Proposal to use the SUMEX-AIM

Resource for Computer Simulation of Language Acquisition

John R. Anderson

Human Performance Center

University of Michigan

Ann Arbor, Michigan

The purpose of this research is to understand language acquisition. There has been a great deal of research on first language acquisition in children, second language learning by adults, and learning of artificial languages by laboratory subjects. The principle goal of this research is not getting more experimental evidence. Rather it is to develop a working computer simulation model that can learn natural languages. The model would attempt to explain the already available set of experimental facts. It is also hoped that such a model would be a contribution to the artificial intelligence goal of developing language understanding systems.

Some of the detailed plans of the research are described in the accompanying grant proposal that was awarded by NIMH (grant number 1 RO 1 MH26383-01). The period of this award is May 1, 1975 to May 1, 1977. That proposal states an intention to use Augmented Transition Networks as the basic grammatical formalism. I have already completed some initial learning programs using the augmented transition network formalism. The very earliest of this work is described in the NIMH proposal. More recently I have decided to try to develop a production system formalism as an alternate to the augmented transition network. There are three main reasons for this switch in representational formalism. First, I think it is easier to represent the grammatical knowledge contained in highly inflected languages (eg., Finnish, Latin) by production systems rather than augmented transition networks. Second, I think it is easier to represent human information processing limitations in terms of production systems. Third, I think production systems serve as a means of representing non-linguistic procedures such as inference-making. Therefore, a theory of induction of production systems for language has the promise of generalizing to the induction of other human cognitive skills.

I have been using the SUMEX facility in a pilot project this summer. I have been bringing up a version of my production system called ACT on this facility. It is hoped that in a few months this program will be in a sufficiently developed form that other SUMEX users may use that production system. It uses an associative network representation as its basic data base. This is a variant of the HAM propositional network that I developed earlier and is described in the accompanying proposal (p. 23 - 27). In the ACT system various portions of the network are active at any point of time. The productions look for patterns of activation in the network. If these patterns exist, the productions are executed causing external actions to be taken, building network structure, and possibly changing the state of activation of the network, Activation spreads associatively through the network and there is also a dampening process which deactivates network structure. A preliminary description of the ACT system is given in the accompanying document "An Overview of ACT." It is a chapter from a forthcoming book. The most relevant section in that chapter is from pages 11 to 25.

It was originally projected that this simulation work would be performed on the Michigan Computer System. However, there are a number of advantages of the SUMEX-AIM facility. All the programming will occur in LISP. The INTERLISP system in SUMEX, as surmised from my own experientation, permits programming and debugging to progress at least twice as fast as with Michigan LISP. Also programs in INTERLISP would be more available to other A.I. users than programs in Michigan LISP. The Michigan computer is isolated from the national A.I. community whereas I can take advantage of the connections SOMEX-AIM has through the TYMNET and the ARPANET. Finally, the SUMEX-AIM facility provides free computing resources and so will relieve some of the strain from my tight research budget.

It is intended that there will be continued development and testing of this production system formalism as a model of human information processing. There are plans to build substantial ACT production system models for language generation and understanding and for inference making.

Responses to SUMEX-AIM Questionnaire

A.1. Read the accompanying proposal.

A.2. The research is currently supported by a grant from NIMH (grant number 1 RO 1 MH 26383-01) for the period May 1, 1975 to May 1, 1977. The amount of the award for the first year is $20,000. This is to pay for a programmer, computer time, and rental of a terminal.

A.3. Read the accompanying proposal.

B.1. It is expected that this research will have some general contribution to make to development of language understanding systems, modeling human cognitive processes, and development of production systems.

B.2. None

B.3. There should be no difficulty in making my programs generally available to users of SUMEX-AIM.

B.4. Yes

B.5. Yes

C.1. Read next to last paragraph in accompanying proposal.

C.2. The INTERLISP language on SUMEX is the principle requirement of my research. I do not anticipate requiring any additional systems programs not already available at SUMEX.

C.3. Estimated requirements per month:

100 connect hours

2 CPU hours

1500 file pages

what time? PST?

The principle times of use in Ann Arbor would probably be 0600-0900 and 1800-2100

C.4. I intend to communicate with SUMEX via the TYMNET. I would either use the private node in Ann Arbor or the public node in Detroit. The toll cost to Detroit could be met from my current grant as could the cost of terminal rental.

C.5. Not really relevant

BIOGRAPHICAL SKETCH

(Give the following information for all professional personnel listed on page 3, beginning with the Principal Investigator. Use continuation pages and follow the same general format for each person.)

NAME	TITLE	BIRTHDATE (Mo., Day, Yr.)
John R. Anderson	Junior Fellow	Aug. 27, 1947

PLACE OF BIRTH (City, State, Country)	PRESENT NATIONALITY (If non-U.S. citizen, indicate kind of visa and expiration date)	SEX
Vancouver, B.C., Canada	Canadian - J1 - Aug. 15, 1974 *	☒ Male ☐ Female

EDUCATION (Begin with baccalaureate training and include postdoctoral)

INSTITUTION AND LOCATION	DEGREE	YEAR CONFERRED	SCIENTIFIC FIELD
University of British Columbia Vancouver, Canada	B.A.	1968	Psychology
Stanford University Stanford, California	Ph.D.	1972	Psychology

HONORS

1968--The Governor-General's Gold Medal (Head of graduating classes in Arts and Sciences, University of British Columbia)

MAJOR RESEARCH INTEREST	ROLE IN PROPOSED PROJECT
Language & Human Memory	Principal Investigator

RESEARCH SUPPORT (See instructions)

NSF - Recognition Memory for sentences: a process madel

Sept. 1, 1973 - Sept. 1, 1975 - $40,000

$20,265 for year 1

50% of research effort

grant number - GB-40298

RESEARCH AND/OR PROFESSIONAL EXPERIENCE (Starting with present position, list training and experience relevant to area of project. List all or most representative publications. Do not exceed 3 pages for each individual.)

Research and Professional Experience:

Junior Fellow, University of Michigan, 1973 - present

Assistant Professor, Yale University, 1972 - 1973

Numerous experiments in graduate school in human memory under the supervision of Gordon H. Bower at Stanford University, 1968 - 1972

Publications:

Reber, A. S. and Anderson, J. R. The perception of clicks in linguistic and non-linguistic messages. Perception & Psychophysics, 1970, 8, 81-89.

Anderson, J. R. and Bower, G. H. On an associative trace for sentence memory. Journal of Verbal Learning and Verbal Behavior, 1971, 10, 673-680.

Anderson, J. R. FRAN: A simulation model of free recall. In G. H. Bower (Ed.), The Psychology of Learning and Motivation, Vol. 5. New York: Academic Press, 197?

Anderson, J. R. and Bower, G. H. Recognition and retrieval processes in free recall. Psychological Review, 1972, 79, 97-123.

U. S. GOVERNMENT PRINTING OFFICE : 1971 O - 451-723

Anderson, J. R. A stochastic model of sentence memory. Doctoral dissertation, Stanford University, June, 1972.

Anderson, J. R. and Bower, G. H. Configural properties in sentence memory. Journal of Verbal Learning and Verbal Behavior, 1972, 11, 594-605.

Anderson, J. R. and Bower, G. H. Human Associative Memory. Washington: Winston and Sons, 1973.

Reder, L. M., Anderson, J. R., & Bjork, R. A. A semantic interpretation of encoding specificity. Journal of Experimental Psychology, 1974, 4, 648-656

Anderson, J. R. Verbatim and propositional representation of sentences in immediate and long-term memory. Journal of Verbal Learning and Verbal Behavior, in press.

Anderson, J. R. and Bower, G. H. A propositional theory of recognition memory. Memory & Cognition, in press.

Anderson, J. R. and Bower, G. H. Interference in memory for multiple contexts. Memory & Cognition, in press.

Anderson, J. R. Retrieval of propositional information from long-term memory. Cognitive Psychology, in press.

Anderson. J. R. and Hastie. R. Individuation and reference in memory: proper names and definite descriptions. Cognitive Psychology, in press.

Anderson, J. R. Computer simulation of a language-acquisition system, first report. In R. L. Solso (Ed.) Information Processing and Cognition: The Loyola Symposium, in press.

Anderson, J. R. Language acquisition by computer and child. To appear in: S. Y. Sedelow & W. A. Sedelow (Eds.), Current Trends in Computer Use for Language Research, in preparation.

* Special Note

I am in the second year of an exchange visitor's visa. I can renew the visa for another year. My wife, an American citizen, is currently petitioning to have my status changed to that of a permanent resident. Therefore, I will be able to be at the University of Michigan for the entire period of the proposed research.

COMPUTER SIMULATION OF LANGUAGE ACQUISITION

A. Introduction

1. Direction and goals of the research

Most simply stated, the purpose of this research is to understand language acquisition. There has been a great deal of research on first language acquisition in children, second language learning by adults, and learning of artificial languages by laboratory subjects. This research is not principally concerned with getting more experimental evidence. Rather it is concerned with developing an information-processing model that can be used to explain the already available set of experimental facts. One of the principal concerns governing the design of this model is just that it be able to learn a natural language. I will show that this, in itself, is a very significant goal.

It turns out that algorithms adequate to learn a natural language are quite complex. It is not possible to sit down and simply specify them verbally or with a set of equations. This research makes use of the computer as a tool to develop and test complex models. Therefore, I have been developing a computer simulation model of language acquisition. This model is called LAS (an acronym for Language Acquisition System). Most of the proposed budget is concerned with supporting the development of this program. Input to LAS consists of sentences of the language paired with representations of their meaning. Therefore, it simulates language learning in situations where a learner can figure out the meaning of the sentence from context. The simplest case of such a situation would be one in which the learner is presented with simple pictures and sentences describing them. The program constructs a grammar which allows it to go from sentences to representations of their underlying meaning. The grammar can also be used to generate sentences to convey meanings. It is also hoped that this program will make a contribution to the evolution of computer language understanding systems. Thus, the research really has two purposes, one in psychology and one in artificial intelligence.

I became interested in language acquisition as a consequence of my work with a computer simulation model of human memory. This program is described in a book by myself and Gordon Bower entitled Human Associative Memory. The computer program was an attempt to simulate simple question-answering. The principal purpose of that research was to develop a model of the human fact-retrieval system (called HAM) and test it in a series of experiments. A version of HAM is used within LAS. HAM's system included a simple language understander which was capable of dealing with a restricted but considerable subset of English and which was capable of using memory to disambiguate and to resolve reference. Nevertheless, it was relatively primitive in its capabilities compared to the work of Schank (1973); Winograd (1972); or Woods (1970). As a result of my own experiences and studying the more sophisticated systems, I became pessimistic about the value of representing human language understanding in terms of a computer program. To represent the unbounded linguistic competence of the human would seem to require almost unlimited reams of computer program. Rather, I decided that the only compact way to characterize the linguistic competence of the human was to characterize the language acquisition system that generated the competence.

Outline of Proposal

The concern in this proposal will be primarily with developing a system logically adequate for language acquisition and only secondarily with a system that simulated actual human performance. I do not think the latter is a realistic goal until we have a characterization of the sort of algorithms that are adequate for natural language acquisition. This emphasis on logical adequacy is clear in the organization of the proposal. I will first review the work that has been done on computer language understanding. This is important because LAS is a language understander as well as a learner. Then I will review the formal results on grammar induction. Then LAS·1 will be described. LAS.1 is a first pass version of the LAS program adequate to learn simple languages. Then I will propose an extensive set of developments to be added to the program, aimed both at increasing its linguistic powers and making it a realistic simulation. In describing LAS·1 and the proposed extensions, I will review relevant research in the child language literature. Finally, I will propose a series of experiments with artificial languages to check specific claims LAS makes about language learnability.

2. Computer Language Understanding

Computers have been applied to natural language processing for 25 years. There has been a succession of major reconceptualizations of the problem of language understanding, each of which constitutes a clear advance over the previous conceptions. However, any realistic assessment would concede that we are very far from a general language understanding system of human capability. The argument has been advanced that there are fundamental obstacles that will prevent this goal from ever being realized (Dreyfus, 1972). These arguments are shamefully imprecise and lacking in rigor. The best (e.g., Bar-Hillel, 1962) has to do with the extreme open-endedness of language, that an effectively unbounded variety of knowledge is relevant to the understanding process. It is boldly asserted, without proof, that it is not possible to provide the computer with the requisite background knowledge.

In reviewing the work on natural language systems, I will constantly measure them with respect to the goal of general language understanding. I appreciate that it is a legitimate artificial intelligence goal to develop a language system for some special purpose application. Such attempts are free from the Dreyfus and Bar-Hillel criticisms. However, from any psychological point of view these systems are interesting only as they advance our understanding of how language is understood in general.

Machine Translation

The first intensive application of computers to language was concerned with translation. Compared to the initial projections of success, this massive effort turned out to be a dismal failure (ALPAC, 1966; Bar-Hillel, 1964; Kasher, 1966). Today, it is fashionable to attribute the failure to the then-current impoverished conception of language (e.g., Simmons, 1970; Wilks, 1973). The early attempts took the form of substitution of equivalent words across languages. This was augmented by use of surface structure and word associations but at no point was the word abandoned as the principal unit of meaning. Recent work on language understanding (e.g., Schank, 1972; Winograd, 1973) has abandoned the word as the unit of meaning. It remains to be seen whether current attempts (e.g., Wilks, 1973) at machine translation have better success.

Interactive Systems

The now popular task domain for applications of computers to language is in constructing systems that can interact with the user in his own language. Question-answering systems are the most common; the user can interrogate the program about knowledge in its data base and input new knowledge. Such systems depend critically for their success on three aspects of their design--their parser, the representation of information, and the inference system. The task of the parser is to analyze natural language input and translate it into a form compatible with the internal representation. If the input is something to be remembered by the system, it will be translated into an internal representation and stored in that form. If the input is a question, it will be used to guide an interrogation of the data base for the answer. The inference system is critical in the answering of questions since many answers will not be directly stored but will have to be inferred from what is in memory. Both parsing and inferencing run into <u>time</u> problems.

The central time problem in parsing has to do with the extreme syntactic and lexical ambiguity of natural language. Each word in a sentence admits of $\underline{m}$ syntactic and semantic interpretations where $\underline{m}$ on the average may be as high as 10. If there are $\underline{n}$ words, $\underline{m}^{\underline{n}}$ interpretations must be considered although only one is intended. The fact that language is so ambiguous was a surprising discovery of the early machine attempts at parsing (e.g., Kuno, 1965). Thus, there is exponential growth in processing time with sentence length. To date, no heuristics have been demonstrated that change in general this exponential function of sentence length to something closer to a linear function. The human can use general context to reduce ambiguity to something approximating the linear relation.

There is also an exponential growth factor in the task of inference making. Suppose there are $\underline{m}$ facts in the data base and the desired deduction is $\underline{n}$ steps long. Then, there is something like $\underline{m}^{\underline{n}}$ possible combinations of facts to achieve the desired deduction. This suggests that very deep inferencing (i.e., high $\underline{n}$) is difficult to achieve and this is certainly true of our every-day reasoning. However, it also suggests that inference making should become more difficult as we know more facts (i.e., high $\underline{m}$) which is clearly not the case. The problem facing inference systems is to select only those facts that are relevant.

Resolution theorem-proving (Robinson, 1965) is the most studied of the mechanical inference systems. It is also here that the most careful work has been done on heuristics for selecting facts from the data base. These methods include semantic resolution (Slagle, 1965), lock resolution (Boyer, 1971), and linear resolution (Loveland, 1970; and Luckham, 1970). In practical applications these heuristics have served to considerably reduce the growth in computation time. However, the demonstrations of the optimality of these heuristics are task-specific. There are no general theorems about their optimality. I suspect that they do not in general deal effectively with the problems of exponential growth.

Although there are potentially serious time problems both in parsing and inferencing, these problems have not surfaced in the past programs as one might have expected. This is because these programs have all been rather narrowly constrained. Their language systems only need to deal with a small portion of possible syntactic constructions and possible word meanings. Also, because of restrictions in the domain of discourse, only a restricted set of inferences are needed.

Some of the interactive systems (ELIZA - Weizenbaum, 1966; PERRY - Colby & Enea, 1968) made no serious effort to do a complete job of sentence analysis. Only sufficient analysis was performed to permit success in narrowly circumscribed task domains. Sentences were generated by filling in pre-programmed frames with variable words. The ambition in programs like Colby's or Weisenbaum's was to create the appearance of understanding. Weisenbaum's program characterized a strict Rogerian psychotherapist and Colby's a paranoid patient. When the programs made serious errors of language understanding it was difficult for a naive user to reject the possibility that these might just be manifestations of the strong personalities of the simulations.

Other attempts made more serious efforts at language understanding. They avoided the time problems inherent in parsing and inferencing by dealing with restricted task domains. Slagle's DEDUCOM (1965) dealt with simple set inclusion problems; Green, Wolf, Chomsky & Laughery (1963) with baseball questions; Lindsay (1963) with kinship terms; Kellogg (1968) with data management systems; Woods (1968) with airline schedules; Woods (1973) with lunar geology; Bobrow (1964) and Charniak (1969) with word arithmetic problems; Fikes, Hart & Nilsson (1972) with a robot world; Winograd (1973) with a blocks world. Other systems like Green and Raphael (1968), Coles (1969), Schank (1972), Schwarcz, Berger, and Simmons (1969), Anderson and Bower (1973), Rumelhart, Lindsay and Norman (1972), and Quillian (1969) have not been especially designed for specific task domains but nonetheless succeed only because they worked with seriously limited data bases and restricted classes of English input. Because the parser deals with only certain word senses and certain syntactic structures linguistic ambiguity is much reduced. Those programs that use general inference procedures like resolution theorem proving are notably inefficient even with restricted data bases. Winograd made extensive use of the facilities in PLANNER for directing inferencing with specific heuristic information. The validity of these heuristics depended critically on the constraints in the task domain.

Winograd (1973) has combined good task analyses, programming skill, and the powers of advanced programming languages to create the best extant language understanding system. I have heard it seriously claimed that the Winograd system could be extended to become a general model of language understanding. What is needed would be to program in all the knowledge of an adult and extend the parsing rules to the point where they handled all English sentences. Admittedly, this would be a big task requiring hundreds of man-years of work, but, it is argued, no greater than the work that goes into writing big operating systems. Clearly, this argument is faulty if only because it does not deal with the time problems in general inferencing and general parsing. However, it is also unclear whether human language understanding can be captured in a fixed program. Further, it is dubious whether it is manageable to do the bookkeeping that is necessary to assure that all the specific pieces of knowledge are properly integrated and interact in the intended ways. Our linguistic competence is not a fixed object. This is clear over the period of years as we learn new grammatical styles, new words, and new ways of thinking. I think this is also true over short spans of time. That is, the way humans deal with the time problems inherent in parsing and inferencing is to adjust the parsing and inferencing according to context.

Language Acquisition as the Road to General Language Understanding

The preceding remarks were meant to suggest how an adaptive language system might provide the solution to the fundamental problems in general language understanding. Rather than defining and hand-programming all the requisite knowledge, why not let the language understanding system discover that knowledge and program itself? The language acquisition system is a mechanized bookkeeping system for integrating all the knowledge required for language understanding. By its very nature it treats linguistic knowledge as a constantly changing object. So we know it would change with a changing linguistic community. We might hope that it could adapt over short periods (like hours) to its current context.

Learning systems are frequently regarded as the universal panacea for all that ails artificial intelligence. Therefore, one should be rightfully suspicious whether LAS will provide a viable route to the creation of a general language understanding system. Certainly, the initial version of LAS falls far short of the desired goal. However, with our current state of knowledge it is just not possible to evaluate LAS's pretensions as an eventual language understanding system. It is only by systematic exploration and development of LAS that we ever will be able to determine the viability of the learning approach.

Whatever the potential of the learning approach in artificial intelligence, clearly it is the only viable psychological means of characterizing human linguistic knowledge. It would be senseless to provide a catalog of all the knowledge used in language understanding. <u>A catalog of everything is a science of nothing</u> (a quote from T. Bever). Rather, we must characterize the mechanism that creates that knowledge and how that mechanism interacts with experience.

Woods' System

The linguistic formalisms used by LAS are very similar to Woods' (1970) augmented transition networks. This section on computer language understanding concludes with a description of Woods' system and an exposition of the suitability of his formalisms for the current project. There are three critical features that LAS requires of the formalisms that will express its grammatical knowledge. First, it should be a formalism that can be used with equal facility for language parsing and language generation. This is because it is unreasonable to assume that a child independently learns how to speak and how to understand. Second, we want a formalism for which it is easy to devise a constructive algorithm for inducing grammar. That is to say, some descriptions of grammatical knowledge are computationally easier to induce than others, even though the two formalisms may be equivalent with respect to the language they describe. Third, we want the formalism to be closed with respect to the assumptions it makes about the interpretative system that uses the grammar for speaking and understanding. This is because that interpretative system is taken as innate. Thus, it is not possible to induce new programs for interpreting the grammatical rules, it is only possible to induce new grammatical rules.

A guiding consideration in this research is that these disiderata for a grammatical formulation are satisfied by a finite-state transition network representation. The problem is that natural languages are fundementally more complex than finite state languages. However, Woods has shown a way to keep some of the advantages of the finite state representation, but achieve the power of a transformational grammar. Woods' augmented transition networks are similar to and were suggested by the network grammars of Thorne, Bratley, and Dewar (1968) and Bobrow and Fraser (1970). Transition networks are like finite state grammars except that one permits as labels on arcs not only terminal symbols but also names of other networks. Determination of whether the arc should be taken is evaluated by a subroutine call to another network. This sub-network will analyze a sub-phrase of the linguistic string being analyzed by the network that called it. The recursive, context-free aspect of language is captured by one network's ability to call another. Figure 1 provides an example network taken from Woods' (1970) paper. The first network in Figure 1 provides the "mainline" network for analyzing simple sentences. From this mainline network it is possible to call recursively the second network for analysis of noun phrases or the third network for the analysis of prepositional phrases. Wood (1970) describes how the network would recognize an illustrative sentence:

> To recognize the sentence "Did the red barn collapse?" the network is started in state S. The first transition is the aux transition to state q_2 permitted by the auxiliary "did." From state q_2 we see that we can get to state q_3 if the next "thing" in the input string is an NP. To ascertain if this is the case, we call the state NP. From state NP we can follow the arc labeled det to state q_6 because of the determiner "the." From here, the adjective "red" causes a loop which returns to state q_6, and the subsequent noun "barn" causes a transition to state q_7. Since state q_7 is a final state, it is possible to "pop up" from the NP computation and continue the computation of the top level S beginning in state q_3 which is at the end of the NP arc. From q_3 the verb "collapse" permits a transition to the state

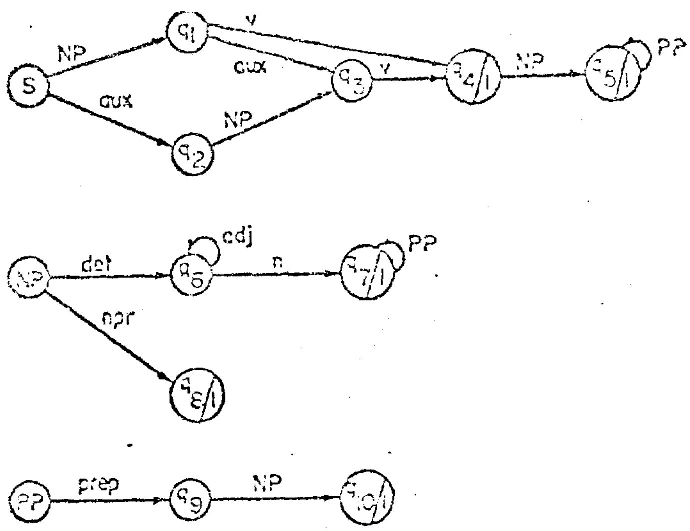


FIG. 1. A sample transition network. S is the start state. q_4, q_5, q_6, q_7, q_8, and q_{10} are the final states. (From Woods, 1970.)

q_4, and since this state is final and "collapse" is the last word in the string, the string is accepted as a sentence (pp. 591-592).

I have illustrated in Figure 1 what is known as a recursive transition network which is equivalent to a context-free phrase-structure grammar. Woods' networks are in fact of much stronger computational power - essentially that of a Turing Machine. This is because Woods permits arbitrary actions. This gives the networks the ability of transformational grammars to permute, copy, and delete fragments of a sentence. Thus, with his network formalisms Woods can derive the deep structure of a sentence. The problem with this grammatical representation is that it is too powerful and permits computation of many things that are not part of a speaker's grammatical competence. In the LAS system all conditions and actions on network arcs are taken from a small repertoire of operations possible in the HAM memory system (see Anderson & Bower, 1973). This way some context-sensitive features can be introduced into the language without introducing psychologically unrealistic powers.

In many ways the network formalisms of Woods are isomorphic in their power and behavior to the program grammars of Winograd. However, there is one critical difference: The flow of control is contained in Winograd's program grammars. That is, a particular program is committed to a certain behavior. This is not the case in the network formalism. The flow of control is contained in an interpreter which uses the grammatical knowledge contained in the networks. Thus by writing different interpretative systems the same network grammar specification can be used in different ways. This is critical to LAS's success where three different interpreters use the same grammatical formalisms to guide understanding, generation, and language induction.

3. Research on Grammar Induction

Apparently the modern work on the problem of grammar induction began with the collaboration of N. Chomsky and G. Miller in 1959 (see Miller, 1967). There have been significant formal results obtained in this field and it is essential that we review this research before considering LAS. The approach taken in this field is well characterized by the opening remarks of a recent highly-articulate review chapter by Biermann and Feldman (1972):

> The grammatical inference problem can be described as follows: a finite set of symbol strings from some language $\underline{L}$ and possibly a finite set of strings from the complement of $\underline{L}$ are known, and a grammar for the language is to be discovered . . . .
> Consider a class C of grammars and a machine M. Suppose some $G \in C$ and some I (an information sequence) in I(L(G)) are chosen for presentation to the Machine M_C. ...
> Intuitively, M_C identifies G if it eventually guesses only one grammar and that grammar generates exactly L(G).
> (pp. 31-33)

The significant point to note about this statement is that it is completely abstracted away from the problem of a child trying to learn his language. There has been virtually no concern for algorithms that will efficiently induce the subset of grammars that generate natural languages. The problem is posed in general terms. The characterization is syntactic. The concern is with inducing a characterization of the well-formed strings of the language. However, this is not the task which the child faces. Rather, he must induce a mapping between conceptualizations and strings of the language. That is, he must understand what is spoken to him and learn how to speak what is on his mind. If a characterization of the well-formed strings emerges, it is really a by-product of the mapping between sentences and meanings. Because of these biases in the formal work on language induction, there has been virtually no concern about the contribution that semantics might have to make to grammar induction.

The grammatical inference problem as characterized by Biermann and Feldman is without any practical solutions. Workable solutions do not exist because the set of possible languages is too unrestricted. Workable solutions are possible to practical problems only when it is possible to greatly restrict the candidate languages or because important clues exist which eliminate many a priori possible languages. Chomsky (1965) argued essentially for this view with respect to the problem of a child learning his first language. He suggested that the child could take advantage of linguistic universals which greatly restricted the possible languages. I will argue that such universals exist in the form of strong constraints between the structure of a sentence and the semantic structure of the referent. These constraints provide critical cues for the induction problem.

Gold's Work

Probably the most influential paper in the field is 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. Gold considers a language to be identified in the limit if after some finite time the learner discovers a grammer that generates the strings of the language. He considers two information sequences - in the first the learner is presented with all the sentences of the language and in the second the learner is presented with all strings, each properly identified as sentence or non-sentence. Then Gold asks this question: Suppose the learner can assume the language comes from some formally characterized class of languages; can he identify in the limit which language it is? Gold considers the classical nesting of language classes - finite cardinality languages, regular (finite state), context-free, context-sensitive, and primitive recursive. His classic result is that if the learner is only given positive information about the language (i.e., the first information sequence), then he can only identify finite cardinality languages. However, given positive and negative information (i.e., the second information sequence), he can learn up to primitive recursive languages.

The proof that the finite state class is not identifiable with only positive information is deceptively simple. Among the finite state languages are all languages of finite cardinality (i.e., with only finitely many strings). At every finite point in the 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 or an infinite cardinality finite state grammar which includes the sample. Logically, it could be either.

It is similarly easy to prove that any language in the primitive recursive class can be induced given positive and negative information. It is possible to enumerate all possible primitive recursive grammars. Assume an algorithm that proceeds through this countably infinite enumeration looking at one grammar after another until it finds the correct one. The algorithm will stay with any grammar as long as the information sequence is consistent with it. Any incorrect grammar G will be rejected at some finite point in the information sequence--either because the sequence contains, as a negative instance, a sentence generated by G, or as a positive instance, a sentence not generated by G. Since the correct grammar has some finite position in the enumeration, the algorithm will eventually consider it and stay with it. Gold's proofs are technically better than the above but these will do for present purposes.

The algorithm outlined in the second proof may not seem very satisfactory. For instance, the position is astronomical of English grammar in an alphabetic ordering of all possible context-sensitive languages using English morphemes as terminal symbols. However, Gold also proved that there is no algorithm uniformly more effective than this enumeration technique. That is to say, given any algorithm one can pick some context-sensitive language for which the enumeration algorithm will be faster.

So, Gold leaves us with two very startling results that we must live with. First, only finite cardinality languages can be induced without use of negative information. This is startling because children get little negative feedback and make little use of what negative feedback they do get (Brown, 1973). Second, no procedure is more effective than blind enumeration. This is startling because blind enumeration is clearly hopeless as a practical induction algorithm for natural language. Shortly, we will see how natural language can be induced despite Gold's results, but first let's review some other research of the same ilk.

Algorithms for Grammar Induction

One of the early attempts to provide a constructive algorithm was proposed by Solomonoff (1964). That is, he attempted to define an algorithm which would construct bit by bit the correct grammar rather than enumerating possible grammars. LAS is a constructive algorithm. His ideas were never programmed and had their logical flaws exposed by Shamir and Bar-Hillel (1962) and by Horning (1969). In part Solomonoff has served as a straw man that served to justify the enumerative approach over the constructive (e.g., Horning, 1969).

Feldman and his students have carried the Gold analyses farther. Feldman (1970) provided some further definitions of languages identifiability and proved Gold-like results for these. Feldman considered not only the task of inferring a grammar that generated the sample, but also the task of inducing the most simple grammar. Grammar complexity was measured in terms of number of rules and the complexity of sentence derivations. Horning (1969) provided procedures for inducing grammars whose rules have different probabilities. Biermann (1972) provided a number of efficient constructive algorithms for inducing finite state grammars when the number of states is known. This is a relatively tractable problem first formulated in 1956 by Moore, however, Moore's algorithms are much less efficient than Biermann's.

Pao (1969) formalized an algorithm for finite state grammar induction that did not require the number of states to be known in advance. A sample set of sentences was provided which utilized all the rules in the grammar. A minimal finite state network was constructed that generated exactly the sample set of sentences. Then an attempt was made to generalize by merging nodes in the network. The algorithm checked the consequences of potential generalizations by asking the teacher whether sentences added by these generalizations were actually in the target language. Pao's work is particularly interesting because she extended these induction procedures to context-free languages. Apparently unaware of Woods' work, she developed a network formalism that was very similar to his. She found that such augmented network grammars could be induced by her algorithms if she provided punctuation information indicating where transitions between networks occur. Basically such punctuation information amounts to indicating the sentence's surface structure. Interestingly, Saporta, Blumenthal, Lackowski, and Ruff (1963) found humans learned artificial context-free languages more easily when surface structure was indicated by spacing.

Crespi-Reghizzi (1970) also obtained encouraging results when his induction program was given information about sentence surface structure. He was interested in the induction of operator-precedence languages which are a subset of context-free languages. For a special subset of operator precedence languages he was able to define an algorithm that worked with only positive information. Except for finite cardinality languages, this is the only available result of success with just positive information.

I think the work of Pao and of Crespi-Reghizzi have promising aspects. They have shown relatively efficient, constructive algorithms are possible for interesting language classes if the algorithms have access to information about the sentence's surface structure. The problem with their work is that this information is provided in an ad hoc manner. It has the flavor of cheating and certainly is not the way things happen with respect to natural language induction. I will show how the surface structure of the sentence may be inferred by comparing the sentence to its semantic referent. Crespi-Reghizzi has also shown how the properties of a restricted subclass of languages can be used to reduce the reliance on negative information. While natural languages certainly have aspects that can be best captured with context-sensitive grammatical formalisms, most context-sensitive languages are ridiculous candidates for a natural language. An efficient induction algorithm should not become bogged down as does Gold's enumeration technique considering these absurd languages.

Grammar as a Mapping Between Sentence and Conception

There is one sense in which all the preceding work is irrelevant to the task of inducing a natural language. They have as their goal the induction of the correct syntactic characterization of a target language. But this is not what natural language learning is about. In learning a natural language the goal is to learn a map that allows us to go from sentences to their corresponding conceptual structures or vice versa. I argue that this task is easier than learning the syntactic structure of a natural language. This is not because there is any magic power in semantics per se, but because natural languages are so structured that they incorporate in a very non-arbitrary manner the structure of their semantic referent. The importance of semantics has been very forcefully brought home to psychologists by a pair of experiments by Moesser 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.

The Role of Semantics

Results like those of Moesser and Bregman have left some believing that there is some magic power in having a semantic referent. However, I will show that there is no necessary advantage to having a semantic referent. The relationship between a sentence and its semantic referent could, in principle, be an arbitrary recursive relation. Inducing this relation is at least as difficult as inducing an arbitrary recursive language. This last statement is in need of a proof which I have provided (Anderson, 1975). It is too involved to reproduce here, but basically it shows that an algorithm to induce an arbitrary semantic relation between referents and sentences, could be used to identify an arbitrary language. Thus, we know from Gold's work that an induction algorithm for the semantic relation could not be more effective than the impossible enumeration algorithm for identifying an arbitrary language. Thus, for it to be possible to induce the semantic relation, there must be strong constraints on the possible form of that semantic relation.

How does this semantic referent facilitate grammar induction? There are at least three ways: First, rules of natural language are not formulated with respect to single words but with respect to word classes like noun or transitive verb which have a common semantic core. So semantics can help determine the word classes. This is much more efficient than learning the syntactic rules for each word separately. Second, semantics is of considerable aid in generalizing rules. A general heuristic employed by LAS is that, if two syntactically similar rules function to create the same semantic structure, then they can be merged into a single rule. Third, there is a non-arbitrary correspondence between the structure of the semantic referent and the structure of the sentence which permits one to punctuate the sentence with surface structure information. The nature of this correspondence will be explained later.

Siklossy 's Work

The only attempt to incorporate semantics as a guide to grammar induction was by Siklossy (1971). He attempted to write a program that would be able to learn languages from the language-through-pictures books (e.g., Richards et al., 1961). The books in this series attempt to teach a language by presenting pictures paired with sentences that describe the depicted situations. Siklossy 's program, Zbie, used general pattern-matching techniques to find correspondences between the pictures (actually hand-encoded picture descriptions) and the sentences. The program does use information in the picture encodings to help induce the surface structure of the sentence, somewhat in the manner of LAS. However, it remains unclear exactly what use Zbie makes of semantics or what kinds of languages the program can learn. The displayed examples of the program's behavior are very sparse with examples of it making generalizations. As we will see, a program must have strong powers of generalization if it is to learn a language. The few examples of generalization all work as follows: Suppose Zbie sees the following three sentences:

1) John walks
2) Mary walks
3 John talks

It will generalize and assume Mary talks is an acceptable sentence. It does not seem that semantics plays an important role in guiding these generalizations.

Siklossy also provides no discussion of how his program's behavior relates to that of a human learning a language. The one example of an attempt to simulate child language learning is Kelley (1967). His program attempted to simulate the initial growth of child utterances from one word, to two words, to three words. Kelley claims to be making use of semantic information, but he never specifies its role in the program's performance. In general the details of the program are not explained. In his examples, the program never gets to the point of producing grammatical sentences and it is unclear whether it could.

4. Rationale

A central assumption in the LAS project is that a language learner can sometimes identify the meaning of sentences and that language learning takes place in these circumstances. The specific goal is to explain how the pairing of the sentence with its semantic referent permits language learning. The form of this explanation is to develop a computer program which can learn a language given an input of sentences paired with semantic interpretations. The computer program builds up a grammar that permits it to understand and generate sentences. Because of the inherent complexity, it is essential that this theory of language acquisition take the form of a computer program. I will argue further for the need of a computer model after describing the current version of LAS.

This project does have as an ultimate goal to provide a faithful simulation of child language acquisition. One might question whether a system constructed just to succeed at language learning will have much in common with the child's acquisition system. I strongly suspect it will, provided we insist that the system have the same information processing limitations as a child and provided its language learning situation has the same information-processing demands as that of the child. The consideration underlying this optimistic forecast is that learning a natural language imposes very severe and highly unique information-processing demands on any induction system and, consequently, there are very severe limitations on the possible structures for a successful system. A similar argument has been forcefully advanced by Simon (1969) with respect to the information-processing demands of various problem-solving tasks.

The current version of the program LAS.1 works in an overly simplified domain and makes unreasonable assumptions about information-processing capacities. Nonetheless, it predicts many of the gross features of generalization and overgeneralization in child language learning. It is terribly "off" in other aspects. It turns out that many of its failures of simulation can be traced to the unrealistic assumptions it is making about task domain and information processing abilities. Many of the proposed developments of the program have as their goal the elimination of these unrealistic assumptions. The assumptions were made to make the problem more tractable in a first-pass attempt.

5. The Program LAS.1

This section describes LAS.1, a relatively small program that was put together in eight months. It has achieved success in a non-trivial natural language induction situation. This proposal will be principally concerned with extending the power of LAS.1 and of producing a simulation which is somewhat more realistic psychologically. However, LAS.1 is a significant first step which is already more successful than any of its predecessors. This section will contain descriptions of the various aspects of the LAS.1 system. First, I will describe the HAM.2 memory system which provides LAS with its semantic powers. Following this will be an exposition of LAS's network grammar formalisms. With this as background, the working parts of the LAS program will be described. These include SPEAK which uses the network formalisms to generate sentences, UNDERSTAND which uses the same networks for sentence understanding, BRACKET which punctuates sentences with their surface structure by comparing them to their perceptual referents, and SPEAKTEST which builds an initial network grammar to parse a sentence, and GENERALIZE which generalizes the initial grammar.

Overview of LAS

LAS is an interactive program 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 (see Anderson & Bower, 1973). It obeys commands to speak, understand, and learn. The logical structure of LAS is illustrated in Figure 2. Central to LAS is an augmented transition network grammar similar to that of Woods (1970). In response to the command, Listen, LAS evokes the program UNDERSTAND. The input to UNDERSTAND is a sentence. LAS uses the information in the network grammar to parse the sentence and obtain a representation of the sentence's meaning. 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 principle purpose of SPEAK and UNDERSTAND in LAS is to provide a test of the grammars induced by LEARNMORE.

The philosphy 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 both direct the child's attention to what is being described and focus the child on 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 more information than is in the sentence. This provides no obstacle to LAS's heuristics. In this particular version of LAS, it is assumed that it already knows the meaning of the content words in the sentence. With this information 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 this case. These additions generalize to other cases so that LAS can understand many more sentences than the ones it was explicitly trained with.

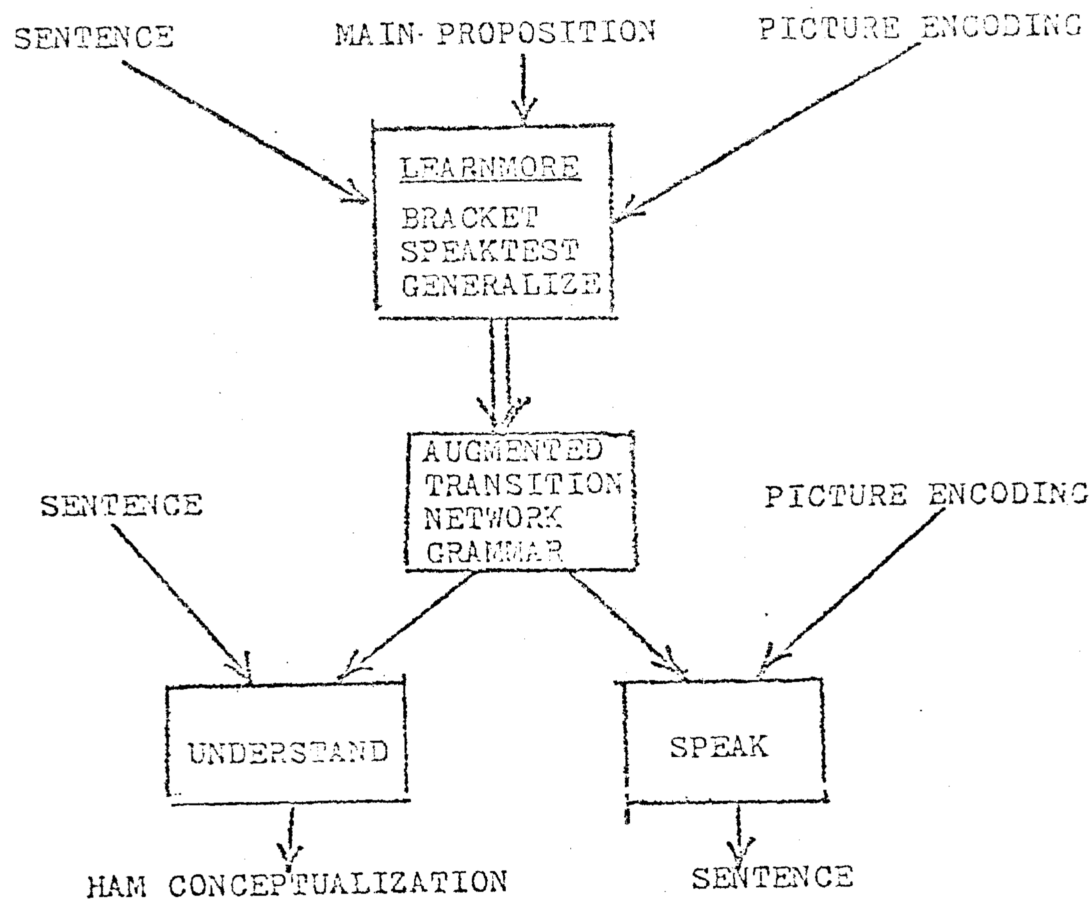


Figure 2. A schematic representation giving the input and output of the major subcomponents of LAS-LEARNMORE,SPEAK, and UNDERSTAND.

The SPEAKTEST program would permit LAS to construct a parsing network adequate to handle all the sentences it was presented with. Also it would make many low-level generalizations about phrase structures and word classes. This would permit LAS to successfully analyze or generate many novel sentences. However, many essential grammatical generalizations are left to be made by the program GENERALIZE. Principally, GENERALIZE must recognize that networks and words occurring at various points in the grammar are identical. Recognition of identical grammars is essential to identifying the recursive structure of the language. GENERALIZE is a program which is only called after fairly stable networks and word classes have been built up. It is only at this point that it is safe to make these critical generalizations.

The HAM. 2 Memory System

LAS. 1 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 for propositional knowledge. This is used for representing the comprehension output of UNDERSTAND, the to-be-spoken input to SPEAK, the semantic information in long-term memory, and syntactic information about word classes. HAM. 2 also contains a memory searching algorithm MATCH1 which is used to evaluate various parsing conditions. For instance, the UNDERSTAND program requires that certain features be true of a word for a parsing rule to apply. These are checked by the MATCH1 process. The same MATCH1 process is used by the SPEAK 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 3 illustrates how the information in the sentence <u>A red square is above the circle</u> would be represented with the HAM. 2 network formalisms. There are four distinct propositions predicated about the two nodes <u>X</u> and <u>Y</u>: <u>X is red</u>, <u>X is a square</u>, <u>X is above Y</u>, and <u>Y is a circle</u>. Each proposition is represented by a distinct tree structure. Each tree structure consists of a root proposition node connected by an <u>S</u> link to a subject node and by a <u>P</u> link to a predicate node. The predicate nodes can be decomposed into a <u>R</u> link pointing to a relation node and into a <u>O</u> 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 <u>X</u> is a subset of the red things, the square things, and the things above <u>Y</u>. The individual <u>Y</u> is a subset of the circular things.

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 <u>W</u>. Figure 3 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 within another. Thus, we cannot express in HAM. 2 such sentences as <u>Yesterday in my bedroom a red square was above the circle</u> or <u>John believes that a red square is above the circle</u>. Representations for such

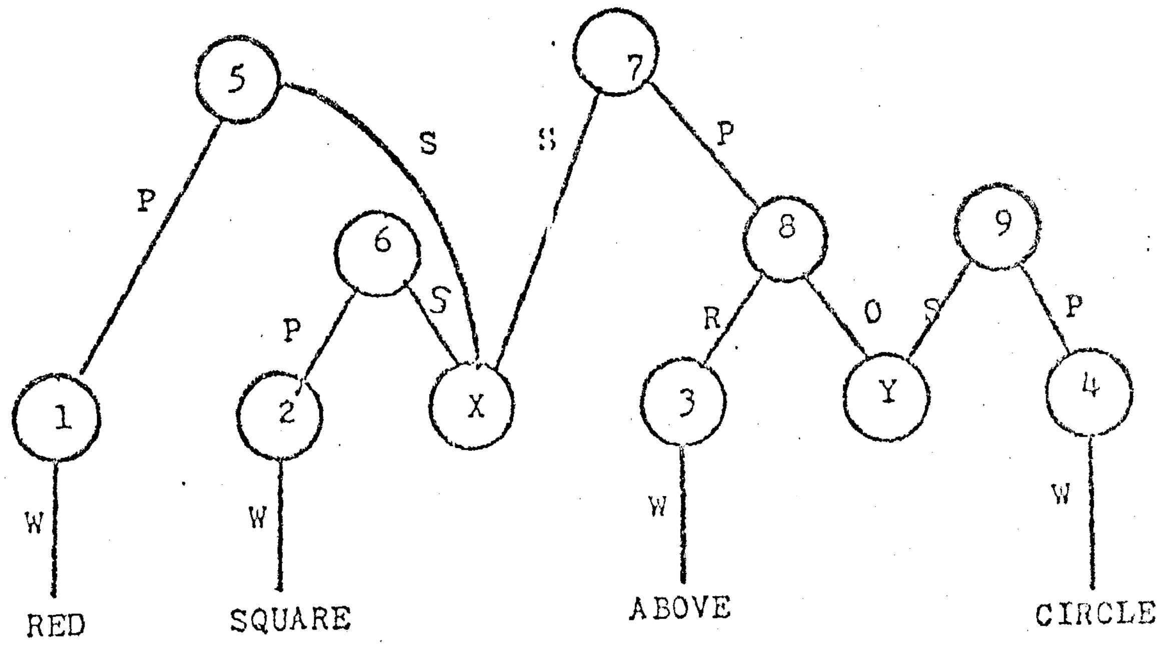


Figure 3. An example of a propositional network representation in HAM.[2]

statements are not needed in the current LAS project because we are only concerned with representing information that can be conveyed by ostension. In ostension, the assumed time and place are here and now. Concepts like belief which require embedded propositions are too abstract for ostension. In future research LAS will be extended beyond the current ostensive domain. At that point, complications will be required in the HAM.2 representations; however, 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 not 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 acessed 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 4 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.

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

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 3:

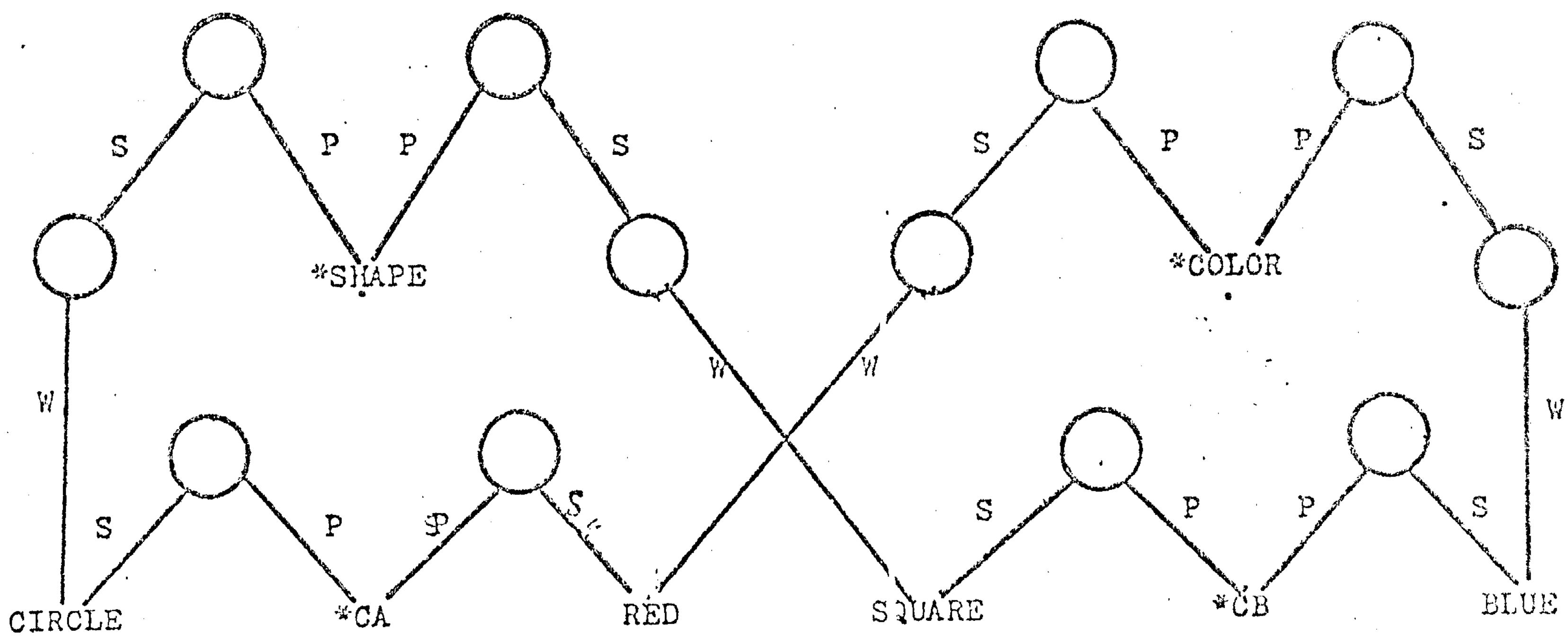


Figure 4. An example of a HAM structure encoding both categorical information and word class information

(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 Y)
(Relatify 8 3)
(Out-of Y 4)

The Network Grammars

Here the formalisms of LAS's network grammar will be described. These formalisms are intended to apply to any natural language. In illustration, the grammars for two test languages will be presented. These test languages will also be used to illustrate the SPEAK and UNDERSTAND programs to be described shortly. The first, GRAMMAR1, is a simple artificial grammar. The second, GRAMMAR2, is a more complex grammar for a subset of English. They are defined by the rewrite rules in Table 1. GRAMMAR1 was designed to be maximally different from English word order. The sentences of GRAMMAR1 are to be read as asserting the first noun-phrase has the relation specified by the last word to the second noun phrase. For purposes of readability, the words of these languages are English but they need not be. GRAMMAR1 is a finite language without recursion. In contrast, in GRAMMAR2 the NP element has an optional CLAUSE which can recursively call NP, generating a potential infinite embedding of constructions.

In both grammars, it is assumed that above and below are connected to the same idea as are right-of and left-of. The words differ in the assigment of their NP arguments to subject and object roles. Thus the difference between the word pairs is syntactic. This is indicated by having the words belong to two word classes RA and RB. Thus, UNDERSTAND with GRAMMAR2 would derive the same HAM representation in Figure 3 for the sentences The red square is above the circle and The circle is below the red square. It would have been possible to generate distinct representations for these two sentences. I think this would have been less psychologically interesting. Basically, the network grammar makes the inferences that A below B is equivalent to B above A and encodes the latter.

TABLE 1

The Two Test Grammars

GRAMMAR1

S	→	NP NP RA
		NP NP RB
NP	→	SHAPE (COLOR) (SIZE)
SHAPE	→	square, circle, et.
COLOR	→	red, blue, etc.
SIZE	→	large, small, etc.
RA	→	above, right-of

GRAMMAR2

S	→	NP is ADJ
		NP is RA NP
		NP is RB NP
NP	→	(the,a) NP* CLAUSE
NP*	→	SHAPE
	→	ADJ SHAPE
CLAUSE	→	that is ADJ
		that is RA NP

TABLE 1 continued

R	→	below, left-of	CLAUSE	→	that is RB NP
			SHAPE	→	square, circle, etc.
			ADJ	→	red, big, blue, etc.
			RA	→	above, right-of
			RB	→	below, left-of

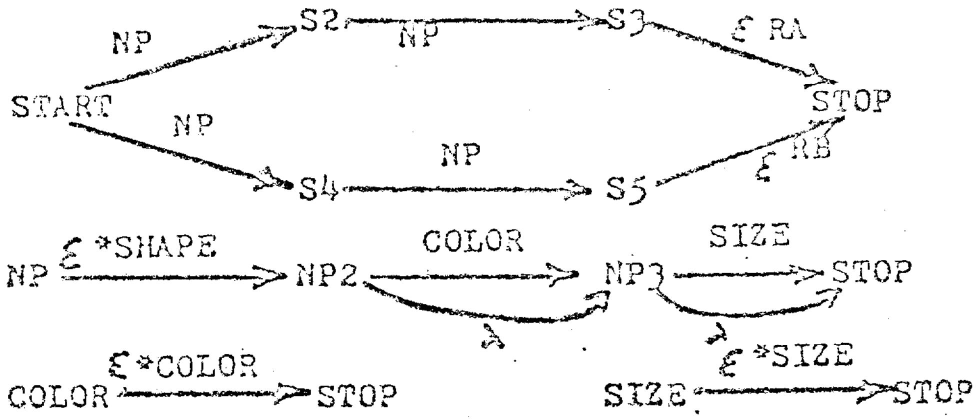
Figure 5 illustrates the parsing networks for the grammars. It should be understood that these networks have been deliberately written in an inefficient manner. For instance, note in GRAMMAR1 that there are two distinct paths in the main START network. The first is for those sentences with RA relations and the second for those sentences with RB relations. If a sentence input to UNDERSTAND has a RB relation, UNDERSTAND will first attempt to parse it by the first branch. The two noun phrase branches will succeed but the relation branch will fail. UNDERSTAND will have to back-up and try the second branch that leads to RB. This costly back-up is not really necessary. It would have been possible to have constructed the START network in the following form:

$$\begin{tikzcd} & & & \text{STOP} \\ \text{START} \arrow[r, "\text{NP}"] & \text{X} \arrow[r, "\text{NP}"] & \text{Y} \arrow[ur] \arrow[dr] & \\ & & & \text{STOP} \end{tikzcd}$$

In this form the network does not branch until the critical relation word is reached. This means postponing until the end the assignment of noun phrases to subject and object roles in the representations of the sentence's meaning. The above network was not chosen because we wanted a more demanding test of the backup facilities of SPEAK and UNDERSTAND.

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 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). A condition can consist of zero or more memory commands also (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 3 were to be spoken using GRAMMAR1. The START network would be called to realize the <u>X is above Y</u> proposition. The embedded NP network would be called to realize the <u>X is red</u> and <u>X is square</u> propositions. In pushing to a network two things must be specified--NODE, which is the embedded network and VAR, which is the memory node at which the main and embedded propositions intersect. The element <u>t</u> is rule 4b is a place-holder for information that is needed by the control mechanisms of the UNDERSTAND program. 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

Networks for Grammar1:



Networks for GRAMMAR2:

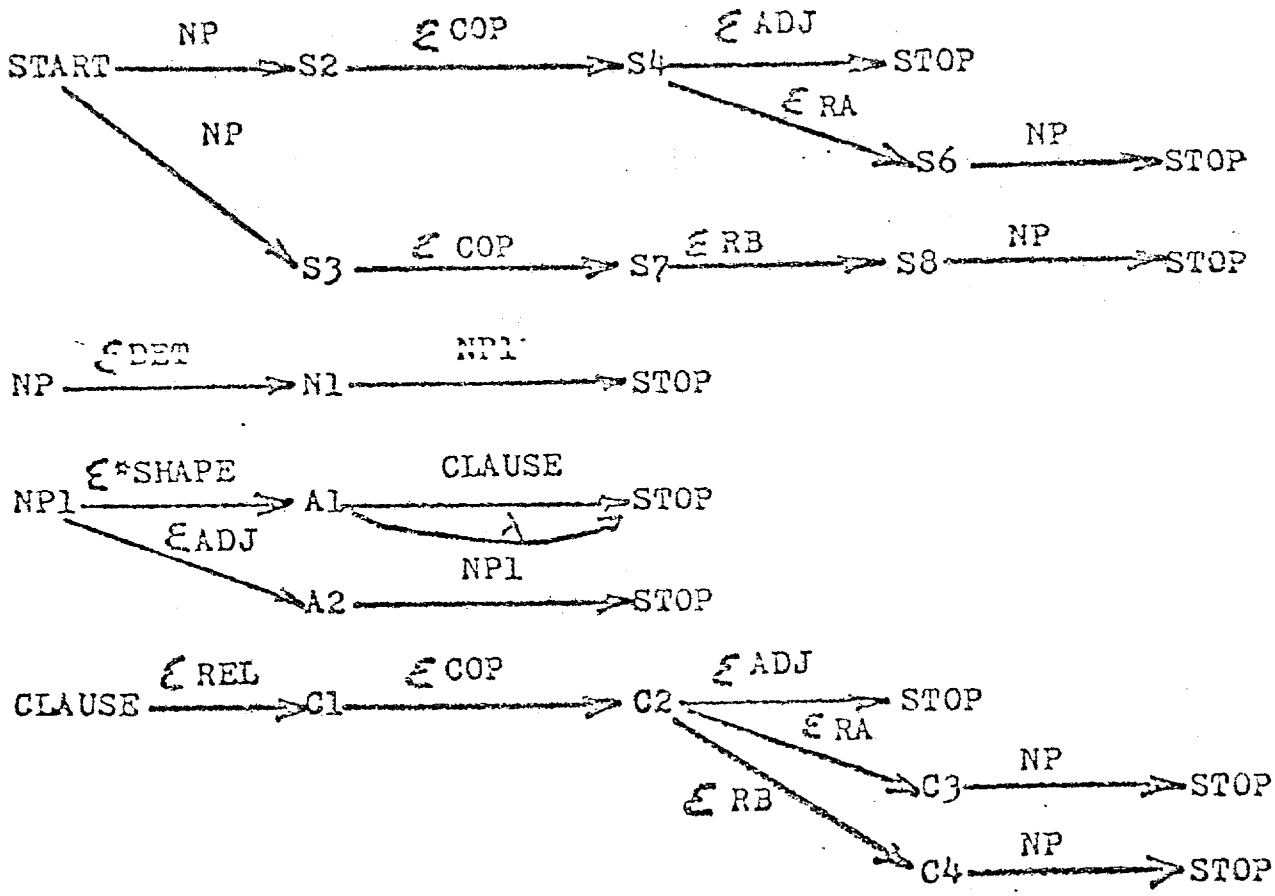


Figure 5. The network grammars used by LAS

memory nodes in the course of parsing.

TABLE 2

Formal Specification of the Network Grammar

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)
	→	X1, X2, X3, X4, X5	(6c)
FUNCTION	→	out-of, objectify, relatify, ideate	(7)

Table 3 provides the encoding of the network for GRAMMAR1.

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 GRAMMAR1 or GRAMMAR2, but as we will see, the grammars induced by LEARNMORE have necessarily a perfect correspondence.

These grammar networks have a number of features to commend them. SPEAK and UNDERSTAND use the same network for sentence comprehension and generation. Thus, LAS is the first extant system to have a uniform grammatical notation for its parsing and generation systems. 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. 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.

It might be questioned whether it is really a virtue to have the same representation for the grammatical knowledge both for understanding and production. It is a common observation that children's ability to understand sentences precedes their ability to generate sentences. LAS would not seem to be able to simulate this basic fact of language learning. 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 under-

Table 3

The contruction of GRAMMAR1

```
(PUT 'DEFPROP 'NSUBR (GET 'PUT 'SUBR))
(PROGN
 (DEFPROP START PATH
  (((PUSH X1 T NP) ((OUT-OF X1 X5)) S2 )
   ((PUSH X1 T NP) ((OBJECTIFY X5 X1)) S4 )))
 (DEFPROP S2 PATH
  (((PUSH X2 T NP) ((OBJECTIFY X5 X2)) S3 )))
 (DEFPROP S3 PATH
 ((((IDEATE WORD X4) (OUT-OF WORD *RA)) ((RELATIFY X5 X4)) STOP
 (DEFPROP S4 PATH
  (((PUSH X2 T NP) ((OUT-OF X2 X5)) S5 )))
 (DEFPROP S5 PATH
((((IDEATE WORD X4) (OUT-OF WORD *RB)) ((RELATIFY X5 X4)) STOP )
 (DEFPROP NP PATH
 ((((IDEATE WORD X4) (OUT-OF X4 *SHAPE)) ((OUT-OF X1 X4)) NP2 ))
 (DEFPROP NP2 PATH
  (((PUSH X1 T COLOR) NIL NP3 )
   ( NIL NIL NP3)))
 (DEFPROP NP3 PATH
  (((PUSH X1 T SIZE) NIL STOP )
   (NIL NIL STOP)))
 (DEFPROP COLOR PATH
 ((((IDEATE WORD X4) (OUT-OF X4 *COLOR)) ((OUT-OF X1 X4)) STOP
 (DEFPROP SIZE PATH
 ((((IDEATE WORD X4) (OUT-OF X4 *SIZE)) ((OUT-OF X1 X4)) STOP )
(TALK)
((IDEATE SQUARE X1)(IDEATE CIRCLE X2))
((OUT-OF X1 *SHAPE)(OUT-OF X2 *SHAPE))
((IDEATE RED X3)(IDEATE GREEN X4))
((OUT-OF X3 *COLOR)(OUT-OF X4 *COLOR))
(LISP SETQ X1 NIL)
((IDEATE SMALL X5)(IDEATE LARGE X1))
((OUT-OF X5 *SIZE)(OUT-OF X1 *SIZE))
NIL
(TALK)
((IDEATE TRIANGLE X1)(IDEATE BLUE X2)(IDEATE MEDIUM X3))
((OUT-OF X1 *SHAPE)(OUT-OF X2 *COLOR)(OUT-OF X3 *SIZE))
(LISP SETQ X1 NIL)
(LISP SETQ X2 NIL)
((IDEATE RIGHT-OF X1)(IDEATE ABOVE X2))
((OUT-OF RIGHT-OF *RA)(OUT-OF ABOVE *RA))
((OUT-OF LEFT-OF *RB)(OUT-OF BELOW *RB))
((IDEATE LEFT-OF X1)(IDEATE BELOW X2))
NIL
```

stood but was not able to speak. Also the child may have the potential to use a certain grammatical construction, but instead use 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 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.

Bloom (1973) has also argued that the general belief that comprehension precedes production in a child is a misperception on the part of the adult observer. 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 spontaneously 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.

The SPEAK Program

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 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 arguments for a recursive call of PUSH are the embedded network and the node that connects the main proposition and the embedded proposition. 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.

As an example, consider how SPEAK would generate a sentence corresponding to the HAM structure in Figure 6 using GRAMMAR2, the English-like grammar in Figure 5. Figure 6 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 asserted that it is a square and that it is above G182. Of G182 it is asserted that it is square, small, and red. Figure 7 illustrates the generation of this sentence from GRAMMAR2. LAS enters the START network intent on producing some utterance about G195. Thus, the topic is G195 (it could have been G246 or G182). 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 8 reflects the construction of this subnetwork. The NP network is called which prints _the_ and calls NP1 which retrieves _square_ and calls CLAUSE which prints _that_, _is_, and _right-of_ and which recursively calls NP to print _the square_. Similarly, recursive calls are made on the NP1 network to express G182 as _the small red square_.

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_ vs. _right-of_ and of _above_ vs. _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 solutional 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 in 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 acceptabil

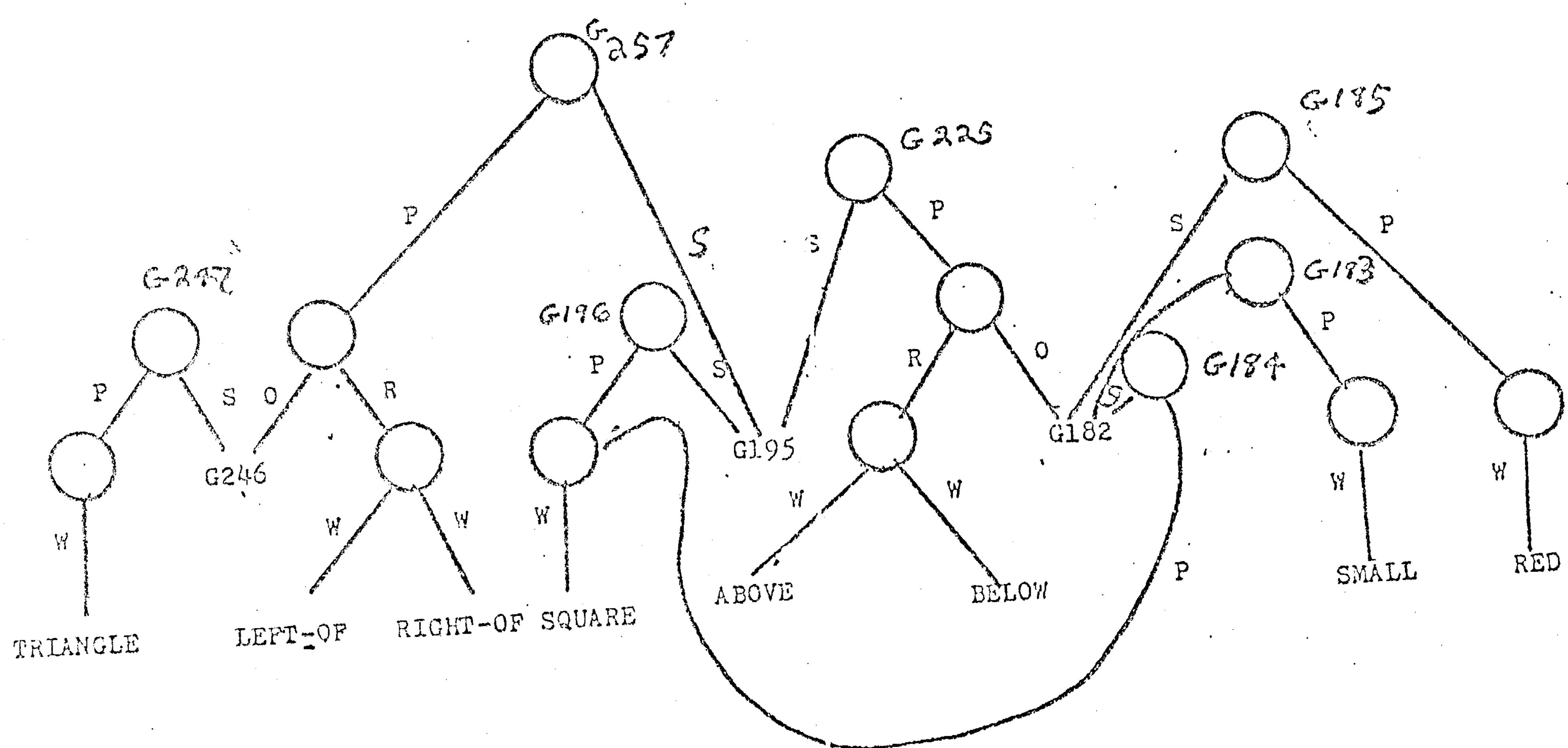


Figure 6. The to-be-spoken HAM network for the SPEAK program.

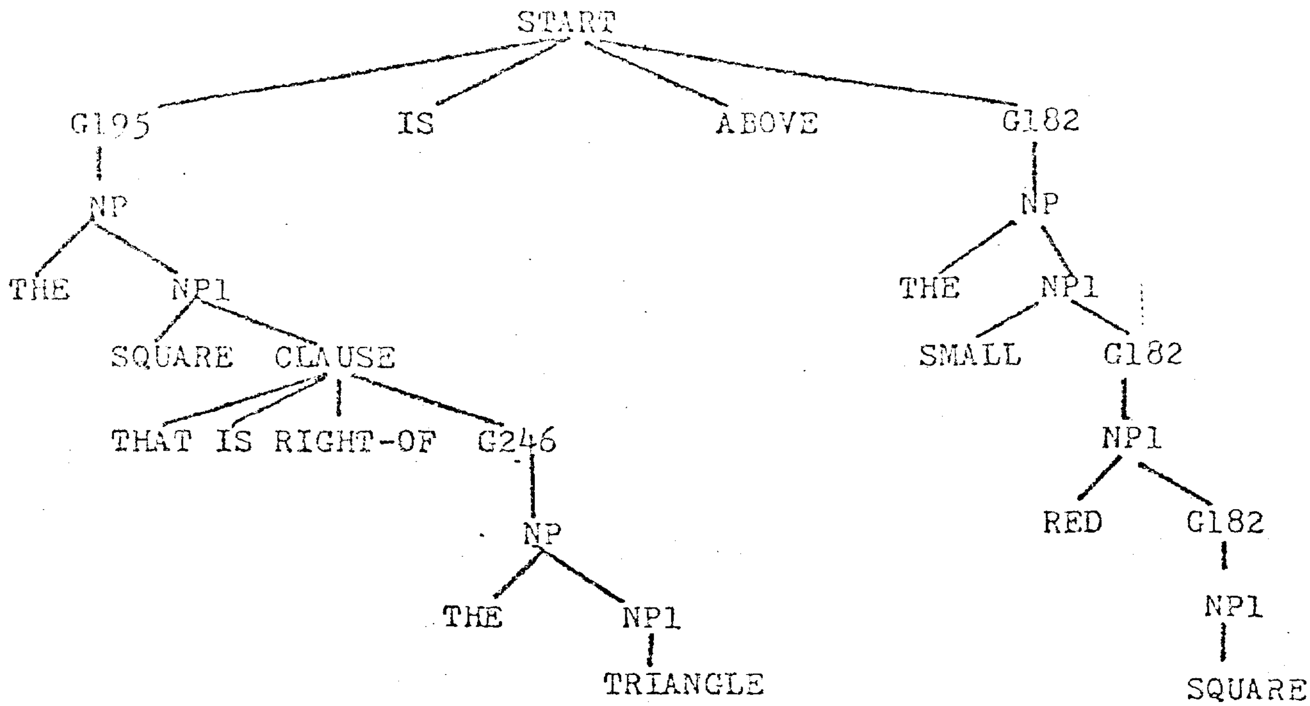


Figure 7. A tree structure showing the network calls and word output. These networks were called in generating a sentence about G195 which expressed the information contained in Figure 6.

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 subnetwork 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. The UNDERSTAND program and its control structure were written by Carol Hafner, a computer science student at Michigan.

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. The complexities of this control structure are described in a more complete report (Anderson, 1975). Here I will just overview the general structure of the program. The program tries to find some path through the START network which will result in a complete parsing of the sentence. It evaluates the acceptability of a particular path by evaluating the conditions associated with that path. A condition may require that certain features be true of words in the sentence. This is determined by checking memory. Alternatively, a condition can require a push to an embedded network. This network must parse some subphrase of the sentence. When LAS finds an acceptable path through a network it will collect the actions along that path to create a temporary memory structure to represent the meaning of the phrase that LAS has parsed. This, for instance, given the sentence, The square that is right-of the triangle is above the small red square, LAS would parse it in the form illustrated for Figure 7, retrieving the HAM structure in Figure 6. That is, in LAS. 1, understanding really is simply generation put in reverse. This is the first displayed example of a reversible augmented transition network. Simmons (1973) comes closest with two different networks, one for generation and one for analysis.

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 an 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 an arbitrary number of adjectives.

This led to the rule in GRAMMAR2 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 ungrammatical phrases. 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.

General Conditions for Language Acquisition

Having now reviewed how LAS. 1 understands and produces sentences, I will present the three aspects of the induction program: BRACKET, SPEAKTEST, and GENERALIZE. Before doing so, it is wise to briefly state the conditions under which LAS learns a language. It is assumed that LAS. 1 already has concepts attached to the words of the language. That is, lexicalization is complete. The task of LAS. 1 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. Because LAS. 1 is not concerned with learning meanings, it cannot be a very realistic model for second language learning where many concepts can transfer from the first to the second language. I will propose extensions of LAS. 1 concerned with learning word meanings.

Another feature of LAS. 1 is that it works in a particularly restricted semantic domain. It is presented with pictures indicating relations and properties of two-dimensional geometric objects. These pictures are actually encoded into the HAM propositional network representation. Along with these pictures LAS is presented sentences describing the picture and an indication of that aspect in the picture which corresponds to the main proposition of the sentence. From this information input, a network grammar is constructed. The semantic domain may be very simple, but the goal is to be able to learn any natural or natural-like language which may describe that domain.

The BRACKET Program

A major aspect of the LAS project is the BRACKET program. This is an algori for taking a sentence of an arbitrary language and a HAM conceptual structure and producing a bracketing of the sentence that indicates its surface structure. This surface structure prescribes the hierarchy of networks required to parse the sentence. For BRACKET to succeed, four conditions must be satisfied by the information input to it:

Condition 1. All content words in the sentence correspond to elements in the co ceptual structure. This amounts to the claim that the teacher is able to direct the learner to conceptualize the information in his sentence. It does not matte to the BRACKET algorithm whether there is more information in the conceptual structure than in the sentence.

Condition 2. The content words in the sentence are connected to the elements in the conceptual structure. Psychologically, this amounts to the claim that lexicalization is complete. That is, the learner knows the meanings of the wor

Condition 3. The surface structure interconnecting the content words is isomorphic in its connectivity to a language-free prototype structure.

Condition 4. The main proposition in conceptual structure is indicated.

Conditions 3 and 4 require considerable exposition. To explain Condition 3 I will first assume that the prototype structure is just the HAM conceptual structure. Later I will explain why something slightly different is required.

Consider Panel (a) of Figure 8 which illustrates the HAM structure for the series of propositions in the English sentence The red square is above the small circle. Panel (b) illustrates a graph deformation of that structure giving the surface structure of the sentence. Note how elements within the same noun phrase are appropriately assigned to the same subtree. 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 HAM structure in Panel (a) is set forth in a particular spatial array, the choice is arbitrary. In contrast, the surface structure of a sentence does specify the spatial relation of links. 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. However, the surface structure of all natural languages is derived from the same graph patterns. Panel (c) of Figure 8 shows how the prototype structure of Panel (a) can provide the surface structure for a sentence of the artificial GRAMMAR1. All the sentences of GRAMMAR1 preserve the connectivity of the underlying HAM structure. By this criterion, at least, GRAMMAR1 could be a natural language.

However, certain conceivable languages would have surface structures which could not be deformations of the underlying structure. Panel (d) illustrates such a hypothetical language with the same syntactic structure as English, but with different rules of semantic interpretation. In this language the adjective phrase preceding the object noun modifies the subject noun. As Panel (d) illustrates, there is no deformation of the prototype structure in Panel (a) to achieve a surface structure for the sentences in the language. No matter how it is attempted some branches must cross.

LAS will use the connectivity of the prototype network to infer what the connectivity of the surface structure of the sentence must be. The network does not specify the right-left ordering of the branches or the above-below ordering. The right-left ordering can be inferred simply from the ordering of the words in the sentence. However, to specify the above-below ordering, BRACKET needs one further piece of information. Figure 9 illustrates an alternate surface structure that could have been assigned to the string in Figure 8 (c). It might be translated into English syntax as Circular is the small thing that is below the red square. Clearly, as these two structures illustrate, the HAM network and the sentences are not enough to specify the hierarchical ordering of subtrees in the surface structure. The difference between the sentences in Figure 8 (c) and 9 is the choice of which proposition is principal and which is subordinate. If BRACKET is also given information as to the main proposition it can then unambigiously retrieve the sentence's surface structure. 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 that he also realized the aboveness relation was what was being asserted in the picture.

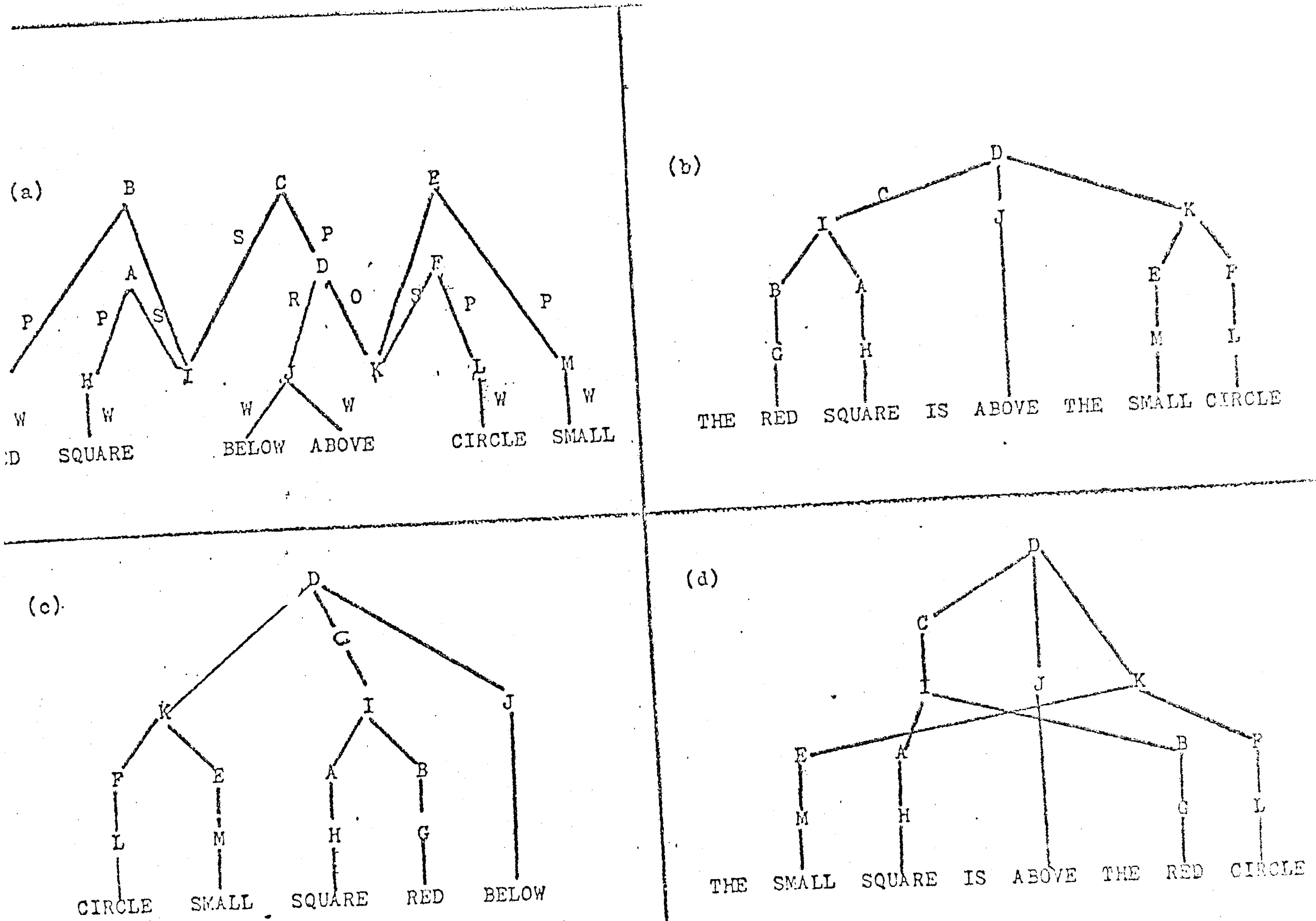


Figure 8. The surface structures of the sentences in (b) and (c) are graph deformations of the HAM structure in (a). Panel (d) the HAM structure [illegible] to deform (a) into a

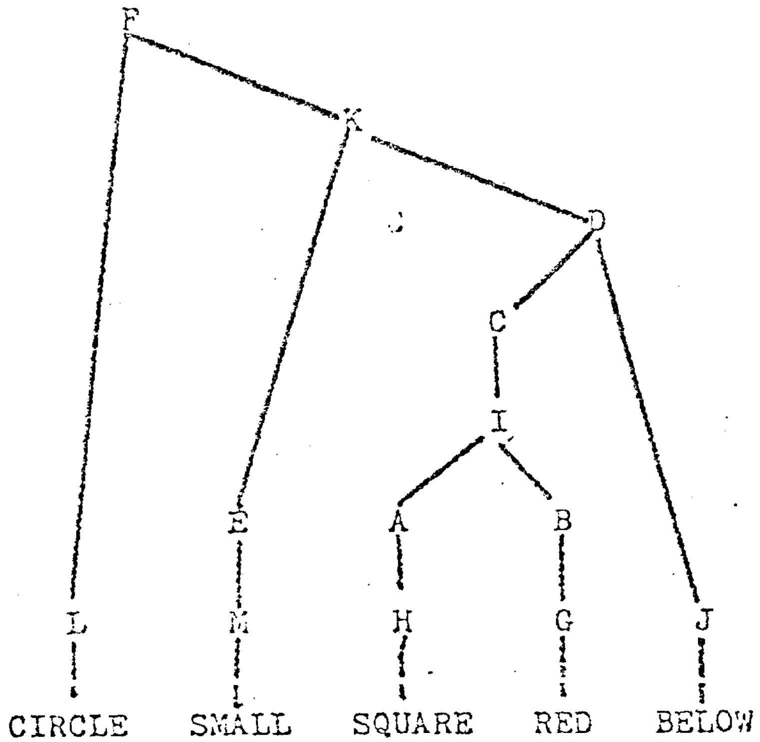


FIgure 9. Alternate surface structure for the sentence in Figure 9c.

More on the Graph Deformation Condition

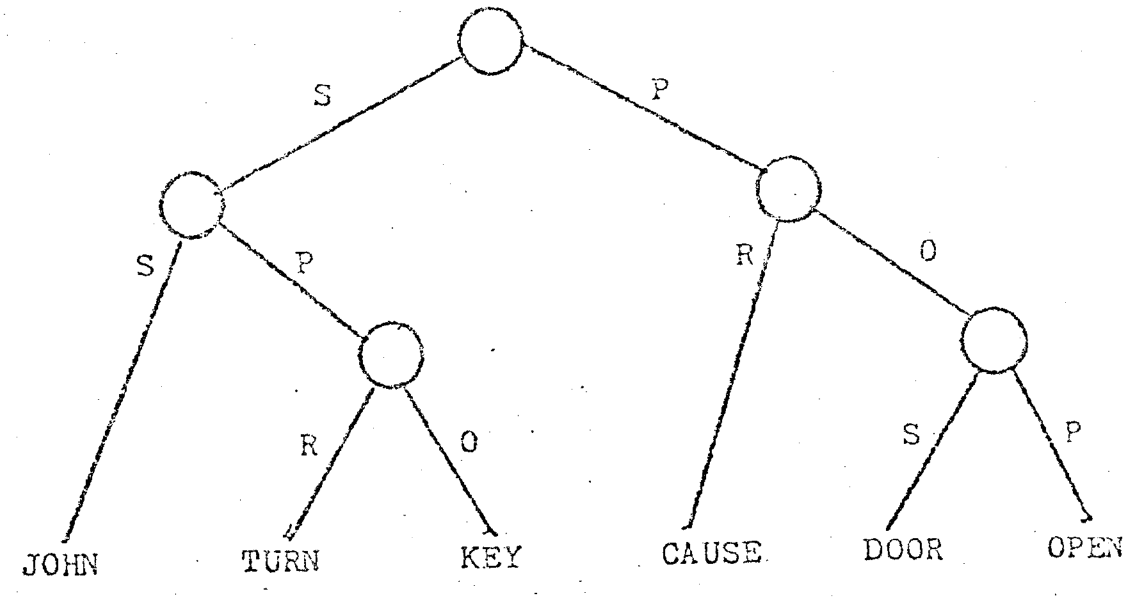
I think that the graph deformation condition has something of the status of a universal property of language. However, to make this claim viable it is clear that something other than the HAM network will have to be adopted as the prototype structure. HAM's binary branching works well enough for the domain of discourse that I have been interested in so far, but it will not generalize to sentences that have verbs that take more than two noun phrase arguments. Figure 10a shows how HAM would represent the sentence John opened the door with a key. This is decomposed into a set of sub-propositions--John turned the key which caused the door to be opened. Because of the binary structure certain elements are grouped together. In particular, John and key are closer together and door and open are closer together. If Figure 10a were the prototype, LAS could not bracket a sentence which alternated words from the two subgroups. For instance, there is no deformation of the structure in (a) that would provide a bracketing for John opened with a key the door. Branches of the HAM structure would have to cross. This English sentence and other English sentences which violate the deformation condition for Figure 10a have all a semi-unacceptable ring to them. However, this is almost certainly a peculiarity of English. Other languages permit free ordering of their noun phrases. What is needed for a prototype structure is something like the case representation in Figure 10b where all arguments are equally accessible from the main proposition node. The problem posed by the verb open is one posed by any verb which takes more than two noun phrase arguments. HAM's representation rules out certain sequences of the verb and its arguments while it is likely that all sequences can be found in some natural language. There are two ways to deal with this dilemma. One could resort to a memory representation like (b). However, there are a number of significant considerations that motivate the HAM representation in panel (a). Moreover, representations like (b) finesse one of the most interesting questions in language acquisition--how we learn the case structure of complex verbs. To address this question we need a representation that decomposes multi-argument verbs into a representation like (a) which exposes the semantic function of the case arguments. Learning the role of the verb open in the language then involves learning how to assign its noun phrase arguments to a structure like (a). I will sketch a system to do this in the proposal section.

If we keep the HAM representations then some changes are required in BRACKET-graph deformation condition. What is characteristic of multi-argument verbs in HAM is that the arguments are interconnected by causal relations as in (a). Thus, BRACKET should be made to treat all the terminal arguments in such causal structures as defining a single level of nodes in a graph structure all connected to a single root node. That is, BRACKET can treat a HAM structure such as (a) if it were (b) for purposes of utilizing the graph deformation condition. In fact, BRACKET already does this in the current implementation.

The Details of BRACKET's Output

So far, only a description of how one would retrieve the surface structure connecting the content words of the sentence has been given. Suppose BRACKET were given A triangle is left-of a square that is above a small red square. A bracketing structure must be imposed on this sentence which will

(a)



(b)

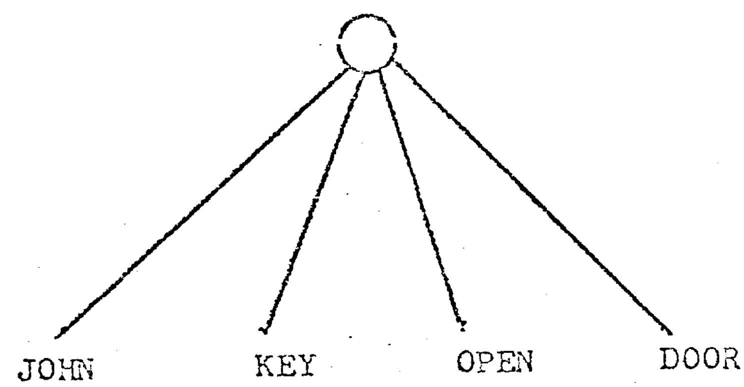


Figure 10. Alternative prototype structures for the sentence John opened the door with a key. The HAM structure in (a) introduces too many distinctions.

also include the function words. Given this sentence and the conceptual structure in Figure 6, BRACKET returned (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 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 subexpression 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 three rewrite rules:

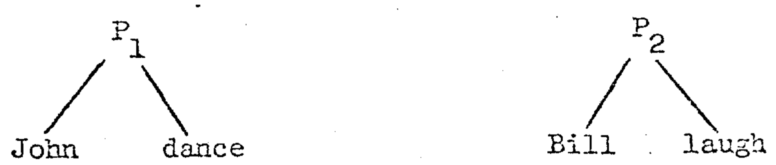
1. S → proposition element *
2. element → word
3. element → (topic S)

That is, each bracketed output is a proposition node followed by a sequence of elements (rule 1). These elements are either rewritten as words (rule 2) or bracketed subexpressions (rule 3). A bracketed subexpression begins with a topic node which indicates the connection between the embedded and embedding propositions. The elements within an expression are either non-meaning bearing words or elements corresponding to subject, predicate, relation and object in the proposition. 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. When a number of embedded propositions are attached to the same node, they are embedded within one another in a right-branching manner.

The insertion of non-function words into the bracketing is a troublesome problem because there is no semantic features to indicate where they belong. Consider the first word a in the example sentence above in Figure 6. It could have been placed in the top level of bracketing or in the subexpression containing triangle. Currently, all the function words to the right of a content word are placed in 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 is a strong cue 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 cues at all as to the structure of that language. Therefore, we would be back to the same impossible language induction task that we characterized in the introduction. 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 a single word. 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. Again it does not seem to be the case that these dependencies exist. So once again we find that the structure of natural language is simple just at those points where it would have to be for a LAS-like induction program to work.

In concluding this section I should point out one example sentence which BRACKET cannot currently handle. They are *respectively* sentences like *John and Bill danced and laughed respectively*. The problem will such a sentence is that underlying it is the following prototype structure:



Thus, *John* and *dance* are close together and so are *Bill* and *laugh*. However, the sentence intersperses these elements just in the way that makes bracketing impossible. There are probably other examples like this, but I cannot think of them. Fortunately, this is not an utterance that appears early in child speech nor is a particularly simple one for adults. Of all the grammatical constructions, the *respectively* construction is the one that most suggests the need to have transformational rules in the grammar.

SPEAKTEST

The function of SPEAKTEST is to test whether its grammar is capable of generating a sentence and, if it is not, appropriately modify 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 to the network. If it cannot, a new path is built through the network to incorporate the sentence.

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 illustrated in Table 4. This is a very simple language, basically GRAMMAR1 of Table 1. It has a smaller vocabulary to make it more tractable. The reason for choosing this language is that it is of just sufficient complexity to illustrate LAS's acquisition mechanisms. In addition, LAS has learned GRAMMAR2, also given in Table 1.

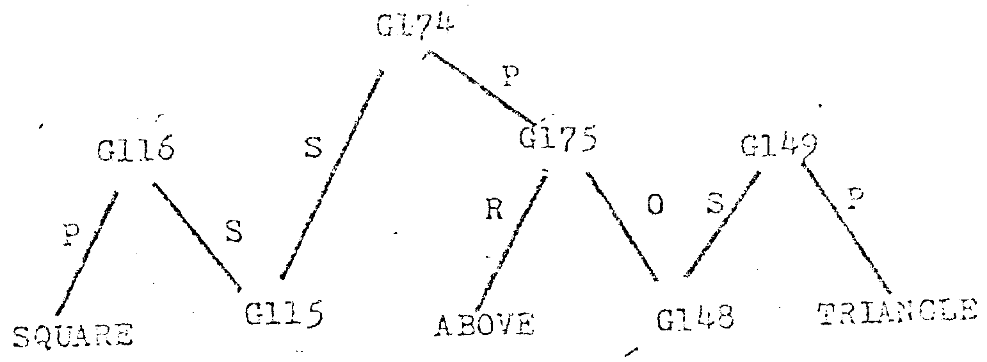
Figure 11 illustrates LAS's handling of the first two sentences that come in. The first sentence is Square triangle above. This sentence is returned by BRACKET as (G174 (G115 G116 square) (G148 G149 triangle) above). G174 refers to the main proposition given as an argument to LEARNMORE. Since this is 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 11. A listing of the exact arc information induced is given below the graphical illustration in Figure 11. 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 subnetworks. The third arc contains a condition on the word above. The restriction made is that it be a member of the word class A199. 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 subnetworks in that path to see whether they can handle the bracketed subexpressions in the sentence. This is accomplished by a recursive call to SPEAKTEST. For the first phrase, SPEAKTEST is called, taking as arguments the network A195, the phrase (G116 square) and the topic G115. In network A195 the word class A211 is created to contain square, and in network A197 the word class A221 contains triangle. These two subnetworks should be the same in a final grammar but LAS is not prepared to risk such a generalization at this point.

Note in this example how the bracketing 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 subexpression indicating a subnetwork should be created. Similarly, the third expression indicated a subnetwork. 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 A199 to include right-of, word class A211 to include triangle, and word class A221 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,

(a)



((SQUARE)(TRIANGLE) ABOVE)

START $\xrightarrow{A195}$ A196 $\xrightarrow{A197}$ A198 $\xrightarrow{\varepsilon A199}$ STOP

A195 $\xrightarrow{\varepsilon A211}$ STOP

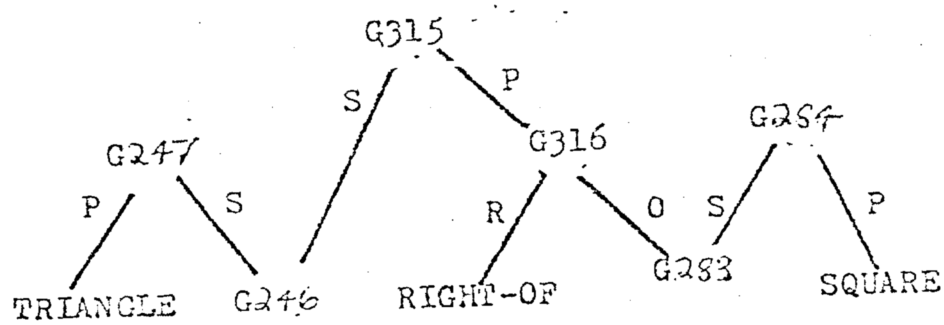
A197 $\xrightarrow{\varepsilon A221}$ STOP

A199 $\longrightarrow$ ABOVE

A211 $\longrightarrow$ SQUARE

A221 $\longrightarrow$ TRIANGLE

(b)



((TRIANGLE)(SQUARE) RIGHT-OF)

A199 $\longrightarrow$ ABOVE, RIGHT-OF

A211 $\longrightarrow$ SQUARE, TRIANGLE

A221 $\longrightarrow$ TRIANGLE, SQUARE

Figure 11. LAS's treatment of the first two sentences in the induction sequence.

the first word of the second sentence. This involved (a) using the same subnetwork A195 that had been created for square and (b) expanding the word class A211 to include triangle. Both decisions rested on semantic criteria. The network A195 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 A195 Within the A195 network the word class A211 contains words which are predicates of the subject node. Triangle has this semantic function and is therefore added to the word class.

In making these generalizations, SPEAKTEST is making a strong assumption about the nature of natural language. This assumption is stated as Condition 5:

Condition 5. Words or phrases with identical semantic functions at identical points in a network behave identically syntactically. This is the assumption of semantic-induced equivalence of syntax. It is another way in which semantic information facilitates grammar induction. It clearly need not be true of an arbitrary language. For instance, decisions made in the subject noun phrase might in theory condition syntactic decisions made in the object noun phrases. LAS, because of its heuristics in SPEAKTEST for generalization, would not be able to learn such a language.

Figure 12 illustrates LAS's network grammar after two more sentences have come in. Sentences 3 and 4 involve the relations below and left-of. LAS treats these as syntactic variants of above and right-of which differ in their assignment of their noun phrase arguments to the logical categories subject and object. Therefore, LAS creates an alternative branch through its START network to accommodate this possibility.

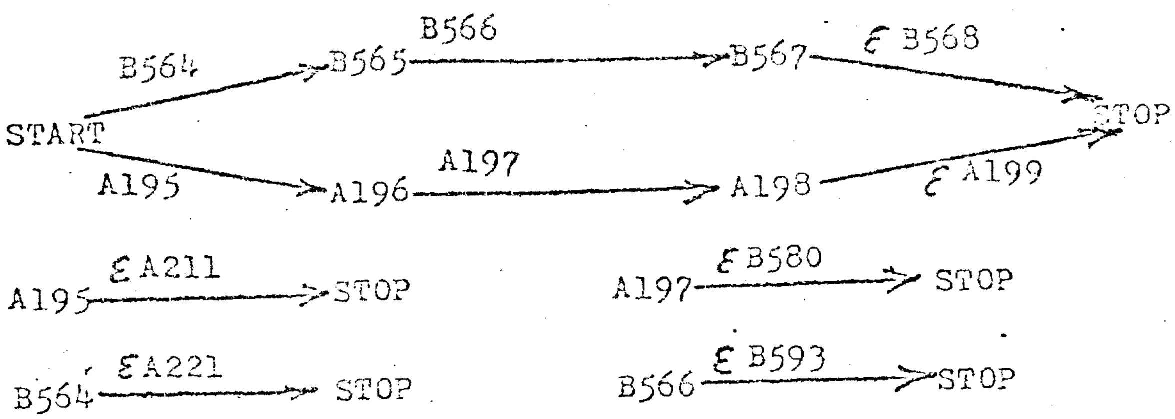
Figure 13 illustrates the course of LAS's learning. Altogether LAS will be presented 14 sentences. Subsequently, it will have to make three extra generalizations to capture the entire target language. Plotted on the abscissa is this learning history and along the ordinate we have the natural logarithm of the number of sentences which the grammar can handle. This is a finite language, unlike GRAMMAR2, and therefore the number of sentences in the language will always be finite. As can be seen from Figure 13, by the fourth sentence LAS's grammar is adequate to handle 16 sentences.

LAS's grammar after the next five sentences is illustrated in Figure 14. These are LAS'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 A195 and A197 sub-networks. Consider the first sentence, square red triangle blue above, which is retrieved by BRACKET as (C329 (C270 C271 square (C270 C272 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 A195 subnetwork. As can be seen in Figure 14, the existing arc containing the A211 word class is kept to handle square. Two alternative arcs are added--one with a push to

Figure 12

LAS's grammar after studying:

1. SQUARE TRIANGLE ABOVE
2. TRIANGLE SQUARE RIGHT-OF
3. SQUARE TRIANGLE BELOW
4. TRIANGLE SQUARE LEFT-OF



A199 = above,right-of
A211 = square,triangle
A221 = square,triangle
B568 = below,left-of
B580 = square,triangle
B593 = square,triangle

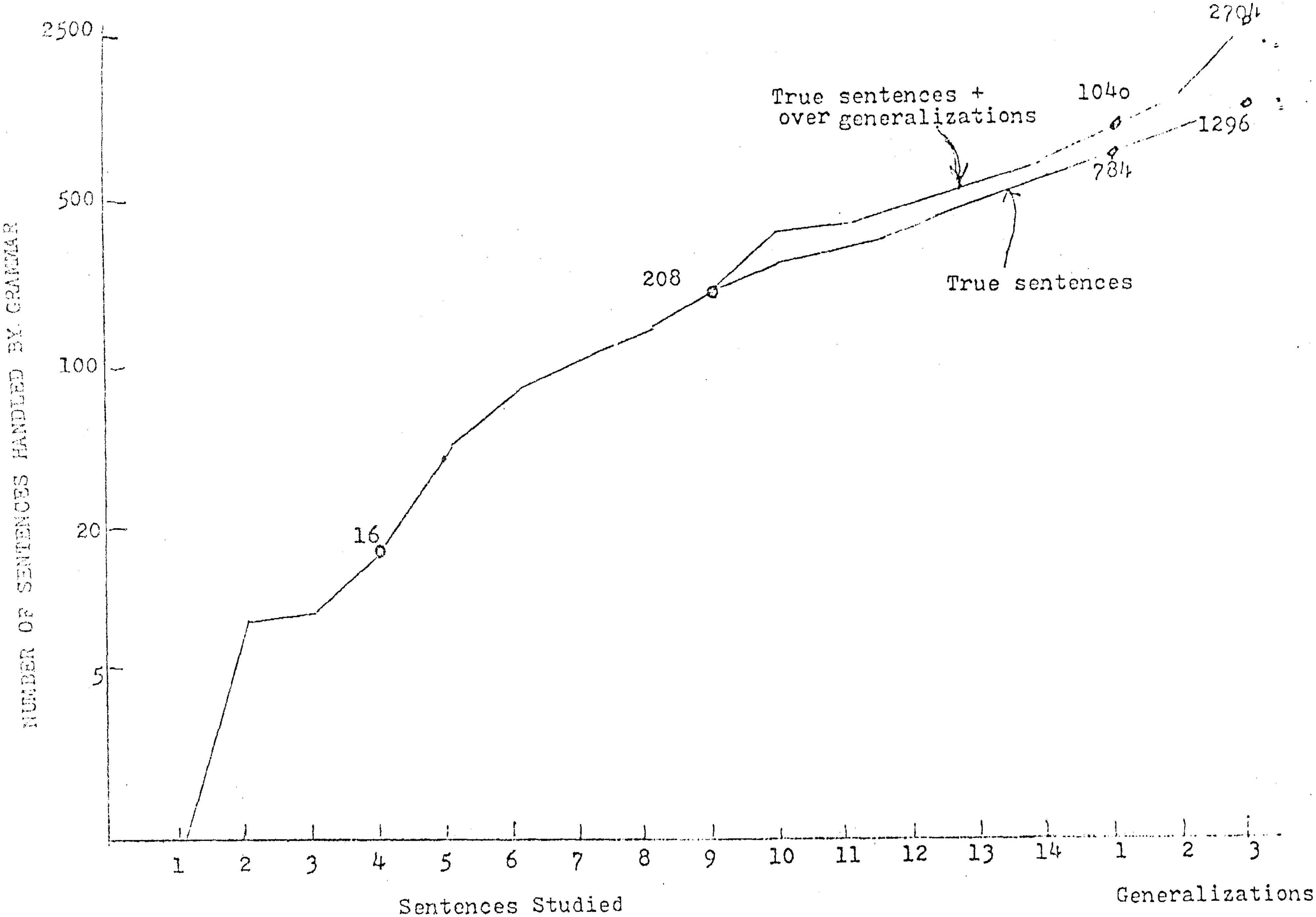
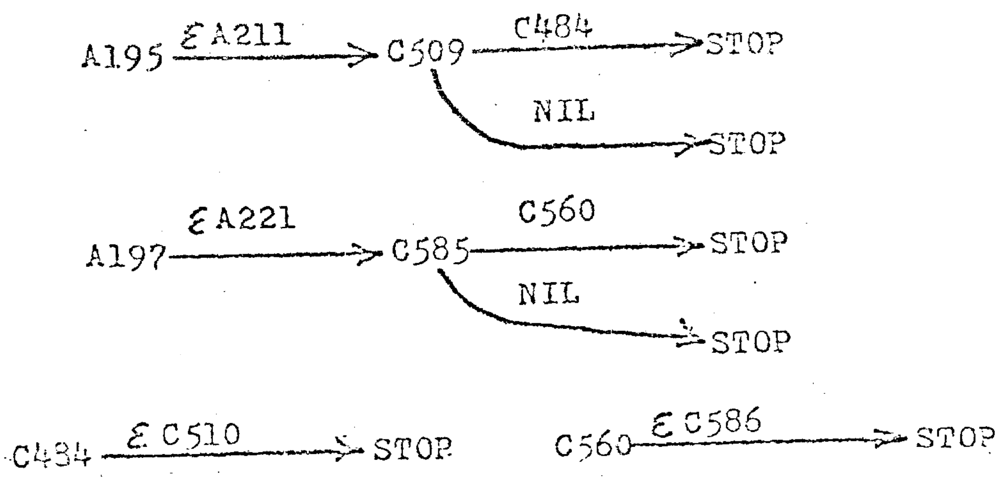


Figure 13. The growth of LAS's grammar with its learning history.

Figure 14

Additions to LAS's grammar after studying:

1. SQUARE RED TRIANGLE BLUE ABOVE
2. TRIANGLE LARGE SQUARE SMALL RIGHT-OF
3. TRIANGLE RED TRIANGLE RED ABOVE
4. SQUARE SMALL TRIANGLE RED RIGHT-OF
5. SQUARE BLUE TRIANGLE LARGE RIGHT-OF



C510 = small,blue,large,red
C586 = small,blue,large,red

the C494 network and the other with a NIL transition. Within the C484 network the word class C510 is set up which initially only contains the word red.

This illustrates the principle of left generalization in LAS: 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, A2, ... A_m have the same semantic function 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. Left generalization may be seen as an elaboration of semantics-induced equivalence of syntax (Condition 5).

Figure 15 illustrates 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 word 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. 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.[1] 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 generalization is, of course, a notorious overgeneralization 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 its principle of semantics-induced equivalence of syntax, LAS will overgeneralize in those situations. Apparently, children are operating under a similar rule.

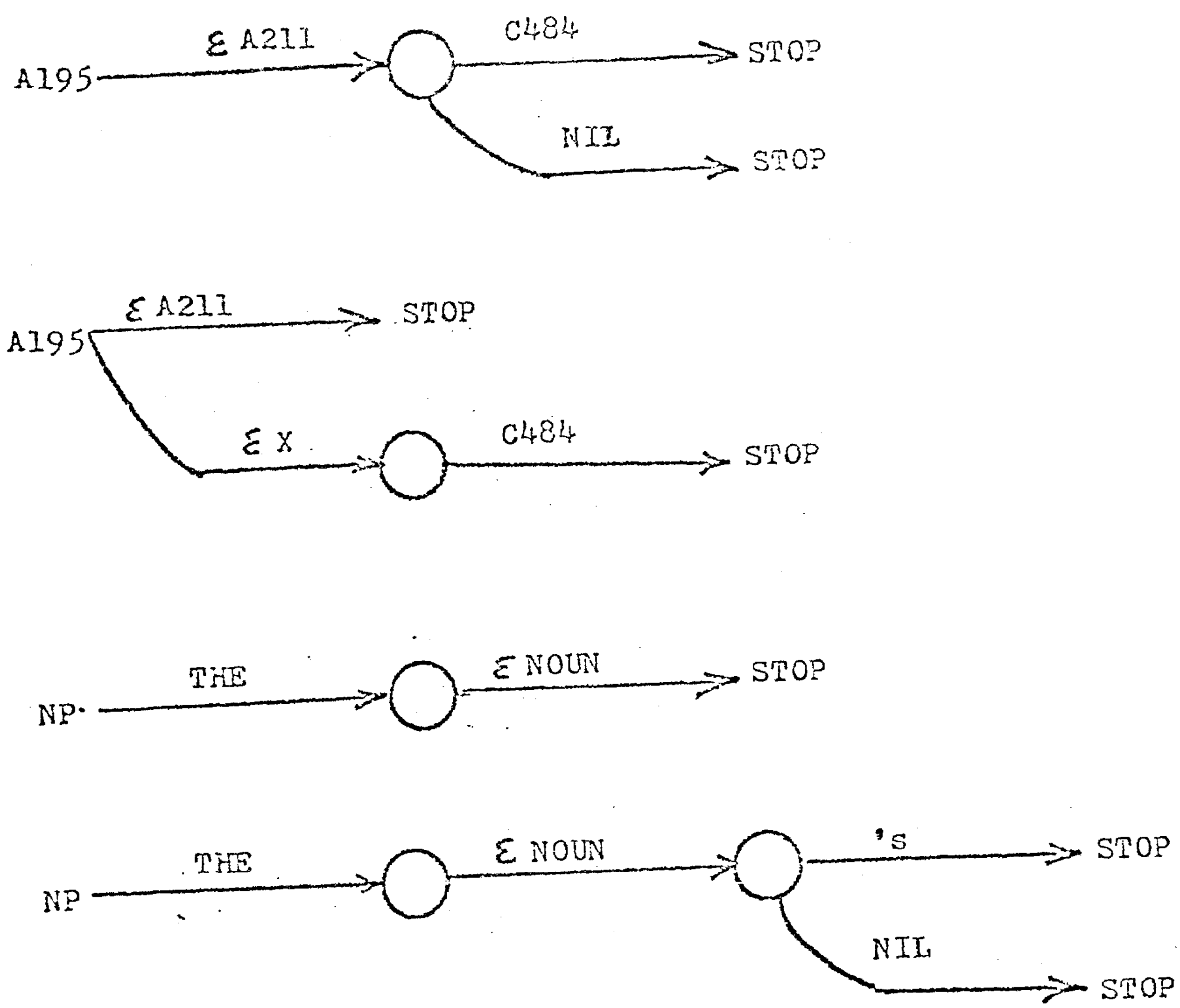
LAS needs to be endowed with a mechanism to allow it to recover from such overgeneralizations. Therefore, one of the future additions to LAS will have to be a RECOVER program. Consider how it would work with this pluralization example. Suppose LEARNMORE receives the sentence The feet are above the triangle. In attempting to analyze the sentence in SPEAKTEST, the plural foots will be generated but will mismatch the sentence. RECOVER has as its function to note such mismatches. Since it is possible that there are two alternate ways of expressing plurality, RECOVER cannot assume its grammar is wrong. Rather it will interrupt the information flow and check the acceptability of The foots are above the triangle. That is, RECOVER will explicitly seek negative information. Upon learning the expression is ungrammatical RECOVER will take foot out of the word class that is pluralized by 's.

---

1 To accomplish this I would have to put within LAS some mechanism that will segment words into their morphemes.

Figure 15

Some possible network grammars



Every bit as much as LAS, a child logically needs negative information to recover from overgeneralizations. The interesting question is where the negative information comes from in the case of the child. Parents do correct the child in such obvious morphemic overgeneralizations (Brown, 1973). Even today I find myself corrected (not by my parents) for my failures to properly pluralize esoteric words. The child may also use statistical evidence for a negative conclusion. In some manner he may notice that the morphemic form foots is never used by the adult and so conclude that it is wrong. Horning (1969) has formalized an algorithm for detecting such overgeneralizations by assigning probabilities to rules.

Figure 16 illustrates LAS's treatment of the last four sentences in the training sequences. These involve some three word noun phrases and also expansion of the noun phrases on the branch of the start network for RB relations. As can be seen from Figure 13, at the point of the 14th sentence LAS has expanded its grammar to the point where it will handle 616 sentences of the target language. 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 squares 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 so and thus UNDERSTAND can not make any mistakes. This is a nice example of how an over-general grammar can be successfully constrained by considerations of semantic acceptability.

The problem of learning to sequence noun modifiers has turned out to be a source of unexpected difficulty. In part, the ordering of modifiers is governed by pragmatic factors, For instance one is likely to say small red square when referring to one of many red squares, but red small square when referring to one of many small squares. Differences like these could be controlled by ordering of links in the HAM memory structure.

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 a 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.

Figure 16

Additions to LAS's grammar after studying:

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

$$\mathrm{C484} \xrightarrow{\varepsilon\,\mathrm{C510}} \mathrm{D713} \xrightarrow{\mathrm{D692}} \mathrm{STOP}$$
$$\mathrm{D713} \xrightarrow{\mathrm{NIL}} \mathrm{STOP}$$

$$\mathrm{B566} \xrightarrow{\varepsilon\,\mathrm{B593}} \mathrm{D1116} \xrightarrow{\mathrm{D1095}} \mathrm{STOP}$$
$$\mathrm{D1116} \xrightarrow{\mathrm{NIL}} \mathrm{STOP}$$

$$\mathrm{B564} \xrightarrow{\varepsilon\,\mathrm{B580}} \mathrm{D1044} \xrightarrow{\mathrm{D1023}} \mathrm{STOP}$$
$$\mathrm{D1044} \xrightarrow{\mathrm{NIL}} \mathrm{STOP}$$

$$\mathrm{D1023} \xrightarrow{\varepsilon\,\mathrm{D1045}} \mathrm{E1394} \xrightarrow{\mathrm{E1368}} \mathrm{STOP}$$
$$\mathrm{E1394} \xrightarrow{\mathrm{NIL}} \mathrm{STOP}$$

$$\mathrm{D1095} \xrightarrow{\varepsilon\,\mathrm{D1117}} \mathrm{E904} \xrightarrow{\mathrm{E884}} \mathrm{STOP}$$
$$\mathrm{E904} \xrightarrow{\mathrm{NIL}} \mathrm{STOP}$$

$$\mathrm{D692} \xrightarrow{\varepsilon\,\mathrm{D714}} \mathrm{STOP}$$

$$\mathrm{D1095} \xrightarrow{\varepsilon\,\mathrm{D1117}} \mathrm{STOP} \qquad \mathrm{D1023} \xrightarrow{\varepsilon\,\mathrm{D1045}} \mathrm{STOP}$$

$$\mathrm{E884} \xrightarrow{\varepsilon\,\mathrm{E905}} \mathrm{STOP} \qquad \mathrm{E1368} \xrightarrow{\varepsilon\,\mathrm{E1395}} \mathrm{STOP}$$

D714 = small
D1045 = red,blue,small
D1117 = blue,red
E905 = small,large
E1395 = large

A list is kept of all the networks created by SPEAKTEST. Once the structure of these networks becomes stable, GENERALIZE is called to determine which networks are identical. 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 $NOUN_1$
$\quad\rightarrow$ the ADJ_1 NP
$NP_1 \rightarrow NOUN_2$
$\quad\rightarrow ADJ_2\ NP_2$
$NP_2 \rightarrow NOUN_3$
$\quad\rightarrow ADJ_3\ NP_3$
$NP_3 \rightarrow NOUN_4$

That is, there are four networks, NP, NP_1, NP_2, and NP_3 whose structure is indicated by the above rewrite rules. It is assumed that LAS has only experienced three consecutive adjectives and therefore SPEAKTEST has only created 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 subnetwork of NP_2. So in the recursive identification of networks, GENERALIZE will have to accept a subnetwork 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 NP1 has been identified with NP2 HAM will have a new network structure where NP* represents the amalgamation of NP1, NP2, and NP3.

NP → the NOUN
the ADJ NP*

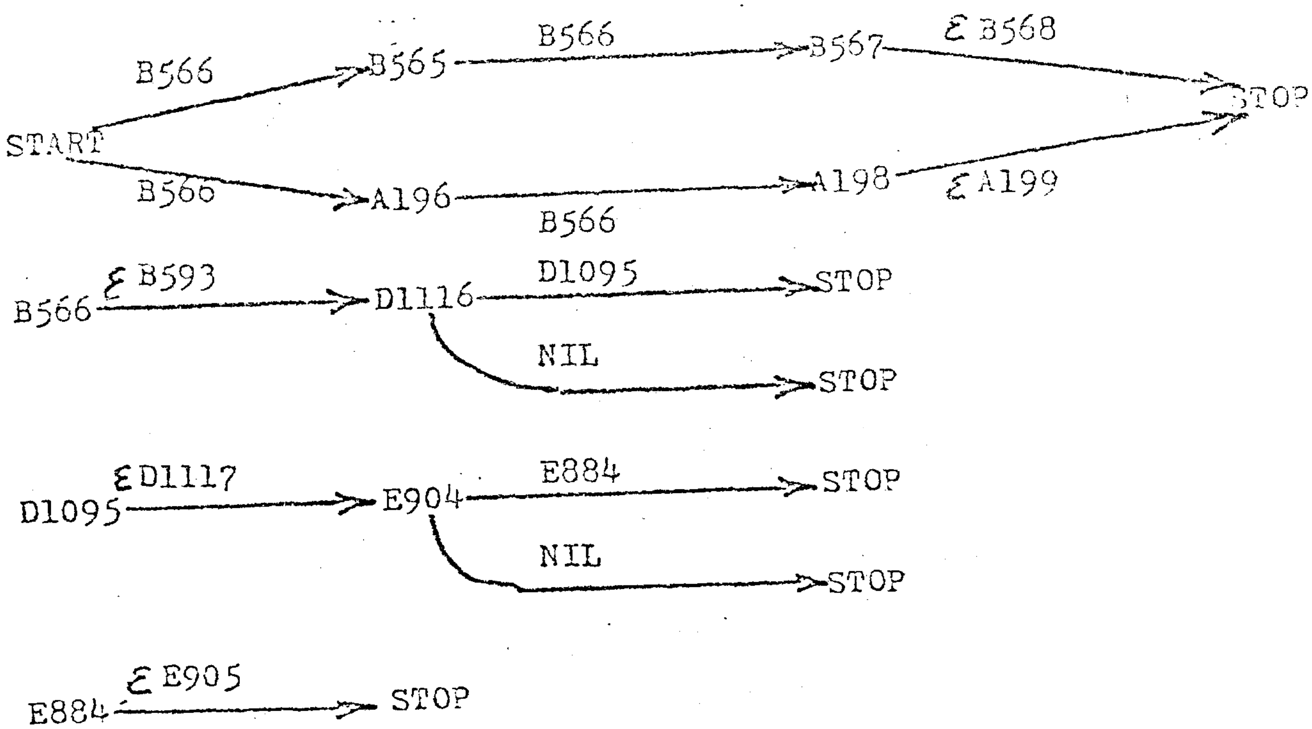
NP* → NOUN*
ADJ* NP*

Note that new word classes NOUN* and ADJ* have been created as the union of the word classes NOUN2, NOUN3, NOUN4 and of the classes ADJ2, ADJ3, respectively.

GENERALIZE was called to ruminate over the networks generated after the first fourteen sentences. GENERALIZE succeeded in identifying A195 with A197. As a consequence, network A195 replaced network A197 at the position where it occurred in the START network (see Figure 12). Similarly, B566 was identified with and replaced network B564. Finally, B566 was identified with and replaced A195 throughout the START network. The final effective grammar is illustrated in Figure 17. It now handles all the sentences of the grammar. It handles more sentences than the grammar that was constructed after the fourteenth

Figure 17

The final grammar



B568 = below,left-of
A199 = above,right-of
B593 = square,triangle
D1117 = blue,red,large,small
E905 = large,small

sentence. This is because the noun-phrase network B566 has been expanded to incorporate all possible noun phrases. Before the generalizations, none of the networks--B564, B566, A195, or A197 were complete. The network B566 became complete through merging with B564 and A195.

At this point, LAS now has a grammar adequate to speak and understand the target language. There are two major assumptions that LAS is making about the relation between sentence and referent which permit it success with these types of languages. The first is the assumption of the correspondence between the surface structure of the language and the semantic structure. This is critical to BRACKET's identification of the surface structure of the sentence which is, in turn, critical to the proper embedding of parsing networks. Second, there is the assumption of a semantics-induced equivalence of syntax. This played a critical role both in the generalization of SPEAKTEST and of GENERALIZE. It was noted with respect to pluralization that such generalizations can be in error and that children also tend to make such errors. However, I would want to argue that, on the whole, natural language is not perverse. Therefore, most of those generalizations will turn out to be good decisions. Clearly, for languages to be learnable there must be some set of generalizations which are usually safe. The only question is whether LAS has captured the safe generalizations.

The importance of semantics to child language learning has been suggested in various ways recently by many theoreticians (e.g., Bloom, 1970; Bowerman, 1973; Brown, 1973; Schlesinger, 1971; and Sinclair-de Zwart, 1973), but there has been little offered in the way of concrete algorithms to make explicit the contribution of semantics. LAS. 1 is a first small step to making this contribution explicit.

Conclusion

This concludes the explanation of the algorithms to be used by LAS.1 for language induction. In many ways the task faced by LAS. 1 is overly simplistic and its algorithms are probably too efficient and free from information-processing limitations. Therefore, the acquisition behavior of LAS. 1 does not mirror in most respects that of the child. Later versions of this program will attempt a more realistic simulation. Nonetheless, I think LAS.1 is a significant step forward. The following are the significant contributions embodied so far in LAS. 1.

1. The transition network formalism has been interfaced with a set of simple and psychologically realistic long term memory operations. In this way we have bridled the unlimited Turing-computable power of the augmented transition network.

2. A single grammatical formalism has been created for generation and understanding. Thus, LAS only needs to induce one set of grammatical rules.

3. Two important ways were identified in which a semantic referent helps grammar induction. These were stated as the graph deformation condition and the semantics-induced equivalence of syntax conditions.

4. Algorithms have been developed adequate to learn natural languages with a simple semantics.

B. Specific Aims

The general mode of developing the program LAS is as follows: A language learning situation is specified by a set of conditions. In LAS. 1 it was specified that LAS already know the meaning of the words and that it be given, as input, sentences with HAM representations of their meaning. The semantic domain was specified to be that constituted by geometric shapes. Once a set of conditions is specified, a set of goals is specified. In LAS. 1 there was only one real goal: to learn any natural-like language that described the domain. Once a set of goals is specified a plan of attack is sketched out. However, the problem is such that the details of that plan only evolve as we attempt to implement the plan as a computer program. Indeed many interesting problems and ideas that were not initially anticipated in LAS. 1 were discovered in attempting an implementation. This is part of utility of computer simulation in theoretical development.

The LAS. 1 program operated in a task domain which was similar, but by no means identical, to that of a natural language learning situation. Its behavior was similar to that of a human learning a language, but again by no means identical. In the next two years I propose to create a program LAS. 2 which comes considerably closer to simulating natural language learning. It has a more elaborate set of goals than did LAS. 1:

1. The program will incorporate realistic assumptions about short-term memory limitations and left-to-right sentence processing.

2. The program will learn the meanings of words.

3. The program should use semantic and contextual redundancy to partially replace explicitly provided HAM-encoding of pictures.

4. The program should handle sentences in a more complex semantic domain.

5. The program should be elaborated to handle such things as questions and commands as well as declarative sentences.

The general methods for achieving these goals in the LAS. 2 program will be sketched out in the proposal section. Also in that section I will propose some experiments to evaluate the LAS program. While it is true that the task faced by LAS. 1 is not really natural language learning, it still is a learning task at which human subjects apparently can succeed. The experiments will determine whether humans have the same difficulties in such tasks as does LAS and whether they make the same generalizations. However, I regard these experiments as of secondary importance relative to program development. It is more important to further articulate our understanding of what algorithms are adequate for natural language learning.

It is probably inevitable that the question will be asked as to whether it is really necessary to expend the resources necessary to construct a computer program. Could not the model just be specified conceptually? The reason why this is not possible has to do with the complexity of any theory that addresses the details of natural language. There is no other way to test the predictions of the theory or to assure that it is internally consistent. The experience with large transformational grammars hand-written for natural language is that they have hidden inconsistencies. These were only exposed by trying to simulate the grammars on a computer (e.g., Friedman, 1971). Consider the description given of LAS. 1 in the preceding section: Although lacking in many details, it was complex and lengthy. Could the reader establish for himself from this description whether the model is really internally consistent? A computer program provides a proof of the consistency and a means of determining the model's behavior. The stated goals of this project are to develop explicit algorithms for natural language learning, specify the relevant details of these algorithms, and evaluate empirically the psychological viability of these algorithms. Without the use of computer simulation none of these goals could be achieved.

C. Methods of Procedure

First I will describe the proposed extension of the LAS program. Then I will describe some experimental tests. In reading the specific extensions proposed for LAS, the reader should keep in mind that they have as their intent achieving the goals set forth in the preceding section.

The Semantic Domain

The first matter to settle upon in the new program is some semantic domain. The LAS. 1 world of shapes, properties, and geometric relations is too impoverished for further work. The following is proposed as a suggestion although there is nothing critical about its exact form. It is critical, however, that some semantic domain be chosen. It is only when there is a specified domain that an explicit goal for success in the program can be specified. The program will be regarded as successful if it can learn any natural language describing this domain.

I have chosen to look at a world close to that of a young child although there is perhaps nothing sacred about this domain. This world is set forth in Table 5. There are three people in this world. In addition to these there are four categories of objects--locations, containers, supporters, and toys. These objects can have four types of properties--number, color, size, and quality. Thus, LAS will have to deal seriously with problems of sequencing adjectives. It will also have to deal with number as a property of objects. The objects permit a much richer variety of relations than in the world of LAS. 1. This will provide a demanding test for the learning of complex multi-argument relations. There can be sentences like Mommy traded Daddy the car for a ball. In this world, people, containers, supporters, and toys can be in locations. People can change their location and that of toys. People and toys can be on supporters, toys can be in containers. People can possess toys, containers, and supporters.

TABLE 5

Categories in the World of LAS. 2

PEOPLE	LOCATIONS	CONTAINERS	SUPPORTERS	
Mommy	bedroom	box	table	
Daddy	kitchen	closet	chair	
LAS	den	dresser	bed	

TOYS	NUMBERS	COLORS	SIZES	QUALITIES
dolly	one	red	big	dirty
car	two	blue	medium	pretty
ball	three	green	small	shiny

Thus the different categories of objects enter differently into different types of relations. This fact will prove important to the predictive parsing facilities that I will want to introduce into LAS. 2.

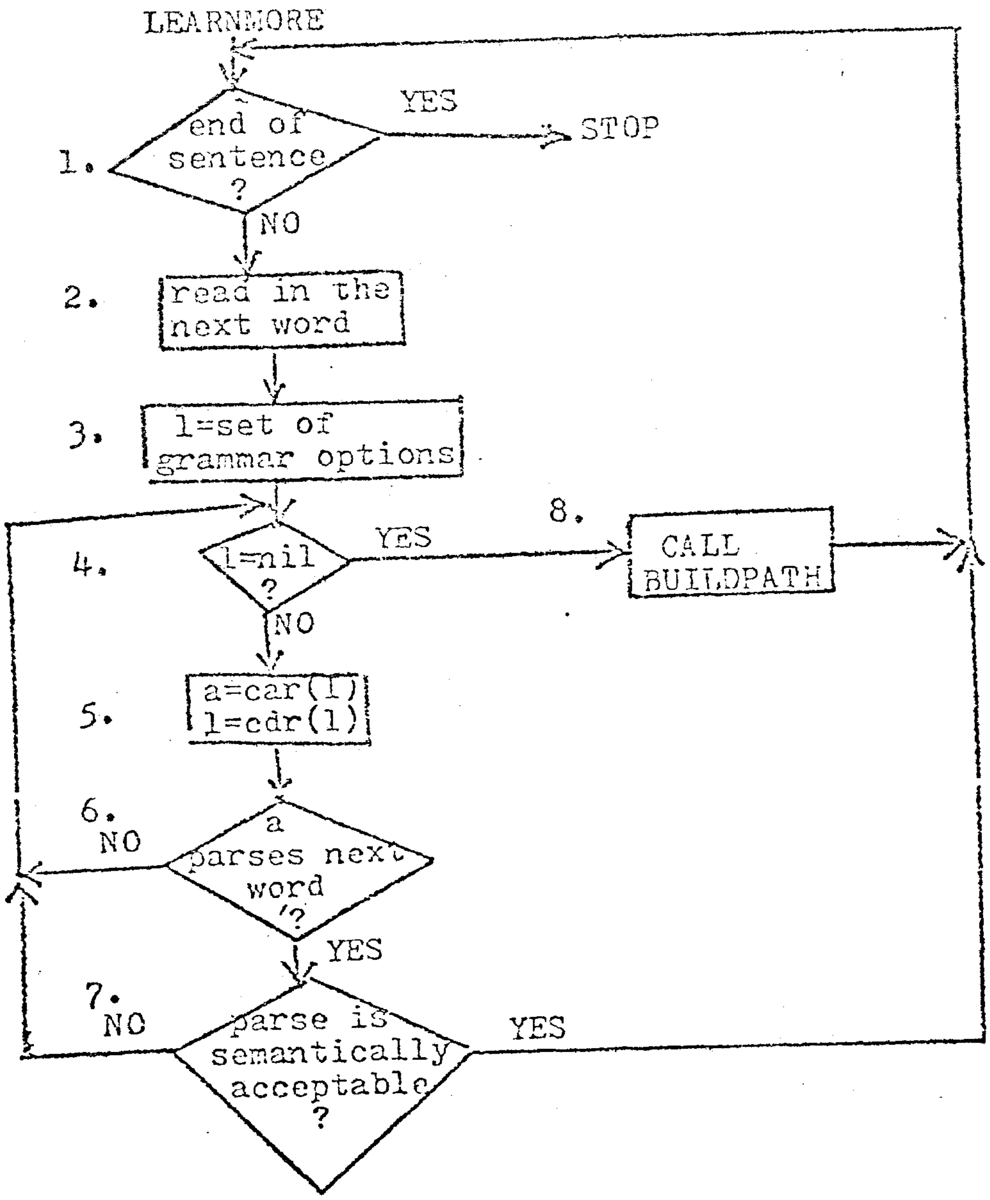
Left-to-Right Processing

Children learn language auditorily. Thus, their induction algorithms must process incoming material in a left-to-right manner. The current LEARNMORE program does not do this. BRACKET completely processes the sentence before SPEAKTEST even begins to work on it. Clearly, BRACKET and SPEAKTEST should be integrated so that the beginning of the sentence is bracketed and considered by SPEAKTEST before the end of the sentence is considered by either. Introducing this left-to-right processing is a preliminary to introducing short-term memory limitations into the induction situation.

Figure 18 illustrates in highly schematic form the left-to-right algorithm proposed for LEARNMORE. Words are considered as they come in from the sentence. LEARNMORE, as in UNDERSTAND, tries to find a path through its network grammar to parse the sentence. The difference between LEARNMORE and UNDERSTAND is that LEARNMORE has available to it a HAM conceptual structure to enable it to better evaluate various parsing options. Suppose LEARNMORE is at some point in processing the sentence. It will also be at some point in a parsing network. Let us consider how it would process the next word. At box 2 it would read in the word. At box 3 it would set l to the various grammatical options (arcs) at that node in the network. Boxes 4 through 7 are concerned with evaluating whether any of these options can handle the current word. Box 4 checks whether there are any options left. Box 5 sets a to the first option and resets l to the remaining options. Box 6 checks whether the word would be parsed by a and box 7 considers whether the action associated with that arc corresponds to a HAM structure. If a passes the tests in 6 and 7, LEARNMORE advances to considering the next word. Otherwise it tries another arc. If it exhausts all the arcs, it will call BUILDPATH (box 8) to build a new arc from the current node.

Figure 17

Flowchart for the proposed new LEARNMORE program



The work currently assigned to BRACKET will have to be assigned to box 7. That is, box 7 will have to determine when an arc should involve a push to an embedded network and when it should pop back up to an embedding network. This will be done by consulting the information in the semantic structure. It would also be possible to consult the pause structure of the sentence for information about phrase structure boundaries.

Note that certain sentences which the old LEARNMORE system could handle will not be handled by this system. For instance, consider the sentence The square that is above the triangle is right-of the square. After the first two words it would not be clear which square it was that we were referring to, the object or the subject of the right-of relation. Thus, buildpath could not assign an appropriate action to the path. In the old LEARNMORE this ambiguity about the referent of square was resolved by letting the whole sentence come in before dealing with it. Presumably, however, children would have difficulty learning from such sentences.

Lexicalization

In this system it will not be assumed that LAS knows the meaning of the words. Rather this will be something that LAS will have to learn from the pairing of sentences with conceptions. First let's discuss the learning of words whose reference is a simple concept or object, e.g., box or mommy, and postpone discussion of complex relational terms like trade. Logically, the task of lexicalization is quite simple and it would not require complex algorithms to succeed. For instance, consider this algorithm: LAS is given a sentence with n_1 words and a conceptualization it describes with m_1 concepts. Store with each word the m_1 concepts. The next sentence that comes has n_2 words and its conceptualization consists of m_2 concepts. If a word in this sentence is new, store with it the m_2 concepts. If the word is old, store with it the intersection of the concepts previously stored with it and the new m_2 concepts. Eventually, ignoring problems of polysemy, a word will become pared down to zero or one concepts. Those with zero concepts are function words and those with one concept have that concept as their meaning.

Of course, this algorithm will run into trouble if LAS does not always conceptualize all the concepts referred to by the sentenee. This can be remedied by having the algorithm wait for a sequence of disconfirming pieces of evidence before rejecting a hypothesized meaning. Incidentally, subjects behave just this way in concept attainment situations (see Bruner, Goodnow & Austin, 1965), not taking negative evidence as having its full logical force about the meaning of the word.

The basic problem with this algorithm is that it makes unreasonable assumptions about the information processing capacities of humans. In pilot research of my own, I have found that adult subjects can learn the meanings simultaneously of a number of words in a sentence. However, they do suffer difficulties when there is high ambiguity about what a word means. Presumably, children would have even greater difficulties extracting word meanings from complex sentences. Broen (1972) and Ferguson, Peizer, & Weeks (1973) report that new items of vocabulary seemed to be introduced through use in set sentence frames such as Where's ..., Here comes ..., There's ... known as deitic phrases. The noun tends to be heavily stressed and repeated. The parent frequently points to help reduce possible ambiguity of reference.

Presumably, later in lexicalization words can be learned by appearing in more complex sentence frames, provided the child knows most of the words and the grammatical structure of the sentence. To combine these various considerations, I propose the following addition to the flow chart in Figure 18 to deal with the reception of words with unknown meaning. In box 2, when an unknown word is read in, LEARNMORE will make a guess about its meaning using knowledge about context and about the word's position in the grammar. It will commit this guess to memory and stick with the guess unless later disconfirmed. The program will only hazard a guess in circumstances of low uncertainty. Thus, it will only guess if it can otherwise parse the grammatical structure in which the word appears. It will not guess if the word is preceded or followed by other words it does not know. Thus, the program, much as adults appear to, will learn on the basis of minimal contrasts between grammatical pattern and a current sentence. Thus, if the program knows the grammatical rule NP → determiner adjective noun and encounters the phrase the glick box it will suppose that glick refers to some property of the box.

Thus, the program will have to acquire its initial vocabulary by means of simple frames, as do young children. With this initial vocabulary information, it can begin to learn grammatical rules. Once in possession of grammatical rules, it will no longer need simple frames to learn new lexical items.

One interesting question is how function words are ever identified as non-meaning-bearing in this scheme. Presumably, this is done on the basis of failing to obtain a constant correlation between the word and any semantic feature. This could be detected by noting how many mistaken guesses had been associated with a word.

Concept Identification and Relation Words

So far I have assumed that all concepts are constructed before language acquisition takes place and that the only problem is to link up these concepts with words. But this is very unrealistic. Consider the verb give in the sentence Mommy gives the dolly to Daddy. The meaning of give is something like to do something which causes one to cease to possess an object and someone else to begin to possess the object. It seems very implausible that a child comes into a language learning situation with such a concept ready made. What probably happens is that he sees Mommy pushing the doll to Daddy or Mommy handing the ball to baby. With these experiences he hears sentences like Mommy gives the dolly to Daddy or Mommy gives the ball to baby. From these examples he induces the appropriate meaning of give. Concept attainment in these situations can be achieved by using the sort of concept identification used by Winston (1970) for inducing geometric concepts. That is, each use of the word give is paired with a HAM network structure given the meaning of the sentence. Winston's heuristics allow us to extract what these network structures paired with give have in common. The concept give, as verb, is then attached to the common structure. For this sort of algorithm to succeed, LAS must be set to regard certain configurations of propositions, interlinked by causal terms, as being associated with a single relational term in the language.

Note also that the effect of such an induction scheme would be to encode the meaning of complex relational terms into the network grammar. That is, in parsing the sentence Mommy gives the dolly to LAS, the network would specify that UNDERSTAND set up, as the meaning of the sentence, a HAM representation of the form Mommy does something which causes her to cease to possess the doll and LAS to begin to possess the doll.

BADEAR--The Telegraphic Perception Hypothesis

The clearest discrepancy between the behavior of LAS. 1 and a child is that LAS. 1 generates no ungrammatical sentences. In contrast, at first the child only generates ungrammatical sentences. The child's early speech has been crudely characterized as telegraphic. That is, children 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. One explanation of the origin of telegraphic speech which is appealing from the point of view of LAS is the following: Suppose that LAS did not receive as input to its induction routine complete sentences, but rather telegraphic sentences. Then, it would quite naturally induce a telegraphic grammar. It seems reasonable to suppose that a child cannot hold in immediate memory the total sentence he has heard but rather only a depleted version of that sentence. If so, then his induction algorithms would be receiving telegraphic sentences as their basic data. Let's call this the telegraphic perception hypothesis.

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). Blasdell 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. Scholes (1969, 1970) found that children 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.

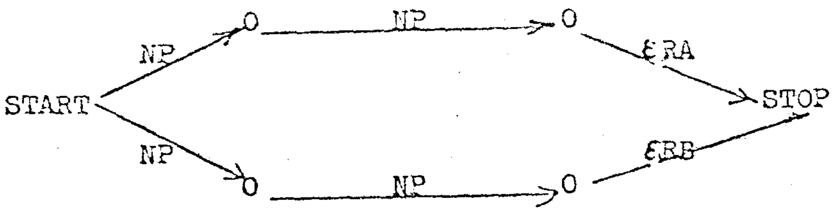
I propose to introduce telepraphic perception into LAS through an aspect of LEARNMORE called BADEAR. The BADEAR program will simulate the variables of stress, meaningfulness, and serial position in providing LAS with a depleted version of the sentence. The locus of the effect of BADEAR will be between boxes 4 and 8 in the flowchart of Figure 2. Basically it will not pass all words onto BUILDPATH. Rather some words will "slip from consciousness" after failing to be parsed. It will tend to omit words when: (a) they are unstressed, (b) their meaning is not known, (c) a critical number of new words in the sentence have already been passed to BUILDPATH. I suspect this critical number is something like one or two.

Factors (a) and (b) would generate the effects of stress and meaningfulness. Factor (c) would yield good memory for the first words of the sentence. What good memory children do show for last words in phrases probably reflects short-term acoustic memory.

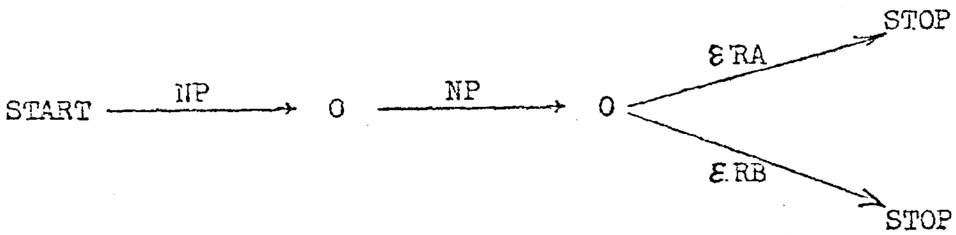
An interesting feature of BADEAR is that, as the grammar is expanded, LAS would be able to receive more of the sentence. Thus, its productions and imitations would grow as does a child's. This would be providing an explicit mechanism for an idea suggested by Braine (1971), Olson (1973, and others. Inducing a grammar from degenerate sentences presents an interesting problem. How is it that LAS ever comes to abandon its rules for generating telegraphic speech? Merely because LAS has learned rules for generating fuller sentences, it does not follow that the old rules are wrong. After all, language permits multiple means for expressing the same thoughts. Perhaps then mechanisms should be incorporated that will strengthen some grammatical rules relative to others. Rules to be strengthened would be those that could be successfully used by UNDERSTAND and that could successfully be used by SPEAK. We might think that the arcs out of a node in a parsing network are ordered on a stack to reflect their relative utilities. Subjects would try rules on the top of a stack first. Ineffective rules like the original ones for two and three word utterances would descend to the bottom of the stack and so become unavailable. This strength mechanism is the same as used to order links in the HAM memory model. This is a different way to bring negative information to bear in grammar induction than that proposed for RECOVER. That is, rather than seeking explicit disconfirmation of rules, the rules are gradually weakened out of existence as more adequate rules take over the roles the old rules used to occupy in sentence understanding and generation.

Grammar Optimization

Note that the START network induced by LAS was one with the following form:



This grammar requires considerable backup if the sentence does not have an RA relation. As suggested earlier it would be more efficient if LAS were given the power to transform the grammar into the following form:



Given that there are serious time problems (see introduction of proposal) in parsing, it is critical that methods be incorporated in the learning program for optimizing the grammar. The merging of arcs, besides making the grammar more efficient, would be another form of generalization. It could be used to further merge and build up word classes.

Further Use of Semantics in Language Acquisition

There are at least two further ways that semantics can be used to aid language acquisition in addition to those embodied in LAS. 1. One concerns using conceptual information as a further aid to word class formation. Words in a particular class tend to have a common semantic core. LAS could use this fact to adjust its threshold for merging words into a class. For instance, suppose LAS considers merging two word classes because they share certain syntactic properties, both of which contain color names. Currently, LAS. 1 makes this decision on the basis of the amount of overlap between the members of the two classes. LAS should lower its overlap threshold because these word classes do share a strong semantic property.

Another use of semantics would be to lessen LAS's reliance on explicitly given semantic interpretations of sentences. It should sometimes be able to guess these interpretations. For instance, suppose a sentence came in with the words ball, box and in. Because of the conceptual constraints between these, LAS should be able to guess their connection. This use of conceptual constraints in the semantic domain could also be used by UNDERSTAND to permit predictive parsing along the model of the Schank's (1972) system. That is, as an alternate to understanding a sentence by use of syntactic information, it is possible to look for conceptual constraints to predict what the interpretation of the sentence should be. This prediction can then be checked for syntactic correctness by use of the network grammar. It would be profitable to try to place a predictive parsing system like Schank's within the rigors of the Woods' network formalisms.

A Procedural Semantics

So far LAS has been principally concerned with representing the meaning conveyed by a declarative sentence. However, language has other purposes than just to communicate meanings from one speaker to another. Consider commands and questions. For instance, consider the sentence Put the dolly in the box. Currently, UNDERSTAND might retrieve the sentence's meaning as Speaker requests of LAS that it put the dolly in the box. This is the declarative meaning of the sentence. However, in addition LAS should evoke an action that causes it to comply or at least take an action to decide whether to comply. This is the procedural meaning of the sentence. The procedural meaning of declarative sentences is very simple: store this sentence. This is already assumed in LAS's treatment of the sentence. However, the procedural meanings underlying other types of sentences are more complex. A large part of the success of Winograd's system is that it was adequately able to deal with the procedural aspects of various sentences' semantics. It is important that LAS begin to deal with these too.

What this would mean, in terms of LAS's network grammars, is enriching the set of actions that can be stored. Currently, the only actions are ones that result in the creation of pieces of HAM structure, i.e., declarative knowledge. LAS will have to store other internal actions that specify what it does with the declarative knowledge. These will include commands to answer the question or obey the order. HAM already has commands that direct it to answer questions but executing orders would be something new. As part of the HAM project, I am working on methods for incorporating procedural knowledge into a network system. It is unclear yet what success I will have here.

It is interesting to note other aspects of natural language whose semantics are procedural. These are well documented in Winograd. Consider for instance the difference between the definite and indefinite article--the red ball versus a red ball. The former indicates an object which the listener knows. Thus the listener's response to the definite article should be to search his memory for the referent of the noun phrase. In contrast, the listener's response to an indefinite noun phrase should be to construct a new representation for it. This difference can be nicely handled in the current HAM system by whether a call to the MATCH routine is evoked.

Winograd has argued convincingly that the semantics of pronouns and other indexicals should be represented by procedures to determine their referents. This is particularly true for terms like you whose meaning is totally relative to speaker and context. Since the referent of you completely changes with speaker, a child would be lost if he tried to associate its meaning with some HAM memory node. He must be prepared to treat it as having as meaning a procedure for determining the referent.

Provided that LAS has the facilities for representing and evaluating procedures, there seem no difficulties in learning those aspects of language which are heavily embued with procedural semantics. Language learning will continue to arise from pairing sentences with semantic interpretations. However, semantic interpretations will now contain a procedural as well as a declarative aspect. Again language learning will consist of learning mappings between sentences and the now-enriched semantic representations.

Experimentation

As stated before, I do not think that experimental research should yet be the principal focus of the project. There is still much further research that needs to be done in the way of specifying algorithms that are capable of language induction. Nonetheless, in parallel with this research, I would like to perform experiments to get some initial assessments of the viability of the proposed algorithms. The type of information relevant to evaluating LAS is only acquired by looking at artifical languages. With these artificial languages it is possible to test LAS's predictions about language learnability and generalization.

Criticisms of Experiments with Artificial Languages

For ethical reasons it is not possible to expose young children, just learning their first language, to an artificial language which LAS had identified as degenerate and probably not learnable. This means that all experimentation with artificial languages must be done on older children already well-established in their first language or on adults. Consequently, the first language may be mediating acquisition of the second language. There is evidence (see Lennenberg, 1967) that there is a critical initial period during which languages can be learned much more successfully than in later years. Lennenberg speculates that there is a physiological basis for this critical period. Thus, one might wonder whether the same processes are being studied with older subjects as in the young child. Personally, I also doubt that the mechanisms of language-acquisition are the entirely same with the young child in first language learning as with the older subject in second language learning. However, it does seem probable that there should be considerable overlap in the mechanisms for the two situations. The reason for this belief has already been stated: Both for the adult and the young child, language acquisition presents largely the same set of severe and unique information-processing demands. The algorithms that deal with induction problems therefore are probably not very different in any system that successfully learns the language.

Other criticisms (e.g., those of Slobin, 1971; Miller, 1967) of studies of artificial language learning focus on the fact that these languages are artificial. Natural language is much more complicated than an artificial laboratory language; it takes years to acquire; it serves more complex functions; the child's motivations are more complex than the laboratory subject. However, these criticisms miss the whole point of laboratory experimentation which is to isolate and study significant aspects of a complex natural phenomena. Another criticism of the past artificial languages studies (e.g., those studies of Braine, 1963b; Miller, 1967; Reber, 1969) is that they lack a semantic referent. Clearly, this makes an enormous difference to the sort of algorithms a subject can employ. The critical heuristics used by LAS would be useless without semantics. Moesser and Bregman (1972, 1973) have shown that the existence of a semantic referent has a huge effect on language acquisition. Except for control conditions, all of my experiments will involve a semantic referent.

LAS's Predictions about Language Learnability

Critical to LAS's induction algorithm is that the graph deformation condition be met concerning the relation between the surface structure of the sentence and the HAM conceptual structure. That is, the surface structure must preserve the original connectivity of concepts. In Section A5 we described languages which violated this assumption. Consider the following language:

S → NP NP relation
NP → noun (Color) (adjective) (clause)
CLAUSE → te NP relation
NOUN → square, circle, triangle, diamond
Color → red, blue
Size → small, large
Relation → above, below, right-of, left-of

This is an expanded version of GRAMMAR1 described in Table 1. (The element te serves the function of a relative pronoun like that.) An example of a sentence in this language is Square red te triangle big above circle blue small right-of. An experiment I will do 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 will be presented along with the sentences. However, the relationship between the sentence's semantic referent and the surface structure will violate LAS's constraints. The adjective associated with the ith noun phrase will modify the $(n + 1 - i)$th shape in the sentence (where n is the number of noun phrases). For example, the adjectives associated with the first noun phrase will modify the last

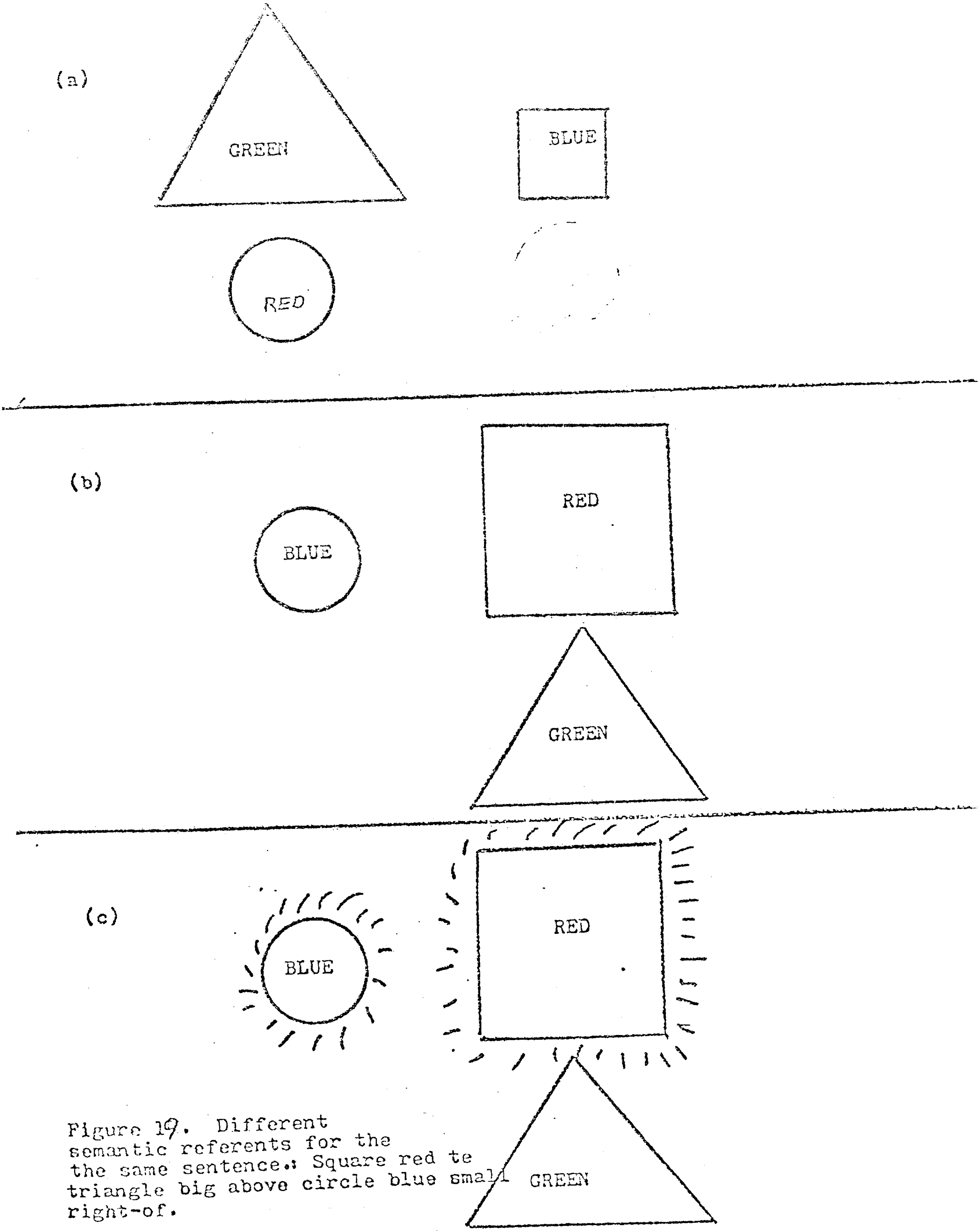


Figure 19. Different semantic referents for the the same sentence.: Square red te triangle big above circle blue small right-of.

shape. Similarly, the ith relation will describe the relation between the (m + 1 - i)th related pair of shapes (where m is the number of relations). So for instance the second relation right-of will describe the relationship between the first pair of shapes square and triangle. The appropriate picture for the example sentence is given in Fiuure 19a.

3. Good semantics. Here the adjective in each noun phrase will modify the noun in that phrase.. Relations will relate the appropriate nouns in the surface structure. The appropriate picture for the example sentence in this case is given in Figure 12b. LAS could bracket sentences given this picture if it could guess the main proposition.

4. Good semantics plus main proposition. The picture in this condition will be the same as in 3 but the two shapes in the main proposition will be highlighted. In this condition LAS would be guaranteed of successfully bracketing the sentence because the main proposition is given.

In some ways this experiment is like Moesser and Bregman's. 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. If English words were replaced by nonsense syllables this would require a simplification of the language to make induction tractible. The predictions of LAS are, of course, that best learning occurs in Condition 4, next best in 3, and failure of any learning in 1 and 2. It would not be surprising to see subjects perform better in 1 than in 2 since in they might partially be able to imagine an appropriate semantics.

The procedure would have subjects in all conditions study the same sequence of sentences but vary the accompanying semantic information according to condition. After a study phase they would be tested for grammaticality judgments about a set of sentences, some of which violate one of the rules for generation. Since the syntax of the language is the same in all four conditions, the same sentences will be grammatical in all four conditions. Even though the syntactic information given during study will be the same in all conditions, marked differences in syntactic knowledge should appear across conditions. The current plan is to alternate sequences of study trials with sequences of test trials, so the subject might study six sentences, with the semantic information (appropriate to his condition, if any). Then he would see six test pairs, one sentence of each pair violating some syntactic rule. For each pair of he would have to choose the grammatically correct pair. By frequently alternating study and test, it would be possible to carefully monitor the growth of information in the conditions.

Many readers may not be surprised by the prediction of better learning in Conditions 3 and 4. Hopefully, the significance of such an outcome would be clear. It would show that semantics is important to induction of the syntactic structure of a natural language. However, it would also show that semantics is useless if the relation between the semantic referent and the syntactic structure is arbitrary. The surface structure of the sentence must be a graph-deformation of the underlying semantic structure. Failures to appreciate the contribution of semantics to language induction and failure to understand the nature of this contribution of semantics to the induction process have been fundamental in the stagnation of attempts to understand the algorithms permitting language induction. These facts may be obvious when pointed out but they have been unavailable to the linguistic theorists for fifteen years.

There are other experiments of this variety which can be done to see how well humans can learn languages which do or do not meet the constraints demanded by LAS's induction algorithms. These constraints have the same purpose as Chomsky's (1965) proposals for linguistic universals. That is, they constrain the set of possible hypotheses about language structure so that the target language can be identified. However, the constraints used by LAS are not the same as those suggested by Chomsky. For instance, Chomsky proposed that transformations which reversed the order of words in a sentence would be unacceptable. This is because such a rule does not refer to the sentence's constituent structure. However, a language which contained sentences of a natural language and their reversals would be learnable by LAS. It would just develop one set of rules for sentences in one order and another independent set for reverse order sentences. It would be interesting to see whether human subjects could learn such a language.

In the example of the induction of GRAMMAR1 we found that there was no way for LAS to detect non-semantic contingencies between syntactic choices in the first noun-phrase and in the second noun-phrase pushed to in the main network. For instance, it is possible that a morphemic embellishment of the adjectives in the second noun phrase may depend on a choice of morphemic embellishment for the noun in the first noun phrase. Human subjects should also find it hard to detect such syntactic contingencies.

Predictions about Generalization

There are another set of predictions, besides those concerned with language learnability, which it will be useful to explore. LAS makes predictions about the situations under which humans will tend to generalize rules and when humans will not. Suppose LAS learned the following grammar:

S → VERB NP NP
NP → (PREPP) N_1 (ADJ)
PREPP → PREP N_2
N_1 → boy, girl, etc.
N_2 → room, bank, etc.
ADJ → tall, nice, etc.
PREP → in, near, etc.
VERB → like, hit, etc.

A typical sentence in this language would be Like in room boy tall girl nice which means The tall boy in the room likes the nice girl. This language is given English terms only to make its semantics clearer. Suppose, in fact, words in the language were das meaning man, jir meaning woman, fos meaning boy, and tuk meaning girl. Suppose the subject studies the following pair of sentences:

1. Like das tuk.
2. Like fos jir.

Then, it is interesting to consider his judgments of the acceptability of sentences like:

3. Like das tuk.
4. Like das jir.
5. Like jir das.

Accepting (3) only involves recalling sentence (1), but accepting (4) would involve a generalization: LAS would currently make this generalization because it would merge das and fos into a single word class and it would similarly merge tuk and jir. If the subject accepted (5) he would be making a more interesting generalization not currently predicted by LAS. He has never encountered jir in the first noun slot or das in the second noun slot. Nonetheless, he assumes they are acceptable in these positions on the basis of their semantic similarity to words which are in these classes.

Neither (4) nor (5) need be acceptable sentences. The words jir and das could, for instance, take a different case inflection when they appear in different slots. This would make (5) unacceptable. Sentence (4) could be unacceptable because jir took a different morphemic embellishment when preceded by das. It would be interesting to see how learnable a language would be that contained such violations of the potential generalizations.

One can explore other questions about generalization in this artificial language. Suppose a subject studied sentences like (6). Would he accept sentences like (7)?

6. Like in room boy tall girl
7. Like girl in room boy tall

That is, will rules generalize from the subject noun phrase to the object noun phrase. As LAS is currently constituted such generalizations would not occur until it had built up fairly stable noun phrases. Again suppose LAS had initially only encountered simple sentences such as (8):

8. Like boy man

From sentences such as (8) LAS would learn the class of nouns that occurred in first and second noun phrase slots. Suppose then sentence (9) was studied. On the basis of it, would sentence (10) be accepted as grammatical? That is, would the prepositional phrase in bank generalize to other nouns in the same class as woman?

9. Like boy in bank woman
10. Like girl in bank man

This would be an example of right generalization which does not occur in LAS. In contrast, LAS does perform left generalization. That is, after studying (11) LAS would accept (12).

11. Like boy woman nice
12. Like boy man nice

It will be interesting to see if humans show any preference for left generalization over right generalization.

It is critical that these artificial language experiments be done with a number of age groups, from young children (e.g., ages 4 and 5) to adults. While one can never really study first language acquisition with artificial languages, it is important to get an appreciation of what the developmental trends are. Since young children cannot handle written languages, much of this language training will have to be done with auditory presentation of the to-be-learned language. There has been little work done on artificial language learning by such young children, so probably much pilot research will be necessary to establish workable procedures.

D. Significance

LAS is a program with two purposes, one concerned with psychology and one concerned with artificial intelligence. I think this mixed purpose is fruitful because it promotes a cross-fertilization of ideas from two fields and so helps prevent theoretical stagnation. There is no guarantee that LAS, in the broad outline currently conceived, will ever achieve the goal of an adequate simulation of a child's acquisition of language. However, a certain outcome of this will be a clearer understanding of the information-processing demands of language-acquisition and of the role of a semantic referent in grammar induction. If LAS fails we will learn what is wrong with one explicit set of induction algorithms. Even that would be a significant contribution to the current theoretical development in a field rich in data but almost totally lacking explicit information-processing theories. I hope, of course, that the processes uncovered in the LAS project will be the same as those used by humans in language learning. A successful simulation program would constitute an enormous advance in our understanding of cognitive development.

The contributions of LAS to the artificial intelligence field are less certain and more distant. Nonetheless, generality in language understanding systems is an important goal and one for which a learning system approach seems ideal. It is therefore important to understand the contribution language learning systems can make in this field. It would be a significant advance to know in detail why a learning system approach was not the answer to language understanding or at least why LAS was not the right sort of learning system. Of course, if LAS does prove to be the basis for a viable language understanding system, its contribution to artificial intelligence will also be of considerable importance.

E. Facilities Available

I shall have available the entire facilities of the Human Performance Center, University of Michigan. My current appointment expires June 30, 1976, but can be extended for one to three years. My principal resource will be the Michigan Terminal System which supports a rich variety of programs. Most of the programming will be performed in Michigan LISP (see Hafner & Wilcox, 1974) which is a relatively economical and an error-free version of LISP.

REFERENCES

ALPAC (Automated Language Processing Advisory Committee). Language and machines - computers in translation and linguistics. National Academy of Sciences, Washington, D. C., 1966.

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.

Bar-Hillel, Y. Language and information. Reading, Mass: Addison-Wesley, 1964.

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

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

Bierman, A. W. and Feldman, J. A. A survey of results in grammatical inference. In (Ed.), Frontiers of pattern recognition. New York: Academic Press, 1972.

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

Bloom, L. One word at a time. The Hague: Mouton, 1973.

Bobrow, D. G. A question-answering system for high school algebra word problems. AFIPS Conference Proceedings, 1964, 26, 577-589.

Bowerman, M. Early syntactic development. Cambridge, England: University Press, 1973.

Boyer, R. S. Locking: A restriction of resolution. Ph.D. Dissertation. University of Texas at Austin, 1971.

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

Braine, M. D. S. On two types of models of the internalization of grammars. In D. J. Sloben (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 & Hearing Association, 1972, 17.

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

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.

Bruner, J. S., Goodnow, J., and Austin, G. A. A study of thinking. New York: Wiley, 1956.

Charniak, E. Computer solution of word problems. Proceedings of the International Joint Conference on Artificial Intelligence. Washington, D. C.: 1969, 303-316.

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

Colby, K. M. and Enea, H. Inductive inference by intelligent machines. Scientia, 103, 669-720 (Jan. - Feb. 1968).

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

Coles, L. S. Talking with a robot in English. Proceedings of the International Joint Conference on Artificial Intelligence. Washington, D. C. 1969, 587-596.

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, 1970.

Dreyfus, H. L. What computers can't do. New York: Harper and Row, 1972.

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

Feldman, J. A. Some decidability results on grammatical inference and complexity. A. I. memo No 93.1, Computer Science Department, Stanford University, 1970.

Fergusen, C. A., Perzer, D. B., & 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.

Fikes, R. E., Hart, P. E. and Nilsson, N. J. Some new directions in robot problem solving. Stanford Research Institute, August, 1972.

Fillmore, C. J. The case for case. In E. Bach and R. J. Harms (Eds.), Universals in linguistic theory. New York: Holt, Rinehart and Winston, 1968.

Friedman, J. A computer model of transformational grammar. American Elsevier, 1971.

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

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

Green, C. D. and Raphael, B. Research on intelligent question answering systems. Proceedings of the ACM 23rd National Conference. Princeton: Brandon Systems Press, 1968, 169-181.

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.

Kasher, A. Data retrieval by computer: A critical survey. Hebrew University of Jerusalem, January 1966. Technical Report No 22 to Office of Naval Research, Information Systems Branch.

Kelley, K. L. Early syntactic acquisition. P-3719, the Rand Corp., Santa Monica, California, 1967.

Kelloggs, C. H. A natural language compiler for on-line data management. Proceedings of the 1968 Fall Joint Computer Conference, 473-492.

Kuno, S. The predictive analyzer and a path elimination technique. Communications of the ACM, 1965, 7, 453-462.

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.

Lindsay, R. K. Inferential memory as a basis of machines which understand natural language. In E. A. Feigenbaum and J. Feldman (Eds.), Computers and thought. New York: McGraw-Hill, 1963.

Loveland, D. W. A linear format for resolution. Proceedings of the IRIA Symposium on Automatic Demonstration. New York: Springer-Varlay, 1970, 147-162.

Luckham, D. Refinemants in resolution theory. Proceedings of the IRIA Symposium on Automatic Demonstration. New York: Springer-Varlay, 1970, 163-190.

Miller, G. A. The psychology of communication. New York: Basic Books, 1967.

Minsky, M. (Ed.), Semantic information processing. Cambridge, Mass.: MIT Press, 1968.

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.

Moore, E. F. Gedanken experiments on sequential machines. Automata Studies, Princeton, 1956.

Olson, G. M. Developmental changes in memory and the acquisition of language. In T. E. Moore (Ed.), Cognitive development and the acquisition of language. New York: Academic Press, 1973.

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

Quillian, M. R. The teachable language comprehender. Communications of the Association for Computing Machinery, 1969, 12, 459-476.

Reber, A. S. Transfer of structure in synthetic languages. Journal of Experimental Psychology, 1969, 81, 115-119.

Richards, I. A., Jasuilko, E. and Gibson, C. Russian through pictures, Book I. New York: Washington Square Press, 1961.

Robinson, J. A. A machine-oriented logic based on the resolution principle. Journal of the ACM, 1965, 12, 23-41.

Rumelhart, D. E., Lindsay, P, and Norman, D. A. A process model for long-term memory. In E. Tulving and W. Donaldson (Eds.), Organization of memory. New York: Academic Press, 1972.

Saporta, S., Blumenthal, A. L., Lackowski, P. and Reiff, D. G. Grammatical models of language learning. In R. J. DePietro (Ed.), Monograph Series on language and linguistics. Vol. 16. Report of the 14th Annual Round Table Meeting on Linguistic and Language Studies, 1963, 133-142.

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

Schlesinger, I. M. Production of utterances and language acquisition. In D. I. Slobin (Ed.), The ontogenesis of grammar. New York: Academic Press, 1971.

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.

Schwarcz, R. M., B rger, J. F., and Simmons, R. F. A deductive question-answerer for natural language inference. Communications of the Association for Computing Machinery, 1970, 13, 167-183.

Shamir, E. and Bar-Hillel, Y. Review 2476. Computing Reviews, 1962, 3, 5.

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

Simmons, R. F. Natural language question-answering systems: 1969. Communications of the Association for Computing Machinery, 1970, 13, 15-30.

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 Francisco: Freeman, 1973.

Simon, H. A. The sciences of the artificial. Cambridge, Mass.: MIT Press, 1969.

Sinclair-deZwart, H. Language acquisition and cognitive development. In T. E. Moore (Ed.), Cognitive development and the acquisition of language. New York: Academic Press, 1973.

Slagle, J. R. Experiments with a deductive question-answering program. Communications of the Association for Computing Machinery, 1965, 8, 792-798.

Slagle, J. R. Automatic theorem proving with renamable and semantic resolution. Journal of the ACM, 1967, 14, 687-697.

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

Solomonoff, R. J. A formal theory of inductive inference, Part II. Information and Control, 1964, 7, 224-254.

Weizenbaum, J. ELIZA - a computer program for the study of natural language communications between man and machine. Communications of the ACM, 1966, 1, 36-45.

Wilks, Y. The Stanford MT and understanding project. In Schank and Colby (Eds.), Computer models of thought and language. San Francisco, 1973.

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

Winston, P. H. Learning structural descriptions from examples. _MIT Artificial Intelligence Laboratory Project AI-TR-231_, 1970.

Woods, W. A. Procedural semantics for a question-answering machine. _Proceedings of the 1968 Fall Joint Computer Conference_, 457-471.

Woods, W. A. Progress in natural language understanding: An application to lunar geology. _AFIPS Proceedings, 1973 National Computer Conference and Exposition._

Woods, W. A. Transition network grammars for natural language analysis. _Communications of the ACM_, 1970, _13_, 591-606.