WORKING PAPER 114

One System for Two Tasks: A Commonsense Algorithm Memory that Solves Problems and Comprehends Language

by

Chuck Rieger

Massachusetts Institute of Technology
Artificial Intelligence Laboratory
November 1975

ABSTRACT: Plan synthesis and language comprehension, or more generally, the act of discovering how one perception relates to others, are two sides of the same coin, because they both rely on a knowledge of cause and effect - algorithmic knowledge about how to do things and how things work. I will describe a new theory of representation for commonsense algorithmic world knowledge, then show how this knowledge can be organized into larger memory structures, as it has been in a LISP implementation of the theory. The large-scale organization of the memory is based on structures called a bypassable causal selection networks. A system of such networks serves to embed thousands of small commonsense algorithm patterns into a larger fabric which is directly usable by both a plan synthesizer and a language comprehender. Because these bypassable networks can adapt to context, so will the plan synthesizer and language comprehender. I will propose that the model is an approximation to the way humans organize and use algorithmic knowledge, and as such, that it suggests approaches not only to problem solving and language comprehension, but also to learning. I'll describe the commonsense algorithm representation, show how the system synthesizes plans using this knowledge, and trace through the process of language comprehension, illustrating how it threads its way through these algorithmic structures.

This is the edited text of the "Computers and Thought Lecture" delivered to the 4th International Conference on Artificial Intelligence, held in Tbilisi, Georgia, USSR, September 1975.

Work reported herein was conducted partly at the University of Maryland, under support of a University Research Board grant, and partly at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-75-C-0643.

INTRODUCTION

I want to talk today about human language comprehension and problem solving.

Investigations into how the human mind comprehends natural language have led model builders into progressively deeper cognitive issues. Until only very recently, most research had been directed toward the unit of the sentence, insofar as it represented an isolated thought. The primary areas of research were the sentence's syntactic analysis, its meaning representation, context-free inferences which could be drawn from it, its significance as a command or query within a microworld, and so forth. It is only recently that much attention has come to be directed toward the problem which, I feel, lies at the heart of language comprehension: the problem of understanding the interrelationships among the thoughts which underlie the sentences of a piece of text, of a story, or, more generally, of any sequence of perceptions.

Each step in the evolution of natural language model building over the past twenty years has required inclusion in the model of larger and larger memory structures for the model to consult in order to perform its job: from the lexical entries of the early translators, to the syntactic rules of parsers, to the semantic and conceptual case frameworks of meaning-based parsers, to the context-sensitive data bases of micro-worlds. I <u>certainly</u> do not intend to reverse this evolutionary tendency today; instead, I want to show you still larger structures which seem to be required by a model which is capable of relating one thought to the next.

Let us take as an informal definition of natural language comprehension "the art of making explicit the meaning relationships among sequences of thoughts which are presumed to be meaningfully relatable." My interest in language comprehension, as this defines it, has led me to the doorstep of another discipline which, historically at least, has developed quite independently from language comprehension. This is <u>problem solving</u>, or the art of influencing the world and self via planful actions. The independence of these two disciplines - language comprehension and human problem solving - sometimes was so acute that I can recall days when the language comprehenders and problem solvers would hiss at one-another when passing in the hallway! Perhaps those days will soon be over.

The thesis I wish to propose today is that these two core elements of human intelligence - language comprehension and problem solving - ought to be regarded as two sides of the same coin... that they are simply two ways of using one repertoire of memory processes and one memory organization. My belief, in other words, is that there ought to be a way of organizing world knowledge so that the very same memory structures can be used to solve problems and understand thought interrelationships. Watch any child as his level of problem solving expertise increases hand-in-hand with his ability to understand and connect his perceptions of the world as he must do, say, when listening to a story.

I come from the frenetic "build-a-model-then-rip-it-apart" persuasion within Artificial Intelligence, because it has been my experience that, no matter how clever one is, he never uncovers the real problems by gedankensexperiments. Rather, he thinks a while, builds a model, runs it, watches it fail, thinks some more, revises it, runs it again, an so on. So I will be describing some theoretical ideas today in the context of a computer model with which I have been preoccupied this past year. The resulting system is dual purpose in that it serves as the core of a problem solver and language comprehender.

Specifically, I want to talk about the three central questions which have served as motivation for my recent research: (FIG. 1)

(1) First, how can we represent the kinds of knowledge about the world which underlie human problem solving and language comprehension abilities? That is, what kinds of primitive concepts do we need to express this kind of world knowledge? In particular, I want to focus on dynamic knowledge... that which relates to actions, states and the notions of causality and enablement in the activities we all do day to day. Let's call this type of knowledge <u>commonsense algorithmic knowledge</u>. How are patterns of this knowledge built up from the set of primitive concepts?

(2) Second, how is this knowledge organized in a large system to provide a flexible and context-sensitive problem solver? This includes the question of how specific plans are actually synthesized by the problem solver which has access to this base of abstract commonsense algorithmic knowledge.

(3) Third, how can this commonsense memory organization be used as the basis of a language comprehension system?

My research shares many common goals with the research of others in the field. Bob Abelson, for instance, has long been interested in the nature of plans and themes from the point of view of a social psychologist interested in how humans incorporate notions of cause and effect in their representation of the world. In his Conceptual Dependency theory, Roger Schank has been interested for a number of years in developing a way to represent a conceptual knowledge of actions via a small set of primitives. More recently Schank and Abelson have been investigating techniques for representing the types of larger stereotyped patterns of "how-to-do-it" knowledge called "scripts" to be used for understanding multi-sentence short stories. In the realm of understanding children's stories, Gene Charniak has also been developing a taxonomy of knowledge about actions in the context of applying that knowledge to understanding connected text. In fact, each of us is attacking the problems of representing a knowledge of how things work in the world, then using that knowledge to understand perceptions. This being the case, we

1. HOW TO REPRESENT CHUNKS OF COMMONSENSE ALGORITHMIC WORLD KNOWLEDGE

2. HOW TO ORGANIZE MANY SMALL CHUNKS OF THIS KNOWLEDGE INTO USEFUL LARGER PATTERNS

3. HOW TO USE THIS MEMORY WHILE COMPREHENDING LANGUAGE

FIGURE 1

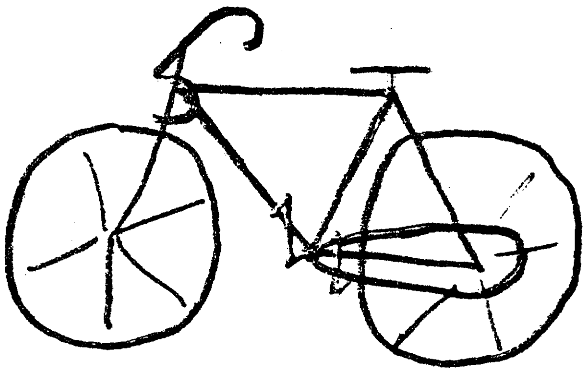
are essentially generating language-related and problem solving-related instantiations of the much broader theory of intelligence proposed by Marvin Minsky: the "frames" theory. However, our methodologies for representing and using action-based world knowledge, and our focusses differ considerably. Today, I want to show you mine.

REPRESENTING ALGORITHMIC KNOWLEDGE

I suppose a good place to begin is to clarify what I mean by dynamic, or algorithmic world knowledge. Consider your knowledge of a bicycle (FIG. 2). You know for instance that it is roughly so tall, you know the shape of its handlebars, the relative positions of its seat, fenders and reflectors; you know that its pedals are free to pivot, that its chain is greasy, and you know its approximate price. These are not the kinds of things I mean by the term "commonsense algorithmic knowledge", but rather pieces of static knowledge about bicycles; they relate to the physical characteristics of a bicycle. Consider on the other hand your knowledge of the function of the bicycle and its various components... things like: the function of a bicycle is to translate an up-down pumping motion of the legs into a statechange in X-Y-Z location of the bicycle and rider; the kickstand is to provide support when the bicycle is not in use, thus interfering with gravity's desire to cause the bicycle to accelerate toward the earth; the horn, when air rushes through it as the result of squeezing the bulb, produces a sound which can cause others to become aware of the rider's presence, and so forth. These are examples of what I mean by the term "commonsense algorithmic knowledge", at least as it relates to mechanical objects and devices; it explains the "why's" and "how to's" of the various components, how the parts interface functionally with each other, and how and why a potential rider might want to interact with them. As you will soon see, I intend the term "commonsense algorithmic knowledge" to cover many other more diverse forms of cause and effect patterns outside the limited realm of mechanical devices.

Just for fun, let's see how we might represent part of our knowledge about the bicycle's horn (FIG. 3). What I am about to describe is a pattern built up from primitive event connectors which illustrate some of the commonsense algorithm representation I have been developing. I will describe this representation in more detail in a moment. Here is the description of how the bicycle horn works:

> Actor P's performance of the action "grasp" will, provided the horn's bulb is not already flat, and provided P has situated his fingers around the bulb, cause a negative change in the volume of the bulb. This negative change will eventually threshold at the state in which the bulb is flat, contradicting a gate condition on the causality and shutting off the flow of causality. Meanwhile, provided there is air in the bulb and there are no leaks in the bulb itself, synonymous with the bulb's negative change in



STATIC KNOWLEDGE

HEIGHT
SHAPE
RELATIVE POSITIONS OF COMPONENTS
CHARACTERISTICS OF COMPONENTS
PRICE
⋮

DYNAMIC KNOWLEDGE

FUNCTION OF THE WHOLE DEVICE
TRANSLATE PUMPIN
INTO FORWARD MOTION
FUNCTIONS OF EACH COMPONEN

NOT INDEPENDENT, BUT DISTINGUISHABLE

FIGURE 2

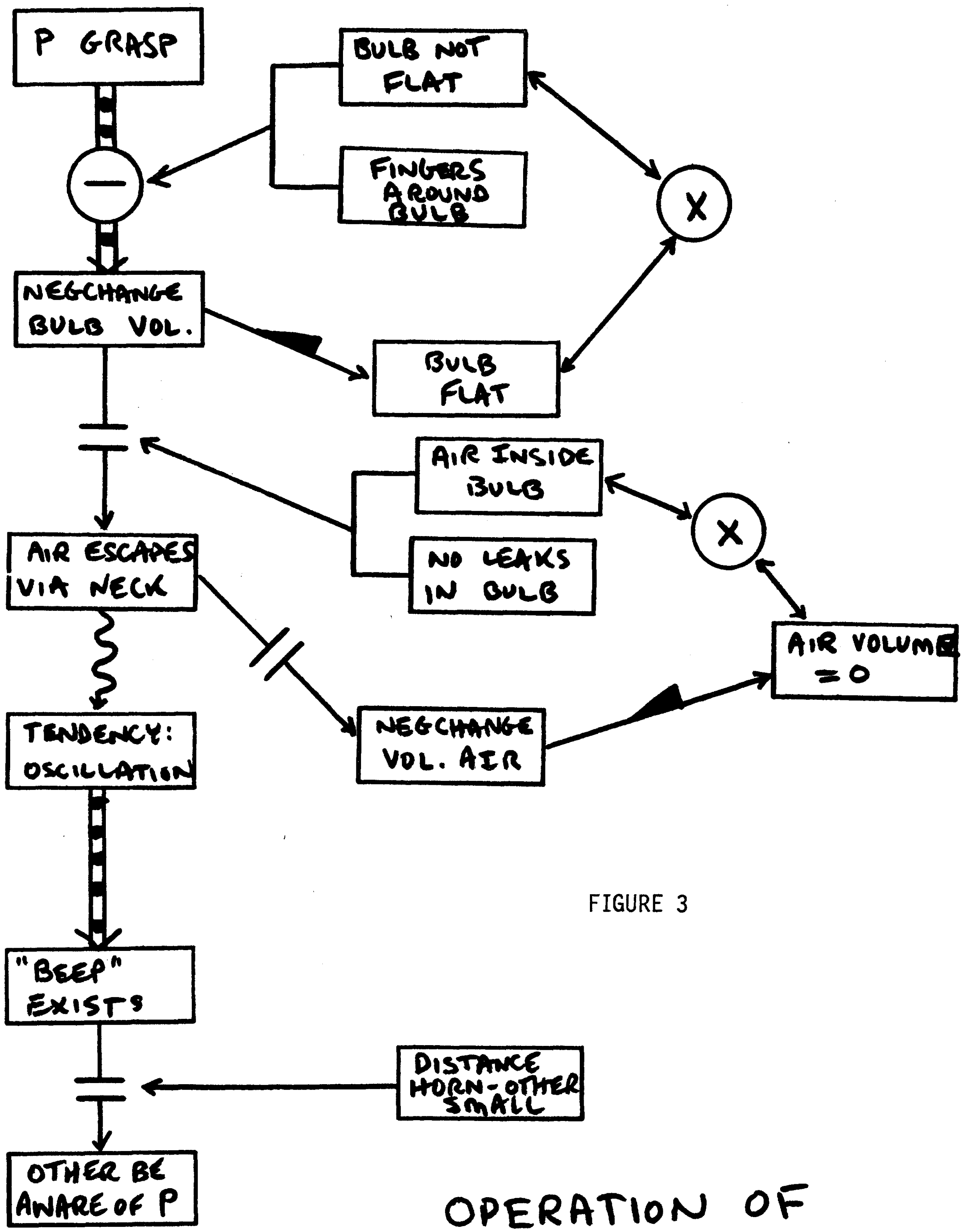


FIGURE 3

OPERATION OF A BICYCLE HORN

volume is a statechange of the location of the air from inside the bulb through the neck of the horn. This rush of air provides the requisite continuous enablement for the tendency "oscillation" to produce a "beep". The rush of air out of the bulb is synonymous with a negative change in amount of air in the bulb, this negative change eventually thresholding at the point at which there is no air left, contradicting a gate condition, and hence indirectly severing the continuous enablement needed by "oscillation". Meanwhile, provided there is a person nearby, the existence of the "beep" will amount to that person becoming aware of the horn honker!

And that, my friends, is the theory of a bicycle horn. Lest it be said that we model builders never verify our theories by psychological experimentation, I will now attempt to verify the correctness of this representation... (honk horn).

Now that I have everyone's renewed attention, and we have a sound experimental basis upon which to proceed, let me briefly describe the commonsense algorithm approach to representing algorithmic world knowledge.

The essence of the representation (FIG. 4) is a set of 26 connecting links. The events which the links connect are assumed to be classifiable into five categories: actions, states, statechanges, tendencies and wants. Each of the 26 links has its own syntax of allowable event types. An <u>action</u> is something a potential actor can actually do: "grasp" is such a thing, while "honk a horn" is not, but is instead a reference to an entire commonsense algorithm pattern such as that I have just experimentally verified. <u>States</u> and <u>statechanges</u> are descriptions of actorless conditions in the world, such as STATE: (LOCATION IVAN HOME(IVAN)) and STATECHANGE: (LOCATION IVAN HOME(IVAN) OFFICE(IVAN)). States and statechanges can be caused by actions and their existence can enable other actions in turn. A <u>tendency</u> is an actorless action which, by definition, must occur whenever its set of enabling conditions is satisfied. For example, "earth gravity" is a tendency which causes an object to accelerate toward the earth, provided that the object is close to earth and unsupported; "metabolism" is another tendency which causes an organism to grow hungry, provided that the enabling condition "there is no food in its stomach" is satisfied. A <u>want</u> is a state or statechange which is desired by a potential actor.

The 26 connecting links are designed to express such concepts as causality, enablement, concurrency, iteration, gating, and intent among events of the five types I have described. There are eight links relating to forms of causality (FIG. 5): the four nominal causal forms (FIG. 5, top, one-shot/ continuous, gated/ non-gated), and the four counterpart byproduct forms (bottom). Suppose, for example, we wish to capture the essence of face-to-face verbal communication by a commonsense algorithm pattern. Then we write: (FIG. 6, top). If, on the other hand, we wish to express one way of causing an object to be attached to one's hand, we write: (FIG. 6, bottom).

COMMONSENSE ALGORITHM REPRESENTATION

26 CONNECTIVE LINKS
DEALING WITH CONCEPTS OF

CAUSALITY
ENABLEMENT
CONCURRENCY
ITERATION
INTENTION
GATING
THRESHOLDING

AMONG EVENTS OF TYPES:

ACTION
STATE
STATECHANGE
TENDENCY
WANT

FIGURE 4

THE 8 CAUSAL LINK FORMS

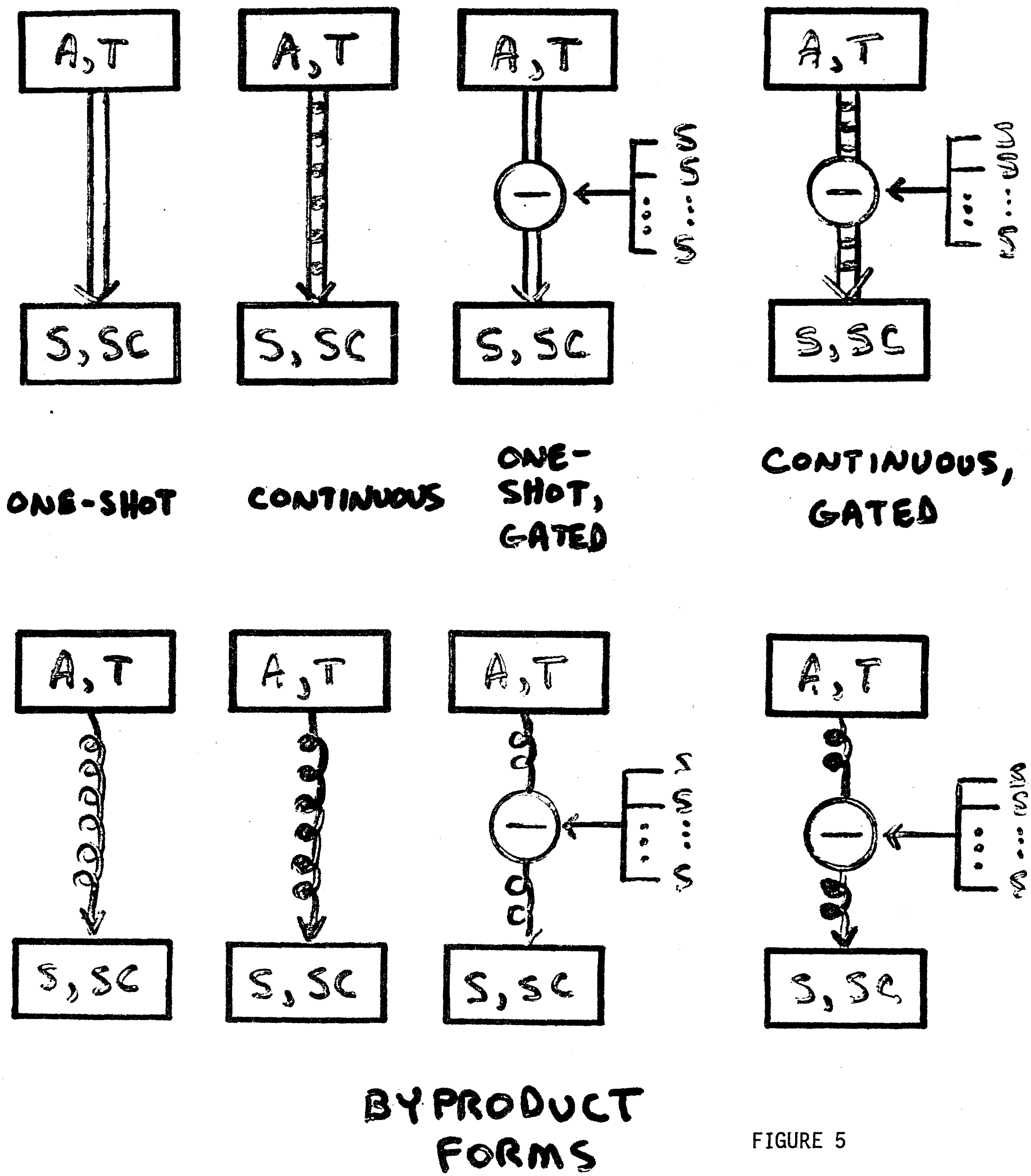
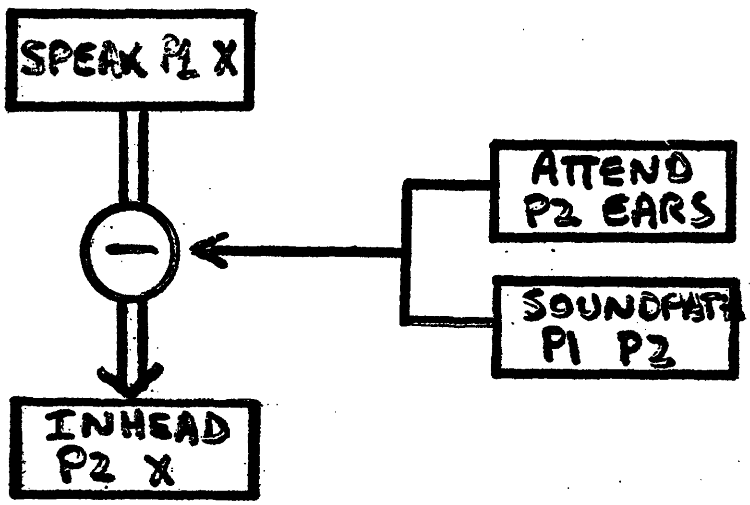


FIGURE 5

A PATTERN FOR VERBAL COMMUNICATION



A PATTERN FOR PICKING UP A SMALL, HOLDABLE OBJECT

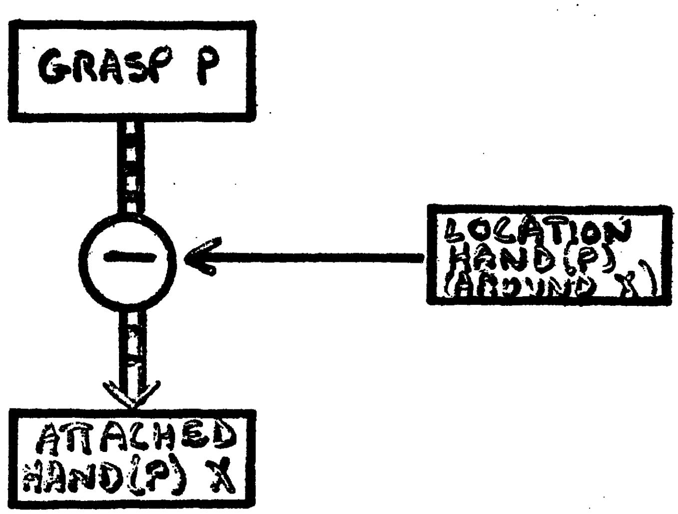


FIGURE 6

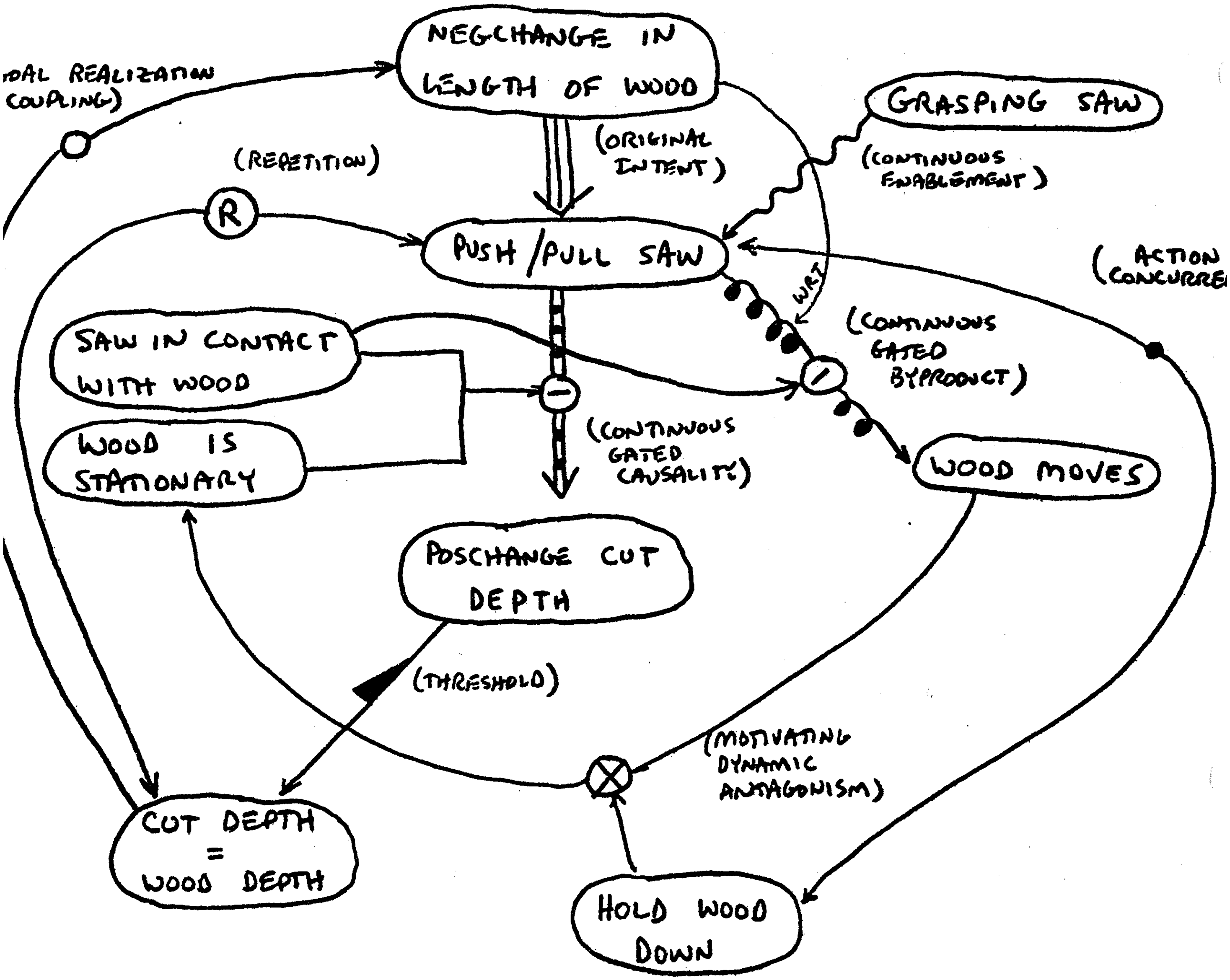
The four causal byproduct forms allow the explicit representation of other states and statechanges which are likely to result from an action, even though they are not related to the attainment of the intended goal of the action. For example, when we saw a board in half, we impart force to the board, causing it to move if unsupported; we cause small particles of wood fiber to begin existing in an unsupported state, and so on (FIG. 7a), or when a professor screams at his graduate students, he sometimes gives himself a sore throat (FIG. 7b), and so forth. These byproduct links will be extremely useful when synthesizing plans, and can play a role in the language comprehension ability of the system.

There is another causal-like link, the state coupling link (FIG. 8, top), which gives us the ability to assert an implicit relationship of causality without having to explain the actual intervening actions. This link also gives us the ability to denote the equivalence of two different ways of viewing some state or statechange. For example, to express a very common method of causing a statechange in location of a small object, we might write: (FIG. 8, bottom left), or to express the synonymy of fluid entering a container with an increase in volume of fluid in the container, we would write: (FIG. 8, bottom right).

There is a link to capture the notion of good continuation of a statechange toward some desired level; this is the threshold link. For example, one way to cause oneself to be located at a place is to walk toward it, hoping eventually to reach it: (FIG. 9).

There are links for representing the requisite preconditions of an action, or of an entire algorithm expressed as a commonsense pattern when we choose to regard the entire pattern as primitive for, say, problem solving purposes. For example, if for planning purposes, we choose to regard the "drive a car" commonsense algorithm as a primitive with a known solution, then all we need to know is the algorithm's cumulative set of preconditions: (FIG. 10).

I have described the most important aspects of the commonsense algorithm representation of dynamic world knowledge. Within this framework of event types and connective links, the act of problem solving (FIG. 11) will be one of reacting to wants by constructing patterns which map out planful activities such as executing primitive actions, setting up preconditions and gate conditions which allow those actions to achieve the desired effects, and unleashing desirable tendencies while performing compensatory actions to suppress other undesirable ones.



Sawing a board in half to decrease its length.

FIGURE 7a

SCREAMING AT
A GRADUATE STUDENT

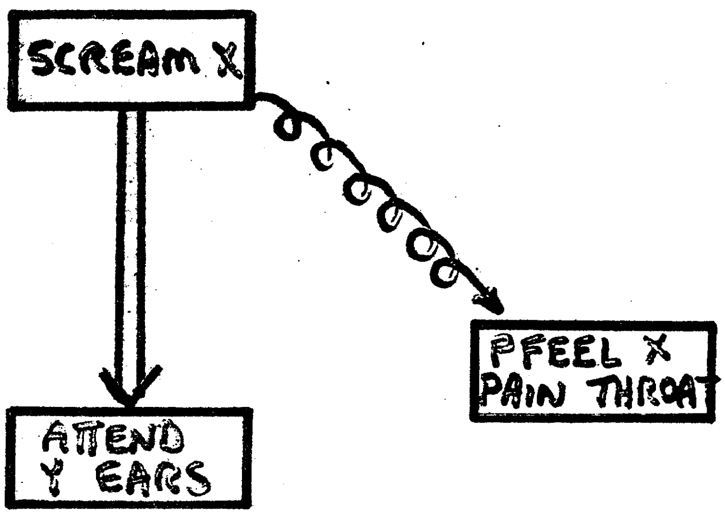
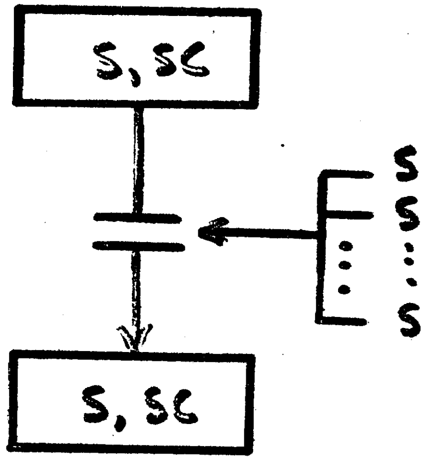


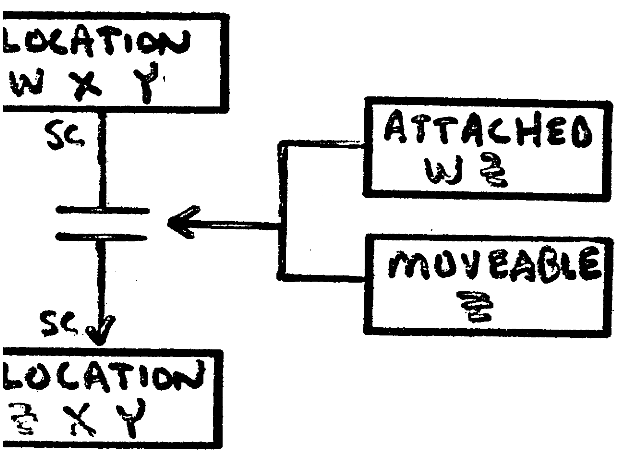
FIGURE 7b

STATE COUPLING LINK

EQUIVALENCE OF STATES OR STATE CHANGES (UNSPECIFIED CAUSALITY)



ONE WAY TO MOVE AN OBJECT



EQUIVALENCE OF FLOW WITH INCR IN VOLUME

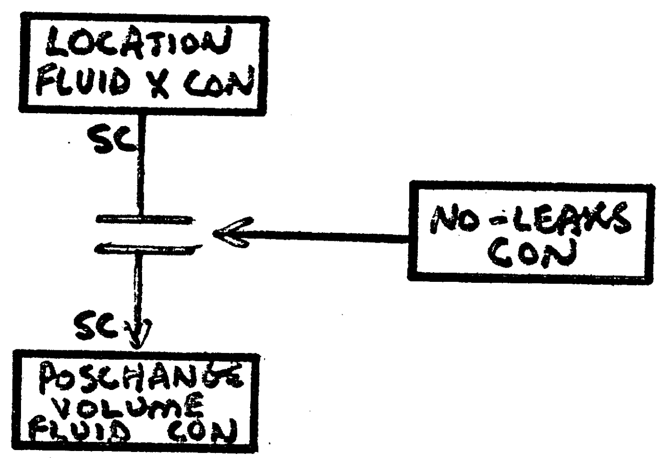


FIGURE 8

THRESHOLD LINK (GOOD CONTINUATION)

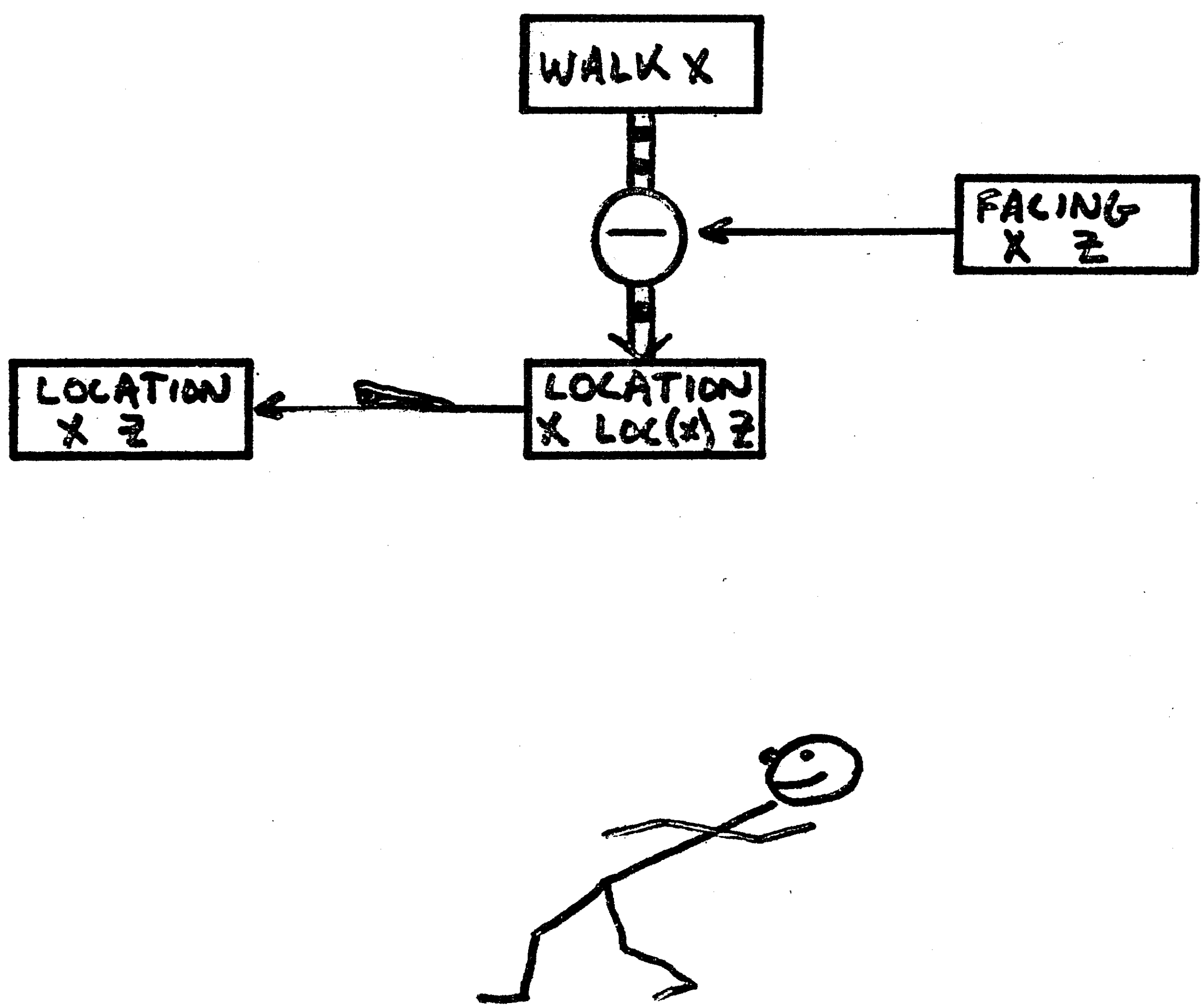
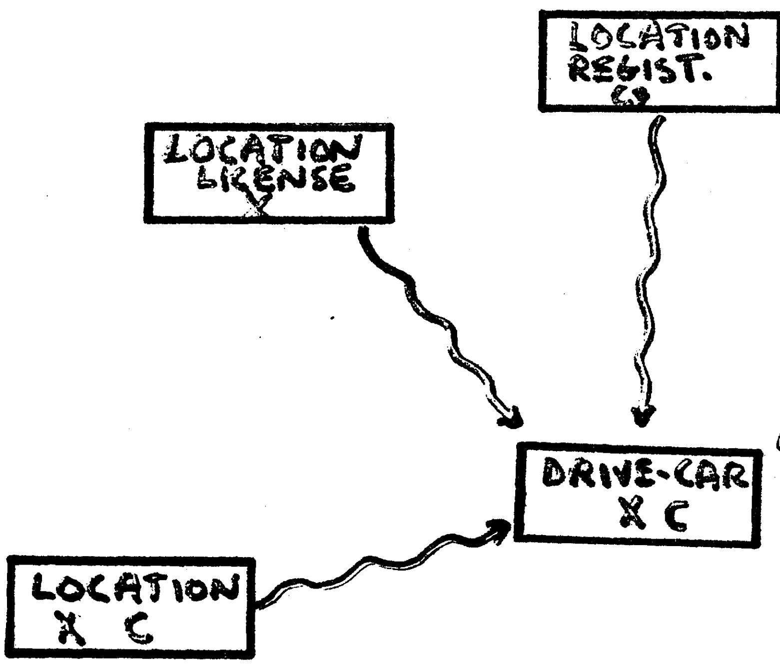


FIGURE 9

ENABLEMENT LINKS

(PRECONDITIONS FOR ACTIONS)



(RELATES TO MECHANISM DESCRIPTION OF CAR)

ENABLEMENTS FOR AN ENTIRE ALGORITHM, VIEWED AS A BLACK BOX

FIGURE 10

THE ACT OF PROBLEM SOLVING IN THE COMMONSENSE ALGORITHM ENVIRONMEN WILL CONSIST OF THINGS SUCH AS:

- EXECUTE A PRIMITIVE ACTION
- SET UP ENABLING CONDITIONS OR GATE CONDITIONS
- UNLEASH TENDENCIES
- SUPPRESS UNDESIRED TENDENCIE
- CONVERT GRAPH COMMONSENSE ALGORITHMS TO LINEAR STEP SEQUENCE

FIGURE 11

PROBLEM SOLVING

I will stop the description of the representation formalism even though there is a lot more to say, because I want to move to the second main topic: (FIG. 11a) how to organize these small, isolated patterns of knowledge such as I have been showing you into larger complexes which will provide the basis for both problem solving and language comprehension. That is, imagine that we have the capability for representing, individually, the thousands upon thousands of small commonsense algorithmmic patterns we all surely must possess. I want now to consider the structure of the larger fabric of the memory in which these patterns might be embedded.

Let me introduce the apporoach to the organization I have been developing by describing the theoretical considerations which led up to it (FIG. 12).

(1) First, if we assume, with regard to problem solving, that we have this ability to represent small, isolated patterns of algorithmic world knowledge, then the primary role of the larger structures must be to provide a matrix into which we can fit and interrelate the patterns. And this matrix should serve as a basis for making intelligent selections of alternative patterns when synthesizing a solution for a given problem. I mean by the phrase "intelligent selections" that the organization ought to lend itself to interaction with both a static knowledge of the unchanging aspects of the world, and with the context in which the problem is being solved... that is, in which the commonsense algorithm pattern selection process is occurring. Where there are numerous alternative causal patterns for achieving some desired goal, the system ought to have good reasons for preferring one alternative over the rest in any given environment.

(2) Second, during the problem solving process, the solutions to subproblems ought somehow to be sensitive to the context and purposes of the larger problem. For example, suppose my goal is to insult Ivan. I decide the best way to proceed, based on what I know about him, is to make a dirty joke about his wife; but I don't know any dirty jokes, and hence have the subproblem of first learning an appropriate dirty joke. Of the many algorithm patterns organized under the "how to learn about something" part of my algorithmic knowledge, one is: "politely ask someone else." Certainly, in solving this subgoal, I would not want to go to Ivan and politely ask him for a dirty joke about wives! The structure of knowledge ought, therefore, to be such that certain aspects of higher level goals automatically diffuse throughout the system, guiding the problem solver away from some alternative solutions to subproblems, toward others. In seeking a dirty joke, I will either avoid Ivan altogether, or interact with him in ways which are compatible with my higher level goals as they concern him.

(3) Third, the organization ought to adapt dynamically as it discovers more and more about the environment - the context - in which it is solving problems. In particular, it would be desirable for knowledge discovered by one part of the system, during the solution of some

1. HOW TO REPRESENT CHUNKS OF COMMONSENSE ALGORITHMIC WORLD KNOWLEDGE

→ 2. HOW TO ORGANIZE MANY SMAIL CHUNKS OF THIS KNOWLEDGE INTO USEFUL LARGER PATTERNS

3. HOW TO USE THIS MEMORY WHILI COMPREHENDING LANGUAGE

FIGURE 11a

THEORETICAL CONSIDERATIONS FOR THE LARGE ORGANIZATION

1. SHOULD PROVIDE THE ABILITY TO MAKE INTELLIGENT SELECTION FROM AMONG ALTERNATIVE CAUSAL PATTERNS FOR SOME GOAL

2. SOLUTION TO SUB PROBLEMS OUGHT TO BE SENSITIVE TO THE CONTEXT AND PURPOSES OF THE MAIN PROBLEM

3. ORGANIZATION OUGHT TO ADAPT DYNAMICALLY TO ENVIRONMENTAL CONTEXT

4. OUGHT TO BE EXPLICIT FOR USE BY PROCESSES OTHER THAN PROBLEM SOLVING

SO: MUST PROVIDE A WAY OF ENCODING A KNOWLEDGE OF WHAT IS RELEVANT TO SELECTING OR SEARCHING

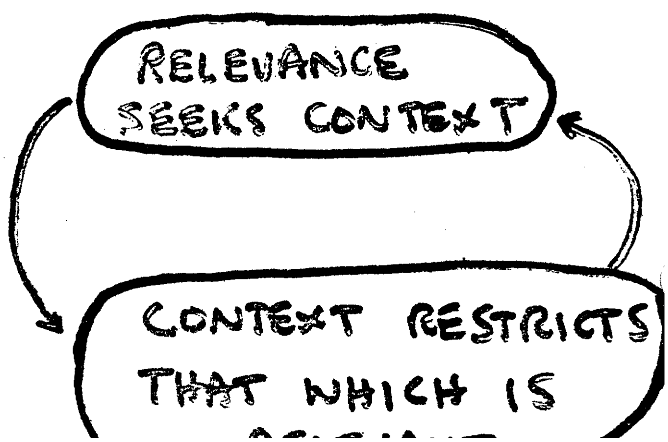


FIGURE 12

particular problem, automatically to enhance the efficiency with which other parts of the system solve their problems. That is, as the system discovers things about its environment during the course of solving particular problems, its general level of problem solving expertise in that environment ought somehow to be heightened.

(4) Fourth, as regards use in language comprehension, the organization ought to be explicit, rather than embedded in procedures where the various causal and enablement relationships are only implicit in the structure of the procedure, and where the relations would normally be one-way subroutine calls or one-way pattern directed interfaces. The explicitness of the structures will be essential when the system needs to search through memory when seeking relationships among thoughts during language comprehension.

There is one other theoretical consideration which was perhaps an even more fundamental design criterion than these four. This is that the essence of both problem solving and language comprehension lies in knowing what are the relevant questions to ask during a selection or searching activity... that is, in knowing which aspects of the environment could possibly be relevant to the attainment of some goal. If a process which, say, selects one commonsense algorithm pattern over the rest does not know, in a context-free way, what is ultimately relevant to the functioning of the patterns among which it is selecting, how can that process ever be made to do different things in different environments? Perhaps this is a matter for the philosophers, but it has suggested to me that the heart of the problem of organizing world knowledge lies in encoding a knowledge of what other knowledge bears relevance to the solution of any given task, be it synthesizing a solution to a problem, or searching through algorithmic structures during language comprehension. The system ought to behave in a way such that a knowledge of what is relevant can be used to seek out context, but one in which context, once discovered, can feed back, restricting that which is relevant.

These four criteria - (1) the ability for intelligent algorithm selection, (2) the ability for higher level goals to influence the way in which subproblems are dealt with, (3) the ability for discoveries about the environment, the context, made during the solutions of particular problems to enhance the general ability to solve problems in that environment, and (4) explicitness - suggested an organization which I will call a bypassable causal selection network as the larger matrix into which to fit the thousands of small commonsense algorithm patterns (FIG. 13).

The purpose of such a network is, given a goal state or goal statechange, to select a most relevant algorithm pattern for achieving that goal state or statechange. Such a network consists of a set of nodes organized into a tree structure. Each node has associated with it a test and a set of one or more alternative branches, one of which will be followed on the basis of the test result. Tests are either memory queries about unchanging world knowledge, or queries about the environment in which the selection is being made, that is, in which the network is being

BYPASSABLE CAUSAL SELECTION NETWORKS

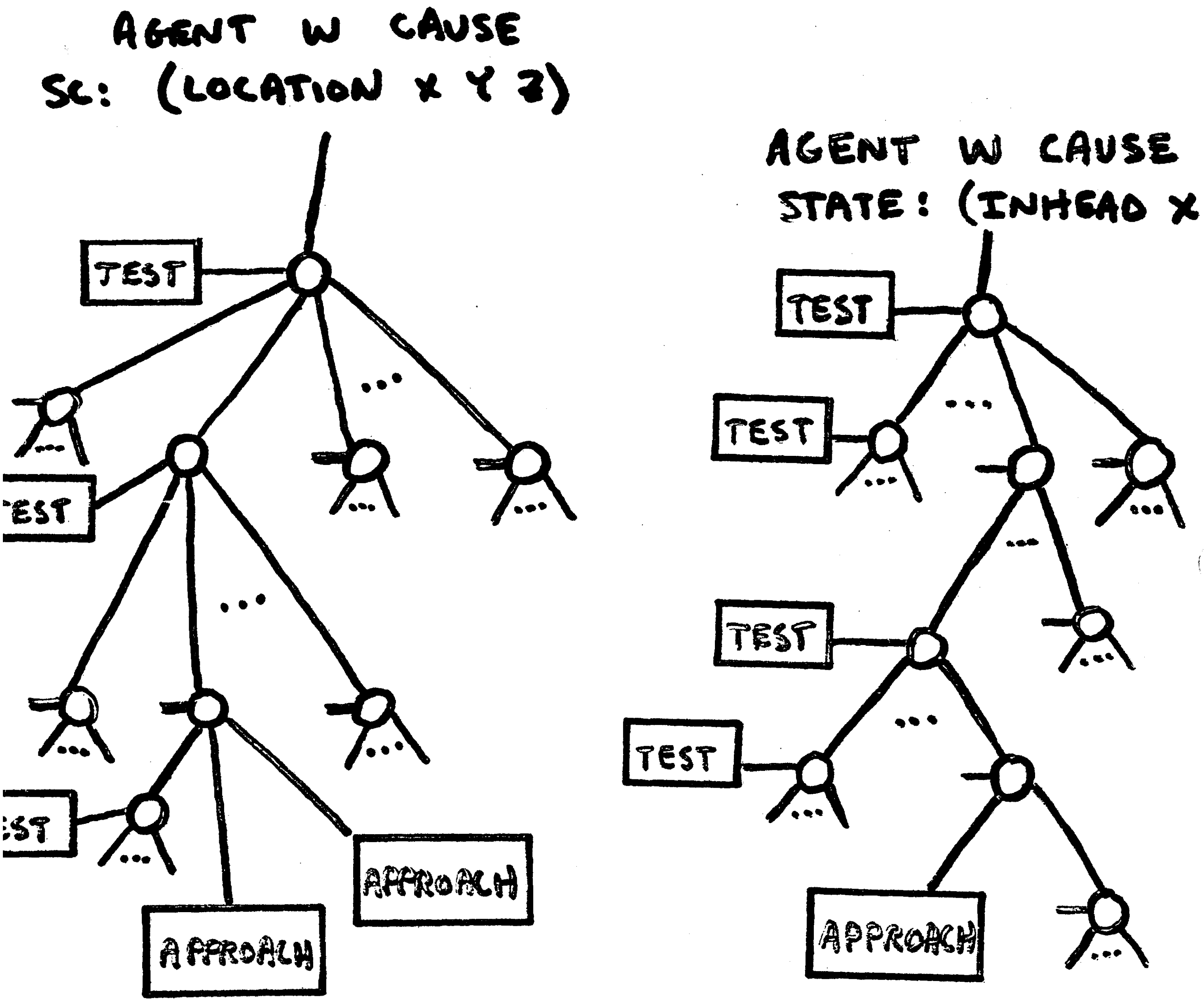


FIGURE 13

traversed. There is one of these bypassable causal networks for each state concept and statechange concept in the system. For example, there is a causal selection network for deciding how to cause a statechange in the location of an object from one place to another: (FIG. 13, left), there is a causal selection network for deciding on an appropriate approach to causing a piece of knowledge to begin existing in one's mind, or in the mind of another actor: (FIG. 13, right), and so forth. This implies that there will be a relatively large number of networks, each being a specialist at solving problems relating to one particular state or statechange concept.

At the terminal nodes of these networks are what I will call approaches; these are the patterns, expressed in the commonsense algorithm representation I have shown you, which map out general, top-level plans for solving problems of the class for which the network is an expert. For example, among the hundreds of patterns at the terminal nodes of the AGENT W CAUSES STATECHANGE (LOCATION X Y Z) selection network will be linkers to commonsense algorithm patterns for driving cars, walking, taking elevators, grasping and carrying objects, hoisting things, throwing things, swallowing things, and so on. The kinds of tests asked at the nodes of this network in order to select among the hundreds of alternatives a pattern for solving a particular instance of a STATECHANGE LOCATION goal will be things like class memberships of W, X, Y, Z, the distance between Y and Z, the nature of Y and Z (in a building, in the woods, under water, in the body), the weight and size of X, and so on. In other words, the causal selection network is the mother who asks all the right questions before choosing which son - which commonsense algorithm pattern - will be best for the job.

Each of the approaches at the bottom of a causal selection network is a commonsense algorithm pattern of one of three types (FIG. 14): (1) an abstract algorithm, (2) a mechanism description, or (3) a sequential abstract algorithm. An abstract algorithm is a pattern having one of the three forms: (FIG. 14, left, center, right). For example, at the bottom of the AGENT W CAUSES STATECHANGE (LOCATION X Y Z) causal selection network, two of the hundreds of patterns we would find are: (FIG. 15). One of these (FIG. 15, bottom left) would be in a part of the network which deals with changing the location of hand-graspable objects, the other (FIG. 15, bottom right) would be in a part of the network which deals with causing objects to come to be located inside the body. If we look at the "swallow" algorithm pattern, this says that, providing the object X is in the mouth, and it is small enough, the primitive action "gulp" will ordinarily cause X to change location to the stomach. In general, you can see that the gate conditions on the causal or state coupling link in an abstract algorithm will prescribe subgoals to be solved by other causal selection networks in the system.

Notice in the swallow algorithm the existence of two recommendations attached to one of the gate conditions. Recommendations are direct pointers to other abstract algorithms which exist at the bottoms of other networks. A recommendation represents a stereotyped way of solving the gate condition in the context peculiar to the abstract algorithm approach in which it

3 TYPES OF APPROACH:

ABSTRACT ALGORITHM
MECHANISM DESCRIPTION
SEQUENTIAL ABSTRACT ALGORITHM

ABSTRACT ALGORITHM FORMS:

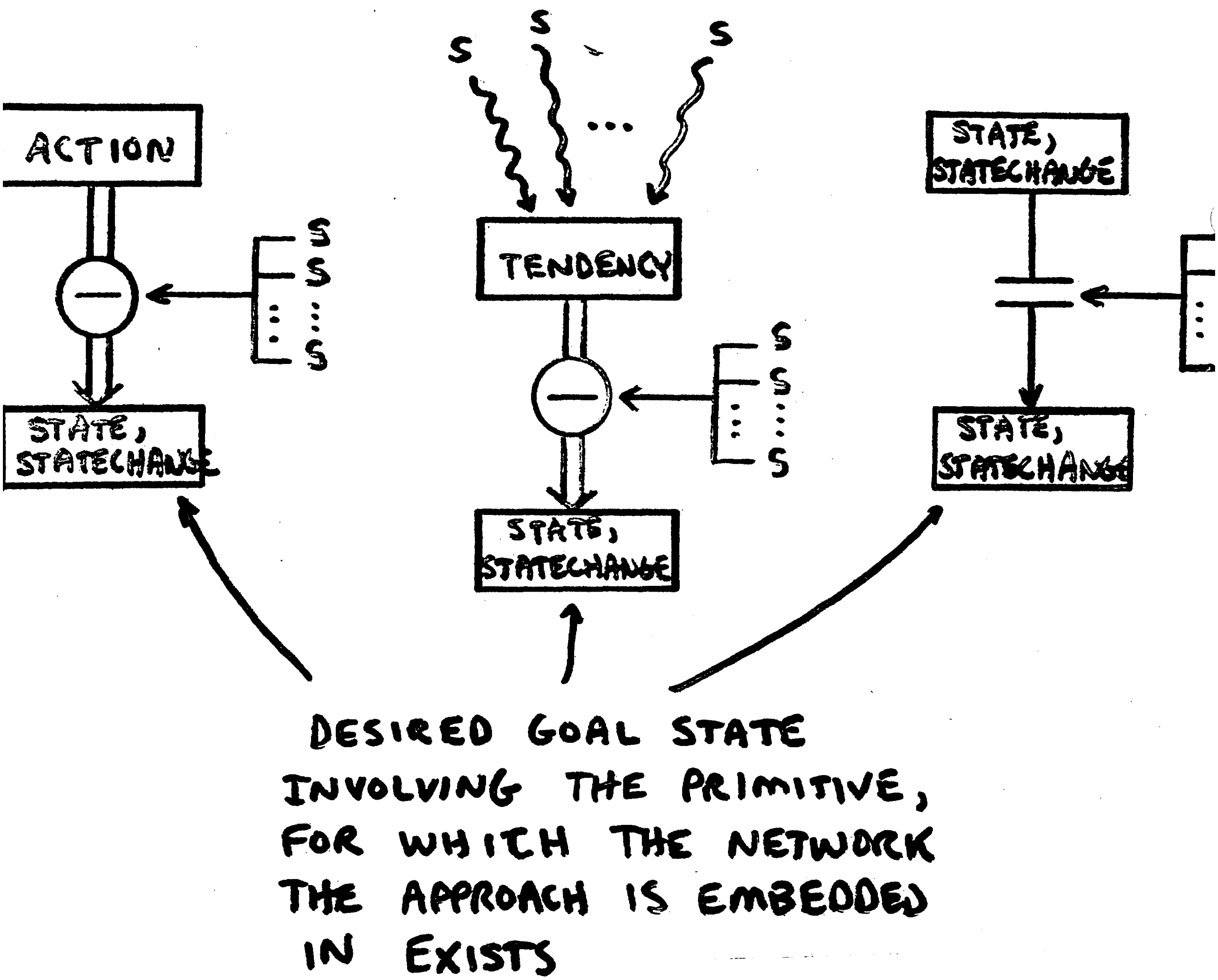


FIGURE 14

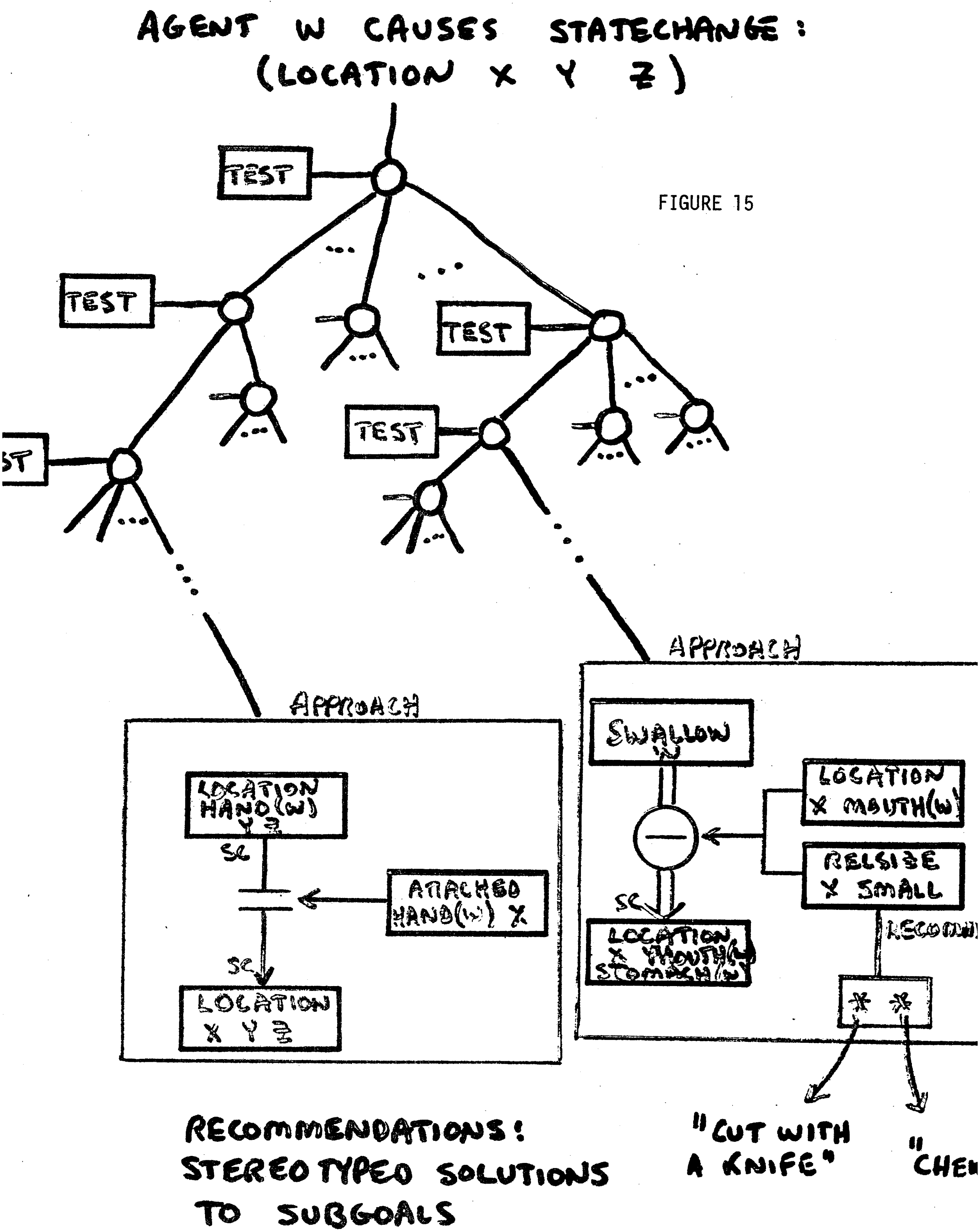

AGENT W CAUSES STATECHANGE:
(LOCATION X Y Z)
TEST
TEST
TEST
TEST
ST
APPROACH
LOCATION HAND(W) Y Z
SC
ATTACHED HAND(W) X
SC
LOCATION X Y Z
APPROACH
SWALLOW W
LOCATION X MOUTH(W)
RELSIZE X SMALL
SC
LOCATION X MOUTH(W) STOMACH(W)
RECOMM
* *
RECOMMENDATIONS:
STEREOTYPED SOLUTIONS
TO SUBGOALS
"CUT WITH A KNIFE"
"CHE

FIGURE 15

occurs. Having the ability to store these recommendations allows the abstract algorithm patterns in effect to grow into larger and larger, more rigid forms. Without recommendations, the gate subgoals will be solved by applying other appropriate causal selection networks. But with recommendations, since a recommendation obviates the causal selection process for a subgoal by pointing directly to another abstract algorithm pattern at the base of another causal network, as the system records, through experience, more and more recommendations for specific gate conditions in specific abstract algorithms, the synthesis of larger and more complex plans becomes possible for the same investment of effort... that is, effort spent in applying causal selection networks... as was once needed for smaller, less stereotyped solutions.

I will not get into the other two types of approach - mechanism descriptions and sequential abstract algorithms - except to show you an example of each. A sequential algorithm is essentially a commonsense algorithm pattern with explicit sequencing information... a linearized chunk of a plan which the system has successfully employed at one time and stored away with the simplifying sequencing information for future use. As such, it keys on action sequences rather than on unsequenced goal states. FIG. 16 illustrates a sequential algorithm for setting an automobile into motion.

A mechanism description captures the internal cause and effect relationships of the events which occur when the mechanism operates. From such a description, the system can figure out both how to use the mechanism and what using it can accomplish. FIG. 17 is the mechanism description of a reverse trap flush toilet. I have been carrying this particular mechanism description around with me for several months; you never know when you might need one!

Let's return now to the causal selection network as a whole. The downward traversal of a network for any given state or statechange goal during problem solving results in the asking of many questions, and, finally, in the selection of an approach based on the answers to these questions. The network tests access a context-layered data base, or, when simple lookups fail, a deduction mechanism which is itself organized as a system of discrimination networks. I will not describe these mechanisms today.

To illustrate how the causal selection networks function, and the kinds of tests they must make, let's confront the problem solver with a goal and observe what happens in response. Suppose I am at work and become thirsty. In other words, I have the goal: (LOCATION FLUID STOMACH(SELF)). This goal will cause the problem solver to apply the STATECHANGE LOCATION causal network for AGENT SELF CAUSES STATECHANGE (LOCATION FLUID ? STOMACH(SELF)), this network being an expert at synthesizing plans for all sorts of statechanges in location of objects (FIG. 18). In this case, by asking questions about W, X, Y and Z, the network will lead the problem solver to a mechanism description at the bottom of the

SEQUENTIAL ABSTRACT ALGORITHMS

T CAR
NTO X
IOTION

GO TO CAR

OPEN DOOR

GET INTO POSITION — ON FRONT SEA

CLOSE DOOR

INSERT KEY IN IGNITION

DEPRESS CLUTCH, ACCELERATOR

TURN KEY UNTIL ENGINE STARTS

RELEASE ACCELERATOR

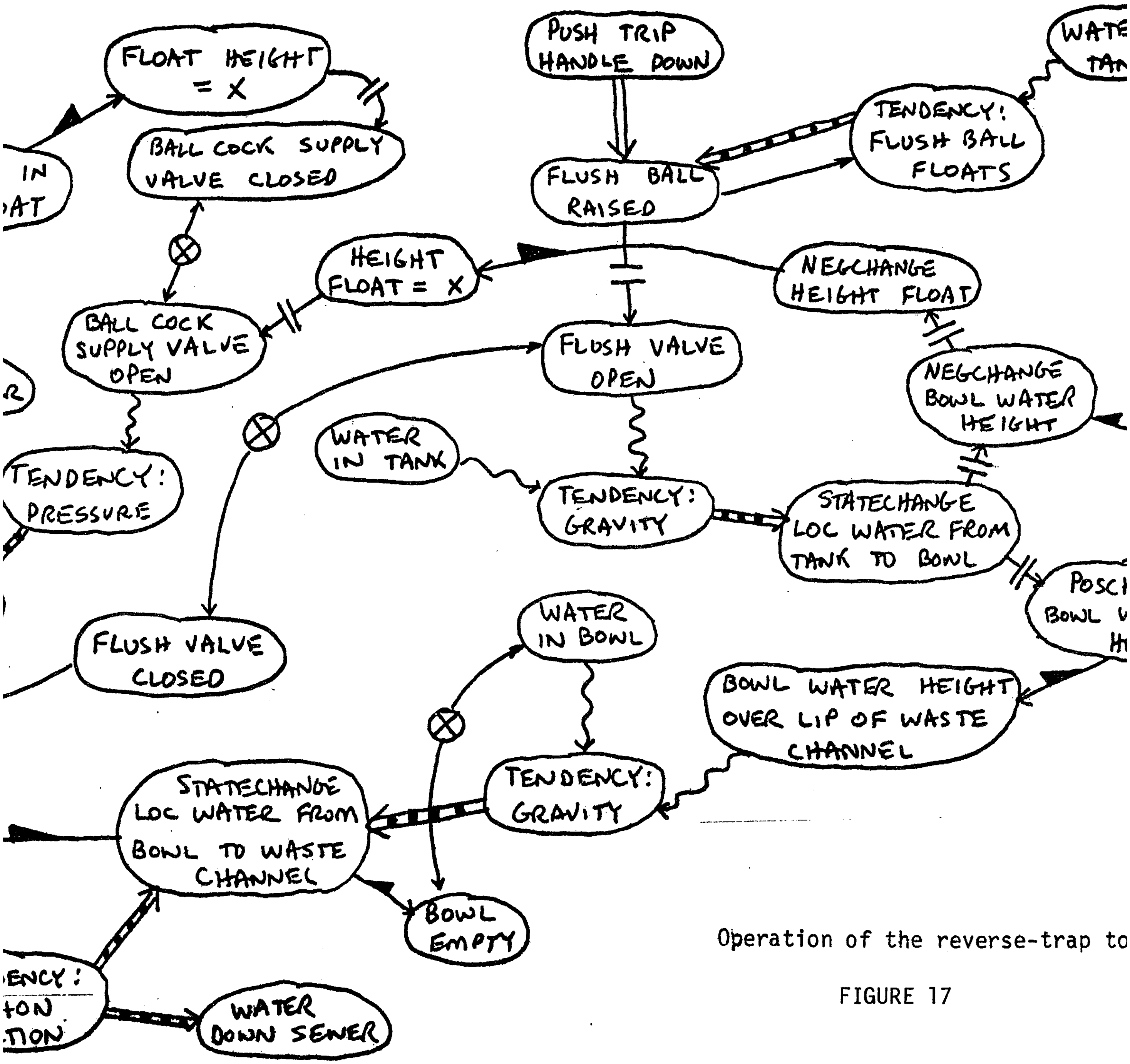
DEPRESS FOOT BRAKE, RELEASE HAND BR

PUT GEAR SHIFT LEVER INTO X POS

RELEASE CLUTCH, RELEASE FOOT BRAKE
DEPRESS ACCELERATOR

FIGURE 16

CRASH



Operation of the reverse-trap to

FIGURE 17

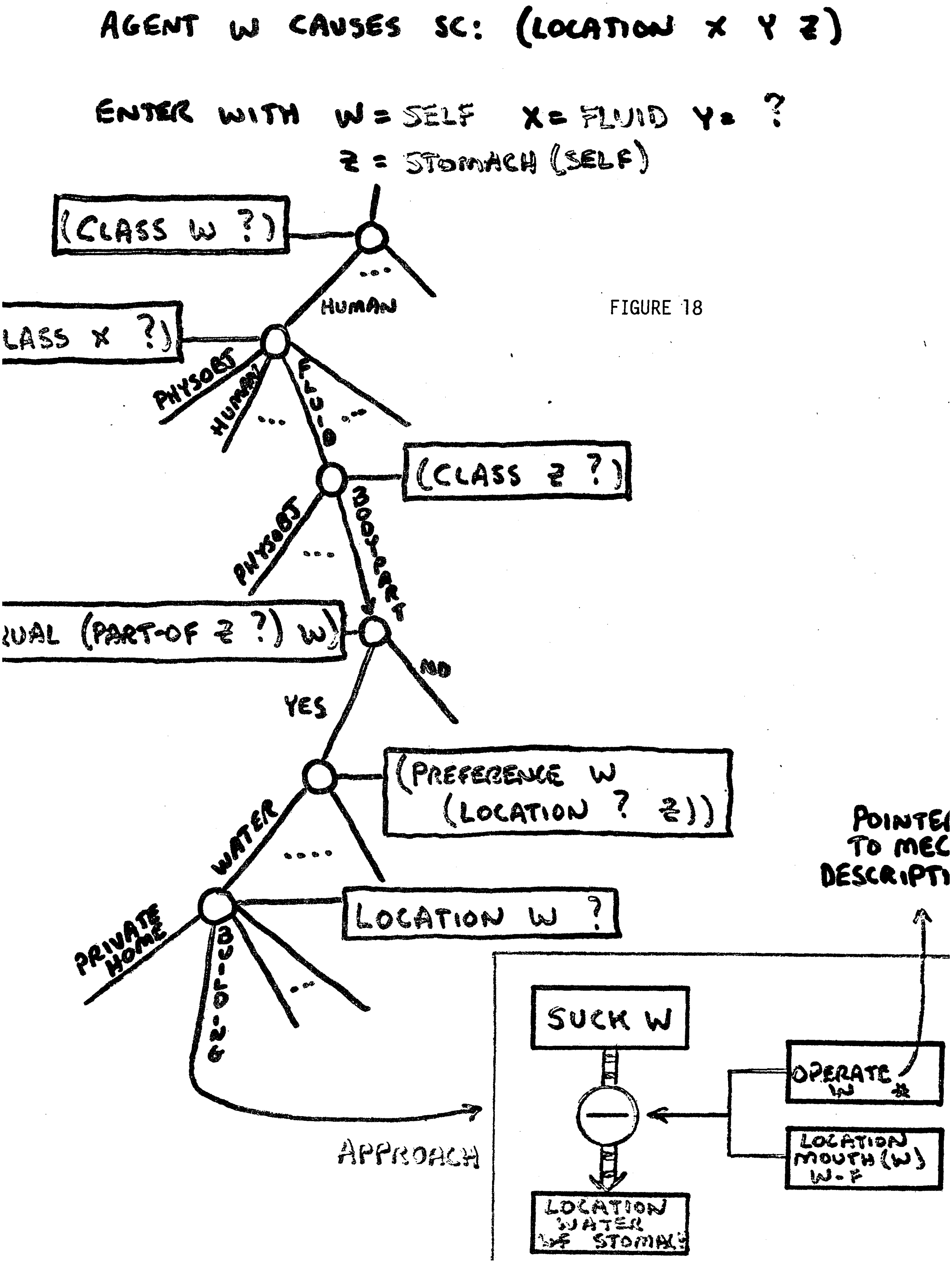

AGENT W CAUSES SC: (LOCATION X Y Z)
ENTER WITH W = SELF X = FLUID Y = ?
Z = STOMACH (SELF)
(CLASS W ?)
HUMAN
LASS X ?)
PHYSOBJ
HUMAN
FLUID
(CLASS Z ?)
PHYSOBJ
BODYPART
UAL (PART-OF Z ?) W)
NO
YES
(PREFERENCE W
(LOCATION ? Z))
WATER
LOCATION W ?
PRIVATE HOME
BUILDING
APPROACH
SUCK W
OPERATE W
LOCATION MOUTH (W) W.F
LOCATION WATER STOMACH
POINTE
TO MEC
DESCRIPTI

FIGURE 18

network as being the most appropriate way of attaining this goal: "use the water fountain out in the hall." In another environment, the network might have selected another quite different approach, such as drinking from a cup in the kitchen, drinking from a stream in the wilderness, going into a store and ordering a glass of water, and so forth. And of course, we have not even considered the rest of this extremely large causal selection network which would also be applied to the solution of vastly different varieties of STATECHANGE LOCATION goals involving people and various kinds of physical objects instead of fluids and bodyparts.

I imagine these causal selection networks as being extremely large, with perhaps thousands of nodes each. But their largeness, I believe, will be mostly a matter of extreme breadth rather than depth. I do not believe, for example, that they need ever exceed a depth of, say, 10 or 15. This characteristic will be important for the language comprehension search processes.

Note that, having selected the "drink from a water fountain" approach, specific subgoals are spelled out, the most significant one in this case being to achieve the statechange (LOCATION SELF OFFICE(SELF) WATER-FOUNTAIN). (Since my mouth is attached, it will come along with me... the system knows about this through a statecoupling pattern!) Each subgoal will result in a planning process similar to this one. The end product of plan synthesis will be a large structure called a <u>synthesized algorithm</u> and will provide input to the linearization processes which will transform the synthesized algorithm, in graph form, into a linear, executable sequence of steps.

Before leaving this aspect of the system, let me reemphasize this notion of relevance. In the water fountain selection example, something in the system had to know that the current location of self bore extreme relevance to the process of deciding how to go about quenching self's thirst. It is the purpose of the networks to carry out an orderly probe of the environment along relevant lines.

Now let me explain why I have called these causal selection networks "bypassable". Consider the system's character as a complete system: if we look across all the networks in the system, we see that any particular test of the environment is apt to occur at numerous points throughout the system of networks. To take a very simple example, the test (CLASS X ?), which inquires about the class membership of an object, except for different variable names, will occur at possibly hundreds of points throughout the system of networks. The presence of this test at different points within one network, or at points in different networks, will be for reasons peculiar to each network's requirements; the way one network uses this information may be quite a bit different from the way another network uses it. Nevertheless, if we disregard the reasons for the test's existence in the various networks, there will be quite a bit of overlap in the knowledge about the environment needed by the various networks.

This overlap can serve as the basis of a very interesting mechanism as follows (FIG. 19). First, we allow all network tests to be shared. What I mean by this is that suppose a given test occurs at 50 points throughout the system of networks. Then, instead of planting 50 instances of that test, we store one central copy and reference the central copy from each point that test is needed in the various networks. Suppose also that the central copy knows of all these occurrences. Then, when any network asks this question and obtains an answer in the current environment, the effects of that piece of knowledge can be distributed to all other points of that test's occurrence throughout the system. This distribution process is achieved by planting what I will call a conditional bypass around each occurrence of that test throughout the system of networks (FIG. 20). For example, if during the course of solving some problem, any network asks and discovers that self's location is (LOCATION SELF BUILDING), then every other network which might ask the question (LOCATION X ?), and which has BUILDING as an alternative, will receive a conditional bypass around the node in the network at which this question would be asked. In this case, the bypasses thus planted would be conditional upon X = SELF; the bypass will not be seen for other values of X.

In the implemented model, this is precisely what occurs. As each node in a network makes its test, the result is recorded as an active datum in the current environment, and then conditional bypasses are distributed to all parts of the system which could conceivably use the newly-acquired information.

The significance of a bypass in a network is a key point of the model: a bypass will provide the problem solver with a shortcut in environments where the bypass remains valid. As more and more bypasses accumulate at scattered points throughout the system of networks via this distribution process, contiguous sequences of bypasses will begin to form (FIG. 21). If this bypass implantation process is overseen by a transitive closure seeker, then the system will automatically replace contiguous sequences of compatible bypasses by progressively longer, single-step bypasses. If the causal selection process is sensitive to these bypasses, preferring to follow them whenever it can, then as more and more is discovered about the environment as the results of solving specific problems in that environment, the percentage of the total system which is bypassed increases. What this means is that the system automatically will tend increasingly to prefer the selection of approaches which are most appropriate in that environment, being guided, without any choice, through longer and longer stretches of bypassed causal selection networks. Some bypasses might grow to be quite long, even up to the point where some entire networks become effectively invisible because of the existence of one long bypass from the top node directly to some approach, A, at the bottom (FIG. 22). Of course, this bypass might never actually be needed; but, should that network ever be applied in an environment in which this total bypass exists, approach A would be selected "without thought" so to speak. As an overall organism, this bypassable network memory organization behaves in a way such that, as more and

TEST SHARING

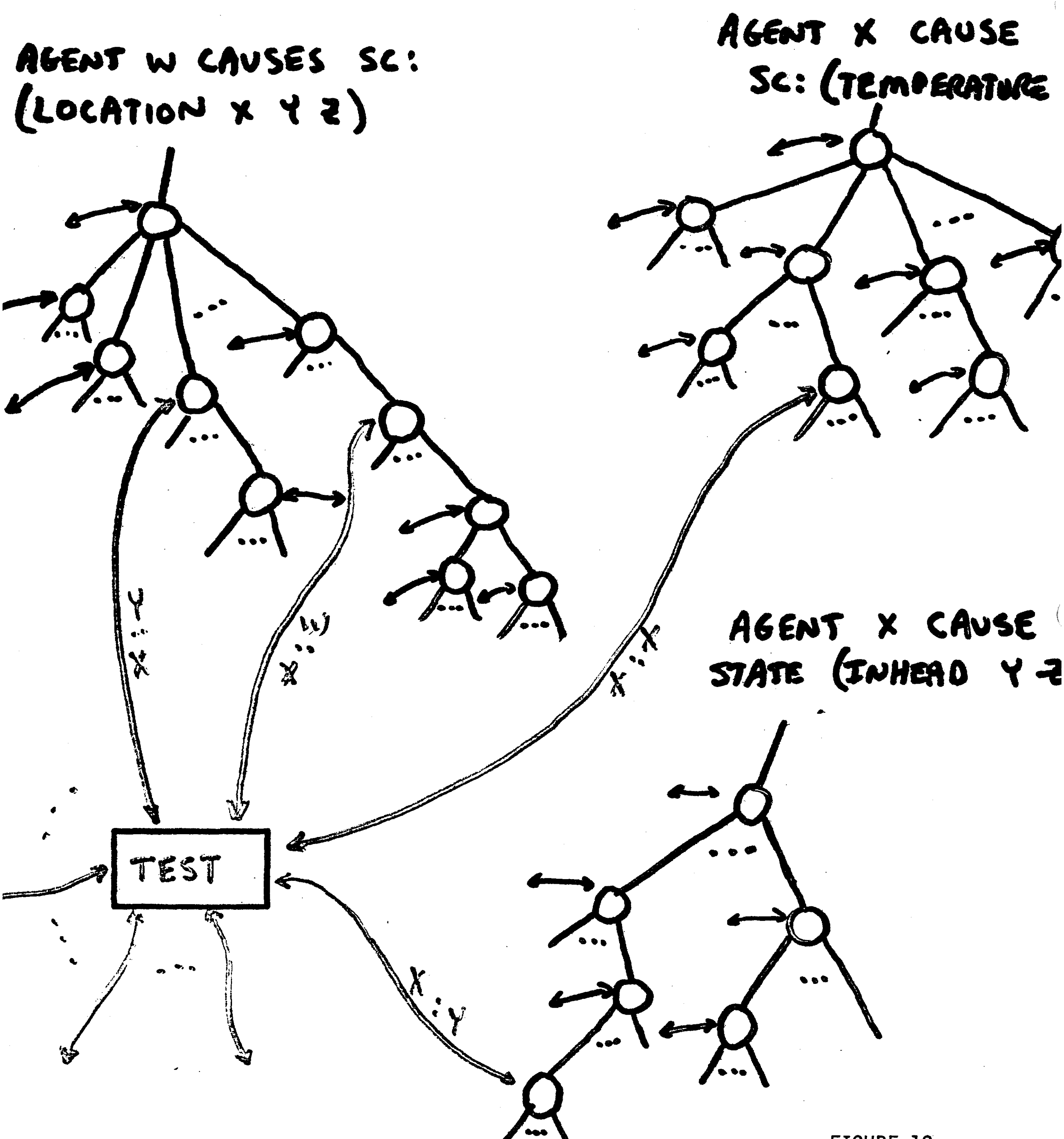


FIGURE 19

CENTRALIZED, SHARED TESTS ALLOW DISTRIBUTI
OF ENVIRONMENTAL AWARENESSES TO ALL
PARTS OF THE SYSTEM WHICH CAN USE THE

BYPASS IMPLANTATION

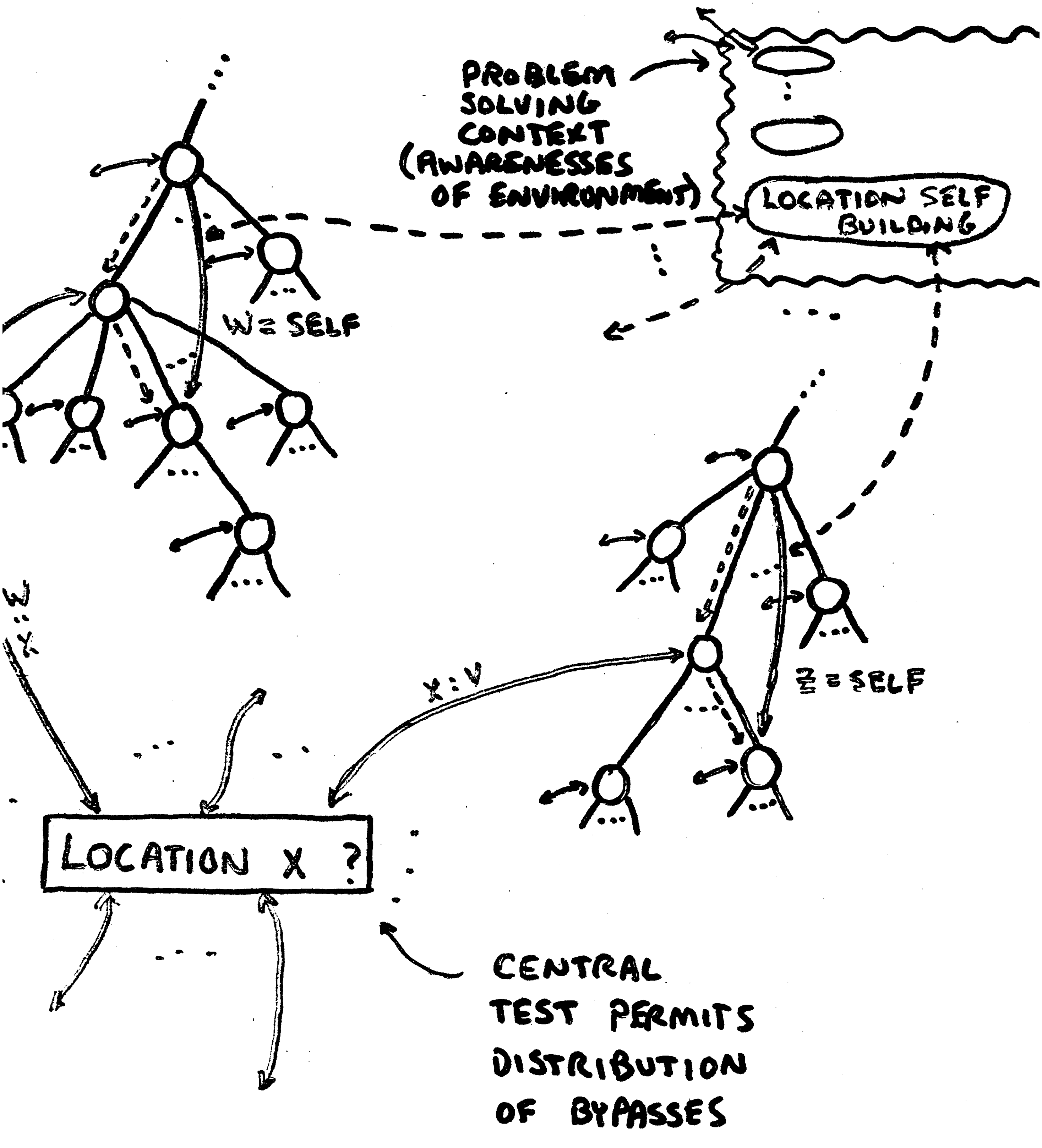
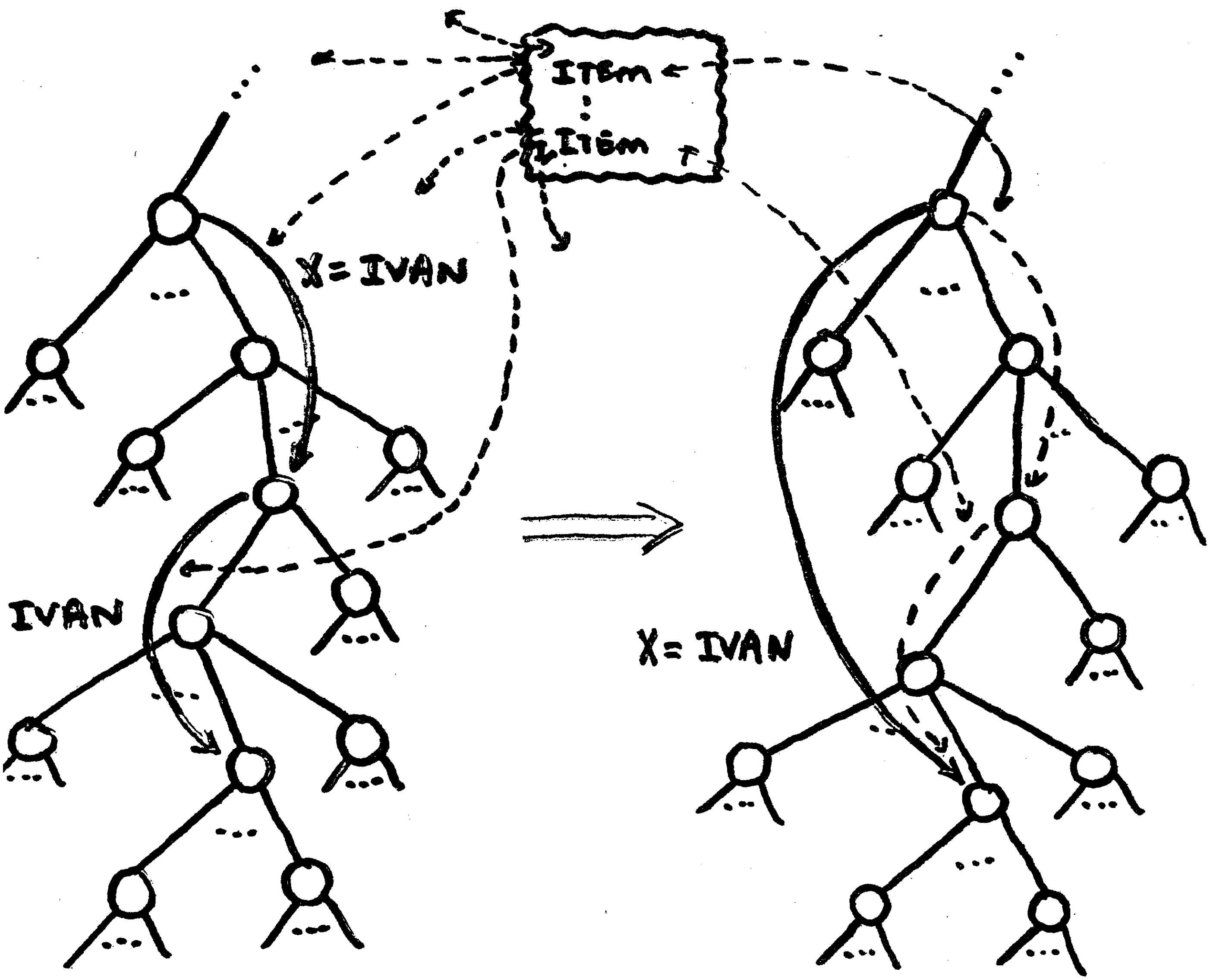


FIGURE 20

CLOSING BYPASS SEQUENCES



CLOSURE IS POSSIBLE ONLY FOR COMPATIBLE SEQUENCES

FIGURE 21

BYPASSING AN ENTIRE CAUSAL SELECTION NETWORK

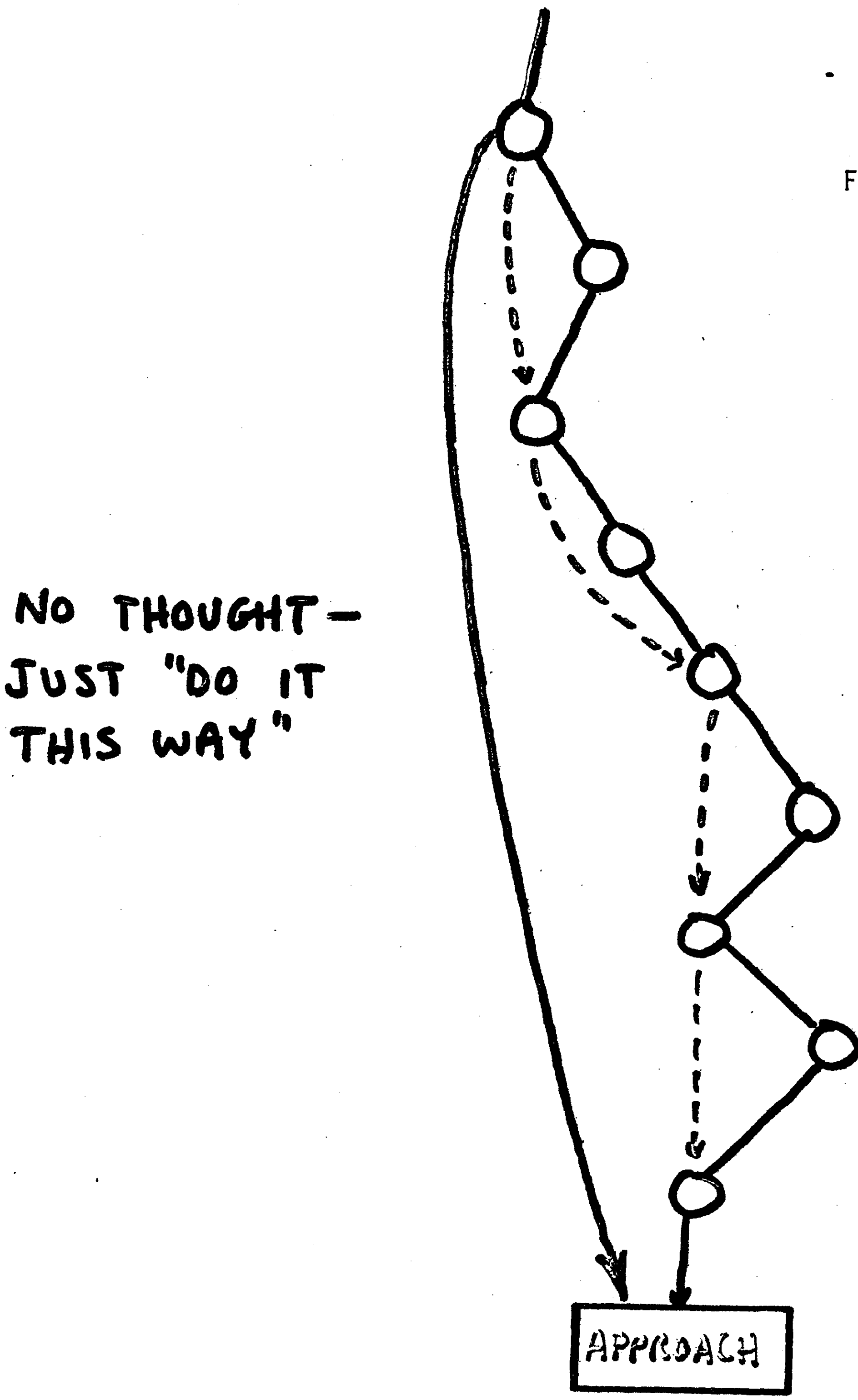


FIGURE 22

more of the environment becomes known, the solutions to problems tend to grow increasingly predetermined, or stereotyped.

If we take this bypass system back to the solution of the goal I posed a while ago, (LOCATION FLUID STOMACH(SELF)), you can see that by the time the "use a water fountain" approach has been selected at the top level, enough will have been determined about the environment - namely, that self is a human who is currently in an office building where there are predetermined ways of getting around - that when it comes time to solve the <u>subproblem</u> STATECHANGE (LOCATION SELF OFFICE(SELF) WATER-FOUNTAIN) in another part of the network system, there will already be bypasses pointing the problem solver <u>toward</u> approaches like "climb the stairs", or "take the elevator", and <u>away</u> from approaches such as "take an airplane", "swim", and so on.

No doubt the word "frame" is throbbing through your networks (FIG. 23), since we generally imagine a frame to be some sort of entity which imposes constraints on the way things are perceived, interpreted, or dealt with. This bypass arrangement is indeed frame-like. But its frames aren't discernable as distinct packets of knowledge. Instead, they are distributed bundles of bypasses which can blend in essentially infinite variety. Notice that because of this, we get a continuity of context where, instead of "switching from one frame to another", the system can flow smoothly from context to context as individual aspects of the environment change. This is because "context" in the bypass system amounts to the set of questions posed and answered while solving problems so far in the environment, and of course, the cumulative set of bypasses introduced by this set of questions and answers. This serving as the definition of problem solving context, if we assume the existence of an overseer in the system who is responsible for monitoring the continued truth of items in this context, then as aspects of the context become no longer true, the bypasses associated with those aspects can be removed from the system of networks. This will cause the overall problem solving behavior of the system to grow slightly less automatic in the slightly altered environment. In other words, as the context changes, bypasses fade in and out. At any given moment, the set of bypasses in the system will guide the problem solver toward solutions which are most appropriate in that environment.

This model suggests a wealth of related mechanisms. I will mention two. The first is: suppose some item, such as (CLASS SELF HUMAN), to take the same very obvious example again, is found to remain in the problem solving context more or less continuously over time. Then perhaps it would be reasonable to discard it from the context, allowing only its bypasses to remain permanently. That is, after a while I simply <u>know</u> I am human, and, knowing this, am always one bypassed step closer to selecting an approach to problems whose networks care about my class membership. And this phenomenon of relatively permanent bypass implantation seems to occur in far subtler ways. One day I was repairing the plumbing under our kitchen sink. I had removed a section of the drain pipe, and as I did so, a considerable quantity of water rushed from

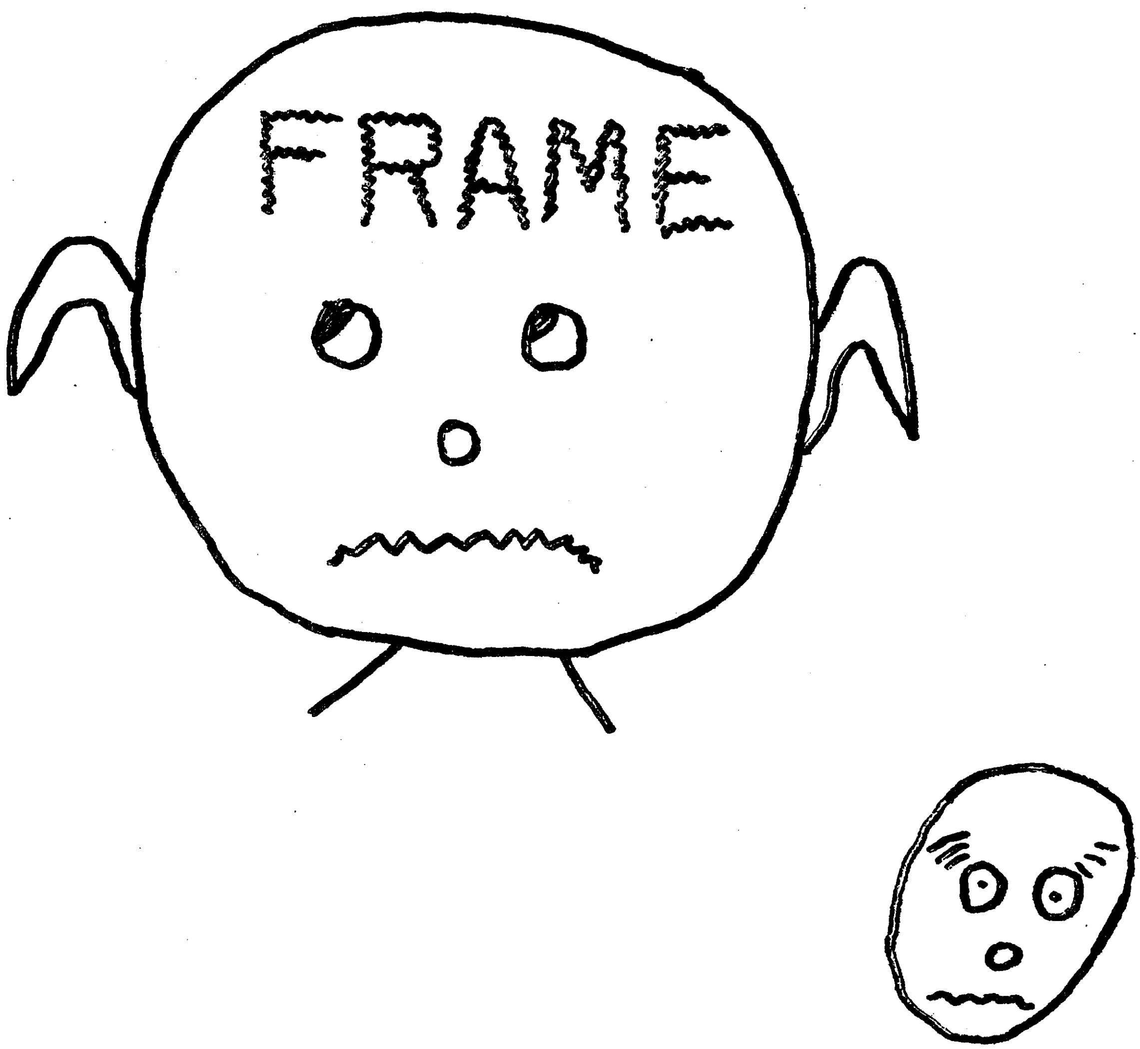


FIGURE 23

the sink into the catch pail I had fortunately positioned underneath the operation. I worked several minutes more, but being annoyed by the bucket full of water, decided to dispose of it. Now, I won't swear to it, but I think I called up my causal selection network for how to dispose of objects, and it immediately told me that, since I was standing right next to a sink, and since the object to be disposed of was a fluid, that I should simply pour the water down the sink. I did so without a moment's hesitation, even as I held the section of disconnected drain pipe in my other hand! I spent the next few minutes mopping up under the cabinet and cursing my causal selection networks.

The second mechanism one might envision in the bypass system is one which, as a particular overall bypass configuration - that is, set of awarenesses of the environment and their related bypass sets - was seen to recur frequently, could freeze that configuration, in effect naming it and forming a more packet-like snapshot of the environment (FIG. 24). Such frozen bypass configurations could be stored away. Later, when evidence presented itself that the context represented by some frozen configuration was again likely to be present, the entire frozen chunk could be called in, its bypasses being implanted en-mass throughout the system of networks. This would cause a more sudden, discontinuous leap in the system's awareness of context, with a concomitant, system-wide increase in the degree of stereotypy in problem solving behavior. Perhaps that elusive packet of knowledge I envisioned when I first heard the term "frame" might simply be one of these distinguished configurations of bypasses which has occurred often enough, and with small enough variation, to have been frozen, and thus in effect named. To jumble metaphors a bit, perhaps this more continuous bypass system is the broth from which those more discrete animalcules, which I once imagined frames as being, emerge.

There is much, much more to be said about the organization of the causal selection networks and techniques of plan synthesis using them. There are also many interesting questions about learning in such an environment: how do the selection networks grow and evolve, how are abstract algorithm patterns initially composed from sequences of perceptions of the world, and so forth. These are current topics of research. But rather than continue along one of these lines, I will stop talking about the use of these commonsense algorithm memory structures in problem solving, and turn now to their use in language comprehension (FIG. 25).

LANGUAGE COMPREHENSION

I view language comprehension as that process which elucidates the interrelationships among a collection or sequence of thoughts by consulting the kinds of world knowledge stored in an algorithmic memory of the sort I have been describing. And this process of elucidating the interrelationships should feed back, causing still other interrelationships to be perceived and awareness of context to expand. I feel the basic character of the language comprehension process

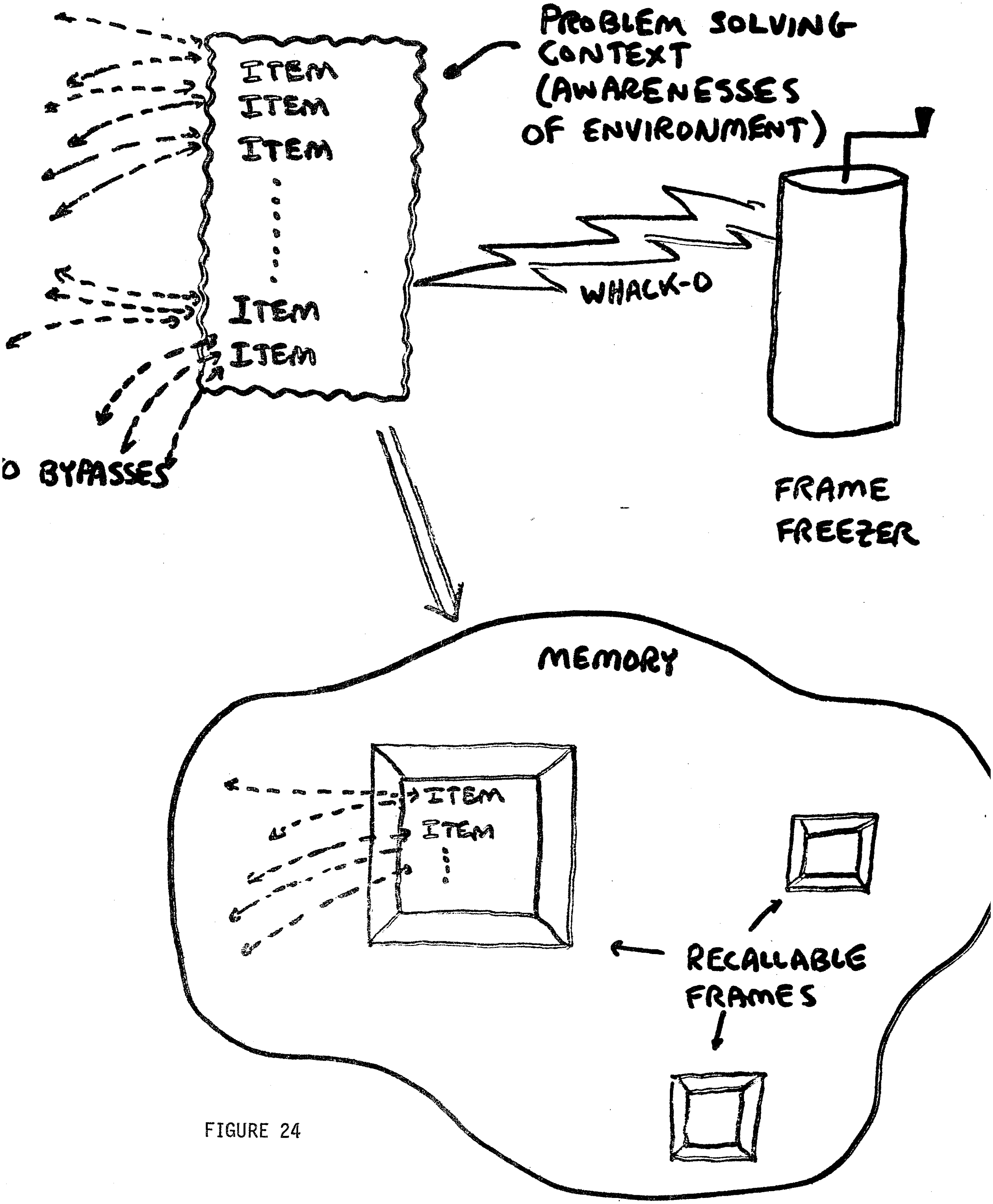

PROBLEM SOLVING CONTEXT (AWARENESSES OF ENVIRONMENT)
ITEM
ITEM
ITEM
ITEM
ITEM
O BYPASSES
WHACK-O
FRAME FREEZER
MEMORY
ITEM
ITEM
RECALLABLE FRAMES

FIGURE 24

1. HOW TO REPRESENT CHUNKS OF COMMONSENSE ALGORITHMIC WORLD KNOWLEDGE

2. HOW TO ORGANIZE MANY SMALL CHUNKS OF THIS KNOWLEDGE INTO USEFUL LARGER PATTERNS

→ 3. HOW TO USE THIS MEMORY WHILE COMPREHENDING LANGUAGE

FIGURE 25

is one of prediction/fulfillment, wherein every perception gives rise to general - and I want to emphasize the word "general" - expectations about what might follow, or it fits into some existing expectation, or both. While I will be talking in terms of language comprehension, where the source of incoming thoughts is linguistic, the approach I will be describing ought to apply to any collection of thoughts, regardless of their origin.

Let me first define language comprehension in a slightly more concise way (FIG. 26):

> Given a context $C(T_1,...,T_i)$ which has been established by thoughts $T_1,...,T_i$, make explicit the relationship between the next thought T_{i+1} and $C(T_1,...,T_i)$. Call this explicit relationship the interpretation of T_{i+1} in this context: $I(T_{i+1},C(T_1,...,T_i))$.

Now, the job will be to define $C(T_1,...,T_i)$, which will represent the <u>predictive</u> component of the theory, and $I(T_{i+1},C(T_1,...,T_i))$, which will represent the <u>fulfillment</u> component. For the sake of simplicity, let's restrict the problem to the case $C(T_1)$, where we want to discover $I(T_2,C(T_1))$. Examples of this task are shown in FIG. 27.

How should we proceed? Suppose we can use T_1 to generate some expectancies about the kinds of commonsense activities we might expect the various potential actors in the situation we are perceiving to engage in. If these expectancies can be kept "defocussed" enough to provide a good target for the processes which will search for subsequent relations between T_1 and T_2, yet well enough organized to make searching through them practical, we will have the basis for computing $I(T_2,C(T_1))$.

The commonsense algorithm memory organization I have been describing provides both the essential breadth, or defocussedness, of expectancies, <u>and</u> enough internal organization to make searching practical. Suppose, by an inference process, we can predict each potential actor's probable reactions in a given situation; in other words, that we can infer a set of likely goals each potential actor might wish to accomplish in the situation. In the commonsense algorithm system, goals can be expressed as WANTs of states and statechanges. As such, an expected goal will be essentially a pointer, with suitable variable instantiations, to the top of what is implicitly an <u>extremely</u> large structure in the memory... that is, a pointer to the top of some causal selection network which explicitly ends in abstract algorithm approaches, but which <u>implicitly</u> extends deep into other causal selection networks via the various subgoals and recommendations mentioned in the various abstract algorithms at its bottom. If a prediction component can identify the <u>tops</u> of some networks as being actors' likely goals, then it has effectively identified an <u>entire realm</u> of things those actors might do to realize those goals, namely, those abstract algorithms at the bottom of the network, all the abstract algorithms at the bottoms of the networks involved as subgoals within each of the first level of approaches, and so forth. "Implicit" is a very important word here; rather than having to make thousands of <u>explicit</u>

LANGUAGE COMPREHENSION:

GIVEN A CONTEXT $C(T_1, \ldots, T_i)$ WHICH HAS BEEN ESTABLISHED BY THOUGHTS $T_1, \ldots, T_i$, ELUCIDATE THE RELATIONSHIP BETWEEN THE NEXT THOUGHT, T_{i+1}, AND $C(T_1, \ldots, T_i)$. CALL THIS RELATIONSHIP THE INTERPRETATION, $I(T_{i+1}, C(T_1, \ldots, T_i))$, OF T_{i+1} IN THIS CONTEXT.

SIMPLEST CASE: $I(T_2, C(T_1))$

FIGURE 26

EXAMPLES OF $I(T_2, c(T_1))$

1. a_1 IVAN FELT THE FIRST DROPS OF RAIN
   b_1 IVAN DIVED UNDER THE BUS.

   a_2 IVAN HEARD A THUD AND SAW OIL LEAKING
   b_2 IVAN DIVED UNDER THE BUS.

   a_3 IVAN HEARD HIS ANGRY WIFE COMING.
   b_3 IVAN DIVED UNDER THE BUS.

2. a_1 ANNA WANTED PETER TO NOTICE HER.
   b_1 SHE PICKED UP A ROCK.

   a_2 WHEN PETER SAW HER, HE STUCK OUT HIS TONGU
   b_2 SHE PICKED UP A(NOTHER) ROCK!

3. a PETE STOLE JAKE'S CATTLE.
   b JAKE SADDLED HIS HORSE.

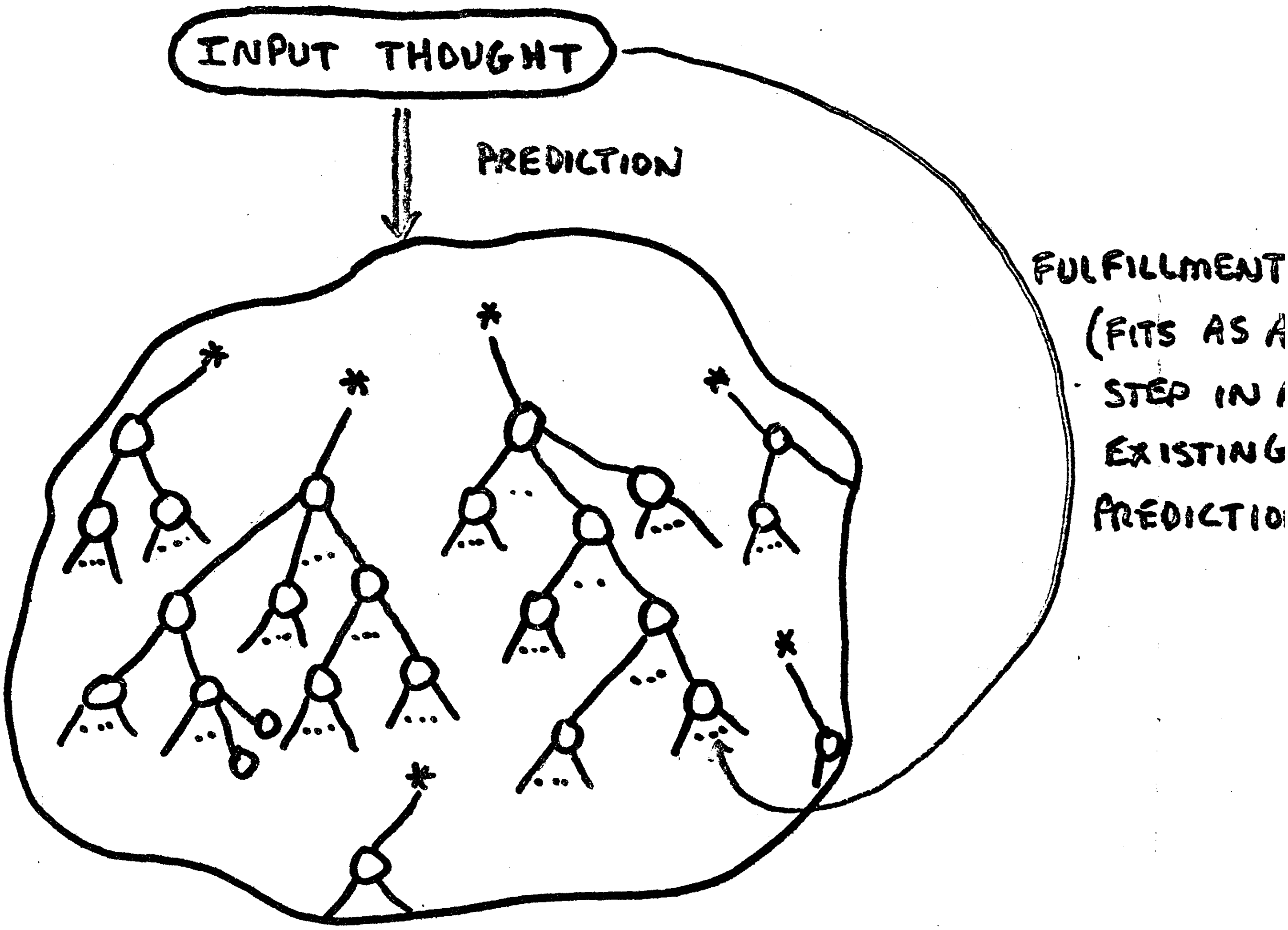
FIGURE 27

expectancies, we gain implicit access to them by pointing at the top of some network.

Now, suppose we have at each moment some collection of such predicted WANTs (FIG. 28). Then, the essence of the language comprehension reflex will be to identify how each subsequent thought fits into one of these implicitly large structures as a step toward achieving some expected goal. If the system can do this, then the relationship between the thought which gave rise to the expectancies and the thought which has been identified as a step toward achieving one of these expectancies will be that upward path through layers of abstract algorithm approaches and causal selection networks which connects the fulfilling thought to the expectancy-generating thought. This path will be the desired interpretation, I(T,K), of thought T in the context K, where K, the language comprehension context, is conveniently defined to be the composite set of various actors' expected WANTs which have been inferred from the sequence of preceding thoughts.

This being the general outline for language comprehension, let me first describe how the set of expectancies arises from the incoming thoughts. For this purpose, there are two other types of networks in the system: inducement networks and prediction networks (FIG. 29). Structurally, these two varieties of network are similar to the causal selection newtorks, in that they participate in the same bypass and context mechanisms, and in that they are organized around state, statechange, and additionally, action concepts in the system. Their differences lie in their use and in the types of information at their terminal nodes.

An inducement network's purpose is to determine of a given event or state those internal, psychological states that event or state could induce in a potential actor. In other words, inducement networks are designed to relate what a potential actor experiences to what he might feel internally in reaction to those experiences. For example (FIG. 29, left), if we take KISS as an action concept expressible in the system - not primitive, since it references an abstract algorithm - then there will be an inducement network whose job it is to infer the internal states the event AGENT W CAUSES ACTOR X TO KISS OBJECT Y might induce in INDUCEE Z, who is assumed to be aware of the event. Suppose then, some thought tells us that agent IVAN caused actor IVAN to perform the action (KISS IVAN NATASHA) while BORIS was watching. Then, by entering the KISS inducement network with INDUCEE= BORIS, AGENT= IVAN, X= IVAN and Y=NATASHA to discover how this kissing event might affect BORIS, the system might discover, as the result of asking relevant questions as it worked its way down the KISS inducement network, that BORIS is likely to experience an induced state of extreme jealousy toward IVAN as a result: (MFEEL BORIS JEALOUSY IVAN). So that what we find at the bottoms of inducement networks are sets of internal states which an event might induce in the INDUCEE. Of course, exactly which induced states, if any, are predicted will be a function of the answers to questions posed by the inducement network... questions which selectively probe relevant aspects of the situation. If, for example, the network discovers no emotional ties

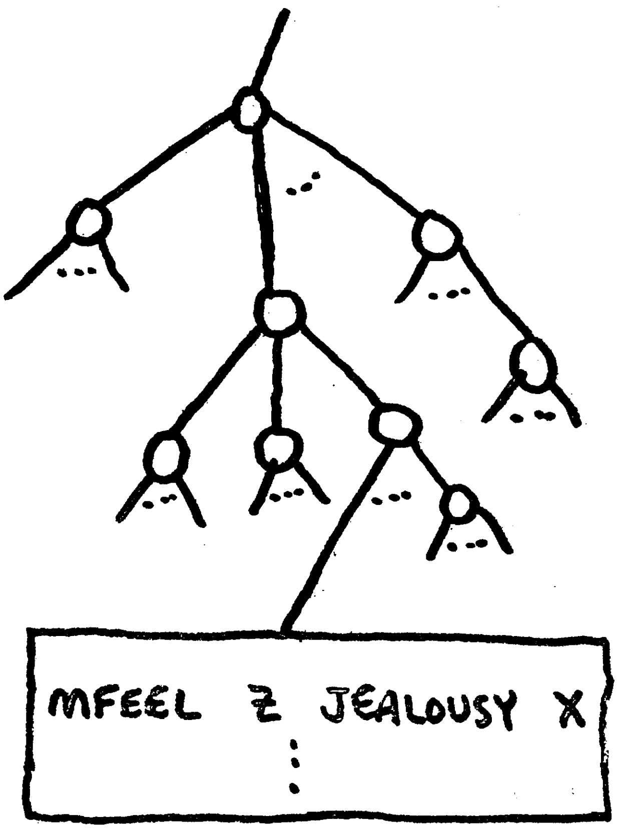


EACH INPUT CAN CONTRIBUTE
SOME PREDICTIONS (EXPECTANCIES),
FULFILL OTHERS

FIGURE 28

INDUCEMENT NETWORKS

WHAT STATES MIGHT
AGENT W CAUSE (KISS X Y)
INDUCE IN Z ?



W = IVAN
X = IVAN
Y = NATASHA
Z = BORIS

PREDICTION NETWORK
HOW MIGHT W
REACT TO
(MFEEL X Y Z) ?

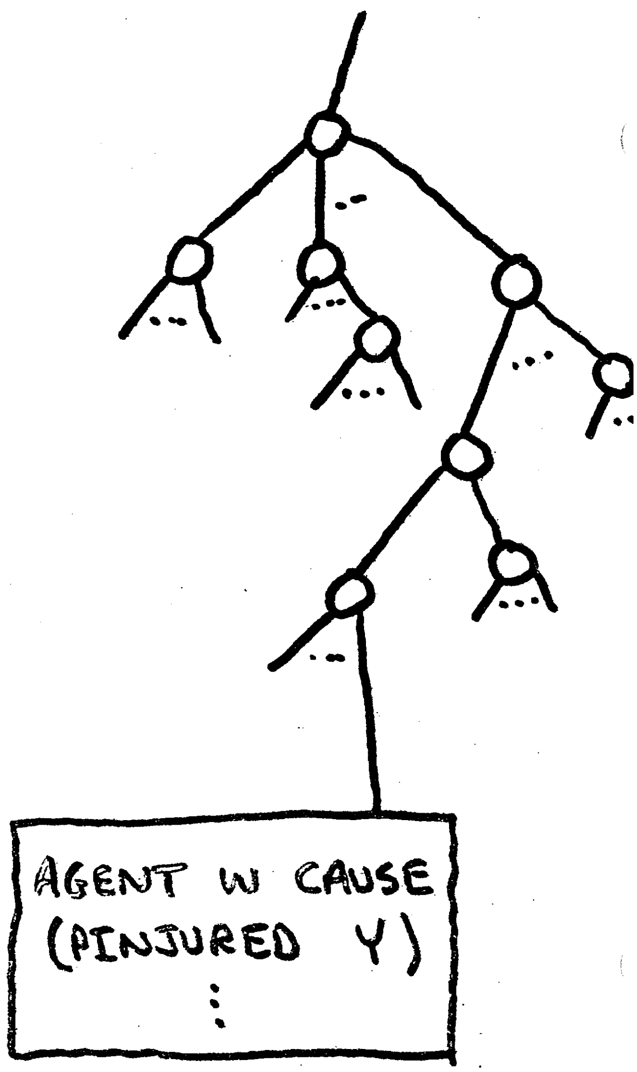


FIGURE 29

between BORIS and NATASHA, no induced states may be predicted at all. On the other hand, if we run the KISS inducement network with INDUCEE= IVAN, we would perhaps arrive at the induced state "IVAN is self-satisfied." And who knows what we would get from applying the network with INDUCEE= NATASHA!

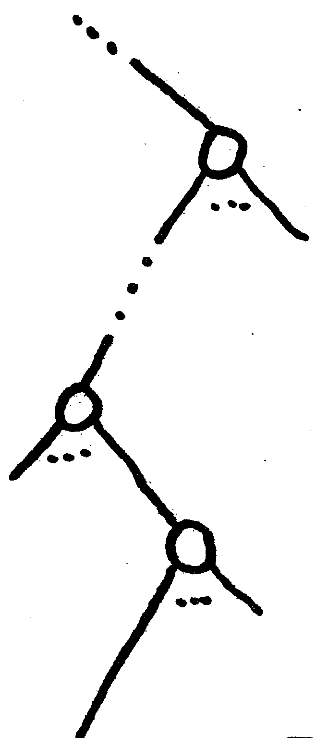
So, the inducement networks provide a method for inferring how various conditions in some situation might affect a potential actor. The system will put these networks to use during comprehension as follows. As each new thought enters, the comprehender applies the appropriate inducement network to it, varying the INDUCEE over all known potential actors in order to discover possible induced states for each. In this case, if IVAN causes (KISS IVAN NATASHA), we will run the KISS network once for INDUCEE= IVAN, once for NATASHA, and once for BORIS, if these are the three known potential actors.

If the inducement networks can thereby infer some induced states, these states will then serve as the input to the <u>prediction</u> networks (FIG. 29, right). It is the role of a prediction network to relate internal states of actors to goals they may be motivated to attempt as the result of being in those inferred states. Suppose, for example, the KISS inducement network decides that (MFEEL BORIS JEALOUSY IVAN) is likely. By applying the (MFEEL X Y Z) prediction network to this state, relevant questions will once again be posed to discover what, if any, X's and Z's responses might be, assuming again that each is aware of the MFEEL condition. For example, the section of the MFEEL prediction network dealing with JEALOUSY will ask questions about X and Z's ages, social relationship, degree of the MFEEL, and so forth. If, for example, the prediction network discovers that BORIS and IVAN are school children, who are rivals and are off on some remote part of the playground away from teacher, then it might predict that BORIS might employ some sort of physical retaliation against IVAN... that he might set about accomplishing the state (PINJURED IVAN). On the other hand, if IVAN is BORIS' boss at the factory, some other reaction to the induced MFEEL would probably be expected.

As in the causal selection networks, we would like to have the mechanism for accumulating larger and larger stereotyped patterns at the bottoms of both the inducement and prediction networks. To illustrate, we may wish to obviate the application of the MFEEL prediction network by including as a <u>recommendation</u> in the KISS inducement set a direct reference to some sort of physical retaliation (FIG. 30). This would amount to saying that, in the <u>specific</u> context of an MFEEL JEALOUSY which has been caused by an act of kissing where emotional relationships are involved, there may be a stereotyped reaction to the induced MFEEL-JEALOUSY which is tightly attached to this inferred internal state. In other words, how one reacts to an induced state is often also dependent upon <u>how</u> that state arose. The recommendations allow the system to capture this dependency.

We have just been considering the manner in which the comprehender reacts to

RECOMMENDATIONS AT BOTTOMS OF INDUCEMENT AND PREDICTION NETWORKS



(MFEEL Z JEALOUSY X)

RECOMMENDATION: (AGENT Z CAUSE (PINJURED X

FIGURE 30

external thoughts as they arrive. In reality, all inducements and predictions generated internally are fed back into other appropriate networks in order to derive second-order and higher effects of the various inferences made by these networks. When, for example, the AGENT BORIS CAUSES (PINJURED IVAN) prediction arises, it can, being fed back into the PINJURED inducement net, give rise to expectancies about what IVAN might do in anticipation of this behavior on BORIS' part.

Now, let's return to the larger picture of the comprehender. As the sequence of thoughts arrives, each is processed via this inducement/prediction network sequence, which gives rise to a collection of likely goals all potential actors might possess. At this point, the prediction component has implicitly established contact with the tops of various causal selection networks (FIG. 31). As each new prediction is made, the comprehender will do one of two things: either stop the forward, predictive activity at that point, or go ahead and attempt to continue by predicting <u>how</u> the potential actor might actually attempt to realize his goal. That is, the prediction can be continued by applying the causal selection process to identify a likely approach. For example, if the system has reason to believe that BORIS will want to cause IVAN to become PINJURED, it will attempt to continue the prediction by trying to guess how BORIS might go about the attack. Sometimes the environment will be constrained enough that such predictions are possible; other times, there will be no good basis for further prediction. In the latter case, the system will stop, finding that the majority of causal selection network questions being posed are not yet answerable.

Now let's look at the fulfillment side of the comprehender. In addition to being filtered through this prediction component, each input thought triggers a process of upward searching through layers of causal selection networks and abstract algorithm approaches at their bases (FIG. 32). Suppose, for example, we hear next, after this kissing incident, that BORIS grabbed a rock: BORIS CAUSE (ATTACHED HAND(BORIS) ROCK). First, the system locates all patterns in all abstract algorithms in the system which match this input pattern, preferring those matches which are most specific first. Locating all the occurrences of, say, an (ATTACHED X Y) pattern in the algorithmic base is possible in the system via a cross-indexing of each concept's occurrence throughout the system. For the sake of illustration, suppose the searcher finds that the input thought matches (ATTACHED X Y) patterns at 25 points in various abstract algorithms at the bases of various causal selection networks. Then, each of these occurrences constitutes a place where the new thought could potentially fit as a step in some higher activity, hopefully one of the predicted activities from the prediction networks. Having identified these points of occurrence of the (ATTACHED X Y) pattern, the comprehender's goal becomes to search upward through the causal selection networks from these 25 points, hoping to encounter, before too long, some goal in the prediction set. If such an upward path can be found, the comprehension system will have found a relationship between the new thought and the context, and hence comprehended according to my definition.

CONTINUED FORWARD ELABORATION OF PREDICTIONS VIA CAUSAL SELECTION NETWORKS

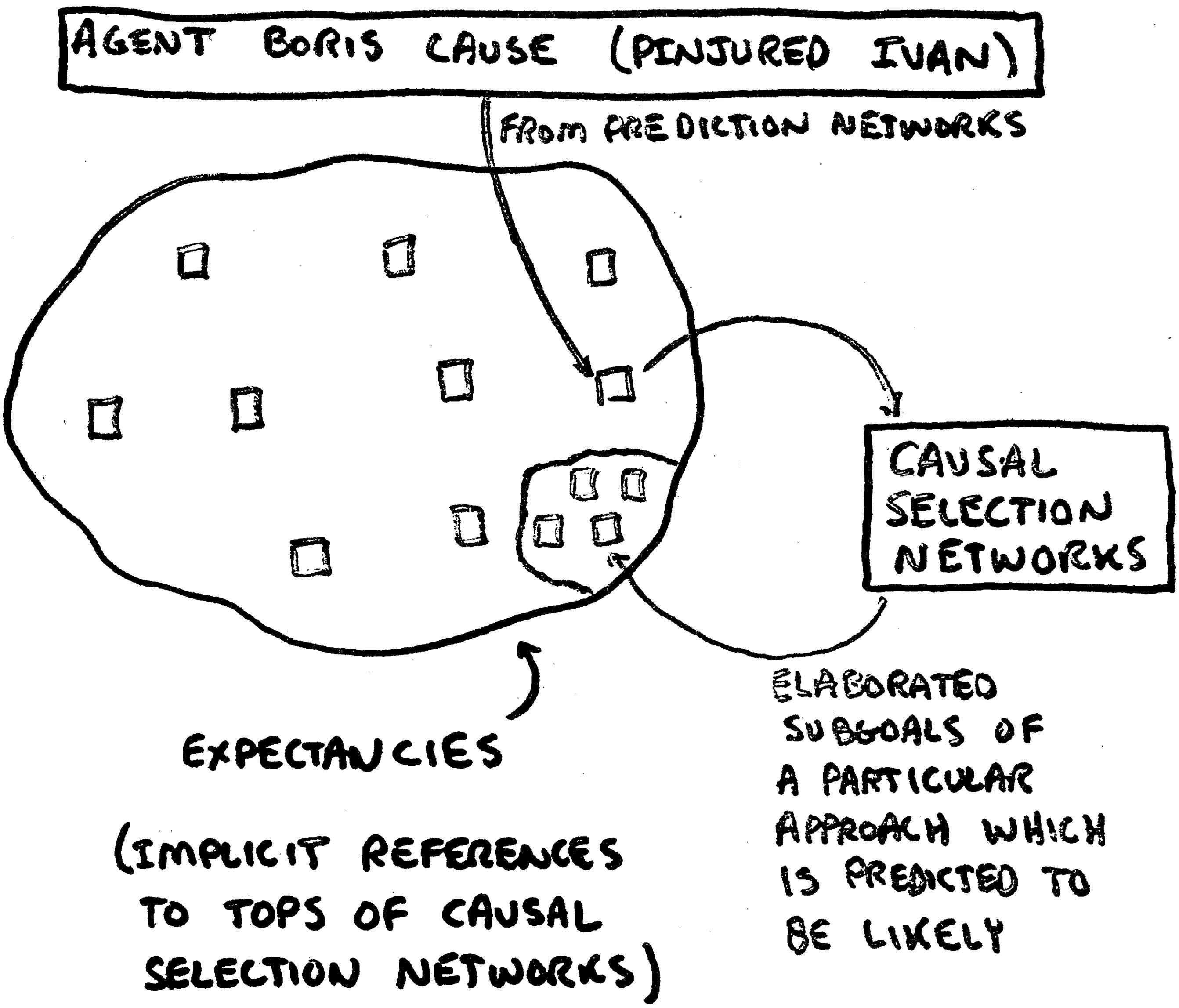


FIGURE 31

FULFILLMENT: LOCATING POINTS OF OCCURRENCE OF A THOUGHT THROUGHOUT THE CAUSAL NETWORKS

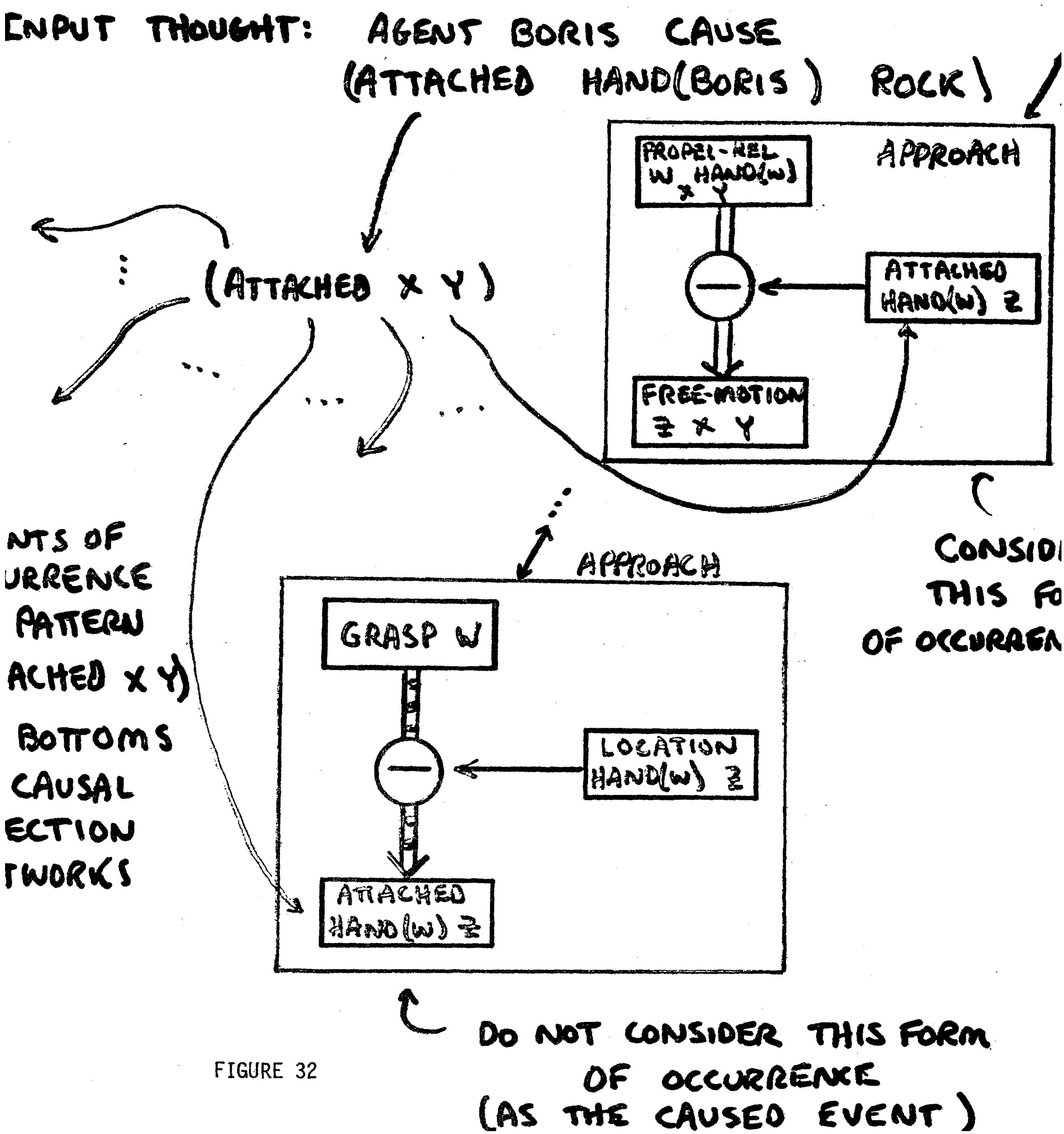


FIGURE 32

This upward searching has the appearance of being a very time consuming operation. Indeed it would be without some means of deciding which of the 25 paths seem to be the most fruitful. But the structure of the causal selection networks offers a convenient means of doing just that. Let's consider what happens, starting from one of these 25 points of (ATTACHED X Y)'s occurrence in an abstract algorithm at the base of some causal selection network (FIG. 33). Abstract algorithms are backlinked to all points where they occur as the terminals of networks. Starting from each occurrence at the bottom of some network, the searcher works its way upward through the causal selection network one node at a time. At each node, the test stored at that node is posed. If the result of the test is such that, had the test been made by the problem solver during plan synthesis, it would have routed the system to the offspring node from which the searcher has just climbed in its upward search, then the search continues, retaining the path as still fruitful. If, on the other hand, the test would have routed the problem solver to some other alternative node at that point, then the path is considered less fruitful, and will eventually be dropped from consideration as an interpretation path after failing several network tests in this manner. Paths which survive this kind of reverse filtering eventually reach the top of the causal selection network. If the goal state or statechange represented by the successfully climbed network is found <u>not</u> to be in the prediction set, this process is begun anew for the higher level goal represented by the network just climbed (FIG. 34); that is, occurrences of this newly-inferred higher level goal are located at the bottoms of yet other causal selection networks, and upward paths from each of those points sought. The process continues until a state or statechange in the set of predictions is reached, or until path length exceeds a cutoff value, indicating that, even if a path were to be discovered, it would be a rather remote interpretation. Path length is defined to be the number of causal selection networks passed through during the upward search.

Paths which survive this process will constitute possible interpretations of the new thought in the context of the preceding thoughts. In case there are several interpretations, the interpretation path finally preferred will be the shortest... that is, the one with the fewest subgoals intervening between the expectancy and the fulfillment. In case there are several shortest paths, the one which fared best during the application of the causal network tests on the upward search will be preferred.

Concerning the bypass mechanism's role during language comprehension, bypasses are not implanted until after the comprehender has obtained an interpretation path of which it is reasonably confident. Then, it reinforces that path by traversing it downward, distributing bypasses throughout the system from each test on the path. This amounts to <u>using</u> the interpretation path to infer pieces of the environmental context which were not known before; these can enhance the system's performance within that environment in the future. Also, as an interpretation path is found, the expectancy from which it has been derived is debunked from the

FOLLOWING AN UPWARD PATH FROM AN APPROACH TO THE TOP OF A CAUSAL SELECTION NETWORK

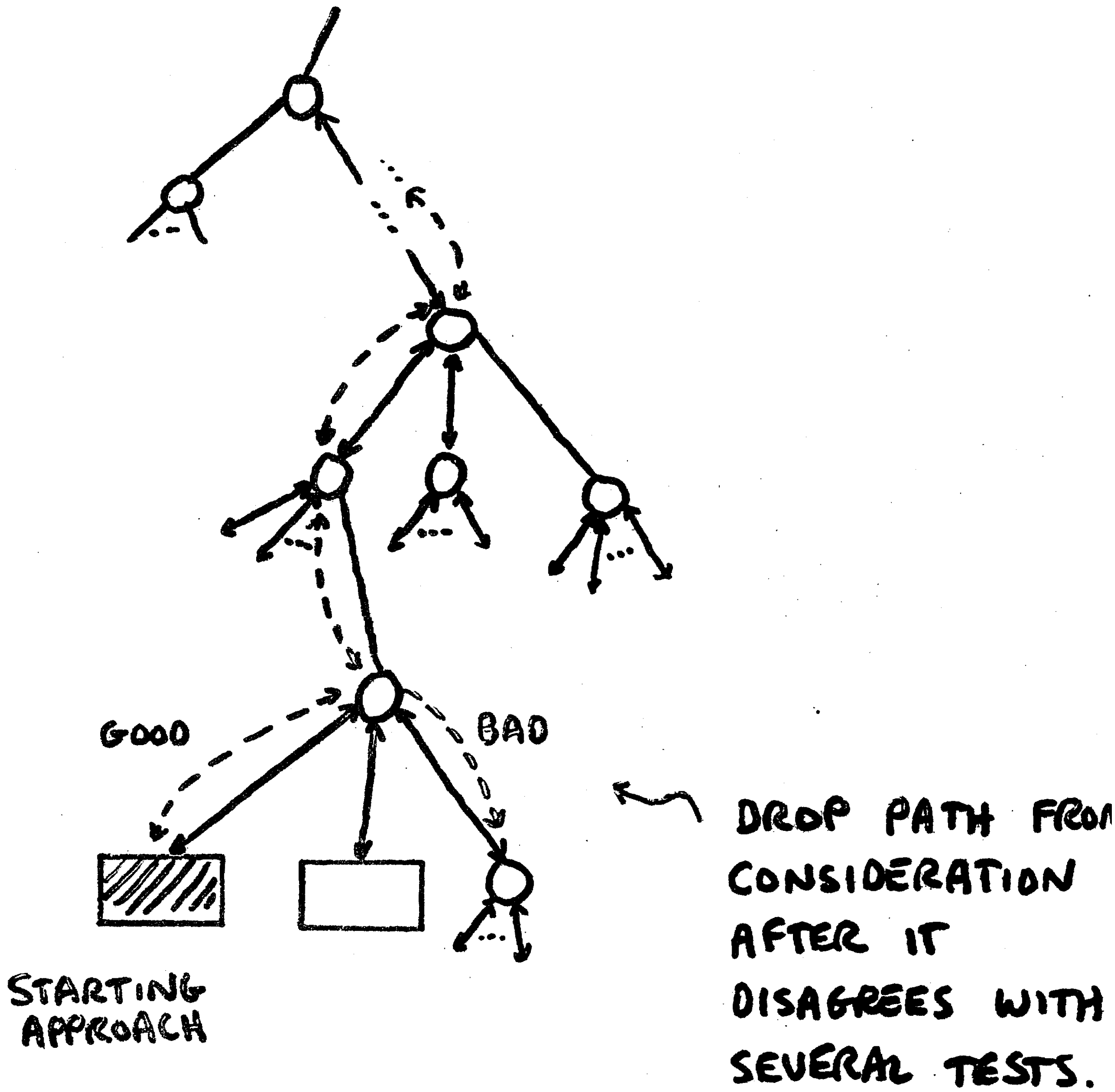


FIGURE 33

CONTINUING UPWARD PATH SEARCH THROUGH SEVERAL NETWORKS

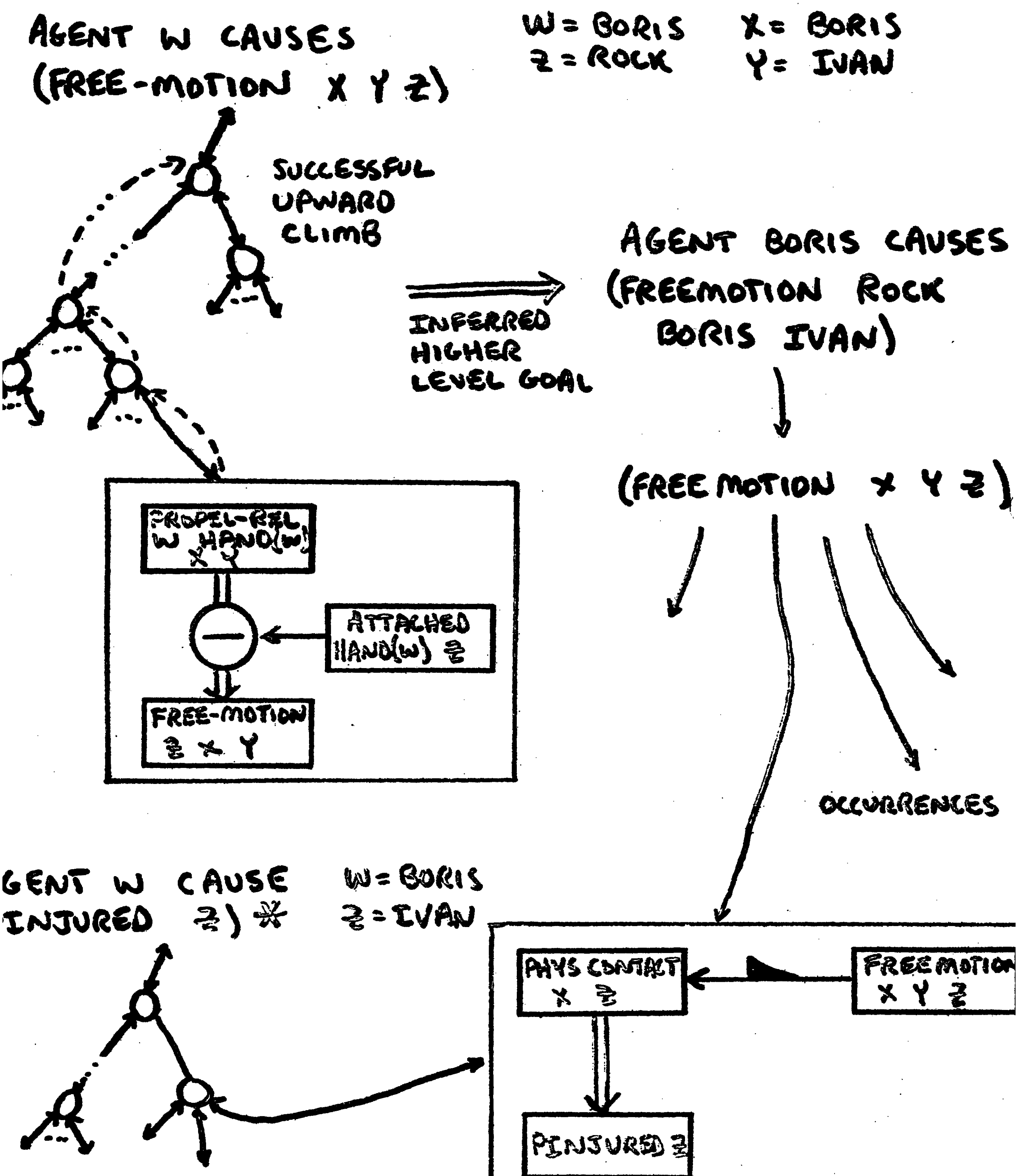


FIGURE 34

set of expectancies and replaced by the more specific expectancies, if any, which represent those uncompleted steps in the abstract algorithm approach in which the fulfilling pattern occurred. Concerning recommendations throughout the causal selection network system, the upward searcher will prefer to follow a stereotyped recommendation rather than climb upward through a causal network.

CONCLUSION

I have described the essential character of the model. Let me now stimulate your various networks by reading a joke which I have been carrying around with me this last year. I promise to lay it to rest after this last reading:

> "A man was out shopping for groceries, pushing his three year old son around in the cart. As they passed by the oranges, the kid took a swipe and knocked over the whole pile. The father said 'Cool it Oscar'. As they walked passed the broom display, the kid yanked a handful of straws out of one of the brooms, in response to which the father again said levelheadedly 'Cool it Oscar'. On their way around the corner to the meat, the kid let loose an epithet he had wittingly absorbed from the family's conversation at the dinner table the evening before. To this, the father merely repeated calmly, 'Cool it Oscar'. Finally, as they were checking out, an elderly lady who had been observing them remarked to the father that she greatly admired his restraint in dealing with his son. To that the father replied doggedly, 'Lady, I'm Oscar!'"

It is time to stop. Everything I have described today has been implemented in a small system which can synthesize plans as well as discover $I(T_2,C(T_1))$. Being small, the system cannot handle complex plan synthesis or comprehend full stories yet. But I am happy with the model as a foundation for these cognitive processes, and with the underlying theory as an explanation of some aspects of human thought.

RELATED READING

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

Charniak, E., "A Partial Taxonomy of Knowledge about Actions", in Proc. 4IJCAI, Sept. 1975

Fahlman, S., "A Planning System for Robot Construction Tasks", M.I.T. AI-TR 283, May 1973

Goldstein, I., "Understanding Simple Picture Programs," M.I.T. AI-TR 294, Sept. 1974

McDermott, D., "Assimilation of New Information by a Natural Language Understanding System", M.I.T. AI-TR 291, Feb. 1974

Minsky, M., "A Framework for Representing Knowledge", in The Psychology of Computer Vision, Winston (ed), McGraw Hill, 1975

Rieger, C., "Conceptual Overlays: A Mechanism for the Interpretation of Sentence Meaning in Context", in Proc. 4IJCAI, Sept. 1975

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

Schank, R., and Abelson, R., "Scripts, Plans and Knowledge", in Proc. 4IJCAI, Sept. 1975

Scragg, G., "Answering Process Questions", in Proc. 4IJCAI, Sept. 1975

Sussman, G., "A Computational Model of Skill Acquisition", M.I.T. AI-TR 297, Aug. 1973