Artificial Intelligence, Language, and the Study of Knowledge*·†

IRA GOLDSTEIN AND SEYMOUR PAPERT

Massachusetts Institute of Technology

This paper studies the relationship of Artificial Intelligence to the study of language and the representation of the underlying knowledge which supports the comprehension process. It develops the view that intelligence is based on the ability to use large amounts of diverse kinds of knowledge in procedural ways, rather than on the possession of a few general and uniform principles. The paper also provides a unifying thread to a variety of recent approaches to natural language comprehension. We conclude with a brief discussion of how Artificial Intelligence may have a radical impact on education if the principles which it utilizes to explore the representation and use of knowledge are made available to the student to use in his own learning experiences.

Artificial Intelligence (AI) represents a new style of thinking about cognition that is having an important impact on the study of language. The goals of this essay are: to convey a sense of the intellectual personality of Artificial Intelligence; and to identify some trends in its development which we see as especially relevant to the study of language and of the role of an AI approach to language in education.

1. ARTIFICIAL INTELLIGENCE AS THE STUDY OF INTELLIGENT PROCESSES

Before we embark on the substance of this essay, it is worthwhile to clarify a potential source of confusion. For many, AI is identified as a narrowly focused field directed toward the goal of programming computers in such a fashion that

they acquire the appearance of intelligence. Thus it may seem paradoxical that researchers in the field have anything to say about the structure of human language or related issues in education. However, the above description is misleading. It correctly delineates the major *methodology* of the science, that is, the use of computers to build precise models of cognitive theories. But it mistakenly identifies this as the only purpose of the field. Although there is much practical good that can come of more intelligent machines, the fundamental theoretical goal of the discipline is understanding intelligent processes *indepen-dent* of their particular physical realization.

Naturally, understanding intelligence abstractly has intimate interactions with understanding it in its natural form. Thus, AI and psychology have close ties. Indeed, AI has been called "theoretical psychology" (Newell, 1973). AI draws from psychology a set of basic concerns—understanding language, perception, memory, problem solving. Psychology, in turn, acquires a new framework of computational ideas for expressing cognitive theories. Indeed, a major purpose of psychology becomes the discovery not of the class of programs that could possibly explain a given intelligent behavior (this is the archetypical concern of AI), but rather the particular program that a given individual actually possesses.

1.1   A Knowledge-Oriented rather than Power-Oriented
       Theory of Intelligence

Given the general perception of AI as the computational study of intelligence, it still remains to identify the particular set of questions and theoretical constructs that a researcher in the field brings to bear on cognitive problems. In the earliest years of the field (i.e., during the late 1950s and early 1960s), the hope was that a few simple but very powerful techniques could be identified that together could be used to create intelligent programs. During this period, theorem proving and search played a primary role. Indeed, to many AI has become identified as the science of *heuristic search,* that is, the study of procedural techniques for exploring state spaces too large to be explored in an exhaustive fashion. During these early years, an AI paper on language might have begun by representing the state space by means of a formal grammar and then have focused on the issue of achieving successful parsings in reasonable times (Green, Wolf, Chomsky, & Laughery, 1963; Lindsay, 1963).

Today there has been a shift in paradigm. The fundamental problem of understanding intelligence is not the identification of a few powerful techniques, but rather the question of how to represent *large amounts of knowledge in a fashion that permits their effective use and interaction.* This shift is based on a decade of experience with programs that relied on uniform search or logistic techniques that proved to be hopelessly inefficient when faced with complex problems embedded in large knowledge spaces. The current point of view is that the problem solver (whether man or machine) must know explicitly how to use

its knowledge—with general techniques supplemented by domain specific pragmatic knowhow. Thus, we see AI as having shifted from a *power-based* strategy for achieving intelligence to a *knowledge-based* approach. Minsky and Papert (1974) characterize these two points of view as follows:

> The *Power* strategy seeks a generalized increase in computational power. It may look toward new kinds of computers ("parallel" or "fuzzy" or "associative" or whatever) or it may look toward extensions of deductive generality, or information retrieval, or search algorithms—things like better "resolution" methods, better methods for exploring trees and nets, hash-coded triplets, etc. In each case the improvement sought is intended to be "uniform"—independent of the particular data base.
>
> The *Knowledge* strategy sees progress as coming from better ways to express, recognize, and use diverse and particular forms of knowledge. This theory sees the problem as epistemological rather than as a matter of computational power or mathematical generality. It supposes, for example, that when a scientist solves a new problem, he engages a highly organized structure of especially appropriate facts, models, analogies, planning mechanisms, self-discipline procedures, etc. To be sure, he also engages "general" problem-solving schemata but it is by no means obvious that very smart people are that way directly because of the superior power of their general methods—as compared with average people. Indirectly, perhaps, but that is another matter: a very intelligent person might be that way because of specific local features of his knowledge-organizing knowledge rather than because of global qualities of his "thinking" which, except for the effects of his self-applied knowledge, might be little different from a child's. (p. 59)

It is worthwhile to observe here that the goals of a knowledge-based approach to AI are closely akin to those which motivated Piaget to call his research center "Centre d'Epistemologie Genetique"; or, more precisely, to his reason for calling himself an "epistemologist" rather than a psychologist. The common theme is the view that the process of intelligence is determined by the knowledge held by the subject. The deep and primary questions are to understand the operations and data structures involved. The physical (biological or electrical) mechanisms underlying them are not seen by us or by Piaget as the source of intelligence in any structural sense (although intelligence could not happen without some such embodiment.)

Perhaps the knowledge paradigm is most clearly characterized by a developmental perspective. Naturally, intelligence is impossible without a certain basic "hardware" capability. But our position and Piaget's is that this capability is already present in the very young child. The subsequent development of intelligence is not due to the emergence of new biological capabilities in the individual. Rather, increased performance is due to the acquisition of knowledge. This knowledge can range from specific facts about a domain to general problem-solving schemata. Indeed, it may even be knowledge about how to learn (for example, facts about debugging techniques). Some facts result in only local improvements in the individual's capabilities, while others may have global effects, giving rise to a stage transition in Piagetian terms. From the computer

standpoint, we see the need for some improvements in speed and memory capabilities, especially for such special purpose applications as vision and speech. But the fundamental difficulties facing researchers in the field today are not limitations due to hardware, but rather questions about how to represent large amounts of knowledge in ways that still allow the effective use of individual facts.

1.2   Artificial Intelligence as a Procedural Theory of Knowledge

The reader may perhaps wonder at this point how AI is to have an identity of its own if the knowledge strategy is pursued, that is, if it asks only for explicit representations of the knowledge of particular domains. Is not this simply the concern of the specialized field of study associated with that domain? How does this notion of AI, for example, distinguish it from linguistics when the domain is that of language competence?

The answer lies in identifying two concerns of AI which are uniquely *procedural* in nature. The first is that the knowledge of a particular domain must be represented in a form that is usable. A linguist may content himself with the specificiation of some attribute of grammatical knowledge by defining the appropriate transformation; an AI researcher must go one step further by asking how this transformation can be woven into the total web of linguistic information in such a way that the comprehension or generation process is possible in reasonable computational times. Thus, AI brings to the traditional study of a given field the quest for identifying and formalizing *pragmatic* knowledge, namely, knowledge of "how" traditional facts are to be used.

There is a second concern which distinguishes AI from traditional studies of a given domain. This is its concern for identifying the manner in which competence in different domains interact. Again drawing upon an example from the study of language, a grammarian can legitimately focus on attributes of grammar divorced from the semantics or use of language. The essential concern of AI, however, is to construct frameworks in which all of these diverse kinds of knowledge can successfully interact in order to allow the complete comprehension process to take place. Thus, both concerns which make AI unique arise from its responsibility to structure knowledge into procedural systems that can actually solve problems.

1.3.   Organization of This Paper

Our goal in the remainder of this paper will be to provide the reader with insight into the structure of theories currently being formulated in AI within this knowledge-based paradigm. In Sections 2 and 3 we illustrate the AI approach by discussing a particular example—frame theory—which many people in the field are currently exploring and find quite exciting. In Section 2, we introduce the subject by approaching frame theory as an AI extension of the traditional linguistic notion of case grammars from a narrowly syntactic theory to a general

representation for knowledge. In Section 3, we discuss the various attributes of AI theories that frames exemplify. In Sections 4 and 5, we deepen our view of AI as a computational style of epistemology by discussing two aspects of cognition and language to which AI has made unique contributions—the role of self-knowledge and the importance of control structure. In Section 6, we conclude our theoretical discussion by briefly viewing in perspective the intellectual history of the field.

These topics do not by any means exhaust the contribution of AI to the study of language. But we believe they provide an excellent illustration of the AI methodology. And that seems to us the only serious purpose to be served by a paper of this length that is accessible both to students of the field and to nonspecialists. Two companion essays to this paper were prepared for the National Institute of Education by Winograd (1976) and Wilks (1976) and complement the discussion presented here.

Throughout the paper, we avoid extensive excursions into formalism and jargon. More seriously, we also ignore, or pass over in a cursory way, a variety of serious technical issues. We do this in order to focus attention on the overall enterprise of a computational investigation of knowledge. We feel especially justified because of the tendency in the literature to focus on narrow technical issues. Nevertheless, to compensate for this broad focus, we provide in Section 6.2 a list of the technical questions which we feel are critical for the reader who wishes to pursue this point of view more deeply. Also, the reader should recognize that the various works we cite as illustrating our position have many important differences. However, we feel that the critical need at this time is to develop a feeling for the paradigm which characterizes AI as a method for investigating knowledge and language, and it is to this goal that we direct our efforts. Finally, we and others in the AI community expect to develop further the ideas raised in this paper. The interested reader is invited to write us for references to the latest research in this area.

1.4 Educational Applications

The final part of the essay (Section 7) will touch schematically on some of the consequences for education of AI research on natural language. These consequences are of very different sorts. Some of them, no doubt for most people the most obvious, are consequences of actually possessing machines capable of understanding what a student says. Obviously such machines would greatly improve the quality of "computer-aided instruction" (in the sense of tutorial programs, programmed instruction administered by computers, "drill and practice," computerized advice, etc.).

Much deeper consequences, however, are related to the intellectual content of AI research on natural language. As a new kind of cognitive theory it has obvious implications for the education. It is our personal opinion that these consequences will very deeply transform the theory and practice of education.

Although we will not try to argue for this position at length, we shall try to convey the flavor of the perceptions on which it is based.

2. FRAMES AS A UNIFYING APPROACH TO REPRESENTING KNOWLEDGE

Frame Theory is an AI approach to representing knowledge that generalizes the linguistic notion of case grammars in a manner that allows the representation not only of syntactic knowledge, but also procedural knowledge for how to construct a parse as well as related semantic, contextual, and thematic facts. Historically, frame theory is based upon more than purely linguistic considerations. In the original formulation by Minsky (1975), the notions of frames and frame systems were developed in equal measure from problems arising from the construction of adequate theories for vision, for memory, for logical reasoning, and for the comprehension of language. However, we shall narrow our focus in this section to consider the utility of frames solely in relation to linguistic knowledge. In Section 3, we step back from the realm of language to consider a variety of larger issues that characterize the AI approach to representing knowledge.

2.1   Case Grammars
The case frame approach to linguistics as developed by Fillmore (1968) and others constrains the enquiry to the realm of syntactic knowledge. Within this sphere, the basic idea is to organize structural knowledge of the sentence around the verb. The result is a *case grammar* which specifies for each verb the cases which it requires. To illustrate this, we use the traditional linguistic device of choosing a representative sentence around which to organize our observations:

(S1)   John tickled the girl with the feather.

Following Chafe's formulation, a case-oriented approach would represent (S1) as follows:

Tickle: agent   John
          patient   the girl
          instrument   a feather

2.2   Understanding a Sentence

To make the linguistic enterprise tractable, the linguist can ignore the issue of understanding an utterance beyond the narrow concern of identifying its syntactic structure. But AI is forced to address this larger issue if computational models are to be designed that actually act appropriately in response to the utterance.

Thus, AI is willing to consider the senses of understanding (S1) illustrated by the following three listeners:

> A five-year-old child, who has a clear image of John using the feather to evoke giggles.
> A slightly more sophisticated listener capable of taking account of a context such as: "All the girls in the room had strange hats. One was decorated with a large feather, another with a towering confection of artificial fruit. *John tickled the girl with the feather.*"
> A linguist who understands the sentence as an example of a well known ambiguity in parsing.

It is obvious that all the subjects use some grammatical knowledge. For example the order of the words tells them that it was John who did the tickling and not the girl. But grammatical knowledge alone is not sufficient even to account for the syntactic parsing of the sentence, that is, the interpretation of the phrase "with the feather" as adjectival or adverbial. This point is made clear by contrasting the sentence with (S2):

> (S2) John tickled the girl with the funny nose.

Even our five-year-old would not be likely to read this as if John used the nose as a tickling instrument. The difference is not puzzling to common sense; after all the child knows that feathers are good for tickling while noses are more likely to be used for smelling or as personal identifiers. But the difference should trouble the cognitive theorist who knows how to describe syntactic structures in fine detail but has no better description than naive common sense of the child's knowledge about feathers and noses. And the difference is relentlessly present for the computer-oriented theorist who absolutely *needs* to find ways to formalize it.

2.3 The Procedural Idea of a Frame-Keeper

Let us begin our generalization of case grammars by discussing informally how a procedural case theory might be organized. Following this, we will discuss work by Riesbeck (1974) that represents one formal realization of this point of view.

In our informal minitheory there is associated with each important word in the sentence an active agent which is strictly defined as a computational procedure but which we might just as well anthropomorphize (for the moment anyway) as a person. Let's call them *frame-keepers* for reasons that will soon become apparent. The process of understanding the sentence will consist of an interaction between these frame-keepers wherein each tries to fulfull certain needs. For example, the needs of the tickle-frame keeper include:

Find the agent (the person responsible for the tickling)!

Find the patient (the person being tickled)!

Find the instrument (the means by which the agent is tickling the patient)!

So the tickle-frame keeper is seen as actively seeking to fill these *slots* written into his definition. If all of them can do so to their satisfaction (an idea which we will pursue shortly) the sentence is to be considered as parsed.

The essence of the AI extension of the case frame idea is that the frame for the verb contain not only syntactic information about the expected location of the underlying cases, for example, in a declarative sentence that the subject is the agent, the object the patient, and the "with" prepositional phrase the instrument, but also semantic and pragmatic facts. Thus, we imagine that for our tickle-frame, the frame-keeper has access to world knowledge that states that feathers are typical tickling instruments, while other kinds of objects, for example, dumbbells, are not. Hence, it would be the *tickle-frame keeper* that would be in a position to reject for (S2) that the "with" phrase refers to the tickle instrument because it knows that one generally does not tickle with noses.

In Section 2.4, we discuss an existing program that associates both syntactic and semantic knowledge with individual words in order to comprehend the sentence. In Section 2.5, we indicate how the frame idea can be extended to nouns. Given that extension, the process of comprehending (S2) becomes clearer as not only does the tickle-frame reject the nose phrase, but the girl-frame accepts it as a legitimate descriptor of people. Thus, the system has two sources of evidence both supporting one another regarding the intended parse.

2.4   A Formal Realization of the Frame Idea

Our discussion of the tickle-frame and frame-keepers represents the kind of concern for *active knowledge* that AI brings to linguistic studies. Riesbeck (1974, 1975) has designed and implemented a program that, although not frame-based, shares many important properties with the approach outlined above. Riesbeck's parser maps natural language into the *conceptual dependency* structures designed by Schank (1973). Sentence (S3) is typical of the sentences that Riesbeck's system analyzes:

(S3)   John gave Mary a beating.

Procedural information is associated with each word in the sentence, which is triggered when the parser encounters the word in its left-to-right analysis. This information constructs an instantiated case-framelike structure. It also contains expectations that serve to disambiguate multiple word senses and properly decode the syntactic cues that indicate the fillers for each slot in the case-frame associated with the verb.

The system, called MARGIE, of which Riesbeck's parser was a part, exhibited its understanding by generating paraphrases of the original sentence and by

making deductions based on it (Schank, 1972). The basic operation was to comprehend the sentence by mapping it into the conceptual dependency representation developed by Schank (1973). Figure 1 illustrates the conceptual dependency diagram generated for the above sentence. The details of Schank's representation are not important here, except to note that MARGIE understood the sentence as referring to a series of actions by John involving the motion of his hand. Riesbeck's parser has not erroneously interpreted ''give'' to mean that John is transferring the possession of something to Mary.

2.5 Frames for Representing Knowledge about Nouns

The case-frame idea in its original linguistic form was oriented toward representing knowledge about verbs. Pursuing the goal of more comprehensive theories of knowledge, AI has generalized the idea to allow frames for nouns.

Consider again the sentence ''John tickled the girl with the feather.'' We imagine that part of the knowledge for properly understanding this sentence is that the verb-frame for tickle has as a *typical* instrument ''feathers.'' Such verb-centered knowledge is supplemented by frames for such nouns as ''girl'' which contain pointers to subframes which describe typical appearances, purposes, activities, etc. The appearance subframes would indicate that girls do not ''have'' feathers. They might under closer scrutiny indicate that ''feathers'' are a possible but unlikely kind of dress. However, the strong default for ''feathers'' as a typical tickling instrument would predominate.

A related example shows where noun-frame information might dominate the verb-frame default. Consider the three sentences:

(S4) I frightened the dog with the loud noise.
(S5) I frightened the dog with the brown fur.
(S6) I frightened the dog with the loud bark.

We believe most readers would understand (S4) by assuming that the ''loud noise'' was the instrument of ''frighten.'' This would be because of both the verb-frame knowledge that loud noises often can be used to frighten and the
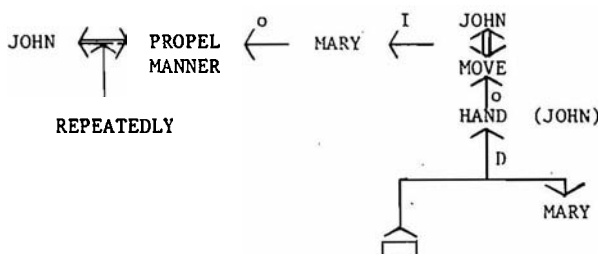

FIG. 1. Conceptual dependency representation of the meaning of the sentence: ''John gave Mary a beating.'' (After Riesbeck, 1974, p. 108.)

noun-frame knowledge associated with dogs that indicate that "noise" is not "possessed" by them. On the other hand, in (S5), the noun-frame knowledge for dogs would surely indicate that they typically have "fur," hence the phrase "with the brown fur" would be understood to be modifying the noun "dog." In (S6), we suspect that our population of readers would be divided, depending on whether their frame for "frightening" contained the typical instrument of emitting a loud noise, describable as a bark, with a stronger emphasis than their frame for "dogs" contained the default property of emitting noise by barking. (This extension of frames to nouns has not yet reached the point of development in the form of programs. However, Drake (1975) extends the details of the theory beyond the outline provided by Minsky to a form that is closer to the implementation state.)

In the previous paragraph, we wrote as if the tickle frame could have *direct, explicit* knowledge about feathers. A fundamental and pervasive class of problems concerns the relative feasibilty of achieving this as compared to various schemes for *indirect* reference to feathers via some *superset* or *description*. For example, the tickle-frame might know that "long, soft objects" are good instruments for tickling and somehow use this as a means of finding a connection to feathers. Most early workers in AI (like most contemporary psychologists) were strongly averse to the direct approach and so a great deal of attention has been given to the invention of "search" or "information retrieval" methods to mediate the indirect approach. Current thinking (especially among frame theorists) leans toward very much more direct knowledge than was previously considered feasible. This places a heavy demand on *memory,* and less on retrieval mechanisms. This appears to some to be "heavy-handed." But we do know that human memory has a very large capacity and no one has been able to propose very sophisticated retrieval mechanisms. So perhaps it is nature (rather than the theorists) who has adopted the "heavy-handed" solution.

2.6 *Extending the Frame Idea to a Theory of Context*

Procedural frames which are not restricted to verbs are powerful tools for dealing with the set of problems traditionally called "context." Indeed, once we break away from the verb-centered sentence as the unit for analysis, it becomes possible to talk about "scope" in a possibly very large text.

This, too, is a topic of current research and there are few programs implemented as yet. One of the systems that does exist, however, is Schank's successor to the MARGIE system that generates paraphrases of simple stories (Schank, 1975). The essential theoretical idea in the new system is the existence of frames (called scripts by Schank) that describe typical activities in terms of his conceptual dependency primitives. An example is the following informal sketch of a frame for going to a restaurant:

    Script: restaurant.

      Role: customer, waitress, chef, cashier.
    Reason: to get food so as to go down in hunger and up in pleasure.

Scene 1: Entering
          PTRANS—go into restaurant
          MBUILD—find table
          PTRANS—go to table
          MOVE—sit down

Scene 2: Ordering
          ATRANS—receive menu
          ATTEND—look at it
          MBUILD—decide on order
          MTRANS—tell order to waitress

Scene 3: Eating
          ATRANS—receive food
          INGEST—eat food

Scene 4: Exiting
          MTRANS—ask for check
          ATRANS—give tip to waitress
          PTRANS—go to cashier
          ATRANS—give money to cashier
          PTRANS—go out of restaurant   (Schank, 1975, p. 117)

The image of understanding a story is mapping the sentences of the story into the actions described in the frame. Unstated facts described in the frame are assumed. Thus, mapping a story in which the sentence, "John left a tip," appeared into this frame would allow the system to know that the "tip" was left for the waitress, a fact contained in the frame but not in the story.

2.7 Frames for Thematic Knowledge

Our previous examples have discussed extensions of the case-frame idea to incorporate into the frame a larger fraction of the knowledge associated with comprehending a sentence than is included within the purely grammatical realm. We would like to conclude by extending our representation beyond the sentential level to include the kinds of knowledge that link sentences together. This represents the frontier of current research and so we will have few formal examples and have to content ourselves with some observations which indicate the direction in which we expect this research to go.

To develop this notion of thematic frames, let us consider the problem of understanding a fairy tale. Our goal is to represent the knowledge that links sentences in a story together into a coherent whole. To begin, we will first examine some work by Rumelhart (1975) which addresses this question, but utilizes a phrase-structured grammar (augmented by semantic interpretation rules) to represent the structure of the story. Typical of the rules that his system possesses are:

RULE 1: STORY → SETTING + EPISODE.

The setting is a statement of the time and place of the story as well as an

introduction for the main characters, for example, "Once upon a time, in a far away land, there lived a good king."

    RULE 2: SETTING → (STATIVES)*.

Episodes consist of events followed reactions:

    RULE 3: EPISODE → EVENT + REACTION.

An event is the most general category of the grammar:

    RULE 4: EVENT → EPISODE or CHANGE-OF-STATE or ACTION
                or EVENT + EVENT.

A reaction consists of an internal and an overt response:

    RULE 5: REACTION → INTERNAL RESPONSE + OVERT RESPONSE.

There are a large number of internal responses. Two are:

    RULE 6: INTERNAL RESPONSE → EMOTION or DESIRE.

Overt responses can be either an action or a series of attempts:

    RULE 7: OVERT RESPONSE → ACTION or (ATTEMPT)*.

Attempts also have an internal structure:

    RULE 8: ATTEMPT → PLAN + APPLICATION.

The plan is developed to accomplish a desire. The application is the actual execution of the plan.

    For each of these rules, there are semantic interpretation rules that indicate, for example, that Rule 1 means that both the setting and the episode must be present while Rule 3 (with the same syntactic "+" connective) indicates that the event is the cause of the reaction and precedes the reaction in time.

    Let us now ask how the skeleton represented by this story grammar might be extended into a frame system. The basic change is the addition of large amounts of highly structured and particular knowledge. This information which we shall call *default knowledge* includes information about the various aspects of the story, for example, setting, episodes, internal and external responses. Indeed, the proposal is that the system knows a few "favorite" stories in extensive detail. Thus, with only a brief introduction to the story such as "Once upon a time and long ago," the listener is able to bring to mind an elaborate setting corresponding to his primary default—perhaps the Camelot setting, the Black Forest setting, or the humble cottage, depending on the idiosyncrasies of the frame system of the individual listener. Of course, conflicts will arise from differences between the story and the default, causing one of two results: either exceptions are entered into the evolving database representing the system's comprehension of the story, or, if the conflict is too radical, the entire default

setting is replaced with another. In either case, the result is that far more detail than is actually stated in the story is assumed by the listener through this mechanism of default knowledge.

Just as there can be default knowledge about states such as setting, so too can there be default knowledge about the expected sequence of episodes in the story, in which each episode is itself a frame (e.g., a frame for the initial state—a damsel in distress, followed by a sequence of frames describing a typical rescue operation by a handsome prince and hindered by a wicked witch). This organization of frames into state action sequences is is essentially the *script* idea of Schank's discussed earlier. Default knowledge about expected themes, morals, and narrative structures may also be included. We expect, for example, for a fable to have a moral, for the prince and princess to live happily ever after in a children's fairytale (though not in the original primitive folklore).

The point of this discussion is to articulate this model of comprehension in which the process is not one of literally understanding the text, but instead is more one in which the text triggers rich, highly structured knowledge packets that supplement the literal content, provide expectations for the remainder of the story, and place the story in a context of related knowledge. Understanding is seen as essentially a process of evoking and then debugging existing knowledge packets.

3. ISSUES IN COMPUTATIONAL EPISTEMOLOGY

At this point, we have seen how the linguistic theory of case grammars has been extended by AI to become a knowledge-oriented theory of language. We now step back from a purely linguistic orientation in order to gain a perspective on the general theory of intelligence and knowledge embedded in this point of view. Specifically, we shall examine:

1. The manner in which frames exhibit a readiness to use new, flexible and complex data types.
-2. The fashion in which frames represent a synthesis of procedural and declarative knowledge.
3. The sense in which the use of an anthropomorphic metaphor has become a technical device in AI theory.
4. The trend toward the representation of knowledge in larger, more structured units (which frames exemplify).
5. A concern for allowing useful interactions between diverse kinds of knowledge.
6. A desire for expertly controlled processes.

3.1 Representing Declarative Knowledge

A central concern of AI is the construction of new data types and the exploration of their capabilities. The frame idea is still evolving in AI, and so it

is difficult to give a precise definition. Minsky (1975) gives the following description of a frame in his original paper on the subject:

> We can think of a frame as a network of nodes and relations. The "top levels" of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many *terminals*—"slots" that must be filled by specific instances or data. Each terminal can specify conditions its assignments must meet. (The assignments themselves are usually smaller "subframes.") Simple conditions are specified by *markers* that might require a terminal assignment to be a person, an object of sufficient value, or a pointer to a subframe of a certain type. More complex conditions can specify relations among the things assigned to several terminals. (p. 212)

We see from the above description that the traditional *declarative* data structures of objects, properties, and relations plays an important role in the definition of a frame. This aspect of frame theory has its roots in earlier work in AI on the representation of knowledge as attribute–value pairs (e.g., the BASEBALL program, Green *et al.*, 1963) and as semantic nets (Quillian, 1968). The notion of a semantic net has already had an important impact on cognitive psychology (cf. work by Anderson and Bower, 1973, and Norman, Rumelhart, and the LNR Research Group, 1975). The additional ideas added by frame theory which generalize the notion of a semantic net are:

1. The organization of these relations, not into a uniform net, but into a highly structured set of contexts (the frames themselves).
2. The inclusion of defaults for properties and relations representing the expected values and supplying common sense knowledge about the context (possibly not supplied by the actual text).
3. The infusion of procedural knowledge which we shall discuss below.

3.2   Representing Procedural Knowledge

In Sections 2.2 and 2.3, we saw informally how procedural expertise can be associated with a frame in the form of a *frame-keeper,* a procedure whose action is to examine the current context for data that can fill the slots of the frame. To further clarify the concept, we begin with a gedanken example of how an AI program might emulate a certain aspect of the way a mathematician thinks. The example goes right into the domain of the logician and proposes a very different way to cope with the existence of paradoxes such as the Burali–Forte paradox in mathematics and such paradoxes in ordinary language as the use of self-reflexive assertions such as "All Cretans are liars" (spoken by a Cretan and meaning: Cretans never tell the truth). The traditional logician's recourse is to seek an axiomatization from which no such contradictions can be deduced. Our theory of what a working mathematician actually does is to use principles of reasoning which could easily lead to paradoxes if they were to be applied in that direction. But the mathematician knows this and refrains from using them in directions known to be dangerous. It is as if he posted a sentry near dangerous places in his

intellectual terrain. The job of the sentry is to issue warnings if any reasoning process comes too close to the danger spot. In AI we see our job as developing theories in which the concept of such a sentry can be easily formalized.

The anthropomorphic idea of posting a sentry is typical, we maintain, of the way in which sensible people have thought intuitively about this kind of problem. The contribution of computer science is to show how to translate this intuitive matter into a technical and well-defined form. If one represents a knowledge system as a set of propositions, the idea of introducing the sentry is outrageous, or at best a literary conceit. But if the knowledge system is represented in the first place as a set of interactive processes, the sentry is merely one more process. Naturally one has to be sure that this new process is well defined. But that is a technical problem. The conceptual change is that it is now perfectly natural and coherent to introduce such entities. From this perspective, a fundamental contribution of AI to epistemology is made clear; and that is, the systematic introduction of active agents into epistemological theory construc-tions, so that, for example, an item of knowledge, a concept, or whatever, is seen as an active agent rather than a passive manipulatable object. (See Greif and Hewitt, 1975, for recent work in this direction.)

The gedanken example from mathematics seemed to us well suited to illustrate the force of this concept in an intuitive way. But there are real applications applied to quite real problems. A well-known example is Charniak's method for dealing with reference problems in children's stories. To illustrate this, consider the following story fragment from Charniak (1972).

Today was Jack's birthday. Penny and Janet went to the store. They were going to get presents. Janet decided to get a top. "Don't do that," said Penny. "Jack has a top. He will make you take *it* back." (p. 32)

The goal is to construct a theory that explains how the reader understands that "*it*" refers to the new top, not the one Jack already owns. Purely syntactic criteria (such as assigning the referent of "*it*" to the last mentioned noun) are clearly inadequate, as the result would be to mistakenly understand the last sentence of the story as meaning that Jack will make Janet take back the top *he already owns*.

Resolving reference is a traditional problem of linguistics that cannot be handled by purely syntactic techniques. Our solution will involve the realization of our intuitive "frame-keeper" as a precise procedural entity, the *demon*. This is a production-like entity that is activated by some aspect of the state of the computational world within which it is defined. But we will postpone the details of this solution until Section 3.4 in order to first give some perspective on the underlying AI philosophy.

3.3 Representing Knowledge in Highly Structured Packets

The fundamental frame assumption is the thesis that *fewer situations than we think are really as new as we think.* Most situations in which people find themselves have sufficient in common with previously encountered situations for the salient features to be preanalyzed and stored in a situation-specific form. Whenever this happens we are able to use "predigested," highly particular knowledge—an explanation in sharp contrast with the views of intelligence propounded by theorists who would prefer to believe that our minds always work (and have to work!) in a deductive fashion from very general principles.

For our *top* example, it is clear that one cannot know that *"it"* refers to the new *top* without knowledge about the trading habits of our society. One could imagine a different world in which newly bought objects are never returned to the store, but old ones are. The question we raise here is how this knowledge might be represented, stored, and made available to the process of understanding Charniak's story. To see the spectrum of possible theories consider two extremes:

An extreme in the frame direction. The listener has stored away in a long-term memory a frame for birthday parties in which certain events and problems are explictly represented. These include the problem of choosing a present with quite explicit reference to the three or four "possible dangers to be thought about" including "recipient won't like it and will be disappointed" and "recipient already has one and will return it."

An extreme in the antiframe direction. From a strictly logical point of view it is not necessary to have a preconstructed party frame in order to assign the proper referent to *"it."* It is unnecessary to have been to a party or have given a present. It would be sufficient to have stored away for general use items of information such as these about returning gifts:

1. $\{(x$ has been bought at store $y)$ and $(x$ is in good condition$)\}$
    implies (store y will exchange x for $z)$.

2. $\{(x$ is new$)$ and $(x$ has not been used$)\}$
    implies $(x$ is in good condition$)$.

The advantages of this method (if it is workable) are apparent: less memory space, easier updating, modularity, etc. The major disadvantage is that we then have to face the question of how these two items of information are chosen from a very large stock of knowledge, presumably including many other items rather similar to these. How does the interpretive system know which items to use?

An even more serious problem takes the form: how does the system know that

any appeal to knowledge about the world is necessary? Are we to assume that simple syntactic rules (e.g., based on word order and proximity) are *never* used? But if they are *sometimes* used, what restrains them in this case?

By providing possible answers to this question, the frame theory shows itself to be much more than a method for information retrieval or "heuristic search." One such answer is that Charniak's reference problem simply does not arise in the sense that the birthday-present-frame-keeper will already have assigned the new top to a slot for "returned gifts" before the pronoun is processed!

3.4   The Demon Solution to the Reference Problem

Charniak's 1972 research on constructing a model for comprehending children's stories predates the articulation of the frame idea, but presages important notions of that concept in several ways. In particular, his formal realization of a *frame* was in the form of *base knowledge* about a large variety of situations that arise in the context of these stories. The mechanism of his program was for the content of sentences to evoke this base knowledge with the following effect: demons ("frame-keepers" in our terminology) were created to monitor the possible occurrence in later sentences of likely (but not inevitable) consequences of the given situation. Thus, for our story fragment the birthday knowledge creates expectations about the need for participants of the party to buy presents and the possible consequence of having to return these gifts. Hence, these demons expect the possibility of Jack already possessing the present and the resulting need for Janet to return *it,* where *it* is known to be the present.

It is interesting to note that this demon about presents cannot be applied unless several common sense inferences are made. For example, the demon might be a procedural representation of the following fact:

If a person P1 does not want a present *X,* then
he may request the giver P2 to return the present.

For the demon to be applied, the antecedent, that is, that P1 does not want the present *X,* must be triggered by an assertion in the database being created to represent the meaning of the story. (The database may be thought of as containing the kernel sentences of the story with each word mapped into its proper sense and reference problems disambiguated.) But the story does not include explicitly all important facts. Look back at the story. Some readers will be surprised to note that the text itself does not state (a) that the presents bought by Penny and Janet were for *Jack,* (b) that the *top* bought by Janet was intended as a present, and (c) that having an object implies that one does not want another. All of the above facts are inserted into the database by other demons made active by the birthday frame. Thus, Charniak's notion of a demon is particularly satisfying because not only do demons handle reference problems, they are also

used to insert into the story database various reasons, motivations, and purposes not explicitly stated.

In the current view of frame theory, this insertion of unsaid information assumes a paramount role. From this point of view, it is not accidental that so much is left out of the story. Rather the typical situation in comprehension is to be faced with a set of clues that evoke a rich and detailed knowledge structure, the frame, that supplies the unstated details. Naturally, these defaults may be inappropriate for some situations and, in those cases, the text must supply the exceptions. But in general, reference, deduction, and comprehension are made possible only by knowing a great deal about the situation being discussed.

3.5   The Virtue of Particular Knowledge

AI is a young science and the ideas that we have discussed are under constant development, elaboration, and change. However, it has been our goal to give the reader a feeling for the kind of enterprise that AI is rather than a final statement on its theoretical form.

A central point has been the fashion in which many kinds of knowledge play a role in cognitive processes. In particular, we considered for the reference problem the need for *real world* versus *syntactic* knowledge and, more profoundly, *particular* versus *general* knowledge. A major point of our analysis, that is, of the *Knowledge* as opposed to the *Power* theory of intelligence, is that *particular knowledge* can often do the job for which *general knowledge* is traditionally thought to be necessary.

4.   SELF-KNOWLEDGE

In Sections 4 and 5, we leave the context of Frame Theory in order to broaden our insight into the notion of AI as a new kind of epistemology. To do this, we shall discusss two concerns which are central to AI theories but rarely appear in more traditional formulations of knowledge and intelligence: the first is the formal representation and use of self-knowledge and the second is the importance of control structures.

4.1   Kinds of Self-Knowledge

As we shall see, an interest in self-knowledge will not take us outside our concern for understanding the comprehension process. Indeed, we shall find that this kind of knowledge plays an obvious role in allowing the system (whether computer model or individual) to answer questions about its own performance as well as a role in the less obvious situation of handling special cases that arise in parsing the syntactic structure of the utterance.

There are many kinds of self-knowledge. Perhaps the simplest is the ability to

observe one's own behavior, allowing the individual or program to answer elementary questions about its actions. A more skillful use of self-knowledge would involve being able to take account of these observations and direct one's problem solving accordingly. Still a more sophisticated program might, in certain circumstances, utilize this self-observation to *learn* by debugging faults in its procedures, generalizing its knowledge, or perhaps suitably specializing it to form more efficient techniques for particular problems. We shall illustrate each of these kinds of self-referential abilities below in relation to the Blocks World, a favorite miniworld for AI programs in which a one-armed robot is asked to build various structures from blocks placed on a table (Fig. 2).

4.2   The SHRDLU Program and the Blocks World

Winograd's SHRDLU program represented a *tour de force* of design and an important benchmark in the realm of natural language comprehension by
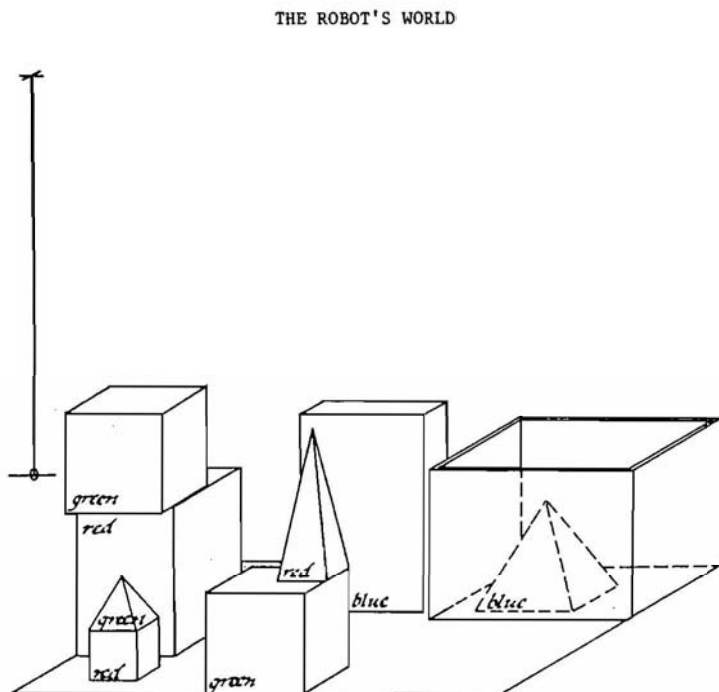
THE ROBOT'S WORLD

FIG. 2. "Pick up the big red block." (After Winograd, 1972, p. 8.)

computer. In this section, we introduce the program briefly, and in the next two illustrate several ways in which it manifests rudimentary self-knowledge.

SHRDLU is a program that converted such sentences as "Pick up the big red block and put it in the box" into commands for a blocks world problem solver. The success of its comprehension was demonstrated by the program successively carrying out the instructions in a simulated world displayed on a cathode ray screen which contained a one-armed robot, a table, a box, and a collection of cubes and pyramids.

The program is capable of comprehending sentences that reached the degree of complexity illustrated by the following examples:

1.   Find the block which is taller than the one you are holding and put it into the box.
2.   Will you please stack up both of the red blocks and either a green cube or a pyramid.
3.   Does the shortest thing the tallest pyramid's support supports support anything green?

In SHRDLU, there are a multitude of different knowledge structures. For example, the system maintains history lists recording each event, why it occurred, and what it accomplished; numerical coordinate data structure recording the dimension and location of objects; a symbolic database in which assertions about the state of the world, about global knowledge, and about active procedures are stored; and a Fodor and Katz style marker hierarchy (Katz & Fodor, 1964) indicating the relation between words. But, perhaps most significantly, a major part of the system's knowledge was represented directly as procedures, thereby achieving far more flexibility and power than more traditional "declarative" representations. These different knowledge packets are used in many ways: to answer questions, to perform actions, to update its knowledge.

4.3   Answering Elementary Why Questions

An elementary example of self-knowledge is found in SHRDLU's capacity to generate names for assertions in its database, events, noun groups, and other entities. With these names, the system can refer back to these entities, associating them with times, places, and other information. This allows the system to answer elementary kinds of questions about its behavior. This ability is demonstrated in Table 1.

4.4   Self-Reference in Grammatical Systems

A second level of self-referential ability in which the system can not only answer questions, but make decisions based on self-knowledge, is also illustrated by SHRDLU. This is the manner in which the system has access not only to the

TABLE 1
Examples of SHRDLU's Ability to Answer WHY Questions[a]

23. had you touched any pyramid before you put the green one on the little cube?

    YES, THE GREEN ONE.

The system deduces from its memory what event is being referred to, and can relate other events to it in time. The analysis includes complex tenses of verb groups, and there are special facilities for keeping track of the various forms of irregular verbs like "have."

24. when did you pick it up?

    WHILE I WAS STACKING UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE.

A time is described by naming the main goal which was being worked on.

25. why?

The system remembers immediately previous sentences to understand questions like this and the following ones.

    TO GET RID OF IT.

By keeping track of selected parts of the original subgoal tree, the system has some understanding of its own motives.

26. why did you do that?

    TO CLEAN OFF THE RED CUBE.

"That" refers to the action mentioned in the answer.

27. why did you clear off that cube?

Here, "that" means the one just referred to.

    TO PUT IT ON A LARGE GREEN CUBE.

28. why did you do that?

    TO STACK UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE.

29. why did you do that?

    BECAUSE YOU ASKED ME TO.

It remembers what it did, not how the request was worded.

[a] After Winograd (1972), p. 13.

current state of the parse, but also to the past history of the parse. This is illustrated by the fashion in which Winograd's program parses conjunctions.

To parse sentences with words like *"and,"* the system invokes a special routine upon detecting the conjunction which notes the last constituent formed and attempts to find a duplicate constituent following the conjunction. If unsuccessful, it looks further back in the history of the parse until a substructure can be found that symmetrically appears following the conjunction. A similar

capability is found in Woods' Augmented Transition Network approach to parsing (Woods, 1970). Elements of this approach are discussed in Section 5.1 of this paper.

It is interesting to note that *"and"* is not handled simply by augmenting the basic systemic grammar of SHRDLU. By some standards, this moving outside the formalism of one theoretical mechanism would be unaesthetic by being contrary to Occam's Razor. But experience has shown that the management of diverse knowledge packages is the rule rather than the exception in the architecture of complex problem-solving systems. The obligation is simply one of clearly defining the interactions between these knowledge packets.

4.5 Debugging Knowledge

Recently, the Blocks World has served as a domain for investigating self-knowledge of a deeper kind. This has been in the form of a system called HACKER (Sussman, 1975) which is capable of learning to solve block construction problems of the complexity that SHRDLU can handle. (Remember that for SHRDLU, the problem-solving procedures are preprogrammed and not learned.) The learning is achieved through HACKER possessing an expertise in debugging. It is our belief that debugging knowledge is absolutely fundamental to the acquisition and operation of any program as complex as human thought and language.

Although this work was not done specifically as part of a natural language comprehension system, it provides the basic understanding to enable such a system to answer questions about why a particular method was used, in a far deeper fashion than SHRDLU's elementary abilities allowed. Knowledge of one's own problem-solving strategies supports an understanding that not only could discuss the chronology of events but could also discuss the underlying plans and their evolution through succeeding stages of generalization and debugging.

To illustrate this, we shall develop an example from Sussman. HACKER initially might be compared to a small child interested in constructing block structures, but not very knowledgeable about the problems involved. HACKER begins with the ability to place one block on top of another (by executing the primitive PUTON) and with an elementary planning capability. A typical elementary goal is to construct a tower from three blocks, expressed formally to the program by the conjunction (GOAL (AND (PUTON A B) (PUTON B C))). HACKER attempts to solve this problem in a "linear" fashion by first planning to achieve (PUTON A B), and then as an independent second goal (PUTON B C). This produces the sequence of block states illustrated below:

Trace of (ACHIEVE (AND (ON A B) (ON B C)))

State 1:   A   B   C

State 2:   B   C   ;PUTON A B has just been executed.
          A

State 3:   A   B   C   ;The top of B has been cleared as a
                        ;required prerequisite of PUTON B C.
                        ;HACKER is one-handed and must lift
                        ;blocks from the top.

State 4:   A   ⚅   ;PUTON B C has just been completed.

Obviously a bug has occurred. The action of clearing the top of B as a prerequisite to achieving (ON B C) resulting in State 3 has undone the first subgoal (ON A B) whose purpose was to place block A on block B.

One solution to this difficulty, which AI programs of the past might have attempted (and which occasionally children pursue), is to return to the original state and try some different construction strategy, without attempting to learn anything from the current failure to guide the next attempt. HACKER contains a new kind of knowledge, namely, knowledge of how to *debug*. This debugging knowledge knows that linear solutions are only first approximations and that sometimes interactions between the subgoals of a conjunction must be taken into account. In this particular case, HACKER recognizes that the prerequisite of one subgoal CLEARTOP conflicts with the purpose of a brother goal. In such situations, it applies the debugging strategy of seeking a reordering of the brother goals such that the goal with the prerequisite is achieved first. In this particular problem, HACKER:

1. Recognizes that the bug is due to the general bug type of "prerequisite clobbers brother goal."
2. Applies the debugging strategy of reordering the goals.
3. Edits in long-term memory the procedure for constructing towers of three blocks to take account of this ordering requirement.
4. Records a comment to itself for subsequent problems of this kind that states that when building block towers of any size, ON chains should be ordered from the bottom up.

Thus, HACKER exhibits both expertise about debugging and about repair, coupled with the ability to examine its own problem-solving goals and actions so that it is able to apply this debugging expertise to its own reasoning. It represents a model of skill acquisition whose competence is based on self-knowledge about its behavior and metaknowledge about the nature of plans and bugs.

We emphasize the importance of debugging skill because a vital element of intelligence is knowing how to handle a vast variety of situations. We cannot do this by knowing all about these situations because each real world situation is different. What we do need is: (1) knowing how to tell which old situation is sufficiently like the present one, and (2) knowing ways to adapt—"debug"—the old procedure for handling the old problem to a new procedure that can deal with

the new situation. Thus, subjects like analogy, similartiy, and metaphor are at the heart of the new formalisms. In the older methods of AI, linguistics, logic, and psychology, these were embarrassing, hard-to-explain phenomena. In the new, debugging technology, they become concrete, manipulable types of knowledge.

Finally, without a theory of debugging, our image of frame-based comprehension would have a fatal flaw. Understanding by using large packets of previously known information requires the ability to *debug* these knowledge structures for the particular application at hand, if the system is to be able to deal with novel situations.

5. CONTROL STRUCTURES

We have seen the way in which AI epistemology is opening new areas of study in its concern for self-knowledge and, in particular, debugging skill. A second dimension along which AI's approach to epistemology is opening new horizons and fundamentally differs from older approaches is its attention to *control structure*. Indeed the idea of control structure could scarcely exist unless one begins to see knowledge as made up of active agents which need to be controlled.

5.1   Controlling Grammatical Computations

"First-order" theories such as those composed of generative grammars are constructed under strong assumptions of modularity and independence. These assumptions are heuristic ones that enable the epistemological analysis to get underway, but include an expectation of a subsequent need to study the interactions. Controlling computation by choosing between the relevance of alternative chunks of knowledge, deciding when a given approach should be terminated, and generally guiding the reasoning process are among the important kinds of interactions which must be eventually understood.

When context-free grammars were first applied to the problem of comprehending text (rather than only generating grammatical sentences), it was immediately recognized that more structure was needed if the parsing times were to be kept within reasonable bounds. The essential problem is to provide information for choosing whether a given rule should be applied. It is the basic concern that characterizes the basic approach of both Augmented Phrase Structured Grammars (Heidorn, 1975) and Augmented Transition Nets (Woods, 1970). The ATN approach has served as the basis of several important language programs and we shall discuss it further below. (See, for example, Simmons, 1973, in which ATNs are used both for parsing into an internal representation and for generation of English output from this representation.)

5.2   Augmented Transition Networks

The Augmented Transition Network formalism for expressing grammars developed by Woods was designed to extend the power of finite-state recognition

grammars. His first step was to generalize the finite state graph to allow recursion, thus essentially obtaining the power of a context-free grammar while still preserving the perspicuity of a network representation. A recursive call was allowed in the state transition graph by the label of an arc being a state name rather than a terminal. This directs the recognizer to jump to that state, run until acceptance, and then return control to the state pointed to by the recursive arc. Essentially a pushdown store has been provided to keep track of the bookkeeping involved in allowing recursion.

Woods' next step along the dimensions of adding control was to generalize these recursive transition networks further and provide for both a preference ordering on the arcs exiting a given node and arbitrary predicates which act as filters and must be satisfied for the node to be pursued.

With the additional augmentation of allowing the system to maintain registers holding various pieces of information found during the parse, the ATN formalism has the full capabilities of a transformational grammar, but incorporates these abilities in a form that admits of far more efficient parsing algorithms than those previously proposed for transformation grammars. This approach has proved successful and now serves as the basis of Woods Lunar system (Woods, 1974) and several systems constructed by Simmons (1973, 1975).

Winograd's SHRDLU also addressed the control issue. Woods added control knowledge to a state-oriented description. Winograd added control knowledge by pursuing an entirely procedural approach. He represented grammatical knowledge as programs. These programs, being procedures, could have conditionals, loops, variables, or recursion. Figure 3 illustrates some simplified programs for recognizing sentences, noun phrases, and verb phrases (expressed as flowcharts rather than in Winograd's *Programmar* formalism).

Woods preferred to preserve some element of declarative representation by clearly naming the states: Winograd preferred the freedom of an entirely procedural description. The approaches are computationally equivalent (Winograd, 1972) and both represent the fashion in which first-order grammatical theories can be extended to support efficient computation.

Of course, most AI theories are computationally equivalent, just as most programming languages, through more or less labor, can be made to emulate each other. This is not, however, the real issue. The critical question is the perspicuity of the language or representation for the particular applications envisioned. Woods emphasizes the virtues of a net description; Winograd prefers expression directly as programs.

5.3   An Issue for the Future—Deterministic Parsing

A concern for control structure raises an obvious question for the future: will it be possible to organize grammatical knowledge (including interactions with other kinds of knowledge) in such a way that backup is eliminated (without recourse to parallelism which is ultimately equivalent to backup) at least for

DEFINE program SENTENCE

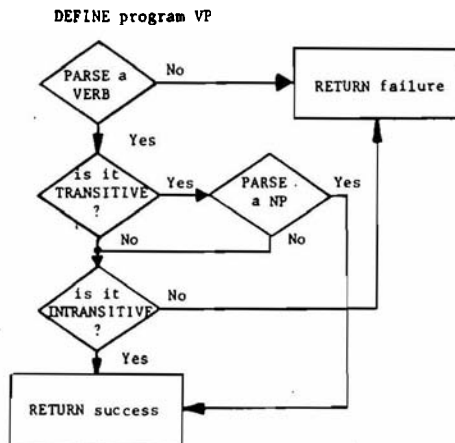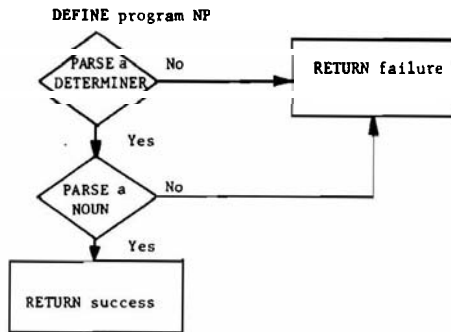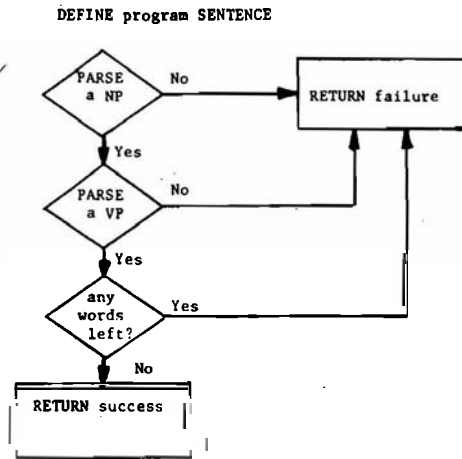DEFINE program NP

DEFINE program VP

FIG. 3. Flowcharts of simplified PROGRAMMAR grammars for noun phrases, verb phrases, and sentences. (After Winograd, 1972, p. 83.)

those sentences which one intuitively feels do not have a multitude of interpretations?

This goal of eliminating exhaustive search represents a natural concern of the computational epistemologist. First define the basic knowledge space. Then analyze its procedural structure. Viewing the process of finding a solution as a search branching at choice points, the question arises: How is this branching controlled? One possibility is to allow for backup. A second is to utilize parallelism. A third is to have diagnostic experts that can guide the decision at choice points so that backup is minimized or perhaps even eliminated.

This evolution has occurred most notably in the development of programs capable of symbolic integration. An early program by Slagle (1963) had knowledge of a basic set of integration formulas but combined this knowledge by means of a search. Recent programs following Moses' design of SIN (Moses, 1967) have a deeper understanding of the nature of integration and do not require any search to find the integral. We do not necessarily expect language programs to evolve to the extreme form of deterministic problem solving that the integration programs have reached, but we do expect a similar kind of evolution in the same direction.

Indeed, insight into grammatical knowledge and knowledge representation techniques has reached the point where one can seriously consider a backup-free approach. The issue is not decided, but the suggestions are promising. For example, a backup-free scheme has recently been proposed by Marcus (1975). He calls his approach "wait and see parsing." The basic idea is that backup is made unnecessary if the system is provided with a limited advance look at only a few words (or phrases) and if sufficient constraint is provided by semantic and pragmatic knowledge. He is currently constructing a case-frame based parser following these guidelines.

Another proposal by Riesbeck (1975) involves a somewhat more radical approach. Riesbeck thinks that grammatical knowledge should be thoroughly integrated with semantic and pragmatic knowledge to such an extent that it is no longer reasonable to speak of separate components. This represents the extreme position along a natural line of evolution of more and more intimate interaction between grammatical and other kinds of knowledge. Riesbeck has constructed a comprehension system that reflects this philosophy. Words in the sentence trigger "productions" (called demons by Riesbeck) whose goal is to build a representation of the sentence's meaning (using Schank's conceptual dependency structures, Schank, 1973). Some of these productions reflect grammatical knowledge such as generating the expectation for a noun following the appearance of an article. Others represent semantic and pragmatic information such as expecting a human recipient following an appearance of the word "give." Riesbeck proposes that such an organization has both conceptual elegance as well as the ability to parse deterministically.

A new "conceptual" grammar by Martin (1975) also promises to generate perspicuous, backup-free conversions from sentences into expressions in his semantic representation. Martin's work suggests the possibility that an internal representation might be closer to the "surface" structure of natural language than most workers have believed feasible.

Clearly there is an important interaction between grammatical knowledge and other kinds of knowledge. A purely pipeline model in which a grammatical analysis is followed by a semantic analysis and finally a pragmatic analysis is untenable. However, it is not yet clear which alternative among the following is preferable: (1) structuring the grammatical knowledge as a modular expert engaged in a dialogue with other experts or (2) distributing the grammatical knowledge with other kinds of knowledge in such a fashion that it is no longer appropriate to talk about separate syntactic, semantic, and pragmatic components. In any case, it is our expectation that in the next few years language systems will be constructed in which backup does not generally occur (except in the case of garden path sentences which are equally confusing to people). This is based on the observation that most procedural theories initially require backup, not because of the intrinsic properties of the problem area, but rather only because sufficient knowledge to disambiguate alternatives has not yet been incorporated into the system. This knowledge has gradually been formalized (as our earlier discussion of work by Winograd and Woods showed). Consequently, we are evolving toward the stage where knowledge is available to make the right choices at each decision point.

6.
        THE CLASSICAL, ROMANTIC, AND MODERN PERIODS OF AI
6.1   A Change in Perspective

In this essay, we have tried to give the flavor of the new perspective from which AI approaches traditional problems in linguistics and epistemology. Specific forms of this general change in perspective are related to a number of theoretical issues: the relative importance of procedural as opposed, usually, to "declarative" or "propositional" knowledge; the shift away from traditional logistical concepts (axiomatizations, logical deduction) for the representation and manipulation of knowledge to more highly structured and particular representations; and liberation from the need for highly uniform transformational methods by allowing procedural sentries or "frame-keepers" to populate the knowledge system. In each case we note a trend away from a traditional form toward a newer form. In each case experience has already shown us that such trends can go too far, and that there is often good reason for using the traditional forms. So, looking back over the brief history of AI, we see a dialectic process with three phases which might be termed the classical, romantic, and modern periods of AI.

In the earliest work (which might be termed the *classical period* extending

from the mid 1950s through the early 1960s), knowledge is represented in very traditional ways: for example as subject–predicate propositions or as important but relatively nonfundamental extensions of this to property lists. Reasoning is accomplished through uniform techniques such as search or theorem proving.

Typical programs of the classical period were often criticized by laymen on the grounds that "people don't think like that." (Of course, this reaction may be totally wrong, since people do not necessarily know what deep processes lie under the conscious layers of thinking.) However, whether or not these objections are right in principle, the programs of what we shall term the *romantic period* (extending from the late 1960s to early 1970s) do not evoke them nearly to the same extent. On the contrary, they sometimes appear to be nothing more than a new language for introspective material. Striking examples are the Evans ANALOGY program (Evans, 1968) and the Winston learning program (Winston, 1970).

This stage might be justifiably termed the romantic period for two reasons. One is the extent to which it represented an introspective endeavor, while the second is a notable tendency to argue for first one approach and then another as a complete theory of intelligence. The period began with an emphasis on network-like systems (Quillian, 1968) and ended with a concentration on the representation of knowledge in an entirely procedural form (Hewitt, 1971; Winograd, 1972).

Finally, in this the *modern period,* we begin to investigate more rigorously the conditions under which different kinds of knowledge are appropriate. Frames are extended from an essentially declarative data type to a procedural system through the incorporation of frame-keepers. Procedures, in order to allow debugging, are embedded in a web of declarative commentary specifying the purposes, requirements, bugs, and effects of programs. Procedural knowledge thus serves as a basis for action while propositional knowledge provides a basis for understanding the behavior of procedures and especially for knowing how to debug or change them.

6.2 *Some Unanswered Technical Issues in Frame Theory*

We have discussed a variety of fundamental epistemological notions—frames, procedural knowledge, control structure—and have utilized these concepts to provide a unifying thread to recent AI natural language research. We conclude this section with a collection of questions surrounding the frame idea, which have yet to be answered and which we feel will provide the focus for the next round of research in the field.

Learning. A theory that makes extensive use of prestored knowledge obviously faces a critical question: how is such knowledge obtained? Or, in other words, how does such a theory allow novel situations to be comprehended?

Reference and reasoning. Given that the system knows that feathers can be used to tickle (i.e., could successfully answer a question to that effect), it still

remains to describe how indirect reference or reference through description is accomplished. Presumably, there are some tickling instruments that we have never before considered, but that we nevertheless are able to understand in that role. How are such inferences made?

Difficulties with sentries. There are a variety of technical problems that arise in implementing the idea of a sentry as a demon. These include handling conflicts (i.e., situations where more than one demon is triggered by the current situation and each has an alternative meaning), forgetting (i.e., when should a demon become inactive), and context-switching (i.e., how are leaving, returning, and combining contexts handled).

Representation. In many ways, the frame idea has served as a metaphor for us to present various ideas about the representation of knowledge—both declarative and procedural. However, there are many representation questions that must be resolved in formalizing the notion. For example, how homogeneous is our representation of language to be? Are sentries (demons) to be represented in the same form as the basic frames? Are current computer languages such as Lisp adequate or do we require new programming constructs?

Control. Is knowledge of control implicit or explicit in a frame system? Control is explicit when in addition to basic facts the system also includes knowledge about when to apply these facts and how to resolve conflicts. Control thus becomes just another kind of knowledge to the system. On the other hand, control is implicit through the representation of knowledge as procedures, with ordering in the program, conditionals and calls determining the sequence of application. Which approach is to be preferred?

Level interactions. We have posited the existence of frame-based knowledge for verbs, nouns, contexts, and themes. Technical questions arise about how these various levels of knowledge interact. Is the organization essentially *hierarchical,* with each level looking only at the results of the interpretation of the level below it. Or do all levels examine the surface linguistic input (perhaps in addition to examining each other)? Is communication occasionally *heterarchical* with higher levels sometimes directing the analysis of lower levels and at other times being in turn directed by the results of the lower level analysis?

This represents the conclusion of our theoretical discussion of AI as a new style of epistemology and its impact on the understanding of language and comprehension. In the next section, we consider the relevance of these ideas to education.

7.. THE IMPACT OF ARTIFICIAL INTELLIGENCE

ON EDUCATION

7.1 *If the Machines Get Smarter, Will the Children Get Smarter?*

The relevance of Artificial Intelligence to education can be discussed along two different dimensions corresponding to the narrow and broader views of the

subject as "machine intelligence" and as "epistemology." The former view evokes the prospect of more useful machines, sometimes described under the label "Intelligent Computer-Aided Instruction" (Brown, 1975). The other view has possible consequences of a much deeper sort, for it can be taken as projecting a total restructuring of the content and the theory of education. But even if this radical prediction is not fulfilled, the cognitive style of AI has very immediate and realistic suggestions about what children should be taught and how.

These two views can be stated in a more dramatic way. The organizers of the HUMRO–NSF conference on "Ten Year Predictions of the Impact of Computers on Education" (HUMRO, 1975) chose as the theme of one session: "If the machines get smarter, will the children get smarter?" We think the deeper question is this: "Suppose that *smart* machines can be constructed. Should educators look to the machines themselves as instruments to make kids smart, or should they rather look at the *principles* which led to the machines becoming more intelligent to see whether these principles could be applied to children?" There is, of course, no need to choose. Our intention in so posing the question is to draw attention to the richer, but almost entirely neglected, second view.

This paper is not the place for a full discussion of the relevance of AI to education. Instead, we shall develop briefly three representative issues:

1.   The impact on education of the development of computer-based tutors with reasonable language comprehension abilities. We shall develop this topic under the heading "Intelligent Computer-Aided Instruction" (ICAI).

2.   The deeper question of whether the content of AI theories of language can have an impact on the very substance of linguistic studies in the schools, not just in the manner in which the current material is taught. We have chosen to call this topic "The Glass Box Approach to Language Studies."

3.   Whether the basic goal of AI in seeking unified theories of knowledge can be expected to have a profound impact on the entire educational curriculum, with the effect of decreasing the currently accepted division of education into disparate subject areas. We shall discuss this topic under the title "The Articulate Learner."

Again we remind the reader that our discussion is intended only to give the flavor of a few kinds of interaction that AI and education might have. For a discussion of related areas of research in psychology, linguistics, and AI, see Winograd (1976). For a deeper discussion of the particular interactions discussed here, see Goldstein (1974), Minsky and Papert (1974), and Papert (1972).

7.2   Intelligent Computer-Aided Instruction

The question cited earlier: "If the machines get smarter, will the children get smarter?" immediately evokes the comment "certainly not automatically." This follows from the deeper observation that the question is not primarily about

machines, but rather about the nature of human intelligence and the effectiveness of teaching. To see this, transpose it into the two subquestions: (1) Could an idealized personal human tutor make a kid smarter? and (2) Could all or some of the functions performed by this tutor be mechanized?

Answers to the first question will vary depending on what one thinks about controversial issues in the theory of (human) intelligence. But there would certainly be less variation in answers to the question: Could a highly intelligent, empathetic personal tutor enhance the intellectual development of children? (Whether this enhancement is to be described as increasing the child's intelligence or the effectiveness of his use of it is not germane here.)

There are very few programs that can lay claim to fulfilling the functions of an intelligent, insightful tutor. However, important ground-breaking research has been done by Carbonell (1970) and Collins, Warnock, and Passafiume (1975) working on the SCHOLAR project, and by Brown and Burton (1975) in their development of the SOPHIE program. The SCHOLAR project has developed a series of successively more sophisticated computer-based tutors for such domains as geography. The project has involved the development of natural language comprehension modules, the construction of domain-oriented problem solvers based on a semantic net approach to representation, and, most notably, the procedural representation of tutorial strategies.

Recent work by Collins, Warnock, Aiello, and Miller (1975) has involved a deep interaction between psychology and AI in the form of studying the kinds of reasoning that people typically use to handle questions in *open domains,* that is, concerning topics about which they know a great deal but not all of the relevant facts. People are typically quite good at making plausible inferences in such contexts. Collins' goal is to represent such reasoning strategies in a procedural way and then make that knowledge available to the SCHOLAR tutor for the purpose of allowing the tutor to help a student debug his own reasoning style.

Brown and Burton (1975) have developed a computer-based laboratory in which a student can learn electronic troubleshooting. The program SOPHIE contains an electronics problem solver that embodies a genuine understanding of the circuits involved, a natural language component that allows English question answering to occur between the student and program, and an understanding of certain kinds of tutoring strategies. We shall not develop the details of Brown's program further, except to remark on the following points:

1. The program is a milestone in computer-aided instruction (CAI) simply because of the extent to which it understands its subject matter. Previous CAI programs (and indeed most current ones) are notorious for their ignorance. The slightest deviation from the expected response on the part of the student in his answer results in the program erroneously judging the answer as incorrect. Brown's program utilizes a powerful simulation of the circuits under study to accurately predict the consequences of various faults. This kind of domain

competence is essential as a foundation for any CAI tutor, and we expect the incorporation of AI problem solvers into CAI programs to become more and more common in the coming years.

2. The tutoring strategies of SOPHIE are concerned with the nature of reasoning involved in building up a model of a problem from incomplete evidence. This is a fundamental kind of human reasoning. It addresses the basic issue of what kinds of questions a student should ask in attempting to understand a new subject, device, or electronic circuit. SOPHIE is able to judge whether a measurement requested by the student (as part of the process of tracking down a fault in the circuit) is a reasonable question on the basis of what the student already knows. If not, that is, if the answer to the question is already entailed by information previously obtained by the student, then the program draws the student's attention to the redundancy and discusses the information which the student knew but failed to consider.

This study of human cognitive processes through the formalisms developed by AI and the incorporation of that understanding into CAI programs is only in the formative stages. But it is our hope that in the coming years, the concept of personal computer-based tutors available to any student at any time will become more and more of a reality.

7.3   The Glass Box Approach to Language Studies

In the previous section, we saw the application of AI research on language comprehension and problem solving to the design of intelligent CAI programs. In this section, we develop a more radical application.

The ICAI tutors, no matter how intelligent, are essentially *black boxes* from the student's perspective. The possibility exists that the concepts developed to design those programs might themselves be worthy objects of study by the students. We propose calling this a *glass box* approach. Unlike the "black box" approach, the point here is precisely to encourage the young students to look inside the box, to ask how it works, and to gain insight into their own language by so doing. In using the phrase "glass box," we follow Peele's (1974) discussion of APL programs whose structure is available to the student to be studied and understood.

An example of the glass box approach is illustrated by suggesting a new kind of *language laboratory*. We envision a computer-based environment in which students would have the opportunity to explore the structure of language by designing, programming, and testing various kinds of language comprehension programs.

The virtues of such a laboratory approach lie in the active nature of the student's involvement, the possibility of the student developing his ideas in a personal way not rigidly limited by the teacher's approach (but reasonably circumscribed by the goals of designing a successful program), and the exposure

to new computational theories of language that are in many ways superior to traditional grammars as conceptual frameworks for the study of language at school.

As an example of the power of these new approaches to language, we mention the possibility that they hold out for a unified approach to the study of knowledge. This was illustrated earlier by our discussion of Frame Theory in which the same basic knowledge-representation techniques were seen to apply to grammatical knowledge as well as other kinds of knowledge concerned with themes, contexts, and semantic meanings. The exciting possibility exists of diminishing the compartmentalization of school knowledge. Linguistic knowledge becomes less of a "different" thing, apart from other knowledge.

Our series of roles for language understanding machines has moved progressively away from a focus on the machine to a focus on the linguistic ideas which make the machine possible. We conclude this essay by generalizing this approach beyond the linguistic domain.

7.4   The Articulate Learner

In this section, we develop further the notion of a unified approach to educational curricula by introducing the notion of *articulate learning*. Our discussion is motivated by the question of why students who find language studies in school challenging and who do well in them often find the mathematics curriculum a confusing and impossible maze. A traditional answer is that there is such a thing as quantitative mathematical abilities and qualitative language-oriented abilities. Such a belief is mirrored in all of the Achievement Tests that students are asked to take in which there are clearly demarcated language and mathematics sections.

We should like to raise the possibility that the traditional mathematics curriculum of grades 1–12 lacks certain dimensions which make it understandably obscure compared to language studies. In particular, its basic failing is in the extreme difficulty which it presents for a student or teacher who attempts to be articulate about the problem-solving process.

In language studies, a teacher can discuss the many dimensions of a story written by a student: its grammar, its theme, its narrative structure, the plot, the setting. In contrast, what can a teacher discuss about the failure of a student to solve an arithmetic problem or an integral? For example, we have seen a child go to the blackboard and solve an arithmetic problem in the following way:

$$
\begin{array}{r}
35 \\
+\ 35 \\
\hline
610
\end{array}
$$

The teacher involved had little to say besides observing that the answer was wrong. We believe that the earlier discussion of AI and epistemology has some

very definite suggestions for altering the mathematics curricula to allow the student and the teacher to be far more articulate about the knowledge and problem solving involved.

For the above example, we imagine an environment in which the first kind of knowledge that a student is exposed to is the structure of programs and the related ideas of planning and debugging. Skill in designing procedures lies at the heart of achieving competence in mathematics (and in writing essays as well). Given access to a vocabulary for programs, plans, and bugs, we believe that student and teacher could be articulate about describing the particular algorithm used by the student in reaching the "610" answer, in identifying the bugs, and in debugging the addition program to yield a correct result. For the above example, we imagine a discussion similar to the earlier description of the HACKER program's reasoning regarding its problem solving in the Blocks World. The student had mastered an addition procedure for single digits. His goal in this problem is to apply that procedure as a *subroutine* to the more complex goal of adding multidigit numbers. His solution indicates that he followed the reasonable *linear plan* of assuming that the sum of multidigit numbers could be achieved by adding the columns *independently*. Linear planning commits the problem solver to have some method of putting together the solutions of the subproblems which were arrived at independently. The answer "610" indicates that our student utilized the simple artifice of concatenating the two partial solutions. The *bug*, of course, was in assuming the complete independence of the column additions. The *debugging* is accomplished by taking account of this interaction through ordering the columnar additions right-to-left and utilizing a "carry" *data structure*.

The reader may find a vocabulary involving such terms as linearity, bug, debugging, independence, and data structure confusing; and must certainly believe that such a vocabulary is hardly usable with an elementary school student. This is true, of course, given the current curriculum. However, our proposal is not simply to add this vocabulary to the traditional curriculum. It is more far-reaching. The proposal is to redesign the curriculum to allow introduction of this and related concepts at an early stage.

This is being done in a preliminary way at MIT's LOGO project (Papert, 1972) in which new computation-based approaches to mathematics, science, and music are currently being developed. As an example, elementary school students are introduced to the above set of programming concepts and to traditional mathematical concepts in the context of programming a computer display to draw pictures of their choice (Fig. 4).

The children construct the programs themselves in LOGO, a programming language expressly designed for students. A successful LOGO experience involves not only the student being self-motivated to construct such programs (because of the enjoyment of creating pictures), but also involves discussions between student and teacher, between student and student, and between the student and
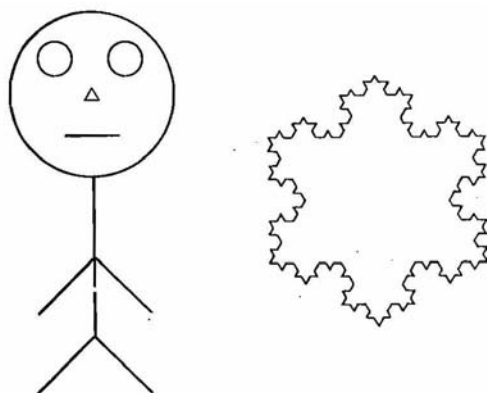
FIG. 4. Typical turtle figures drawn by children.

himself, in which the student becomes verbal about his reasoning and learns to
use a vocabulary describing his plans, his programs, and, most importantly, his
bugs.

The design of new educational environments in which the student is
introduced to the concepts that AI has found essential to the construction of
intelligent programs is only in a beginning state. But our experience with
children in teaching them turtle geometry, computer-based physics, and
computer-based composing leads us to believe that the approach is promising and
potentially revolutionary. Furthermore, it seems to lead to a situation in which
students can be articulate about their own problem solving, regardless of whether
it is in the mathematical or linguistic domain.

7.5   Epistemology and Education

The role of a knowledge-oriented AI approach to the study of language
comprehension has its counterparts in all areas of cognitive science, and each of
these carries implications for education.

We have argued again and again that a major thrust of AI research has been to
find representations of knowledge which lend themselves to effective integration
into larger knowledge systems. The analogous problem for education is obvious:
Are the representations of knowledge which have proven themselves effective in
augmenting machine intelligence also of value in augmenting human intelli-
gence? And, if not, does a similar inquiry into the forms of knowledge suitable
for human thinking yield a different set of modes of representation? Futhermore,
the epistemological paradigm for the study of learning asks questions like: What
knowledge does a good learner use in a given situation? What knowledge would
make learning a simpler process than it appears to an observer who does not
know this knowledge is being used, or to a learner who lacks it?

Very little attention has been paid to problems of this sort. But signs of a rich area of the utmost importance seem to be present in the results of such preliminary investigations as the construction of ICAI programs containing procedural models of human inference and learning, the design of new approaches to learning based on debugging theory, and the design of new curricula such as turtle geometry based on a notion of the articulate learner.

It is our belief that future explorations into the nature of language, of various subject domains, and of the elements of intelligence itself—if pursued from an epistemological perspective grounded in the computational metaphor—will provide the foundation for deeper theories of the nature of language and of the nature of learning. Furthermore, we expect such theories to have practical consequences in the form of programs that comprehend language and in the design of new kinds of computer-based educational environments.

EPILOGUE—CONTROVERSIES ABOUT EDUCABILITY AND INNATENESS

The general issues about the theory of efucation discussed in Section 7 cut across a number of acute current controversies about the relations between linguistic abilities in the developing child and other cognitive functions including learning. Different approaches to linguistics and to cognitive theory appear to lead theorists to very different opinions on questions such as the *domain specificity* of language skills; their relation to particular brain structures; and the extent of their *innateness*. For example, one sees a very sharp opposition between the Chomskian and Piagetian lines of thinking (Chomsky, 1975). For the former, there is a definite *linguistic competence* which is extremely specific to language (i.e., has very little overlap with other cognitive abilities) and which is determined by innate biological structures. Adherents of the Piagetian school, on the other hand, tend to see linguistic competence as sharing major cognitive processes with other intellectual domains and to be explicable more in terms of a psychodevelopmental process.

Any such brief summary of complex theories necessarily oversimplifies the respective positions. Nevertheless, the oppositions, though more modulated, are real and of fundamental and urgent importance. We see work in Artificial Intelligence as having a very direct bearing on this debate. This does not mean that we are able to *resolve* the issues. Rather we believe that *computational models* provide a theoretical overview necessary even for a clear statement of the issues. This belief is reinforced by the increasingly rapid spread of computational ideas into the work of psychologists and even philosophers whose views fall on both sides of these controversies. For example, see Fodor (1975) for the computational formulation of an extreme "nativist" position by a psychologist who disagrees fundamentally with much of the "main-line" AI trends. See Norman *et al.* (1975) and Bobrow and Collins (1975) for the views of psychologists sympathetic to the kind of AI theory treated in this paper.

In conclusion, though the debates about intelligence and cognition rage both in and out of AI, and the reader may have objected to many of the specific observations that we have made regarding frame theory and other issues, we believe that there has already occurred this fundamental and profound change: that the language of computation has become the proper dialect for discussing the basic issues of both psychology and education.

REFERENCES

ANDERSON, J. R., & BOWER, G. *Human associative memory.* Washington, D.C.: Winston, 1973.

BOBROW, D., & COLLINS, A. (Eds.) *Representation and understanding.* New York: Academic Press, 1975.

BROWN, J. S. Proposal for knowledge-based intelligent computer assisted instruction. BBN PROPOSAL NO. P75-CSC-17A, May 28, 1975. Bolt, Beranek, and Newman, Cambridge, MA, 1975.

BROWN, J. S., & BURTON, R. R. Multiple representations of knowledge for tutorial reasoning. In D. Bobrow & A. Collins (Eds.), *Representation and understanding.* New York: Academic Press, 1975.

CARBONELL, J. R. AI in CAI: An artificial intelligence approach to computer-assisted instruction. *IEEE Transactions on Man–Machine Systems,* 1970, MMS-11, 190–202.

CHARNIAK, E. Toward a model of children's story comprehension. AITR-266, Artificial Intelligence Laboratory, MIT, Cambridge, MA, 1972

CHOMSKY, N. Questions of form and interpretation. *Linguistic Analysis,* 1975, 1, 75–109.

COLLINS, A., WARNOCK, E. H., & PASSAFIUME, J. J. Analysis and synthesis of tutorial dialogues. In G. Bower (Ed.), *The psychology of learning and motivation.* Vol. 9. New York: Academic Press, 1975.

COLLINS, A., WARNOCK, E. H., AIELLO, N., & MILLER, M. Reasoning from incomplete knowledge. In D. Bobrow & A. Collins (Eds.), *Representation and understanding.* New York: Academic Press, 1975.

DRAKE, C. Semantic case frames in the understanding of connected discourse. Unpublished Masters dissertation, MIT, Cambridge, Mass., 1975.

EVANS, T. A program for the solution of geometric-analogy intelligence test questions. In M. Minsky (Ed.), *Semantic information processing.* Cambridge, Mass.: The MIT Press, 1968.

FILLMORE, C. The case for case. In E. Bach & R. T. Harms (Eds.), *Universals in linguistic theory.* New York: Holt, Rinehart, & Winston, 1968.

FODOR, J. A. *The language of thought.* New York: Crowell, 1975.

GOLDSTEIN, I. P. Summary of MYCROFT: A system for understanding simple picture programs. *Artificial Intelligence,* 1975, 6, 249–288.

GREEN, B. F., WOLF, A. K., CHOMSKY, N., & LAUGHERY, K. Baseball: An automatic question answerer. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought.* New York: McGraw-Hill, 1963

GREIF, I., & HEWITT, C. Actor Semantics of PLANNER-73. *Proceedings of the Second Symposium on Principles of Programming Languages,* Palo Alto, Calif., January, 1975.

HEIDORN, G. Augmented phrase structure grammar. In R. Schank & B. L. Nash-Webber (Eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing,* MIT, Cambridge, Mass., 1975.

HEWITT, C. Procedural embedding of knowledge in PLANNER. *Proceedings of the Second International Joint Conference on Artificial Intelligence,* 1971.

HUMAN RESOURCES RESEARCH ORGANIZATION. Ten year forecast for computers and communications: Implications for education. *Proceedings of TECHCON Conference,* Arlie House, Warrenton, Va., September, 1975.

KATZ, J. J., & FODOR, J. A.   The structure of a semantic theory in J. A. Fodor & J. J. Katz (Eds.), *The structure of language*. Englewood Cliffs, N.J.: Prentice-Hall, 1964.

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

MARCUS, M.   Diagnosis as a notion of grammar. In R. Schank & B. L. Nash-Webber (Eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, MIT, Cambridge, Mass., 1975.

MARTIN, W.   *Conceptual grammar*. Internal Memo 20, Automatic Programming Group, Project MAC, MIT, Cambridge, Mass., 1975.

MINSKY, M., & PAPERT, S.   *Artificial intelligence*. Condon Lectures, Oregon State System of Higher Education, Eugene, Ore., 1974.

MINSKY, M.   A framework for representing knowledge. In P. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill, 1975.

MOSES, J.   *Symbolic integration*. MAC-TR-47, Project MAC, MIT, Cambridge, Mass., 1967.

NEWELL, A.   Artificial intelligence and the concept of mind. In R. C. Schank & K. M. Colby (Eds.), *Computer models of thought and language*. San Francisco, Calif.: Freeman, 1973.

NORMAN, D. A., RUMELHART, D. E., & the LNR RESEARCH GROUP.   *Explorations in cognition*. San Francisco: Freeman, 1975.

PAPERT, S.   Teaching children thinking. *Journal of Structural Learning*, 1975, **4**, 219–229.

PEELE, H. A.   Computer glass boxes: Teaching children concepts with A Programming Language. *Educational Technology*, 1974, **14**(4).

QUILLIAN, M. R.   Semantic memory. In M. Minsky (Ed.), *Semantic information processing*. Cambridge, Mass.: MIT Press, 1968.

RIESBECK, C. K.   Computational understanding: Analysis of sentences and context. Working paper, Fondazione Dalle Molle per gli studi linguistici e di comunicazione internazionale, Castagnola, Switzerland, 1974.

RIESBECK, C. K.   Computational understanding. In R. Schank & B. L. Nash-Webber (Eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, MIT, Cambridge, Mass., 1975.

RUMELHART, D. E.   Notes on a schema for stories. In D. Bobrow & A. Collins (Eds.), *Representation and understanding*. New York: Academic Press, 1975.

SCHANK, R., GOLDMAN, N., RIEGER, C., & RIESBECK, C.   Primitive concepts underlying verbs of thought. AIM-162, Stanford University, 1972.

SCHANK, R.   Identification of conceptualizations underlying natural language. In R. C. Schank & K. M. Colby (Eds.), *Computer models of thought and language*. San Francisco: Freeman, 1973.

SCHANK, R. C.   Using knowledge to understand. In R. Schank & B. L. Nash-Webber (Eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, MIT, Cambridge, Mass., 1975.

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

SIMMONS, R. F.   The clown's world. In R. Schank & B. L. Nash-Webber (Eds.), *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, Mass., 1975.

SLAGLE, J. R.   A heuristic program that solves symbolic integration problems in freshman calculus. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought*. New York: McGraw-Hill, 1963.

SUSSMAN, G. J.   *A computational model of skill acquisition*. New York: American Elsevier, 1975.

WILKS, Y.   Natural language understanding systems within the A. I. paradigm: A survey and some comparisons. In *Artificial intelligence and language comprehension*. National Institute of Education, 1976.

WINOGRAD, T. *Understanding natural language*. New York: Academic Press, 1972.

WINOGRAD, T. Frame representations and the declarative/procedural controversy. In D. Bobrow & A. Collins (Eds.), *Representation and understanding*. New York: Academic Press, 1975.

WINOGRAD, T. Artificial intelligence and language comprehension. In *Artificial intelligence and language comprehension*. National Institute of Education, 1976.

WINSTON, P. Learning structural descriptions by examples. TR-76, Project MAC, MIT, Cambridge, Mass., 1970. Reprinted in P. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill, 1975.

WOODS, W. Transition network grammars for natural language analysis. *Communications of the Association for Computing Machinery*, 1970, 13(10).

WOODS, W. *Natural language communication with computers*. BBN Report No. 2976, Vol. 1. Bolt, Beranek, and Newman, Cambridge, Mass., 1974.