CONCEPTUAL OVERLAYS:
A MECHANISM FOR THE INTERPRETATION
OF SENTENCE MEANING IN CONTEXT

Chuck Rieger
Department of Computer Science
University of Maryland
College Park, Maryland 20742

Abstract

This paper describes a context mechanism for a natural language understanding system. Since no sentence is ever perceived outside some context, it is reasonable to inquire into the nature of context as it affects the interpretation of sentence meaning at a deep conceptual level. A theory, called <u>conceptual overlays</u>, is described. This theory (1) defines $C(T_1, \ldots, T_i)$, the context established by the meaningful sequence of thoughts $T_1 \ldots T_j$: (2) defines $I(T_{j+1}, C(T_1, \ldots T_j))$, the high-level interpretation of Ti+i *in* the context established by Ti,...,Ti; and (3) specifies an effective algorithm and data structure for computing I(t,K) for arbitrary thought T in context K. In particular, a prototype LISP system, EX-SPECTRE-1, which solves simple cases of $I(T_2, C(T_1))$ is described. The system is based on *an* expectancy/fulfillment paradigm. Expectancies are spontaneously activated by a pattern-directed invocation technique. Each expectancy implicitly *references* large chunks *of* <u>common-sense algorithms</u>. A collection of such implicitly activated algorithms constitutes context, and the interpretive process is one of identifying future input as steps in these algorithms. Context switching and uses of I(T,K) in a language comprehension system are discussed.

Introduction

The goal of this research is to synthesize a domain-independent theory of how context, specifically expectancy, influences perception and interpretation of natural language meaning stimuli. I want here to examine the specific problem of interpreting <u>actions</u> in context, and to describe a general theoretical approach to its solution, because I believe many of the issues of this specific problem overlap with the Issues of most other problems of context.

The statement of the task of interpreting actions in context can be formulated as follows: Given a "meaningful" sequence, $T_1 \ldots T_j$, of syntactically, referentially and conceptually unambiguous sentences (actually, the sequence of <u>thoughts</u> underlying them, expressed in some meaning formalism), assign a meaning interpretation to the 1+1st similarly unambiguous sentence 1n a way which elucidates its relationship to the context, or situation, established by $T_1 \ldots T_j$.

This task engages four issues: (1) What is a reasonable definition of C(T],... ,Tj), the context established by $T_1 \ldots$ Ti? (2) What is a meaningful definition of I(T,K), the interpretation of sentence T in context K? (3) What is the role of inference in the contextual interpretive process (and at what point and to what richness are inferences made)? (4) How do contexts begin, switch, end and interact? This paper concentrates on (1) and (2), describing a method and a prototype system, EX-SPECTRE-I, for obtaining I(T2,C(T-|)), where T2 describes some volitional action by an actor who is acting in the context established by T_1.

Specific Goals and Examples

There have been four goals in the first phase of building this theory and model:

<u>GOAL 1</u>: To elucidate the relationship of action T_2 to the context $C(T_1)$.
<u>Example</u>: T_1: Pete stole Jake's cattle.
T_2: Jake saddled his (Jake's) horse.

<u>GOAL 2</u>: To account for how different contexts prescribe different interpretations of the same thought.
<u>Example</u>: T_{1a}: John saw the thunder clouds and felt the first drops of rain.
T_{1b}: <u>John dived under his jeep</u>.
T_{2a}: John heard a thud and saw the precious water leaking from his jeep's radiator.
T_{2b}: <u>John dived under his jeep</u>.
T_{3a}: John didn't want the cops to see him.
T_{3b}: <u>John dived under his jeep</u>.

<u>GOAL 3</u>: To alter T_1's influence on T_2 by preceding T_1 with some T_0.
<u>Example</u>: T_0: Jake asked Pete to steal his (Jake's) cattle to teach Sara a lesson.
T_1: Pete stole Jake's cattle.
T_2: Jake saddled his horse.

<u>GOAL 4</u>: To account for how a "peculiar" T_2 can be detected and how it can change the influence nominally exerted by T_1.
<u>Example</u>: T_1: Pete stole Jake's cattle.
T_2: Jake smirked.

It is felt that the specific mechanisms underlying these four varieties of binary and ternary interaction will prove to be characteristic of many other context-related mechanisms. At the time of writing, EX-SPECTRE-I has met the first

* There will be many interesting interrelationships between the present theory and the processes by which meaning is extracted from real-world sentences and perceptions which contain syntactic, referential and conceptual ambiguities. The assumption made here that sentences are unambiguous helps separate the influence of context from the influences of the various other processes.

and second goals for restricted examples.

Background

The present theory of conceptual overlays 1s the next step 1n the development of the model of memory and inference described in (R1) and (R2), which proposed that a spontaneous, non-goal-directed substratum of language-independent inference is requisite to even the simplest forms of language comprehension. As will be evident, the present theory of overlays is in accord with the philosophy verbalized by Abel son in (A1) and by Minsky in (MI) that perceptions and events can be interpreted only within relatively large "themes" (Abelson) or "frameworks" (Minsky). The theory also relates to the idea of a "demon", used by Charniak in (CI), and ties in closely with the goals of Schmidt's model of personal causation (S3), which deals with some of the same issues from the point of view of social psychology.

Conceptual Overlays

For the sake of concreteness, acknowledging the potential loss of generality, it will be useful to thread the discussion through the details of one particular example. At least this will connote the main ideas of the theory. The example will be the one used earlier:

T_1: Pete stole Jake's cattle.
T2: Jake saddled his (Jake's) horse.

The goal 1n this example will be to explain how Jake s saddling his horse might relate to Pete's act of theft. The theoretical setting 1s a conceptual memory of the sort described in (RI) and (R2), which receives as its input meaning graphs which have been constructed from input sentences by an autonomous parser of the sort described by Riesbeck in (R4), and which contains the low-level inference reflex mentioned above. In order to communicate some representation-independent ideas, most problems of meaning representation will be unabashedly ignored by using English-like notation. It is assumed that a suitably expressive system of representation (such as Schank's Conceptual Dependency (S2)) is used throughout.

To begin with, we would expect reasonable interpretations of this example to follow the lines: "Jake is going after Pete to get his cattle back," or "Jake is going into town to see the sheriff." These interpretations relate to this specific T_2. Of course, there could have been a limitless range of other T_2's in place of this particular one: "Jake sat down and wept" "Jake took out his rifle," 'Jake smirke-j," "Pete was arrested next day," "Jake lingered over his morning coffee in deep thought," "Pete celebrated," and so forth. The important test of the theory is that it be able to relate these as well. The requirement of the theory is therefore that it provide a format wherein general expectancies can be maintained, providing enough implicit slots so that very diverse subsequent input can fit later. Hence the term "overlay" has been used to suggest a superimposable piece of cellulose with boxes drawn on it which say: "If you see such-and-so occur, here is where it fits into the laroer scneme of things at the moment."

Action Overlays

Entities called action overlays are the vehicles for storing and organizing bundles of expectancies. An action overlay, A, is a data structure consisting of five components: (1) a set of interceptors, 1_A, *(2)* a universe of potential expectancies , E_A, (3) an expectancy selector function, S_A, (4) a set of overlay switchers , WA, and (5) a set of termination handlers! HA. Fig. 1 shows an action overlay which relates to acts of theft. Fig. 2 is the corresponding LISP data structure in use by EX-SPECTRE-1. (Recall that, in principle, the English-liko notation now in use will eventually be replaced by a formal system.)

Interceptors

The set of interceptors, 1A, specifies conceptual patterns which can trigger the action overlay. A, of which they are a part. As each new input, T, arrives in the system (either from the sentence analyzer directly, or as an inference from the inference component of the overlay system), the interceptors of every inactive action overlay in the system are compared to T. If at least one interceptor is satisfied, the overlay becomes active. The process of action overlay activation in EX-SPECTRE-1 is therefore a pattern-directed invocation scheme.

There will be quite a large number of action overlays in a full-scale system, each dealing with some relatively narrow situation and its related expectancies. Because of this, and since situation descriptions will frequently overlap, more than one action overlay will generally be found applicable to a given input, meaning that several situations are in progress simultaneously. As will be seen, multiple overlays exert their influence concurrently on subsequent inputs.

In the overlay of Fig. 2, only one interceotor has been included. When the input

(PETE STEAL CATTLE FROM JAKE)

arrives, this overlay is activated in EX-SPECTRE 1. Activation begins by binding X to PETE, Y to CATTLE, Z to JAKE, and H to the entire pattern for later reference in the selector function.

Potential Expectancies

E_A, the universe of potential expectancies, is a set of "what next" describers which enumerates at a high descriptive level the universe of activities which might possible follow the triggering input without "violating the spirit" of the overlay (and hence cause a switch to another action overlay). The information in Fig. 2 associated with the feature UNIVERSE comprises the set of potential expectancies for this theft overlay. By "violates the spirit" I mean some subsequent activity which would cause the comprehender to be surprised - something which somehow deviates from what is normally expected after a theft.

To suggest that the range of possible "what next" activities can be captured by a relatively small, enumerable set of expectancies might seem unrealistic. I do not selieve it is. To be sure, there is an infinite ange of "what nexts" which is surely impossible to anticipate directly. But the obvious phenomenon of comprehension is that, given any one of the infinitely many subsequent events which might occur or be described next, we are usually able to say in retrospect "Yes, that fits here in what I was vaguely expecting, and here's how: ..." So rather than cope with detail, an expectancy should simply specify the kinds of activities reasonably expected to follow. Algorithmic knowledge can fill out the details, as will be shown. An expectancy simply serves to carve off a manageable chunk of relevant world knowledge from what would otherwise be an enormous, unrestricted potpourri of "what nexts".

Expectancy Selector Function

The universe of expectancies simply enumerates potentials; not all members of EA will be equally applicable across all situations, and within a given situation (that is, the particulars of who stole what from whom), some elements of EA will be more salient than others. Consider acts of theft of the form (X STEAL Y FROM Z).

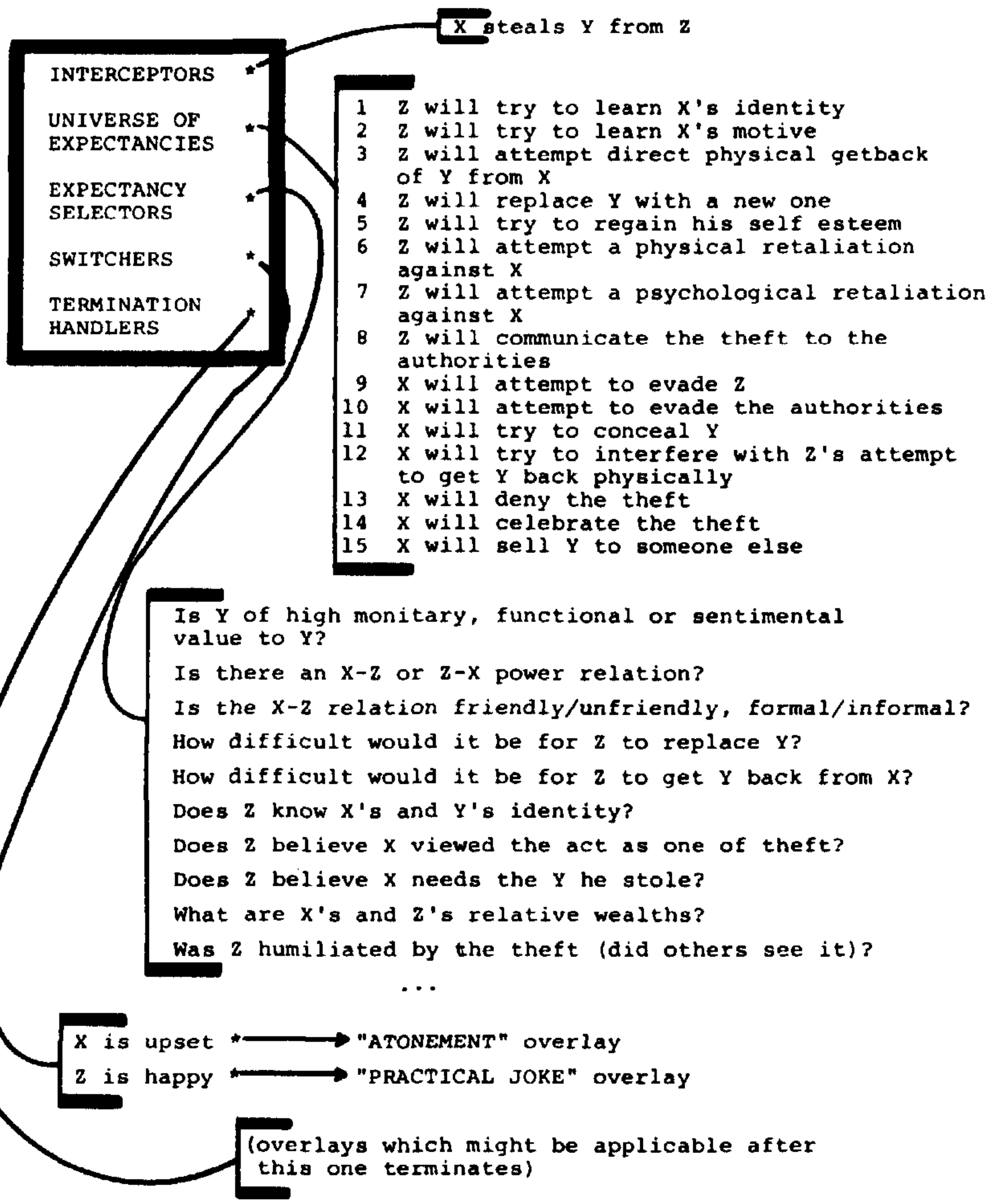

Figure 1. The action overlay relating to thefts.

```
((REFERENCE-NAME OVERLAY1)
 (VARIABLES X Y Z H)
 (INTERCEPTORS (H (X STEAL Y FROM Z)))
 (UNIVERSE
   (1 (Z LEARN (IDENTITY X)))
   (2 (Z LEARN (MOTIVE-OF X IN H)))
   (3 (Z PHYS-GETBACK Y FROM X))
   (4 (Z REPLACE Y))
   (5 (Z CAUSE (POSCHANGE Z)))
   (6 (Z CAUSE (NEGCHANGE X PHYS-STATE)))
   (7 (Z CAUSE (NEGCHANGE X PSYCH-STATE)))
   (8 (Z COMMUNICATE H TO AUTHORITY))
   (9 (X EVADE Z))
   (10 (X EVADE AUTHORITY))
   (11 (X CONCEAL Y))
   (12 (X INTERFERE-WITH
          (Z PHYS-GETBACK Y FROM X)))
   (13 (X DENY H))
   (14 (X CELEBRATE))
   (15 (X SELL Y TO ANOTHER)))
 (SELECTORS SELECTNET1)
 (SWITCHERS ((Z BE HAPPY) OVERLAY2)
            ((X BE UPSET) OVERLAY3))
 (TERMINATORS))
```

NOTE: SELECTNET1 points to the structure shown in Fig. 3.

Figure 2
The LISP data structure for the theft overlay of Fig. 1.

Within this paradigm there are many specific fac tors which can assist in the prediction of the relevance of each potential expectancy in EA. For example, if Z already knows X's identity, then his future activities should not be expected to include the determination of X's identity. Likewise, such factors as X and Z's relationship (student-teacher, husband-wife, jailer-inmate, neighbor-neighbor, etc.), 7's belief about X's motivation for the theft (if ascertainable), and the relative value of Y to Z, will all be good clues for deriding, say, whether Z can be expected to attempt a physical get-back of Y from X or a psychological retaliation against X, or both. If some sort of retaliation seems in order some notions of degree and kind will be of importance - Jake is less likely to stick his tongue out at Pete than to break down his fence! Fig. 1 contains a partial list of relevant factors for the theft overlay which might be reasonable to include in a larger system."EX-PECTRE-1 contains only a subset of these tests, as shown in Fig. 3.

The tests of Fig. 3 are made by the expectancy selector function, SA. SA is currently implemented as a "ternary transition network", serving as a discrimination network. "Ternary" refers to the characteristic that each node in the net branches three ways: "yes", "no" and "don't know". Nodes in the net are memory queries and each arc has associated with it a set of saliency setters - assignment statements which specify estimations of the saliency of various elements of EA when the arc is followed. All saliency estimations *are* preset to 0.5 out of 1.0; the net changes only those for which definite clues are present one way or another.

S_A is intended to model the phenomenon wherein a human comprehender seems to acquire some *very* cursory, rough gestalt of the situation, or in other words, which expectancies in EA are salient and which are not. SA therefore imposes only a rough ordering on EA so that those elements presumed most salient can receive more attention in subsequent PARTOF searching to be described. An "incorrect" judgement of saliency will in principle never preclude comprehension; it will simply increase the processing time required to discover relationships lat-*v on, and perhaps cause an inability to prefer one interpretation over another in a case where there are several competing interpretations. In a sample run, EX-SPECTRE-1 produced the saliency vector indicated by the dashed lines in Fig. 3 for the theft example.

The Expectancy Cloud

Having been roughly assessed for saliency, all members of EA *are* thrown into a central expectancy cloud, EC. Each expectancy in EC is backlinked to its contributing overlay, since the cloud will generally contain expectancies from numerous concurrent overlays, and since final interpretations will need to reference the contributina overlay.

A person is never fully aware at each moment exactly what his expectancies are. Apparently, an expectancy is a subliminal thing. The cloud, EC, therefore, although explicit and composed of discrete and isolatable expectancies, is intended to model a component in the human which is not part of his immediate awareness from moment to moment, but which in retrospect generally seems to have contained most of the necessary information for contextual interpretation.

The expectancies in this cloud define the context at each moment in the model.

Stepwise Indexed Algorithms

An active expectancy in EC describes some activity which is anticipated to have some chance of occurring at some future point. Such an activity can be characterized by a commonsense algorithm, or set of algorithms, where an algorithm is some temporally sequenced, hierarchical collection of subgoals, or steps. As one penertates deeper into the hierarchy, steps become more specific and the alternatives more numerous. As a basic data structure, the AND/OR graph provides an effective declarative (as opposed to procedural) method of specifying commonsense algorithms. (See (NI) for instance).*

Since an expectancy is an anticipated activity, and since activities can be represented

*NOTE: Since the time of first submittal of this paper, a new formalism for representing commonsense algorithms has heen developed and partially implemented. The new formalism renlaces the AND/OR graph, which lacks the structural constraints needed for representing general commonsense algorithms. See (R3).

by algorithms, each expectancy in EC is in fact no more than a pointer to one or more commonsense algorithms; hence each expectancy implicitly references al assible steps in those algorithms. *If a* subsequent input, T, *can be* identified as a step in the algorithms referenced by some expectancy in EC, this identification will relate T to the current context, K (represented by EC) thereby providing an interpretation, I{T,K). This relating of an input as a step in some activity which has been anticipated via an action overlay is the central idea of the theory.

The collection of commonsense algorithms, the algorithm base, used for the contextual interpretation process is the very same collection of algorithms the comprehender would use in actually getting about in the world (first-hand problem solving). This is a crucial point in my opinion, since it directly brings the world model to bear on the process of comprehending lanquage.

However, the interpretive task is the inverse of the executory task; rather than "translate a goal into some plan of action," the task

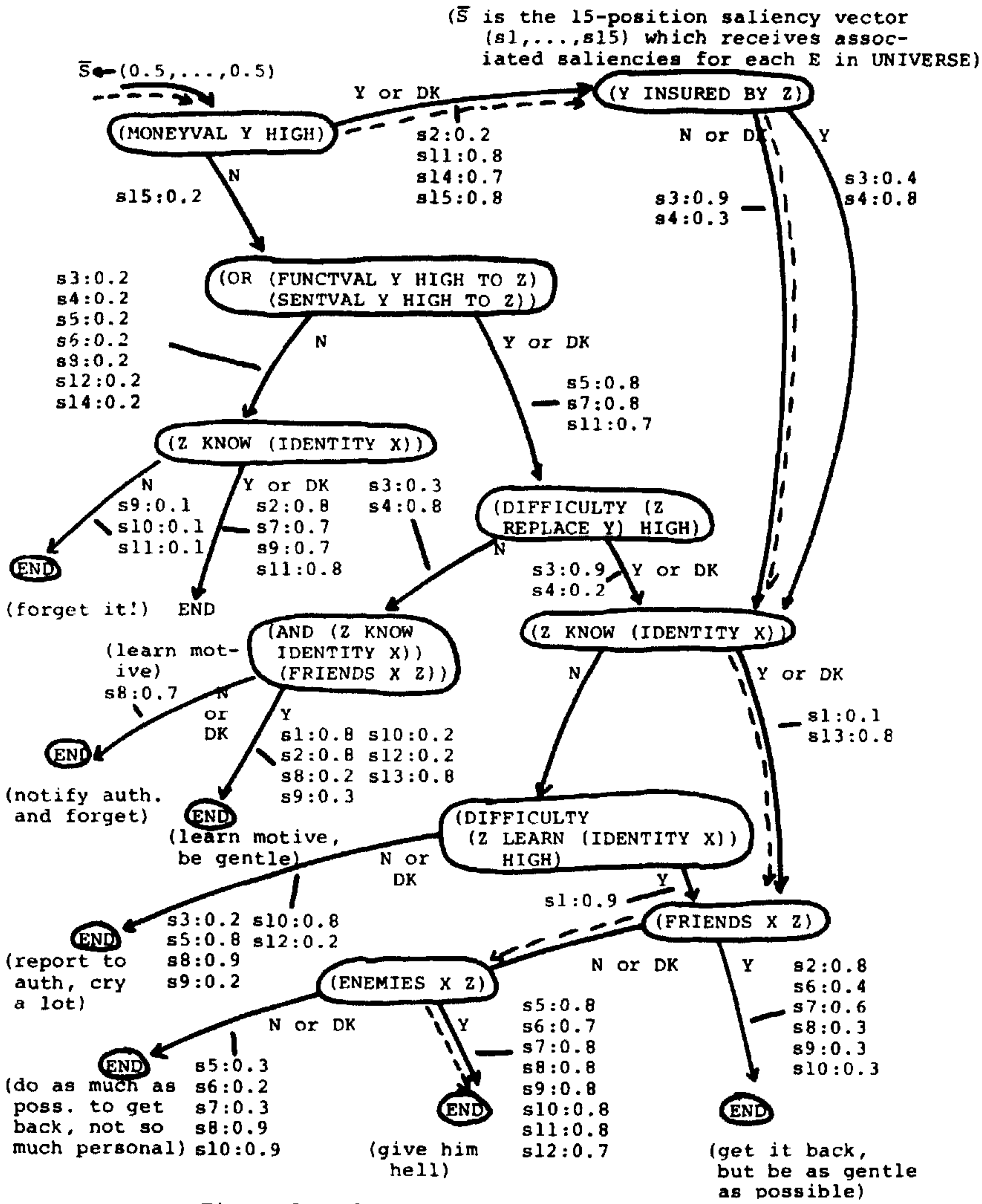

Figure 3. Selector function discrimination net for the theft overlay of Fig. 2.

is "ascribe a aoal to some observed plan of action," or more specifically, 'characterize some observed action as a step in tne attainment of some inferred goal." Because of this difference in use, the data structure used in EX-SPECTRE-1 for storinq commonsense algorithms is an augmented AND/OR graph which I have called a stepwise indexed algorithm (SIA). In an SIA, each specific step in the alaorithm is represented by (1) a pointer to a step schema, and (2) relevant binding information relating the instance of the schema to the schema. For instance, instead of writing (P GOTO DRUGSTORE) as a step in some algorithm for getting rid of a headache, this GOTO step, as well as all other GOTO steps in the entire algorithm base, 1s represented by a pointer to the step schema (X GOTO Y), with the appropriate bindings, ir. this case, X:P and Y:DRUGSTORE. In addition, each step in the SIA is "father-linked" to its parent step. This will permit traversing an SIA bottom-up, from a step to higher level qoals of which it is a part.

Each step schema is back!inked to each of its occurrences in the algorithm base through its occurrence set. This provides an index into the algorithm base to all points where the schema is referenced. For the (X GOTO Y) schema, this occurrence set would probably be quite large in a full-scale system. Hence, finer distinctions among the various types of GOTO (as one example), based on the specific conceptual features of X and Y, would probably have to be made by the indexinq scheme to improve on the search efficiency of the PARTOF function, about to be described.

Fiq. 4 shows the SIA and related steD schema for oart of a physical get-back goal, referenced by one of the expectancies in the cattle theft example.

PARTOF Searching Upward Through SIA's

SIA's and step schemas are nut to use as follows. In addition to beinn scanned by the interceptors of all inactive action overlays, each input to the overlay system is matched to some step schema. If the input is unambiguous, and if a suitably formal system of representational primitives is used to represent the meaning of each input, then at most one schema will match each input. In the cattle theft example, "Jake saddled his horse," represented in EX-SPECTRE-1 as (JAKE PUT SADDLE ON HORSE), matches the step schema (X PUT Y ON Z). This schema represents the more general concept of an actor putting some object on top of another.

Having matched the input to some schema, the task then becomes one of locating occurrences of that schema in some active expectancy in EC by upward searches from each of the schema's occurrences in the algorithm base. This operation is performed in EX-SPECTRE-1 by calling the function (PARTOF <step > <algori thm>) for <step> = (X PUT Y ON Z) and <algorithm> varying over the members of EC, in the order of highest saliency first. Identifications of the schema in some member, E, of the cloud define upward paths from the schema to E through SIA's. Such a path, from a step to an expectancy, is defined to be the **contextual interpretation of the sentence, I(T,K).**

Fig. 4 shows by dotted lines one path of length 3 which EX-SPECTRE-1 discovers as an interpretation of "Jake saddled his horse" in C("Pete stole Jake's cattle"); Jake is going directly after Pete to get his cattle back. Another interpretation not shown, but which is also discovered, is in the "Jake will inform the sheriff" expectancy: Jake is going to see the sheriff.

As a point of engineering, the motivation for searching upward through SIA's (from the step schema to the element of EC), rather than downward (from the members of EC to all possible step schema) has to do with the phenomenon that the typical commonsense algorithm tends to become quite specific after just a few levels. That is, even though the top few levels of an expectancy describe general subgoals, the lower levels beqin quickly to deal with the particular details of the objects and people involved. Put another way, the infinity of detail in a commonsense algorithm is a phenomenon of breadth rather than depth. Because of this, if the occurrence set indexing is refined enough, the number of relevant occurrences of a schema referenced by a particular input can be made quite small. This means that the branching factor in an upward search will be small in comparison with the branching factor coming downward from the expectancy to the step. To illustrate, even though the full occurrence set for the schema (X GOTO Y might be large, the subset of occurrences indexed by the particular form (X GOTO STORE) would only be *2* or 3 in number (e.g. it occurs in an algorithm for buying something, or returning something, and perhaps one or two others).

Multiple Interpretations

In any given context $C(T_1, \ldots, T_j)$, there will generally be several competing interpretations. A human comprehender seems usually to be able to prefer one over the rest. Although it is too early in the research to prescribe exactly how such a preference should be made by the model, two obvious factors are (1) saliencies of expectancies and (2) interpretation~path length. That is, a fit of a step into an expectancy *of* high saliency should be preferred over a fit of that step into one of low saliency. In conjunction with this, it would seem reasonable to prefer shorter interpretation paths over longer, more remote ones. In fact, in a full-scale system, it would probably be necessary to have PARTOF give up its scanning after, say, 10 levels of some SIA; after that point, even if some interpretation could be found, it would almost certainly be obscure.

Related to these ideas is the conjecture that sequences of purposively-constructed communication, say as found in children's stories, are organized so that the average contextual interpretation path length will be relatively constant across all cultures, relative to a typical algorithm base within each culture. Perhaps the constant is 3, perhaps 8; it would be interesting to ascertain whether or not such a metric exists, and if so, to discover its range. That is, on the average, how far do we have to search to connect each thought in with the context? Too short and we become red, too long and we become lost.

Uses of I(T,K)

An interpretation is a useful thing in itself. However, once an interpretation path is discovered, two other context-related operations become possible. First, all steps "to the left" of the identified step, S, in algorithm A - those steps which must logically have preceded S in A - can be generated as inferences. That is, if we know that Jake has already saddled his horse, and this has been Interpreted as part of a physical get-back, then Jake must already know who and what X and Y were, have decided on some plan of action, perhaps qotten his gun, and gone to the barn. Such inferences relate to the

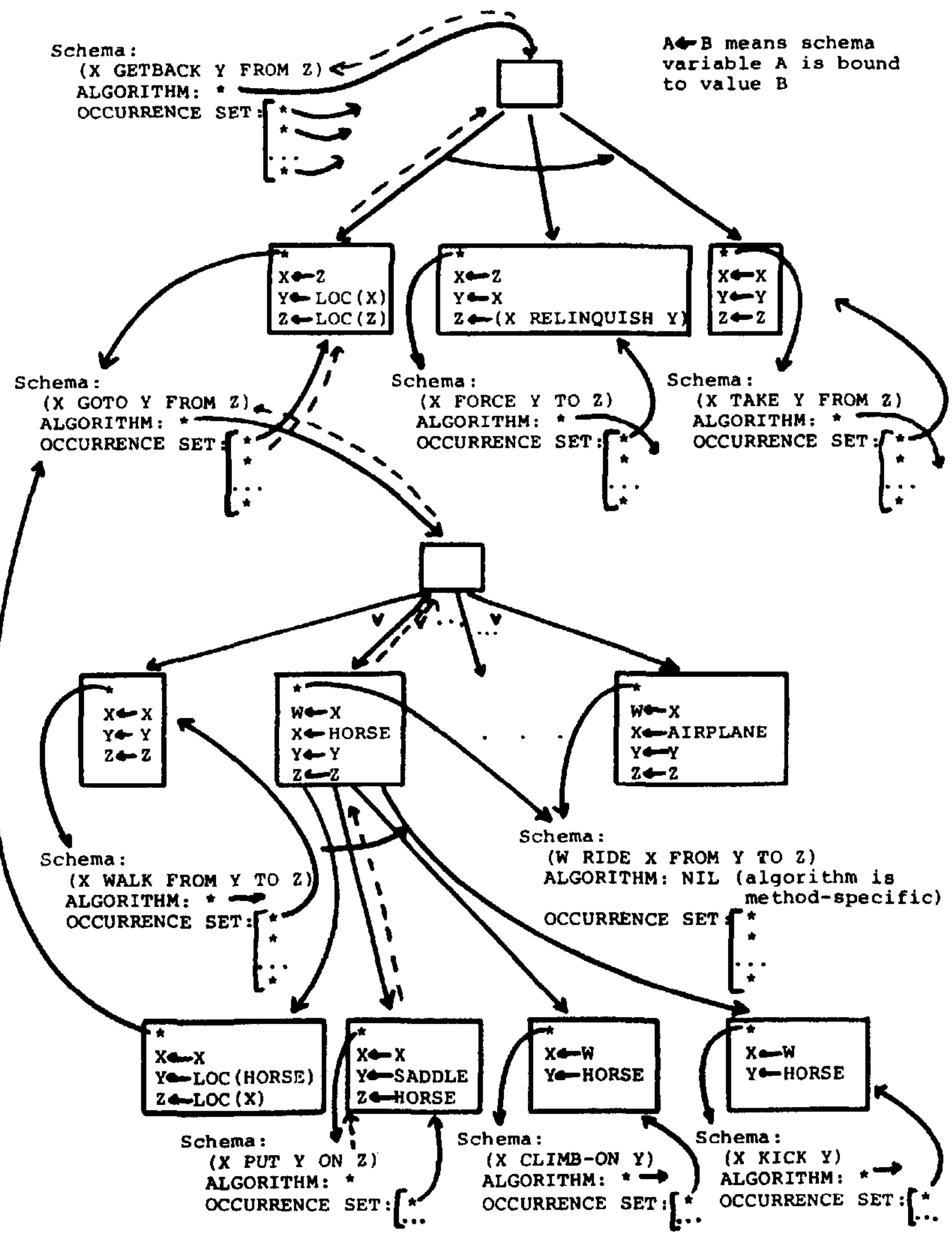

Figure 4. Stepwise-indexed algorithm (SIA) for the physical getback of an object, and related step schema.

class of conceptual inference termed "enablement inference" in the model of (RI).

The second use is that the original high-level expectancy in EC can be replaced by the set of remaining subgoals in A - those "to the right" of S in A. At that point, the system has begun to "track" a particular algorithm, and can descend from general to more specific expectancies which are the subgoals remaining to be fulfilled in the original expectancy. This narrowing *of* expectancies is perhaps a rough approximation of the way expectancies evolve from the general to the specific during the course of a human's comprehension of a story or sequence of events.

Overlay Switchers

In a system which relies upon expectancy as its primary mechanism, it seems to be as important to specify negative expectancies as well as positive ones. That is, to make explicit what classes of behavior would be "anomalous" in the current context, and to specify what the system should do when it detects an anomaly.

The switchers, W_A, specify these negative expectancies by making explicit a set of patterns which are overtly anomalous in the context the overlay represents. These patterns become active at the time A is initially activated, and serve as interceptors for subsequent input which "violates the spirit" of the overlay. Associated with each negative expectancy is a pointer to another overlay to which the system should switch in case that negative expectancy is realized.

In the prototype system, there *are* just two switchers for the theft overlay; one of them is: (Z BE HAPPY): 0VERLAY2, where 0VERLAY2 represents the bundle of expectancies relevant to classes of practical jokes. At the time of writing, this negative expectancy can intercept anomalous input and prepare to switch, but no switching actually occurs yet. Details of how accumulated information can be transfered from one overlay to another have not been considered yet.

Related Issues

There is not room here to discuss all the related topics and questions raised by the conceptual overlay approach to context. For a fuller discussion, the reader is referred to the unedited version of this paper and to the new formalism for commonsense algorithms (R3), both available upon request.

Conclusions

While the task of interpreting actions in context is only one aspect of "the context problem", its applicability is broad; even thouoh much of what we perceive is static information about the world, it usually relates, via relatively low-level inference, to some action or another. In this sense, nearly everything relates one way or another to algorithmic knowledge. And within this domain, one basic mechanism is as applicable to understamling why the president summoned the tape-erasing technician as it is to understanding why Mary wanted the car keys when she was hungry.

Specific conclusions are not yet in order. However, a general conclusion to be drawn is that algorithms play a central role in the interpretation of volitional actions in context. Since algorithms presumably account for a large portion of a person's belief system, the theory is actually one of how interpretation of perceptions is influenced by beliefs. Although the issues discussed have of necessity been limited *in* scope, I believe that the basic idea will prove to be fundamental to most other context phenomena.

Current plans are to continue the development of EX-SPECTRE along the lines of the new commonsense algorithm formalism. Then, after seven more prototype version numbers, it will be time to rename the system!

Acknowledgments

My thanks to Dave Levy and Bob Abelson, who taught me about cattle thefts last summer.

References

AI Abelson, R., "Frames for Understanding Social Actions," paper for Carbonell Conference, 1974

A2 Abelson, R., "The Structure of Belief Systems in Computer Models of Thought and Language, Schank and Colby (eds.), W.H. Freeman, 1973

CI Charniak, E., "Toward a Model of Children's Story Comprehension," Doctoral dissertation, M.I.T. AI TR-266, 1972

MI Minsky, M. , "A Framework for Representing Knowledge," M.I.T. AI TR-306, 1974

NI Nilsson, N. , Problem Solving Methods in Artificial Intelligence, McGraw-Hill, 1971

RI Rieger, C., "Conceptual Memory: A Theory and Computer Program for Processing the Meaning Content of Natural Language Utterances," Doctoral dissertation, Stanford AIM-233, 1974

R2 Rieger, C., "Understanding by Conceptual Inference," Amer. Journal of Computational Ling., microfiche #13, 1974

R3 Rieger, C., "The Commonsense Algorithm as a Basis for Computer Models of Human Memory, Inference, Belief and Contextual Language Comprehension," in Proc. Workshop on Theoretical Issues in Nat. Lang. Processing, 1975

R4 Riesbeck, C. , "Computational Understanding: Analysis of Sentences and Context," Doctoral dissertation, Stanford AIM-238, 1974

SI Schank, R. , Goldman, N., Rieger, C. , and Riesbeck, C, "MARGIE: Memory, Analysis, Response Generation, and Inference on English," in Proc. 3IJCAI, 1973

S2 Schank, R., "Identification of Conceptualizations Underlying Natural Language," in Schank and Colhy (eds.), W.H. Freeman, 1973

S3 Schmidt, C., and D'Addimio, J., "A Model of the Common-Sense Theory of Intention and Personal Causation," in Proc. 3IJCAI, "1973