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.

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 time 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 m syntactic and semantic interpretations where m on the average may be as high as 10. If there are n words, m^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 m facts in the data base and the desired deduction is n steps long. Then, there is something like m^n possible combinations of facts to achieve the desired deduction. This suggests that very deep inferencing (i.e., high 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 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 fundamentally 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

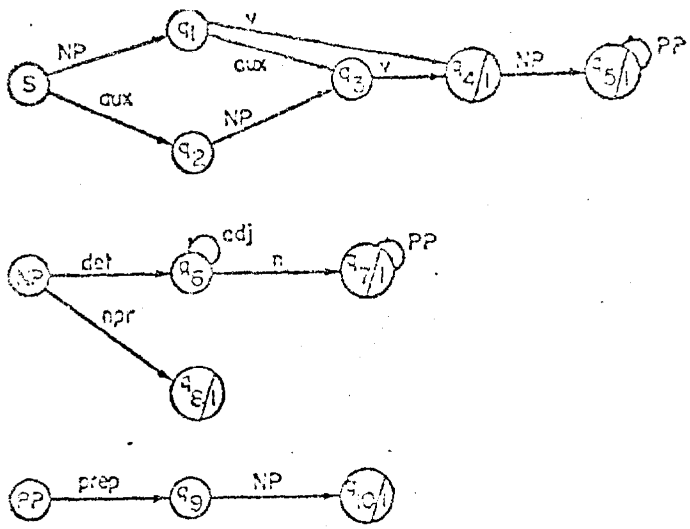

FIG. 1. A sample transition network. S is the start state. q_4, q_5, q_6, q_7, q_8, and q_{10} are the final states. (From Woods, 1970.)

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 grammar 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.

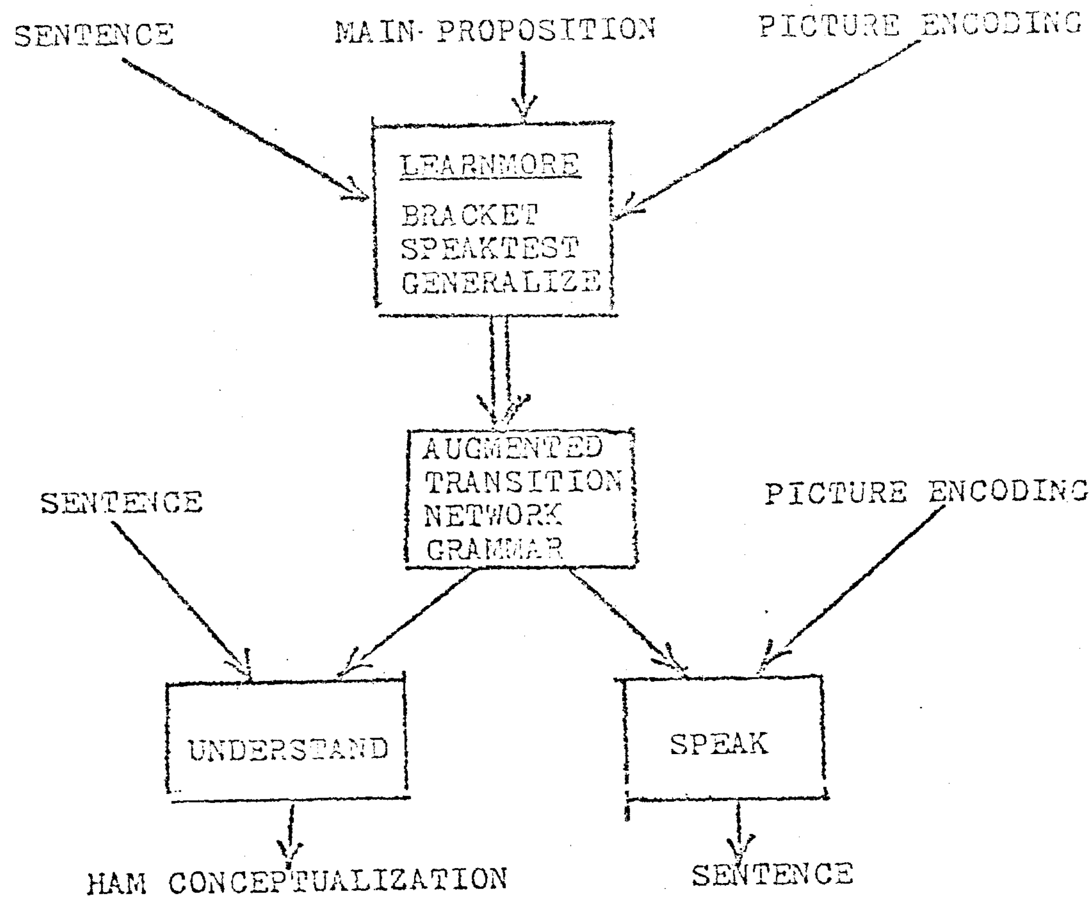

Figure 2. A schematic representation giving the input and output of the major subcomponents of LAS-LEARNMORE, SPEAK, and UNDERSTAND.

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 A red square is above the circle would be represented with the HAM. 2 network formalisms. There are four distinct propositions predicated about the two nodes X and Y: X is red, X is a square, X is above Y, and Y is a circle. Each proposition is represented by a distinct tree structure. Each tree structure consists of a root proposition node connected by an S link to a subject node and by a P link to a predicate node. The predicate nodes can be decomposed into a R link pointing to a relation node and into a O link pointing to an object node. The semantics of these representations are to be interpreted in terms of simple set-theoretic notions. The subject is a subset of the predicate. Thus, the individual X is a subset of the red things, the square things, and the things above Y. The individual Y is a subset of the circular things.

One other point needs emphasizing about this representation. There is a distinction made between words and the concepts which they reference. The words are connected to their corresponding ideas by links labelled W. 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 Yesterday in my bedroom a red square was above the circle or John believes that a red square is above the circle. Representations for such

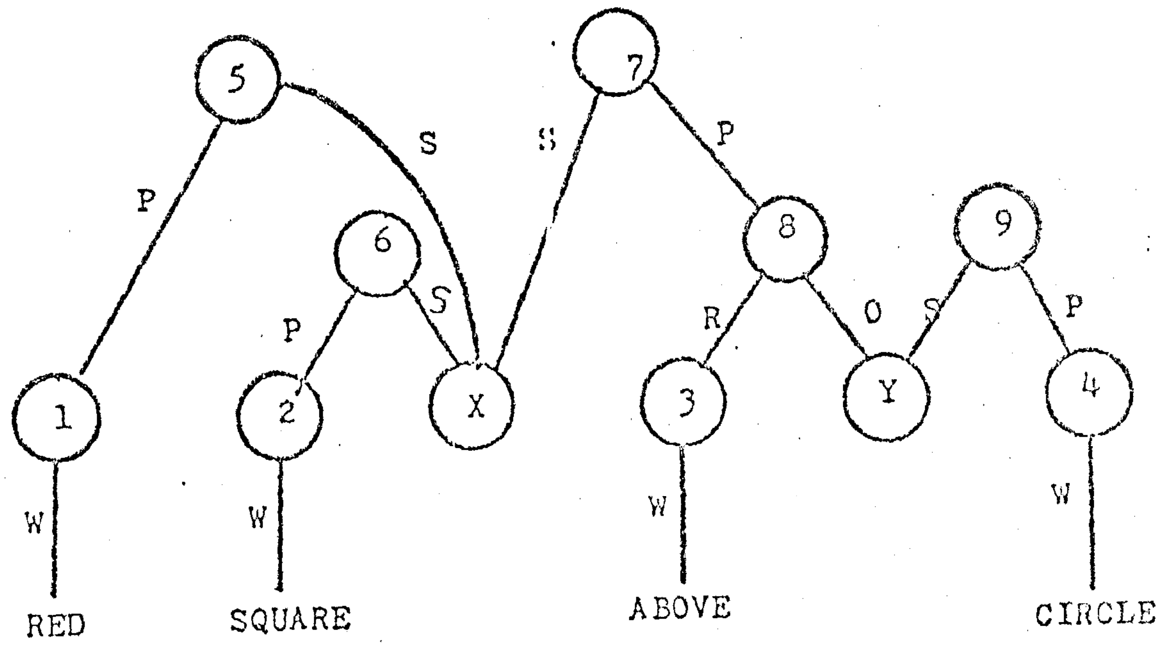

Figure 3. An example of a propositional network representation in HAM.[2]

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: