 14 outlines the control structure of the generation of this sentence. LAS enters the START network intent on producing some

---

Insert Figures 13 and 14 about here

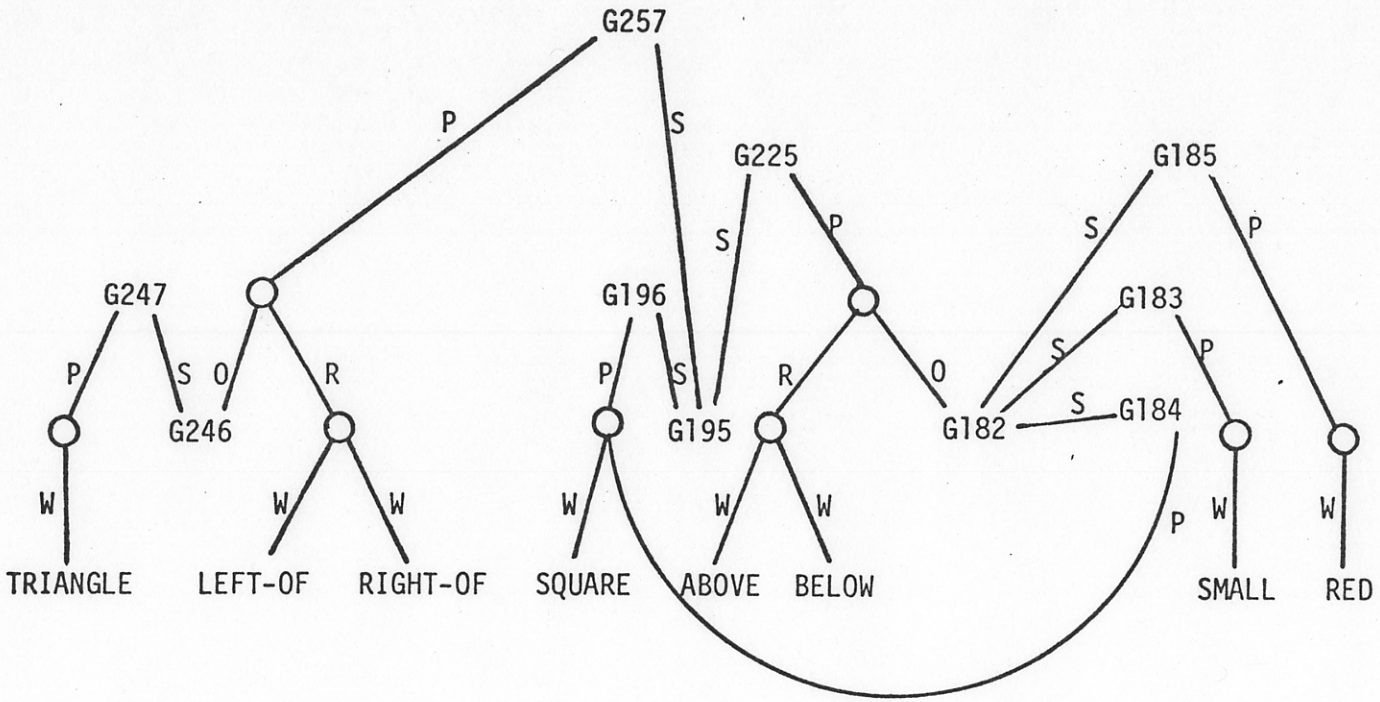


Fig. 13. This to-be-spoken HAM structure is given as input to the SPEAK program.

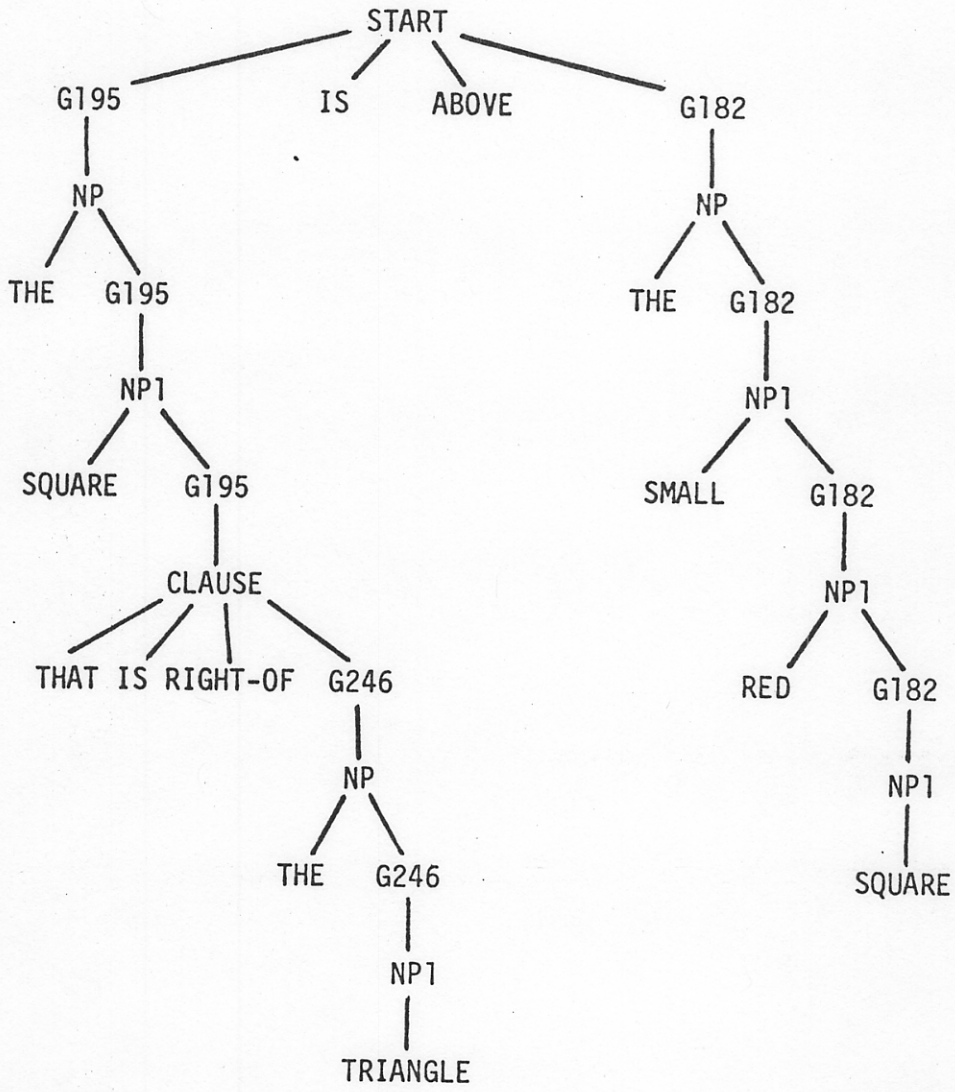


Fig. 14. A tree structure showing the network calls and words output in the generation of a sentence to express the information contained in Fig. 13.

utterance about G195. The first path through the network involves predicating an adjective of G195, but there is nothing in the adjective class predicated of G195. The second path through the START network corresponds to something LAS can say about G195--it is above G182. Therefore, LAS plans to say this as its main proposition. First, it must find some noun phrase to express G195. The substructure under G195 in Figure 14 reflects the construction of this phrase. The NP network is called with its connecting node G195. It prints out the and calls NP1 which retrieves square and calls CLAUSE which prints that, is, and right-of and recursively calls NP to print the triangle to express G246. Similarly, recursive calls are made on the NP1 network to express G182 as the small red square. A noun phrase is concluded once all the to-be-expressed information about the referent of the noun phrase has been expressed.

The actual sentence generated is dependent on choice of topic for the START network. Given the same to-be-spoken HAM network, but the topic G246, SPEAK generated A triangle is left-of a square that is above a small red square. Given the topic G182 it generated A red square that is below a square that is right-of a triangle is small. Note how the choice of the relation words left-of versus right-of and of above versus below is dependent on choice of topic.

It is interesting to inquire what is the linguistic power of LAS as a speaker. Clearly it can generate any context-free language since its transition networks correspond, in structure, to a context-free grammar. However, it turns out that LAS has certain context-sensitive aspects because its productions are constrained by the requirement that they express some well-formed HAM conceptual structure. Consider two problems that Chomsky (1957) regarded as not handled well by context-free grammars: The first is agreement of number

between a subject NP and verb. This is hard to arrange in a context-free grammar because the NP is already built by the time the choice of verb number must be made. The solution is trivial in LAS--when both the NP and verb are spoken, their number is determined by inspection of whatever concept in the to-be-spoken structure underlies the subject. The other Chomsky example involves the identity of selectional restrictions for active and passive sentences. This is also achieved automatically in LAS, since the restrictions in both cases are regarded simply as reflections of restrictions on the semantic structure from which both sentences are spoken.

While LAS can handle those features of natural language suggestive of context-sensitive rules, it cannot handle examples like languages of the form $a^n b^n c^n$ which require context-sensitive grammars. It is interesting, however, that it is hard to find natural language sentences of this structure. The best I can come up with are respectively-type sentences, e.g., John and Bill hit and kissed Jane and Mary, respectively. This sentence is of questionable acceptability.

There are some linguistic constructions like number and active-passive which are more easily described by context-sensitive rules, but which can be generated by context-free rules. Languages which contain such constructions are still called context-free. However, there are true context-sensitive languages like the $a^n b^n c^n$ example which, in principle, cannot be generated by a context-free grammar.

LAS could generate a language of the form $a^n b^n c^n$ if its HAM semantic structure contained to-be-spoken expressions of this form. The LAS networks really are transducers. That is, they translate one information representation, the HAM network, into another representation, the sentence. Thus, questions

about the context-sensitive features of the language generated by LAS can depend on context-sensitive features of the memory structure. Unfortunately, the context-sensitive features of the HAM representation are not well defined.

There is another way that LAS could gain context-sensitive powers from its memory structure. Suppose in speaking a sentence it could write notes to itself in the memory structure and later read these. Given this general read-write capacity it could behave as a Turing machine. However, in speaking LAS can only read from its to-be-spoken memory structure. Its "writing" consists of outputing a sentence and it is prevented from reading that sentence. These restrictions on reading and writing are probably too severe but it would be unrealistic psychologically to give LAS general read-write capabilities.

Actually, LAS does have a certain very limited writing capability within its memory structure. It can tag memory structures as already spoken. This is to prevent it from generating expressions like The blue blue square which is blue is blue in which the same proposition is expressed over and over again. This does give the grammar context-sensitive powers of the sort that cannot be obtained in a context-free grammar.

The UNDERSTAND Program

The search in SPEAK for a grammatical realization of the conceptual structure was limited to search through a single network at a time. Search terminated when a path was found which would express part of the to-be-spoken HAM structure. Because search is limited to a single parsing network, the control structure was simply required to execute a depth-first search through a finite network. In the UNDERSTAND program it is necessary, when one path through a network fails, to consider the possibility that the failure may be in a parsing of a sub-network called on that path. Therefore, it is possible to have to back into

a network a second time to attempt a different parsing. For this reason the control structure of the UNDERSTAND program is more complicated.

Perhaps an English example would be useful to motivate the need for a complex control structure. Compare the two sentences The Democratic party hopes to win in '76 with The Democratic party hopes are high for '76. A main parsing network would call a noun phrase network to identify the first noun phrase. Suppose UNDERSTAND identified The Democratic party. Later elements in the second sentence would indicate that this choice was wrong. Therefore, the main network would have to re-enter the noun phrase network and attempt a different parsing to retrieve The Democratic party hopes. When UNDERSTAND re-entered the noun-phrase network to retrieve this parsing, it must remember which parsings it tried the first time so that it does not retrieve the same old parsing.

Figure 15 provides the flowcharts of the control in three functions which approximately characterize UNDERSTAND. These flowcharts are sufficiently rich in LISP notation that I would warn the non-LISPer to expect less than full comprehension. They are included to make available some of the technical details to the LISPer. The top level function is called PUSH (Part a). The function PUSH simply calls ANALYZE to find a path through the network. It takes the actions collected along the path and performs them to create a temporary memory structure. This structure represents part of the understanding of the sentence. The arguments to PUSH are: STRING, the remaining part of the to-be-understood sentence; TOPIC, the topic for that network; CONTROL, a history of the paths taken in any previous entries into that network; and NODE, the start-node in the network being entered. It returns sufficient

---

Insert Figure 15 about here

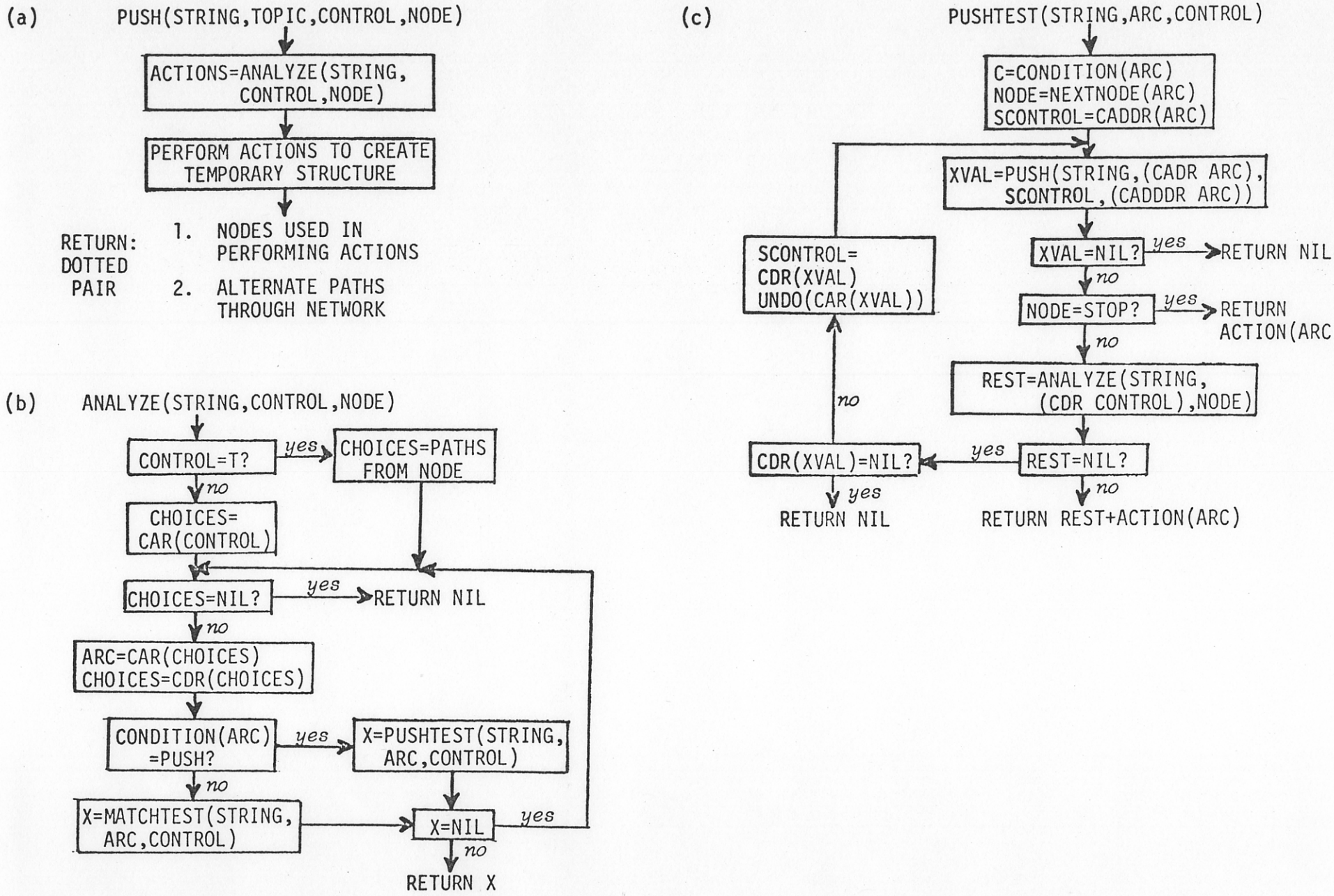


Fig. 15. Flowcharts illustrating the high-level control structure of the UNDERSTAND program.

information so that it can be re-entered. The information includes the nodes in the temporary structure. If that network has to be re-entered, these temporary nodes will have to be erased. Secondly, it returns as control a list of the remaining possible paths (if any) through the network.

The function ANALYZE (Part b) is called with arguments STRING, CONTROL, and NODE, the next node in the parsing network. If the node has not been used in a previous entry into the network, CONTROL is T. Otherwise, it is a listing the remaining paths from NODE which have not yet been tried. ANALYZE evaluates the conditions on the paths possible from NODE to see if any are satisfied by the part of the sentence contained in STRING. If the condition involves matching to long-term memory, MATCHTEST is called. MATCHTEST checks a memory condition. Its logic is similar but simpler than that of PUSHTEST which is illustrated in Figure 15.

The function in Part (c), PUSHTEST, evaluates an arc whose condition involves a push to an embedded network. PUSHTEST calls PUSH to see whether the embedded network will analyze part of the string. If successful, PUSH returns the structure XVAL. This contains information about the memory structure created (on the CAR of XVAL) and about possible alternate paths through the embedded network (on the CDR of XVAL). ANALYZE is then called to evaluate the paths subsequent to the current arc. If these fail, a function UNDO is called to erase any temporary structure created in the embedded push. If there are any further paths through the embedded network ((CDR XVAL) $\neq$ NIL), PUSH is called again but with a new control structure. The functions described here do not correspond to the ones implemented. Some complexities have been omitted for expository sake. A documented listing of the actual program can be obtained by writing to me.

These programs serve to find some path through the network that results in a complete parsing of the sentence. In parsing a sentence it invokes the same hierarchy of networks that it would in generating a sentence. Thus, for instance, given the sentence, The square that is right-of the triangle is above the small red square, LAS would parse it following the control structure of Figure 14, retrieving the HAM structure in Figure 13. This is an example of a reversible augmented transition network. Simmons (1973) has a similar idea but uses two different networks, one for generation and one for analysis. In exchange for this complexity, he gains the savings in only having to write one interpretative process both for understanding and generation.

It is also of interest to consider the power of LAS as an acceptor of languages. It is clear that LAS, as presently constituted, can accept exactly the context-free languages. This is because, unlike Woods' (1970) system, actions on arcs cannot influence the results of conditions on arcs, and therefore, play no role in determining whether a string is accepted or not. However, what is interesting is that LAS's behavior as a language understander is relatively little affected by its limitations on grammatical powers. Consider the following example of where it might seem that LAS would need a context-sensitive grammar. In English noun phrases, it seems we can have almost arbitrary numbers of adjectives. This led to the rule in Figure 11 where NP1 could recursively call itself each time accepting another adjective. There is nothing in this rule to prevent it from accepting phrases like the small big square or other unacceptable constructions. However, in practice this does not lead LAS into any difficulties because it would never be presented with such a sentence due to the constraints on what a speaker may properly say to LAS.

It is useful to compare generation versus analysis in the LAS networks. The flow of control is somewhat different in the two circumstances. In sentence production LAS first finds a path through the START network to correspond to the main proposition that it wants to assert. Then it will proceed to generate the first phrase in the sentence. Thus with respect to the control structure of Figure 14, SPEAK completely generates one level before it expands the left-most substructure. In contrast, in UNDERSTAND, the control structure is generated as the words come in. This means that before any further control structure is evoked at a particular level, the left-most substructure will be completely generated. Network grammars are rather ideal in that they permit with equal facility the breadth expansion required by SPEAK to plan the production and the depth expansion required by UNDERSTAND to follow the spoken sentence.

8. The Program LAS.1

Having now reviewed how LAS understands and produces sentences, I will turn to describing the induction program, LEARNMORE. There were two versions of this program. The first is part of a general system called LAS.1 and the second part of a general system called LAS.2. This section will describe the LEARNMORE system in LAS.1. It consists of three principle programs: BRACKET, SPEAKTEST, and GENERALIZE. Before describing this system, it is wise to briefly review the conditions under which LAS learns a language. It is assumed that LAS already has concepts attached to the words of the language. That is, lexicalization is complete. The task of LAS is to learn the grammar of the language--that is, how to go from a string of words to a representation of their combined meaning. Later projected versions of LAS will deal with learning word meanings.

The philosophy behind the LEARNMORE program is to provide LAS with the same information that a child has when he is learning a language through ostension. It is assumed that in this learning mode the adult can direct the child's attention to that aspect of the situation which is being described. Thus, LEARNMORE is provided with a sentence, a HAM description of the scene and an indication of the main proposition in the sentence. It is to produce as output the network grammar that will be used by SPEAK and UNDERSTAND. It is possible that the picture description provides more information than is in the sentence. This provides no obstacle to LAS's heuristics. With the information of sentence, semantic referent, and main proposition BRACKET will assign a surface structure to the sentence. SPEAKTEST will determine whether the sentence is handled by the current grammar. If not, additions are made to handle the sentence. These additions generalize to other cases so that LAS can understand many more sentences than the ones it was explicitly trained with.

The LAS.1 and LAS.2 programs were evolved to describe a particularly simple two-dimensional world. This world consisted of various geometrical shapes of different sizes and colors. The shapes could bear a number of spatial relations to one another (above, below, left-of, right-of). The ambition of these programs was to learn languages (natural or artificial) adequate to describe this semantic domain. Despite the simplicity of the semantic domain, a lot was learned about language induction.

It is important to have a specified semantic domain in a language learning enterprise. It provides a very explicit criterion for success. 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, Siklóssy) is that they have take 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.

BRACKET - The Graph-Deformation Condition

The BRACKET program is an algorithm for taking a sentence and a HAM conceptual structure and producing a bracketing of the sentence which indicates its surface structure. This program makes use of the Graph-Deformation Condition (GDC) discussed in Sections 4 and 5. The GDC applies to a prototype structure which must be derived from the underlying HAM semantic structure. Recall that the prototype structure differs from the semantic structure because it has in it those relational terms used in the sentence. The relational terms used in LAS.1 are very simple so that it is not difficult to calculate the prototype structure.

I will go through one example of the derivation of a surface structure from the HAM conceptual structure via the prototype structure. It should be emphasized, however, that the prototype structure is not actually calculated in the BRACKET program. It is implicit in the logic of the program. I make the prototype structure explicit for purposes of exposition. If the reader is interested in the programming details he should write to me for a copy of the March, 1974 BRACKET program.

Figure 16a illustrates the HAM structure that might underlie the English sentence The red square which is right-of the triangle is below the small circle.

Insert Figure 16 about here

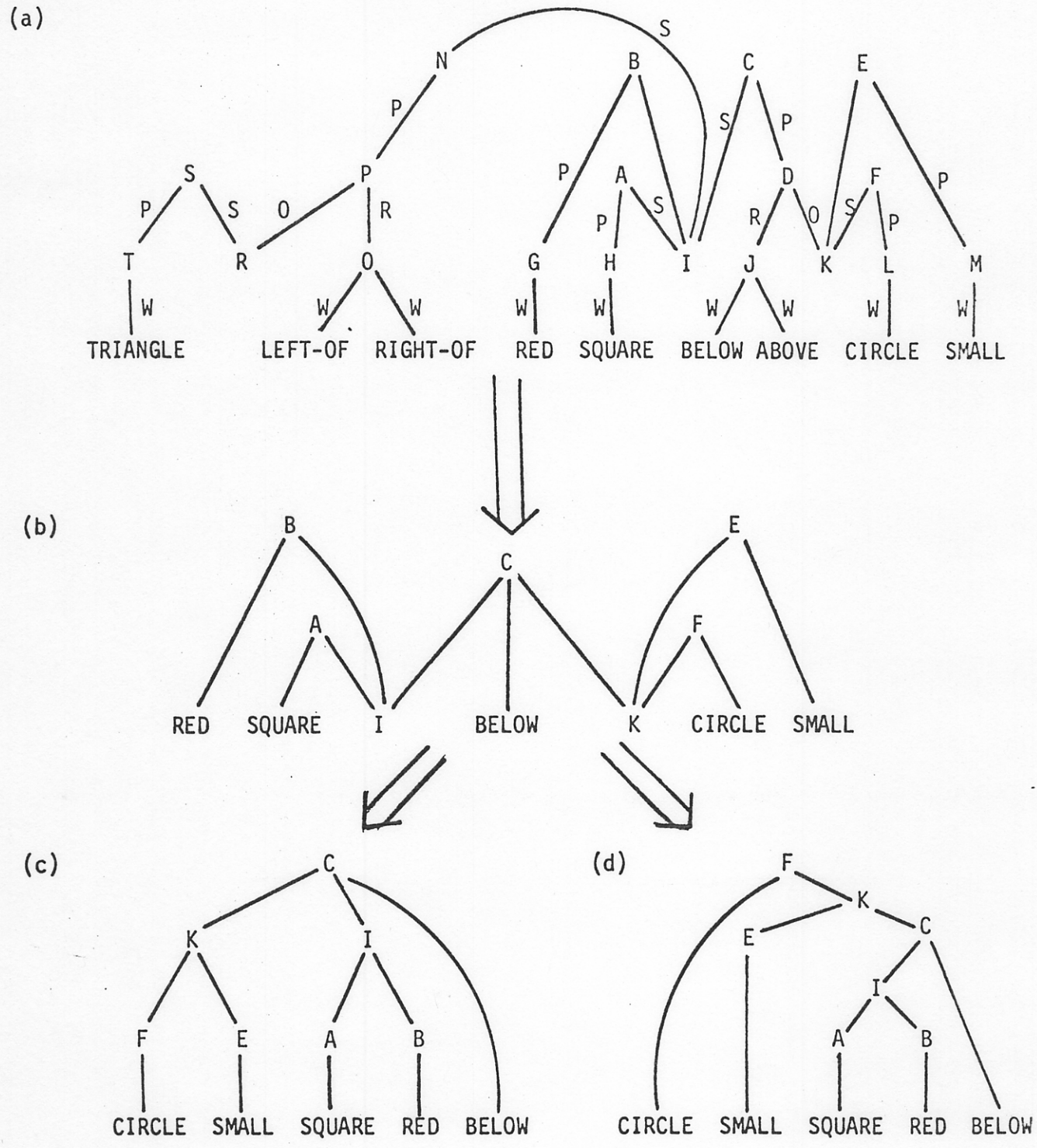


Fig. 16. The prototype structure in (b) is derived from the HAM structure in (a). From this prototype structure two surface structures, (c) and (d), can be imposed on the same string of words.

We will use it to derive the prototype structure in Figure 16b for the sentence Circle small square red below which is a sentence in an artificial language that LAS will be learning. (This is also the prototype structure for the English sentence The small circle is below the red square.) The prototype structure is derived by deleting from the HAM structure all nodes except proposition nodes (A, B, C, E, and F), the individual nodes (I and K) and the words (red, square, below, circle, small). 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.

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. Part (c) indicates one such graph-deformation of the prototype sentence. Note that all the links in (b) 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 which others and which are right of which others. Although the prototype structure in (b) is set forth in a special spatial array the choice is arbitrary. In contrast, the surface structure in (c) does specify the spatial relations of links. From (c) we may derive a bracketing of the sentence indicating its surface structure--((circle small) (square red) below).

It is important to note what has happened to the relational term below in this derivation. Note that both it and above were attached to the same idea node J. In learning the meanings of such complementary words, the learner would see them applied to the same situation. Therefore, he would naturally come to regard them as synonymous. The difference would be

syntactic from the point of view of the learner. Below is heard with the logical object first and above with the logical subject first. Thus, it seems that the natural course for language acquisition would be to evolve a parsing system on the model of Schank's (1972) in which synonymous structures are mapped into the same semantic structure . In the application of the graph deformation condition, LAS can reorganize the semantic structure around whatever relation term appears in the sentence to derive a surface structure.

I claimed (Anderson, 1975) that:

> "It seems reasonable that all natural languages have as their semantics the same order-free prototype network. They differ from one another in (a) the spatial ordering their surface structure assigns to the network, and (b) the insertion of non-meaning-bearing morphemes into the sentence."

Clearly, that claim will have to be modified because of the fact that the prototype structure can vary from language to language. Still the GDC imposes strong constraints on the form of possible languages. This is because there are strong constraints on the prototype structure that may be derived from a semantic structure and the semantic structure is still held to be language-universal. As discussed in Section 4 it seems that there are serious constraints on the set of possible relational structures. It seems to me that there is also an iron-clad constraint with respect to individual nodes like I and K in the semantic structure. Their connections may be deleted in the prototype structure (as was the connection from I to right-of the triangle) but they may not be moved to other nodes. Thus, the connection from I to red cannot be moved to K nor can the connection from K to small be moved to I. This means that a sentence like The small square is above the red circle could not be used to express the HAM structure in (a) by any natural language. The experiment reported in Section 5 demonstrated how difficult it is to learn such a language.

Main Proposition

BRACKET needs to know more than just the prototype structure to infer the surface structure of the sentence. As parts (c) and (d) of Figure 16 show, the same string of words can have the same prototype structure deformed into more than a single surface structure. The difference between (c) and (d) is the choice of which proposition is principal and which is subordinate. The structure in (d) 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 panel (c), the teacher would direct the learner to the picture of a red triangle above a small circle. He would both have to assume 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 Details of BRACKET's Output

So far, for purposes of exposition I have simplified the specification of BRACKET's output. Also the example in Figure 16 was particularly simple because there were no non-meaning-bearing words to consider. Consider, how BRACKET would handle the sentence A triangle is left-of a square that is above a small red square, given the HAM structure in Figure 13. It is left as an exercise for the reader to derive the sentence's prototype structure which is an intermediate and implicit step in BRACKET's computation. BRACKET returned a complex expression of the form (G257 (G246 G247 a triangle) is left-of (G195 G196 a square (G195 G225 that is above (G182 G183 a small (G182 G185

red (G182 G184 square))))))). The embedding of parentheses reflects the levels of the surface structure. The main proposition is G257 which is given as the first term in the bracketing. The first bracketed sub-expression describes the subject noun phrase. The first element in the sub-expression G246 is the node that links the embedded proposition G247 to the main proposition G257. The first two words of the sentence A triangle are placed in this bracketed sub-expression. The next two words is left-of are in main bracketing. There are no embedded propositions corresponding to these two. The remainder of the output of BRACKET corresponds to a description of the element G195. The first embedded proposition G196 asserts this object is a square and the second proposition, G225, asserts that G195 is above G182. Note that the G225 proposition is embedded as a sub-expression within the G196 proposition. The last element in the G225 proposition is (G182 G183 a small (G182 G185 red (G182 G184 square))). This expression has in it three propositions G183, G185, G184 about G182.

The above example illustrates the output of BRACKET. Abstractly, the output of BRACKET may be specified by the following five rewrite rules:

1. S → proposition element*
2. element → morpheme
3. element → CLAUSE
4. CLAUSE → (topic S)
5. CLAUSE → (topic S CLAUSE)

That is, each bracketed output is a proposition node followed by a sequence of elements (rule 1). These elements consist of the relation, its arguments, and non-meaning-bearing morphemes. These elements are either rewritten as words (rule 2) or a bracketed clause (rule 3). A bracketed clause begins with a topic node which indicates the connection between the embedded and embedding

propositions (rule 4). The elements within a clause are expanded according to rule 1. That is, they will be morphemes or bracketed clauses. When a number of embedded propositions are attached to the same node (e.g., G182), they are embedded within one another in a right-branching manner (rule 5). Note that BRACKET induces a correspondence between a level of bracketing and a single proposition. Each level of bracketing will also correspond to a new network in LAS's grammar. Because of the modularity of HAM propositions, a modularity is achieved for the grammatical networks.

The insertion of non-function words into the bracketing is a troublesome problem because there are no semantic features to indicate where they belong. Consider the first word a in the example sentence above. It could have been placed in the top level of bracketing or in the sub-expression containing triangle. Currently, all the function words to the right 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.

Non-meaning-bearing morphemes pose further problems besides bracketing. Consider a sequence of such morphemes in a noun phrase. That sequence could have its own grammar that, in principle, might constitute an arbitrary recursive language. The sentence's semantic referent could provide no clues at all as to the structure of that language. Therefore, we would be back to the same impossible syntactic induction task that we characterized in Section 3. Hence, it is comforting to observe that the structure of these strings of non-meaning-bearing morphemes tends to be very simple. There are not many examples of these strings being longer than two words. For example, in the phrase which the boy ate the words which and the would be regarded as a string of two non-meaning-bearing morphemes. Thus, it seems that the languages constituted by these non-meaning-bearing strings are nothing more than very simple finite cardinality languages which pose, in themselves, no serious induction problems. The various stretches of non-meaning-bearing morphemes in a sentence could also have complex interdependencies thereby posing serious induction problems. For example, gender inflection of a subject noun might depend on gender inflection of an object noun. However, it does not seem to be the case that these dependencies exist. So it seems that the structure of natural language is simple just at those points where it would have to be for a semantics-based approach to work.

SPEAKTEST

The function of SPEAKTEST is to test whether its grammar is capable of generating a sentence and, if it is not, to modify appropriately the grammar so that it can. SPEAKTEST is called after BRACKET is complete. It receives from BRACKET a HAM conceptual structure, a bracketed sentence, the main proposition, and the topic of the sentence. As in the SPEAK program SPEAKTEST attempts to

find some path through its network which will express a proposition attached to the topic. If it succeeds, no modifications are made in the network. If it cannot, a new path is built through the network to incorporate the sentence.

An Example of Grammar Induction

The best way to understand the operation of SPEAKTEST is to watch it go through one example. The target language it was given to learn is given in Table 4. This is a very simple artificial language. In Table 4 are also given the 14 sentences that LAS studied in learning the language. The reason for choosing this language is that it is of just sufficient complexity to illustrate LAS's acquisition mechanisms.

Insert Table 4 about here

Figure 17 illustrates LAS's handling of the first two sentences that come in. The first sentence is Square triangle above. This sentence is returned by BRACKET as (G174 (G115 G116 square) (G148 G149 triangle) above). G174 refers to the main proposition given as an argument to LEARNMORE. Since this is LAS's first sentence of the language the START network will, of course, completely fail to parse the sentence. It has no grammar yet. Therefore, it induces the top-level START network in Figure 17a. Since the first two elements after G174 in the bracketed sentence are themselves bracketed, the first two arcs in the network will be pushes to sub-networks. The third arc contains a condition on the word above. The condition made is that it be a member of the word class RA. This class was created for this sentence and only contains the word above at this point. Having now constructed a path through the START network, SPEAKTEST checks the sub-networks on that path to see whether they

---

Insert Figure 17 about here

TABLE 4

Test Language to be Learned

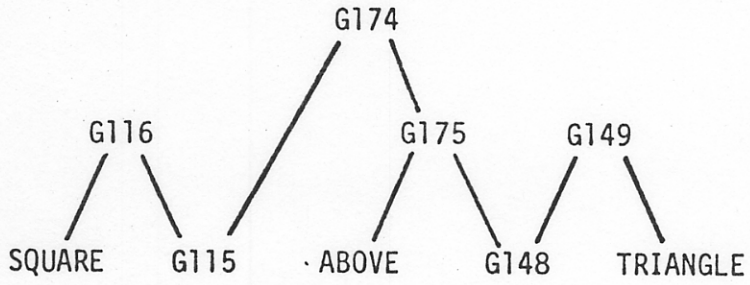
Grammar

S	→ NP NP RA
	→ NP NP RB
NP	→ SHAPE (COLOR) (SIZE)
SHAPE	→ square, triangle
COLOR	→ blue, red
SIZE	→ large, small
RA	→ above, right-of
RB	→ below, left-of

Sentences Studied

1. SQUARE TRIANGLE ABOVE
2. TRIANGLE SQUARE RIGHT-OF
3. SQUARE TRIANGLE BELOW
4. TRIANGLE SQUARE LEFT-OF
5. SQUARE RED TRIANGLE BLUE ABOVE
6. TRIANGLE LARGE SQUARE SMALL RIGHT-OF
7. TRIANGLE RED TRIANGLE RED ABOVE
8. SQUARE SMALL TRIANGLE LARGE RIGHT-OF
9. SQUARE BLUE TRIANGLE LARGE RIGHT-OF
10. SQUARE BLUE SMALL TRIANGLE RIGHT-OF
11. TRIANGLE RED SQUARE BLUE LEFT-OF
12. TRIANGLE SMALL SQUARE RED SMALL BELOW
13. SQUARE BLUE TRIANGLE BLUE LARGE LEFT-OF
14. SQUARE RED LARGE TRIANGLE RED LARGE BELOW

(a)



(G174 (G115 G116 SQUARE) (G148 G149 TRIANGLE) ABOVE)

START $\xrightarrow{NP1}$ SX1 $\xrightarrow{NP2}$ SX2 $\xrightarrow{\varepsilon RA}$ STOP

NP1 $\xrightarrow{\varepsilon N1}$ STOP

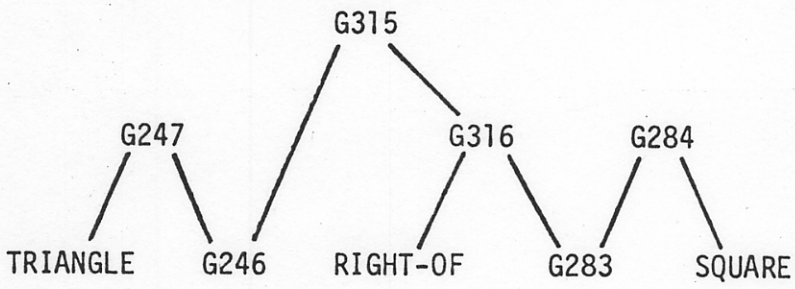
NP2 $\xrightarrow{\varepsilon N2}$ STOP

RA = ABOVE

N1 = SQUARE

N2 = TRIANGLE

(b)



(G315 (G246 G247 TRIANGLE) (G283 G284 SQUARE) RIGHT-OF)

RA = ABOVE,RIGHT-OF

N1 = SQUARE,TRIANGLE

N2 = TRIANGLE,SQUARE

Fig. 17. Treatment of LAS.1 of the first two sentences in the induction sequence.

can handle the bracketed sub-expressions in the sentence. This is accomplished by a recursive call to SPEAKTEST. For the first phrase, SPEAKTEST is called, taking as arguments the network NP1, the phrase (G116 square) and the topic G115. In network NP1 the word class N1 is created to contain square, and in network NP2 the word class N2 contains triangle.

Note in this example that I am using semi-mnemonic labels for nodes and word classes like NP1, NP2, N1, RA. In point of fact these were not the labels generated by the program. However, I have taken the liberty of replacing the program's nonsense labels with these. I hope this will facilitate the still difficult task of following these examples.

Note in this example how the information provided by BRACKET completely specified the embedding of networks. The sentence provided by BRACKET was (G174 (G115 G116 square) (G148 G149 triangle) above). The first element G174 was the main proposition. The second element (G115 G116 square) was a bracketed sub-expression indicating a sub-network should be created. Similarly, the third expression indicated a sub-network. The last element above was a single word and so could be handled by a memory condition in the main network.

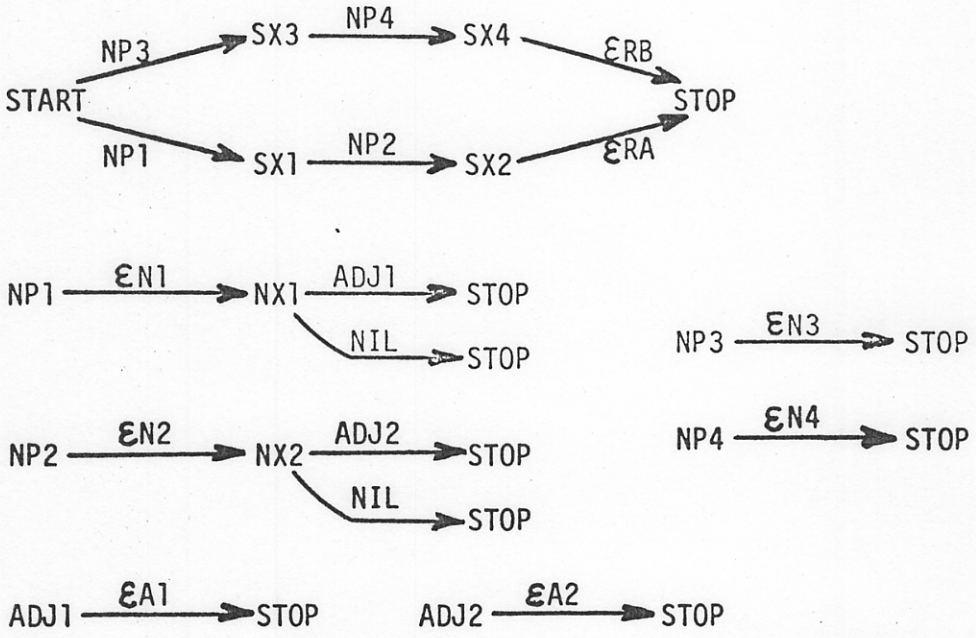
The second sentence is triangle square right-of. This is transformed by BRACKET to (G315 (G246 G247 triangle) (G283 G284 square) right-of). Because of the narrow one-member word classes this sentence cannot be handled by the current grammar. However, SPEAKTEST does not add new network arcs to handle the sentence. Rather, it expands word class RA to include right-of, word class N1 to include triangle, and word class N2 to include square. The grammar is now at such a stage that LAS could speak or understand the sentences triangle square above or square square right-of and other sentences which it had not studied. Thus, already the first generalizations have been made. LAS can produce and understand novel sentences.

This illustrates the type of generalizations that are made within the SPEAKTEST program. For instance, consider the generalization that arose when SPEAKTEST decided to use the existing network structure to incorporate triangle, the first word of the second sentence. This involved (a) using the same sub-network NP1 that had been created for square and (b) expanding the word class N1 to include triangle. Both decisions rested on semantic criteria. The network NP1 was created to analyze a description of a node attached to the main proposition by the relation S. Triangle was a description of the node G246 which is related by S to the main proposition. On the basis of this identity of semantic function, LAS assigns the parsing of triangle to the network NP1. Within the NP1 network the word class N1 contains words which are predicates of the subject node. Triangle has this semantic function and is therefore added to the word class.

This is an example of the principle of semantics-induced equivalence of syntax (PSIES) at work (see Section 4). That is, because triangle, square and right-of serve similar semantic functions in the second sentence as did square, triangle, and above in the first sentence, it is assumed that they are generated by the same syntactic rules.

LAS's grammar after the first nine sentences is given in Figure 18. Sentences 3 and 4 cause another path to be created through the START network to handle sentences with relations like below and left-of which take the object noun phrase first. Sentences 5-9 are the program's first encounters with two word noun phrases. All five sentences involve the relations right-of and above and therefore result in the elaboration of the NP1 and NP2 sub-networks. Consider the first sentence with a two word noun phrase, square red triangle blue above,which is transformed by BRACKET to (C329 (C270 C271 square (C270 C272

Insert Figure 18 about here



RA = ABOVE,RIGHT-OF

N1 = SQUARE,TRIANGLE

N2 = SQUARE,TRIANGLE

RB = BELOW,LEFT-OF

N3 = SQUARE,TRIANGLE

N4 = SQUARE,TRIANGLE

A1 = SMALL,BLUE,LARGE,RED

A2 = SMALL,BLUE,LARGE,RED

Fig. 18. LAS's network grammar after studying the first 9 months in Table 4.

red)) (C303 C304 triangle (C303 C305 blue) above) C270). Consider the parsing of the first noun phrase. Note that the adjective (C270 C272 red) is embedded within the larger noun phrase. This is an example of the right embedding which BRACKET always imposes on a sentence. This will cause SPEAKTEST to create a push to an embedded network within its NP1 sub-network. As can be seen in Figure 18, the existing arc containing the N1 word class is kept to handle square. Two alternative arcs are added--one with a push to the ADJ1 network and the other with a NIL transition. Within the ADJ1 network the word class A1 is set up which initially contains the word red.

This illustrates the principle of left generalization in LAS.1: Suppose a network contains a sequence of arcs A_1, A_2, ..., A_m. Suppose further a phrase assigned to the network requires arcs X_1, ..., X_m, ..., X_n to be successfully parsed. If arcs A_1, A_2, ..., A_m have the same semantic functions as required of arcs X_1, X_2, ..., X_m, then the parsing of the first m elements in the phrase is assigned to the existing arcs A_1, ..., A_m. After arc A_m two alternate paths are built. A NIL arc is added to permit the phrases that used to be parsed by A_1, ..., A_m. Also arcs X_{m+1}, ..., X_n are added to handle the new phrase. LAS is making the generalization that any sequence of constituents parsable by A_1, ..., A_m can be placed in front of any sequence of elements parsable by X_{m+1}, ..., X_n.

Figure 19 shows a more conservative way that LAS might have made this generalization. Instead of network (a), it might have set up network (b). In network (b) a new work class X has been set up to record just those words which can be followed by an adjective. Networks (c) and (d) illustrate how left generalization can and does lead to overgeneralization in natural language. Suppose a child hears phrases like The boy, A dog, The foot, etc.

Insert Figure 19 about here

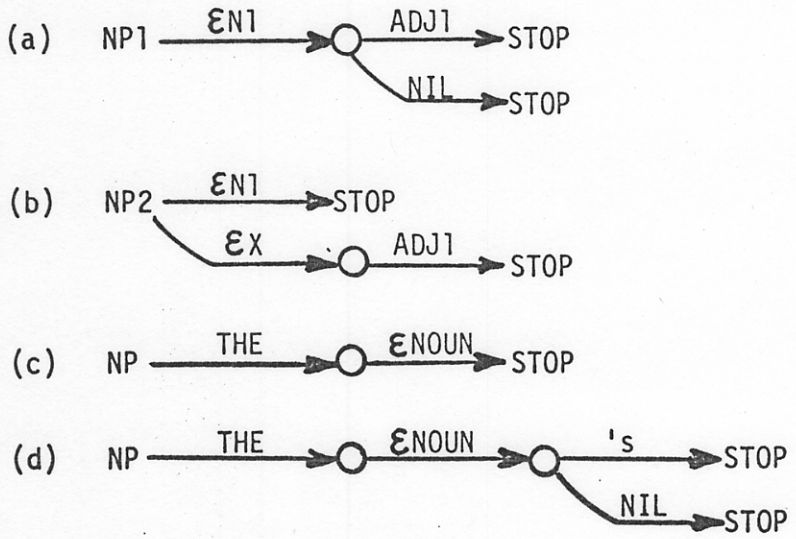


Fig. 19. Part (a) illustrates a structure created by left generalization; Part (b) illustrates a more conservative construction; Parts (c) and (d) illustrate a overgeneralization induced by left generalization.

He would set up a network that would accept any article followed by any noun. Suppose he then hears The boys. This would be represented in LAS as The + boy + 's. Because of left generalization LAS would construct the network illustrated in (d). In this network LAS has incorporated the generalization that foots is the pluralization of foot. This sort of morphemic overgeneralization is, of course, quite frequent in child language (e.g., Ervin, 1964). What is distinctive about such morphemic rules is that there are a number of alternatives and no semantic basis to choose between them. Because of left generalization, LAS will overgeneralize in those situations.

LAS's handling of the last 5 sentences in the sequence is rather uneventful, just resulting in more generalizations of the same kind. The language it is trying to learn has a finite number of sentences in it--1296. After the 14th sentence it has expanded its grammar to the point where it will handle 616 of these. Actually the grammar has produced some overgeneralizations--it will accept a total of 750 sentences. LAS has encountered phrases like square, square small, square red, and square red small. From this experience, LAS has generalized to the conclusion that the sentences of the language consist of a shape, followed optionally by either a size or color, followed optionally by a size. Thus the induced grammar includes phrases like square small small because size words were found to be acceptable in both second and third positions. Interestingly, this mistake will not cause LAS any problems. It will never speak a phrase like square small small because it will never have a to-be-spoken HAM structure with two small's modifying an object. It will never hear such a phrase and thus UNDERSTAND can not make any mistakes. This is a nice example of how an over-general grammar can be successfully constrained by consideration of semantic acceptability.

GENERALIZE

After taking in 14 sentences LAS has built up a partial network grammar that serves to generate many more sentences than those it originally encountered. However, note that LAS has constructed four copies of noun phrase grammar. One would like it to recognize that those grammars are the same. The failure to do so with respect to this simple artificial language only amounts to an inelegance. However, the identification of identical networks is critical to inducing languages with recursive rules.

A list is kept of all the networks created by SPEAKTEST. Once the structure of these networks becomes stable, GENERALIZE is called. It compares pairs of networks looking for those which are identical. The criterion for identification of two networks is that they have the same arc paths. Two arcs are considered identical if they have the same syntactic conditions and semantic actions. Consider what LAS would do if it had the following embedding of networks:

$NP \rightarrow$ the $NOUN_1$

$\rightarrow$ the ADJ_1 NP_1

$NP_1 \rightarrow NOUN_2$

$\rightarrow ADJ_2$ NP_2

$NP_2 \rightarrow NOUN_3$

$\rightarrow ADJ_3$ NP_3

$NP_3 \rightarrow NOUN_4$

That is, there are four networks, NP, NP_1, NP_2, and NP_3 whose structure are indicated by the above rewrite rules. (That is, the symbols in the right-hand portion of the rule indicates the sequence of word classes and pushes on arcs in different paths through the network.) It is assumed that LAS has only experienced three consecutive adjectives and therefore SPEAKTEST has only three

embeddings. The critical inductive step for LAS is to recognize $NP_1 = NP_2$. This requires recognizing the identity of the word classes $NOUN_2$ and $NOUN_3$ and the word classes ADJ_2 and ADJ_3. This will be done on the criterion of the amount of overlap of words in the two classes. It also requires recognition that network $NP_2 = NP_3$. Thus, to identify two networks may require that two other networks be identified. The network NP_3 is only a sub-network of NP_2. So in the recursive identification of networks, GENERALIZE will have to accept a sub-network relation between one network like NP_2 which contains another like NP_3. The assumption is that with sufficient experience the embedded network would become filled out to be the same as the embedding network. After NP_1 has been identified with NP_2, HAM will have a new network structure given below where NP* represents the amalgamation of NP_1, NP_2, and NP_3:

NP → the NOUN

→ the ADJ NP*

NP* → NOUN*

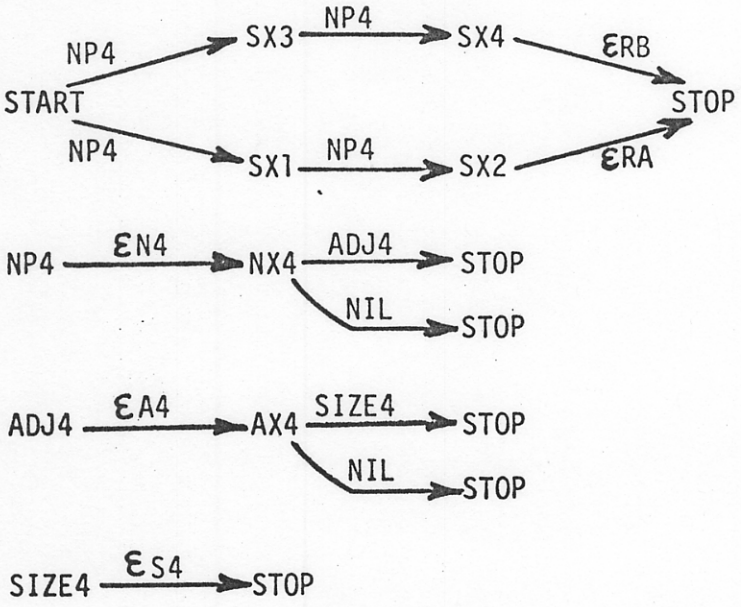
→ ADJ* NP*

Note that a new word class NOUN* has been created as the union of the word classes $NOUN_2$, $NOUN_3$, $NOUN_4$, and the word class ADJ is the union of the classes ADJ_2, ADJ_3.

GENERALIZE was called to ruminate over the networks generated after the first fourteen sentences. GENERALIZE succeeded in identifying NP1 with NP2. As a consequence, network NP1 replaced network NP2 at the position where it occurred in the START network (see figure 18). Similarly, NP4 was identified with and replaced network NP3. Finally, NP4 was identified with and replaced NP3 throughout the START network. The final effective grammar is illustrated in Figure 20. It now handles all the sentences of the grammar. It handles

---

Insert Figure 20 about here



RA = BELOW,LEFT-OF

RB = ABOVE,RIGHT-OF

N4 = SQUARE,TRIANGLE

A4 = BLUE,RED,LARGE,SMALL

S4 = LARGE,SMALL

Fig. 20. The final network for Table 4 after generalization of the noun phrase grammars.

more sentences than the grammar that was constructed after the fourteenth sentence. This is because the noun-phrase network NP4 has been expanded to incorporate all possible noun phrases. Before the generalizations, none of the networks--NP1, NP2, NP3, or NP4 were complete. The network NP4 became complete through merging with NP3 and NP1.

This concludes the discussion of the algorithms LAS.1 uses for language induction. In retrospect, LAS.1 had a lot of problems but it did serve to indicate that the heuristics I had in mind could serve to learn natural-like languages. In addition, I think there were three other significant developments. First the transition network formalism was interfaced with a set of simple and psychologically realistic memory operations. Second, a single grammatical formalism was created for generation and understanding. Thus, LAS.1 needed only to induce one set of grammatical rules. Third, two important ways were identified in which semantics helped grammar induction. These were stated as the GDC and the PSIES.

The PSIES as used here really included a number of generalization principles discussed in Section 4--Principle of Minimal Contrast, the PSIES principle as stated there, left-generalization and some aspects of noun-phrase generalization. As we will see, the ideas about noun phrase generalization in LAS.1 were not adequate. In addition to this problem, there was no attempt to generalize by category merging or by combining relational structures. Also there was no provision for lexicalization or for error recovery.

9. The Program LAS.2

The program LAS.2 had almost the same generation and understanding programs as LAS.1. In fact, except for a slight technical change in UNDERSTAND

(forced by a change in the induced networks), this part of the program is unchanged. However, major changes were made in the induction heuristics. These involved the treatment of noun-phrases and the between-network generalizations that were made by GENERALIZE in LAS.1.

Noun Phrases

Recall that BRACKET produced noun phrases parenthesized in a right-branching manner. Thus, the noun-phrase derived in Figure 20 had the following form:

NP4 → N4
→ N4 ADJ4
ADJ4 → A4
→ A4 SIZE4
SIZE4 → S4
N4 → square, triangle
A4 → blue, red, large, small
S4 → large, small

It quickly became apparent when trying to learn English that right-branching would not always do. Consider the following two phrases:

The red square
The square

LAS.1 would parenthesize these as (The red (square)) and (The square). This leads to the following grammar:

NP → DET X1
→ DET X1 X2
X2 → X3
X1 → red, square
X3 → square

With this grammar, LAS.1 proceeded to produce The red. LAS's problem was that it did not recognize that there was an obligatory noun class with optional modifiers. This knowledge had to be built into LAS.2. In particular, LAS.2 was told that, for the current semantic domain, shapes had the privileged status of being obligatory and all else could be regarded as an optional modifier. As discussed in Section 4, the exact concepts which constitute a noun class is not fixed, but depends on the pragmatics of the situation in which language is used. It is this pragmatic knowledge which is being given to LAS.2. This information was incorporated into BRACKET. In general, BRACKET tries to impose the following structure on noun phrases:

NP → morphemes (MOD) noun (MOD)
MOD → prop (MOD)
→ prop

That is, a noun-phrase consists of a possibly null string of morphemes, a bracketed pre-position modifier, an obligatory noun, and a bracketed post-position modifier. The pre- and post-position modifiers are sets of propositions about the object with which the noun-phrase is concerned. There can be from zero to arbitrarily-many such modifying propositions. As the rewrite rule for MOD shows, these modifying propositions are parenthesized within one another in a right-branching manner. As an example The tall blond man with one black shoe would be returned by BRACKET as (The (tall (blond)) man (with (one (black)) shoe ( ))).

Network Generalization--MERGE

In LAS.1 redundant networks were merged only after fairly stable networks had been built up. This was done by a process of rumination over the existing networks. This process did not seem particularly realistic as a psychological model. The human is a stimulus-driven device and cognitive processing does

not occur without some stimulus. Indeed, GENERALIZE was not really automatically initiated. It was invoked upon my command.

In LAS.2 it is assumed that all networks are simultaneously ready and able to process an incoming phrase. Thus, if there are two networks that can redundantly handle the same phrase, LAS should be able to quickly detect this. Thus, the stimulus for network merging in LAS is when it finds that two networks can handle the same phrase. It then attempts to construct a single network out of the two in the same manner as did GENERALIZE in LAS.1. The program that serves this function in LAS.2 is called MERGE. As was the case with GENERALIZE, in attempting to merge two networks, MERGE may be called upon to merge two other networks, two word classes, or to induce a recursive rule.

The remainder of this section will try to illustrate the consequences of the induction mechanisms incorporated into LAS.2. We will look at the induction of the natural languages to describe the same semantic referent as LAS.1--that is the world of shapes, colors, and sizes. One language will be English and the other French. After learning these two languages, we will observe LAS.2 translate between the two using the induced grammars.

The English Example

Table 5 gives the grammar that defines the language to be learned. It also shows the 11 sentences that were given to LAS in the learning sequence. One thing should be noted about the materials given to LAS. The terms right-of and left-of are hyphenated to create a single lexical item. If they were not, BRACKET would assign of as the first of the non-meaning-bearing morphemes in the noun-phrase. This is one example of the problem of phrase segmentation which is not well solved in LAS.2.

Insert Table 5 about here

TABLE 5

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

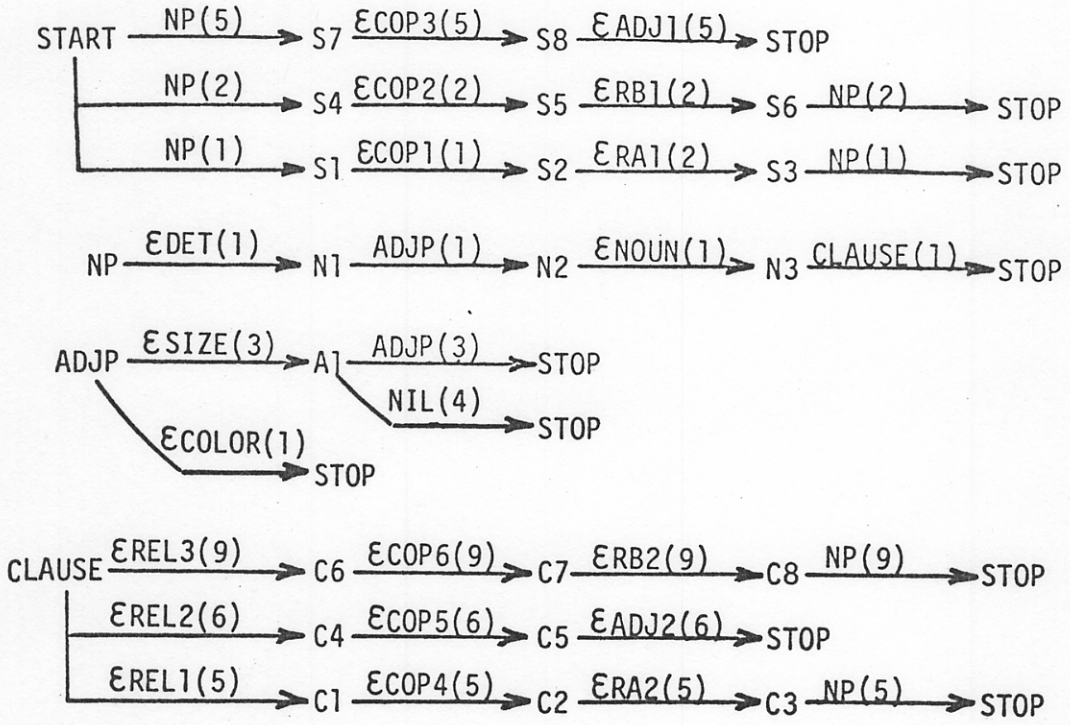
1. The red square is above the red circle
2. The square is below the circle
3. A large blue square is left-of the small red square
4. A small square is right-of a large square
5. The square which is above the red circle is red
6. The circle which is red is small
7. The circle which is right-of the circle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

After studying these 11 sentences, LAS.2 induced the grammar which is given in Figure 21. That grammar is divided into a transition network and categorical component where the latter gives the words in the categories found on network arcs. In attempting to indicate the course of growth of the grammar, I have put a number on parentheses with each arc in the network. This indicates after which sentence the arc was formed.

Insert Figure 21 about here

The first two sentences create paths in the START network for parsing sentences with relations like above and for sentences with relations like below. The noun phrase NP is set up after the first sentence to parse its subject. A different subject noun phrase was set up for the second sentence, but it was merged with NP. Note that NP involves a PUSH to an embedded network ADJP to analyze the pre-positional modifiers of the noun. This PUSH is optional and the simple noun phrase The square can be parsed by NP as well as The red square. Note also that in NP an optional push to CLAUSE has been created for post-positional modifiers. This has been created in anticipation of post-positional modifiers yet to be encountered. Thus, the structure of the noun phrase grammar is being strongly determined by information about noun phrases that LAS.2 has prior to any experience with this language.

Sentence (3) causes some important changes in the noun phrase grammar. Up until then, another noun-phrase network had been set up to parse the second noun in the sentence. The third sentence has a second noun phrase which is bracketed (the (small (red)) square nil).[1] MERGE detects that this can be analyzed by the network NP because in NP word class DET contains the; the push to ADJP will parse (small (red)); NOUN contains square; and the



COP1,COP2,COP3,COP4,COP5,COP6 = IS

ADJ1,ADJ2 = SMALL,LARGE,RED,BLUE

RA1,RA2 = ABOVE,RIGHT-OF

RB1,RB2 = BELOW,LEFT-OF

DET = A,THE

NOUN = SQUARE,CIRCLE

SIZE = LARGE,SMALL

COLOR = RED,BLUE

REL1,REL2,REL3 = WHICH

Fig. 21. The network induced by LAS.2 after studying the 11 English sentences in Table 5.

push to CLAUSE will parse NIL. On this basis MERGE combines the other noun-phrase grammar with NP.

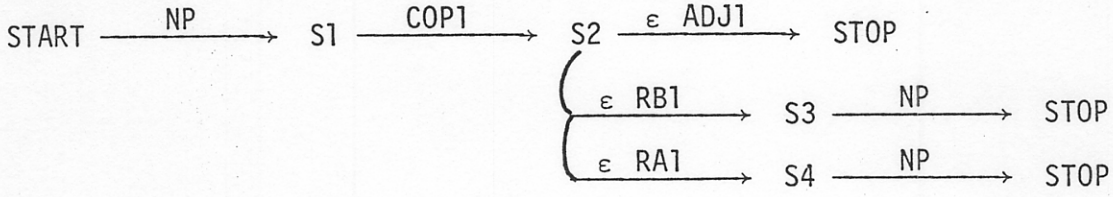
Sentence (3) contains a two-word adjective phrase. The first adjective phrase is (large (blue)). This causes the adjective-phrase network ADJP to become elaborated to involve a push to a sub-network to parse (blue). However, MERGE collapses this sub-network into ADJP because ADJP will parse red. Therefore ADJP is made to call itself. A recursive definition of adjective phrase has been constructed. Therefore, the language produced by the grammar has become infinite. Of course, many of the adjective phrases generated by this recursive rule would not be very interesting or even terribly grammatical. They would be of the form large large small large small blue where each recursive call of ADJP created another size except for the last call which created a color. However, as discussed with respect to LAS.1, this over-generality in the noun-phrase grammar is not a problem. Such phrases would be edited out by considerations of semantic acceptability.

Sentence (4) introduces the possibility of an adjective phrase which consists only of a size. This is encoded in adjective-phrase network ADJP by a NIL transition from A1. Sentence (5) introduces the first post-positional phrase, (which is above (the (red) circle nil)), which initiates the elaboration of the post-positional network CLAUSE. The path created for this sentence is the one that progresses from CLAUSE to C1 to C2 to C3 to STOP. MERGE recognizes the phrase (the (red) circle nil) as parsable by the existing noun-phrase network NP. Therefore, a push to NP is encoded on the path. Note another point of recursion has been created with NP calling CLAUSE which calls NP. This will permit unlimited right-embedding of relative clauses. Sentence (5) also causes the path from START to S7 to S8 to STOP to be built to

accomodate the new main clause which involves an adjective in the predicate.

The reader is invited to inspect for himself the effect of sentences (6) - (11) on the network grammar in Figure 21. They serve to fill in the remaining options for grammatical constructions and to flesh out the word classes to their full size.

The final network grammar in Figure 21 lacks a number of generalizations. It is useful to understand why these were not made. Note the redundancy within the START and CLAUSE networks. For instance in the START network all the sentences begin in the form NP is. Therefore it would seem more efficient if the different branches in the START network were merged to obtain a network of the form:



One of the reasons this generalization did not occur is that the first noun-phrase either identifies logical object or subject in the underlying construction dependent on the subsequent relation word. LAS currently assigns a semantic interpretation corresponding to a noun-phrase as soon as it parses it. Therefore, different branches are required so that the noun phrase may be assigned the different subject and object interpretation. A structure like the above would only be possible if semantic interpretation were delayed until the relational word.

Another point of potential generalization is that the network START for the main clause and the network CLAUSE for the relative clause have marked similarities which are not being capitalized upon. LAS will not merge two networks unless one is completely a subset of another. START begins with an

NP while CLAUSE has an initial element which. To detect such partial overlap, MERGE would have to note the similarity in the networks after the first element. In general, every time LAS tried to parse an element by one arc, it would have to consider whether other arcs could also parse the element.

Another potential generalization has to do with the fact that the same words that are occurring as predicate adjectives are occurring as prenominal adjectives. Yet different word classes have been set up for the words occurring in the two positions. There is also reduplication of the COP word class, the RA word class, and the RB word class. Merging of word classes would be an example of categorical merging discussed in Section 4. Both detection of common word classes and partial network overlap are projected goals for later versions of the LAS program.

Paraphrase Test

As a test of the grammar induced in Figure 21 I wrote a simple program PARAPHRASE. It received a sentence and passed it to UNDERSTAND to build up a conceptualization of it. Then it selected a different topic for the paraphrase sentence. SPEAK was then called with this new topic. A couple of examples of the paraphrases generated are given below:

Original: The square is left-of the circle
Paraphrase: A circle is right-of a square

Original: The large square which is above the small circle is red
Paraphrase: A circle which is below a large red square is small

The French Example

The second language LAS.2 learned was the French subset defined in Table 6. I cannot personally vouch for the correctness of the grammar in Table 6.

Insert Table 6 about here

TABLE 6

The French Subset to be Learned

Grammar

S	→ DET_s NP est Relation DET_o NP
S	→ DET_s NP est ADJ
NP	→ (Size) Shape (Color) (CLAUSE)
CLAUSE	→ qui est ADJ
	→ Relation DET_o NP
ADJ	→ Size
	→ Color
DET_s	→ le, un
DET_o	→ du, d'un
Relation	→ au-dessus, au-dessous, a-gauche, a-droit
Size	→ grand, petit
Color	→ bleu, rouge
Shape	→ square, circle

Sentences Studied

1. Le carre rouge est au-dessus du cercle rouge
2. Le grand carre est au-dessous du petit cercle qui est rouge
3. Le petit carre est a-droit du grand carre qui est bleu
4. Un cercle qui est grand est a-gauche d'un cercle rouge
5. Le carre est au-dessous d'un cercle qui est petit
6. Le carre rouge est au-dessous d'un cercle rouge qui est petit
7. Le cercle rouge a-gauche du cercle bleu est grand
8. Le carre a-droit du carre est bleu
9. Le cercle au-dessous du carre rouge est rouge
10. Le cercle au-dessus du carre est petit

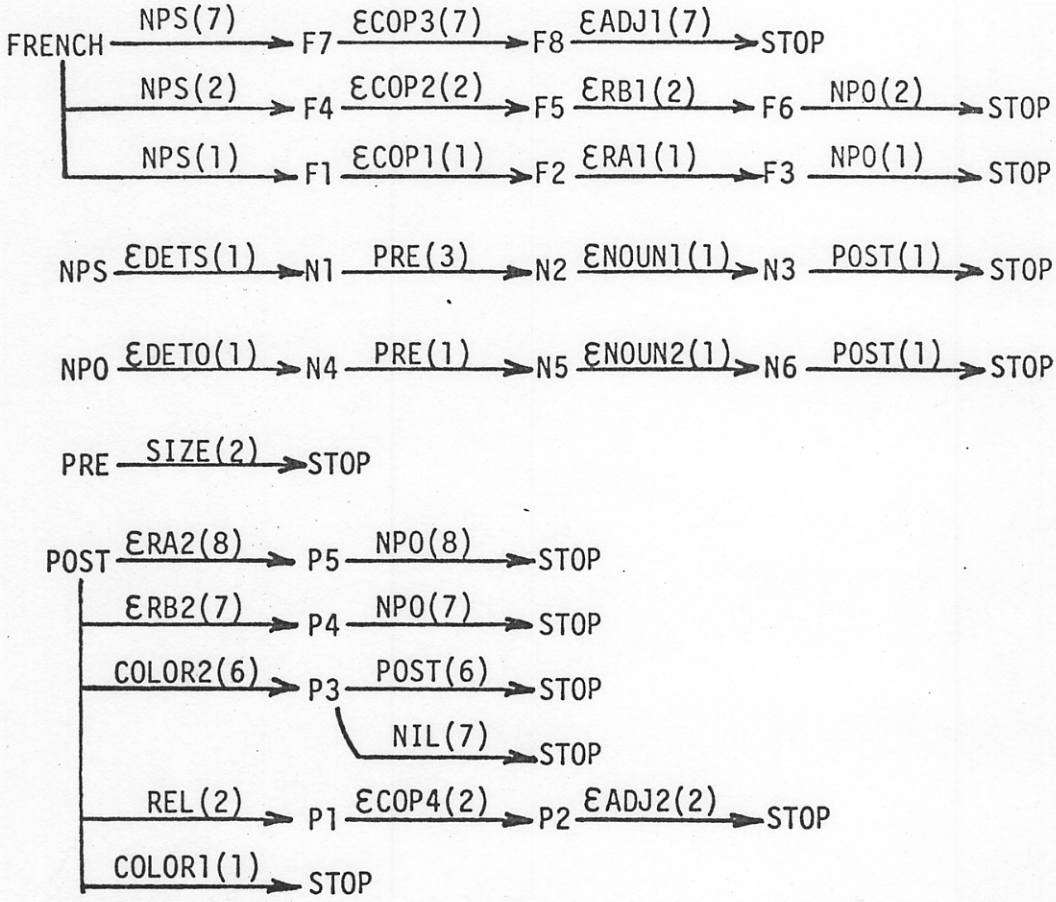
My French informant assured me that they were grammatical, but she giggled a lot at the sentences. The final grammar induced is given in Figure 22. It follows the same conventions used in Figure 21 where the parenthesized digit indicated the sentence which caused the formation of that rule. The induction history has little to reveal that was not already observed with respect to the English example. Therefore, I will not discuss it in detail.

Insert Figure 22 about here

Note that separate noun-phrase grammars are induced for subject, NPS, and for object NPO. These are not merged because they begin with different morphemes. The subject noun-phrases begin with le and un while the object noun-phrases begin with du and d'un. Actually, these object morphemes are contractions of the subject morphemes with the French de. However, LAS does not have the facility to detect these morphemic contractions. Nonetheless, an adequate, if less efficient grammar, is induced.

The Translation Test

LAS, having now learned the two languages is in a position to be able to translate between the two. Translation is possible because synonymous words in the two languages are connected to the same idea node. So, in the program TRANSLATE the French grammar might be used with the UNDERSTAND program to analyze a French sentence. UNDERSTAND creates a representation of the sentence in the HAM memory network. Then SPEAK is called with the English grammar to generate an English equivalent. Examples of translations from French to English and English to French are given below.



COP1,COP2,COP3,COP4 = EST
ADJ1,ADJ2 = ROUGE,BLEU,GRAND,PETIT
RB1,RB2 = A-GAUCHE,AU-DESSOUS
RA1,RA2 = A-DROIT,AU-DESSUS
DETS = LE,UN
DETO = DU,D'UN
NOUN1,NOUN2 = CARRE,CERCLE
SIZE = PETIT,GRAND
COLOR1,COLOR2 = BLEU,ROUGE
REL = QUI

Fig. 22. The network induced by LAS.2 after studying the 10 French sentences in Table 6.

1. Original: The red square is below the blue circle
   Translation: Un carre au-dessous d'un cercle bleu est rouge

2. Original: Le cercle rouge a-gauche du petit carre bleu est grand
   Translation: A red circle which is left-of a small blue square is large

3. Original: A square which is above a large blue square is below a small blue square
   Translation: Un carre au-dessus d'un grand carre bleu est au-dessous d'un petit carre bleu

Note that the translations are rather liberal. For instances, red is a modifier in (1) whereas rouge is a predicate. This is because SPEAK is only given the topic of the to-be-translated memory structure. It is not told which proposition is the main assertion and which propositions are subordinate.

One would hardly want to claim that these examples indicate that LAS has any immediate promise for language translation. The real problems in language translation arise when one must deal with much richer semantic domains. Nonetheless, I think LAS.2 does provide a very scaled-down model of how the learning approach might eventually be applicable to language translation.

References

Anderson, J. R. Computer simulation of a language-acquisition system. In R. L. Solso (Ed.), Information processing and cognition: The Loyola Symposium. Washington: Lawrence Erlbaum, 1975.

Anderson, J. R. and Bower, G. H. Human associative memory. Washington: Winston and Sons, 1973.

Anderson, J. R. and Paulson, R. Information-processing constraints and lexicalization, in preparation.

Bever, T. G. The cognitive bases for linguistic structures. In J. R. Hayes (Ed.), Cognition and the development of language. New York: Wiley, 1970.

Bever, T. G., Fodor, J. A., and Weksel, W. Theoretical notes on the acquisition of syntax: A critique of "contextual generalization." Psychological Review, 1965, 72, 487-482.

Biermann, A. W. An interactive finite-state language learner. First USA-JAPAN Computer Conference, 1972.

Blasdell, R. and Jensen, P. Stress and word position as determinants of imitation in first-language learners. Journal of Speech and Hearing Research, 1970, 13, 193-202.

Blount, B. G. Acquisition of language by two children. Ph.D. dissertation, University of California, Berkeley, 1969.

Bowerman, M. Structural relationships in children's utterances: Syntactic or semantic? In T. E. Moore (Ed.), Cognitive development and the acquisition of language. New York: Academic Press, 1973.

Braine, M. D. S. On learning grammatical order of words. Psychological Review, 1963, 70, 323-348. (b)

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, 153-188.

Broen, P. The verbal environment of the language-learning child. Monographs of the American Speech and Hearing Association, 1972, 17.

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

Brown, R. and Bellugi, U. Three processes in the child's acquisition of syntax. Harvard Educational Review, 1964, 34, 133-151.

Brown, R. and Fraser, C. The acquisition of syntax. In C. N. Cofer and B. S. Musgrave (Eds.), Verbal behavior and learning: Problems and processes. New York: McGraw-Hill, 1963, 158-197.

Cazden, C. B. Environmental assistance to the child's acquisition of grammar. Unpublished Ph.D. dissertation, Harvard University.

Chomsky, N. Syntactic structures. The Hague: Mouton, 1957.

Chomsky, N. Explanatory models in linguistics. In: Logic, methodology and philosophy of science: Proceedings of the 1960 International Congress, E. Nagel, P. Suppes, and A. Tarski (Eds.) Stanford: University Press, 1962.

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

Chomsky, N. The formal nature of language. Appendix A to E. H. Lenneberg (Ed.), Biological foundations of language. New York: Wiley, 1967, 397-442.

Chomsky, N. and Miller, G. A. Introduction to the formal analysis of natural languages. In R. D. Luce, R. R. Bush, and E. Galanter (Eds.), Handbook of mathematical psychology II. New York: Wiley, 1963.

Clark, E. V. Non-linguistic strategies and the acquisition of word meanings. Cognition: International Journal of Cognitive Psychology, 1974, in press.

Clark, E. V. First language acquisition. To appear in J. Morton and J. C. Marshall (Eds.), Psycholinguistics Series. London: Paul Elek (Scientific Books), 1975.

Crespi-Reghizzi, S. The mechanical acquisition of precedence grammars. Report No. UCLA-ENG-7054, School of Engineering and Applied Science, University of California at Los Angeles, June, 1970.

Drach, K. The language of the parent: A pilot study. Lanugage, society and the child. (Working Paper No. 14) Language-Behavior Research Laboratory, University of California, Berkeley, 1968.

Eimas, P. D., Siqueland, E. R., Jusczyk, P., and Vigorito, J. Speech perception in infants. Science, 1971, 171, 303-306.

Ervin, S. M. Imitation and structural change in children's language. In E. H. Lenneberg (Ed.), New directions in the study of language. Cambridge, Mass.: MIT Press, 1964, 163-189.

Farwell, C. B. The language spoken to children. Papers and Reports on Child Language Development, Stanford University, 1973, 5, 31-62.

Ferguson, C. A., Peizer, D. B., and Weeks, T. E. Model-and-replica phonological grammar of a child's first words. Lingua, 1973, 31, 35-65.

Fernald, C. Children's active and passive knowledge of syntax. Paper presented to the Midwestern Psychological Association, 1970.

Fraser, C., Bellugi, U., and Brown, R. Control of grammar in imitation, comprehension, and production. Journal of Verbal Learning and Verbal Behavior, 1963, 2, 121-135.

Gelman, R. and Shatz, M. Listener-dependent adjustments in the speech of four-year-olds. Paper presented at the Psychonomic Society Meeting, St. Louis, Missouri, 1972.

Gold, E. M. Limiting recursion. Journal of Symbolic Logic, 1965, 30, 28-48.

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

Grégoire, A. L'apprentissage du langage. Paris: Droz, 1949.

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

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

Jakobsan, R. Kindersprache, Aphasie und allegemeine Lautgesetze. (Uppsala, 1941) Translated: Child language, aphasia and phonological universals. The Hague: Mouton, 1968.

Kernan, K. T. The acquisition of language by Samoan children. Ph.D. dissertation, University of California, Berkeley, 1969.

Klein, S. Automatic inference of semantic deep structure rules in generative semantic grammars. Technical Report #180, Computer Sciences Department, University of Wisconsin, Madison, May, 1973.

Klein, S. and Kuppin, M. A. An interactive heuristic program for learning transformational grammars. Computer Studies in the Humanities and Verbal Behavior, 1970, 3, 144-162.

Labov, W. The study of language in its social context. Studium Generale, 1970, 23, 30-87.

Lenneberg, E. H. Understanding language without ability to speak: A case report. Journal of Abnormal and Social Psychology, 1962, 65, 419-425.

Lenneberg, E. H. Biological foundations of language. New York: Wiley, 1967.

McNeill, D. Developmental psycholinguistics. In F. Smith and G. A. Miller (Eds.), The genesis of language: A psycholinguistics approach. Cambridge, Mass.: MIT Press, 1966, 15-84.

McNeill, D. The acquisition of language. New York: Harper, 1970.

McNeill, D. Semiotic extension. In R. L. Solso (Ed.), Information processing and cognition: The Loyola Symposium. Washington: Lawrence Erlbaum: 1975.

Miller, W. R. and Ervin, S. M. The development of grammar in child language. In U. Bellugi and R. Brown (Eds.), The acquisition of language. Monographs of the Society for Research in Child Development, 1964, 29, 9-33.

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

Moffitt, A. R. Speech perception by infants. Unpublished Ph.D. dissertation, University of Minnesota, 1968.

Moffitt, A. R. Consonant cue perception by twenty- to twenty-four week old infants. Child Development, 1971, 42, 717-731.

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

Pao, T. W. L. A solution of the syntactic induction-inference problem for a non-trivial subset context-free language. Report No. 70-19, The Moore School of Electrical Engineering, University of Pennsylvania, August, 1969.

Sachs, J. S., Brown, R., and Salerno, R. A. Adults' speech to children. Paper presented at the International Symposium on First Language Acquisition, Florence, Italy, 1972.

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

Scholes, R. J. The role of grammaticality in the imitation of word strings by children and adults. Journal of Verbal Learning and Verbal Behavior, 1969, 8, 225-228.

Scholes, R. J. On functors and contentives in children's imitations of word strings. Journal of Verbal Learning and Verbal Behavior, 1970, 9, 167-170.

Shipley, E. F., Smith, C. S., and Gleitman, L. R. A study in the acquisition of language: Free responses to commands. Language, 1969, 45, 322-342.

Siklóssy, L. A language-learning heuristic program. Cognitive Psychology, 1971, 2, 479-495.

Siklóssy, L. Natural language learning by computer. In H. A. Simon and L. Siklóssy (Eds.), Representation and meaning: Experiments with information processing systems. Englewood Cliffs, N. J.: Prentice-Hall, 1972.

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

Slobin, D. I. The acquisition of Russian as a native language. In F. Smith and G. Miller (Eds.), The genesis of language. Cambridge, Mass.: MIT Press, 1966.

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 and D. I. Slobin (Eds.), Studies of child language development. New York: Holt, Rinehart, and Winston, 1973, 175-208.

Snow, C. E. Mother's speech to children learning language. Child Development, 1972, 43, 549-565.

Solomonoff, R. J. The mechanization of linguistic learning. In Proceedings of the Second International Congress of Cybernetics, Belguim, 1958, 180-193.

Solomonoff, R. J. A formal theory of inductive inference. Part I. Information and Control, 1964, 7, 1-22.

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.

Yasahara, A. Recursive function theory and logic. New York: Academic Press, 1971.

Footnote

1. This research is supported by grant GB-40298 from NSF to myself. I express my deep gratitude to Clayton Lewis for the help he has given me with this work--suggesting ideas, criticizing my own, and pointing out ways to improve the exposition.

RECENT TECHNICAL REPORTS

40. Malin, J. T. An analysis of strategies for solving certain substitution problems. April 1973.

41. Robinson, J. K. Storage and retrieval processes under conditions of proactive interference. April 1973.

42. Mayer, R. E. Acquisition and resilience under test stress of structurally different problem solving procedures. May 1973.

43. Jagacinski, R. J. Describing multidimensional stimuli by the method of adjustment. June 1973.

44. Egan, D. & Greeno, J. G. Theory of rule induction: Knowledge acquired in concept learning, serial pattern learning, and problem solving. August 1973.

45. Pachella, R. G. The interpretation of reaction time in information processing research. November 1973.

46. Rose, A. M. Human information processing: An assessment and research battery. January 1974.

47. Polzella, D. J. The effect of sleep-deprivation on short-term memory. April 1974.

48. Pew, R. W. Human perceptual-motor performance. August 1974.

49. Glenberg, A. Retrieval factors and the lag effect. August 1974.

50. Eft, D. R. The buildup and release of proactive interference. August 1974.

51. Wickens, C. D. The effects of time sharing on the performance of information processing tasks: A feedback control analysis. August 1974.

52. Kieras, D. Analysis of the effects of word properties and limited reading time in a sentence comprehension and verification task. August 1974.

53. Whitten, W. B. Retrieval depth and retrieval component processes: A levels-of-processing interpretation of learning during retrieval. August 1974.

54. Hayes-Roth, B. Interactions in the acquisition and utilization of structured knowledge. August 1974.

RECENT MEMORANDUM REPORTS

14. Halff, H. M. The differential effects of stimulus presentation during error- and success-feedback intervals in concept identification. August 1971.

15. Mayer, R. E. Dimensions of learning to solve problems. February 1972.

16. Martin, E. Serial learning theory. February 1973.

17. Pachella, R. G. The effect of set on the tachistoscopic recognition of pictures. October 1973.