# COMMON-SENSE THEORY FORMATION:

## Towards Understanding Baseball

Elliot M. Soloway*
Edward M. Riseman*

COINS Technical Report 75C-5

June 1975

*Computer and Information Science Department
University of Massachusetts
Amherst, MA 01002

## ABSTRACT

This paper outlines the design of a system that will exhibit a significant degree of independent learning in a complex spatio-temporal domain. This research will address a number of basic issues involved in the cognitive processes of learning and forming theories. The knowledge structure, the organization of logical processes, the ability to form hypotheses and conjectures, and the development of a system whose common-sense reasoning can be analysed are important facets of our research.

The domain in which we will test our ideas is the action-oriented game world of baseball. The goal is to develop general strategies which enable a system to pass from the description of simple low-level action primitives of running, swinging the bat, and throwing the ball, to a higher-level semantic description in terms of runs and innings, etc. Inference of rules that govern behavior becomes a process of inferring the logical and physical constraints placed on the actions of people and their hypothesized goals. The semantic analysis is guided by a database of general knowledge of the world, including information about the intuitive physical laws of the universe, the behavior of people, and action-oriented games. First, biologically motivated syntactic operations filter and reduce the initial mass of data. Then, semantic analysis proceeds in two stages: a) the system heuristically hypothesizes the goals of the individuals in the current action sequence; and b) it generalizes that hypothesized goal sequence according to some measure of similarity to include prior sequences of activity. The hierarchical structure of the database permits the hypotheses under consideration to call in the appropriate level of world knowledge.

This research will open avenues for investigating how people form conceptual structures since the system should exhibit human-like reasoning. It should allow manipulation of both the knowledge base as well as the mechanism for focussing attention, forming conjectures, and verifying generalized hypotheses. This would be a powerful tool for computer aided instruction since learning difficulties can be examined by proper questioning of the system. The ability to abstract concepts from a set of examples will allow an instructor to determine the quality of that set of examples. There are numerous applications to education and learning if the underlying principles can be established in this work.

## TABLE OF CONTENTS

# 1. INTRODUCTION

In our daily lives, we constantly observe and interact with events and objects in the world. Plotting our own course over the sands of space and time is complicated by the desires, needs, and whims which motivate the other animate beings in the world. Obviously, understanding these drives is not simply a matter of solving a set of equations. Rather, to function in the world, we must understand the goals (and their corresponding constraints) which structure the existence of animate creatures. This understanding provides insights that enable us to answer the following types of questions: Why did an event happen? Why was it surprising? What else might have happened?

In order to acquire this understanding, we must make use of a great deal of knowledge. Since we are not born with all of this knowledge, we must possess mechanisms for "learning" it. What are these mechanisms for learning? How can we come to understand the motives that are behind people's actions? We will examine these questions in terms of a mechanical learning system possessing human-like cognitive abilities. In particular, this paper examines a system which can form theories to explain in a common-sense fashion its observations of people's interactions in a constrained spatio-temporal domain. It uses conceptual knowledge to hypothesize, generalize, and integrate with prior knowledge the underlying "rules" which structure that activity.

The problem environments which we will address are action-oriented worlds parameterized in four dimensions: action, actor, location, and time. Namely, consider the game of baseball, and consider the stereotyped Mr. and Mrs. Baseball fan sitting in the bleachers watching their favorite sport.

The Mr. is drinking beer and enjoying the "subtle" managerial ploys--the hit and run, the stolen base, etc. Meanwhile, the Mrs. is complaining that she still does not understand what is going on. In fact, she probably understands much more than she gives herself credit for. With a little more patience and directed attention, she might master many of the rules. Certainly, with the benefit of critical advice, much of the activity should be understandable.

Similarly, the goal of our system is to observe simulated digitized games of baseball and infer the structure and many of the rules that govern play, without requiring a teacher. Certainly our system can be designed with "patience" so that's no problem. In addition, we give the machine a significant degree of semantic information. Initially, the system knows about the behavior of people in game-type situations, the primitive physical acts performed by people, and enough about the laws of physics to understand the physical relationship between the objects and actions involved in the game.

At the outset, let us make clear what we mean by "rules," and what approach we will take in order to discover them. Observe that

> the rules of an action-oriented game are essentially the <u>logical</u> and <u>physical constraints</u> placed on the players as they attempt to achieve their <u>goals</u>.

By physical constraints we mean, for example, that a player can't jump 30 feet into the air to catch a ball or that a player is not able to hit <u>every</u> pitch out of the ballpark, even though he may want to. The system is provided information on the range of the general physical constraints on people's actions. And by logical constraints we mean, for example, that a player must not run to third base before he runs to second base, or that after making three "outs" a team loses its opportunity to make any further score that inning.

Notice that the observation above implies that the rules of the game are not abstract concepts divorced from the motives, constraints, and observed actions of the players. Quite the contrary. This intimate relationship will permit our system to discover the rules through a common-sense reasoning process which uses facts about the world, general facts about people's purposes and of course, the observed actions.

Let us consider the meaning of "rules" more carefully. The concepts that our system is to learn more closely resemble the structure or conventions that are exhibited in baseball games rather than the formal rules of baseball. For example, our system may develop (among others) the conventions that "players often walk to the dugout after an out" or "the catcher throws the ball back to the pitcher." These are not the necessary truths of baseball as stated in the rule book. Nonetheless, they are relevant to the structure of the game, i.e., observed activity on the field. In order to explain and understand that observed activity, our system will not generate a baseball rule. Rather, our system will seek to explain and understand baseball by a) inferring the goals of the players and b) inferring the logical constraints placed on the players as they attempt to achieve their goals.

This approach is partially based on four fundamental characteristics of the real world:

1) people act in accordance with purpose or goals; there are reasons to explain actions.

2) information is redundantly encoded; there are lots of cues or paths to aid in the process of understanding.

3) events that are important recur; not until one understands the significance of repetitive events can one comprehend a novel event.

4) events are interpreted in terms of a higher level context; unless one has the general frame of reference, one will probably not understand the specifics of a situation.

In particular, these four aspects are true in the action-oriented game world of baseball, and they attest to its richness and robustness. The games of a society mirror in a stylized way the real-life activities of a society [30]. They capture people's actions and goals, as applied to a self-contained mini-world. At the same time, issues of time, space, and causality, which are an integral part of all real-life situations, are also at the heart of action-oriented games such as the world of baseball. The techniques we develop in this particular domain for interpreting and learning about people's actions, goals, and constraints should provide insights to corresponding problems in similar complex, real-world environments. The action-oriented worlds of the traffic corner, restaurant, grocery store, etc., all seem open to this analysis. Moreover, forming theories by abstracting an understanding of the rules from observations is similar to the problem of a student forming a "theory" from "observations" of a course protocol.

This paper is an outline of the design of a system currently being developed. Many aspects are treated briefly and a number of areas await careful development. In the following sections we will present the system depicted in Figure 1. The representation of events and knowledge in the system will be followed by a brief overview of the functional units of the system. First, there are heuristic algorithms that focus the attention of the system by filtering and processing the input data. As a TV camera seems to naturally follow the action, our system also is capable of attending to those features of the environment that appear innately and/or motivationally interesting. Next, there is a description of how the active semantic data base generates plausible hypotheses about the goals and constraints of the players. The system's use of "common-sense" inferencing allows its "reasons" for making these decisions to be readily discernable. As experience accumulates, the system can generalize

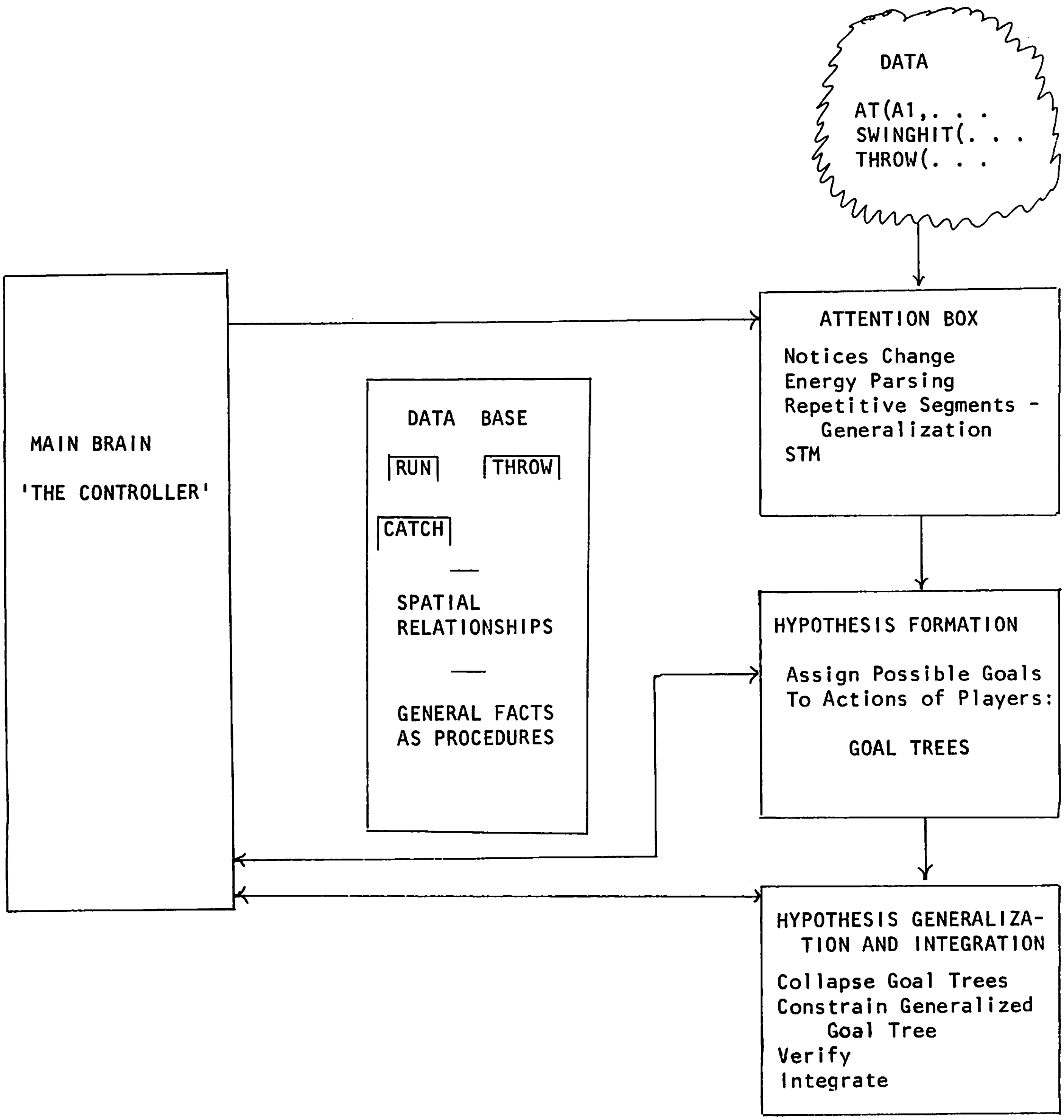


Figure 1. System Overview.

and integrate these hypotheses to encompass repetitive, though somewhat different, events. As the system operates, higher level and more abstract concepts are hierarchically built from simpler and more concrete hypotheses. The goal is to explain an out or an inning in terms of the semantics of gaming.

## II. REVIEW OF RELEVANT LITERATURE

In this brief review of the literature we will be looking mainly at the contributions of AI to learning and theory formation. Though the psychology literature is rich, and does provide a solid background, nonetheless, it usually does not address itself to the major problems with which our system must deal. Notable exceptions to this are the work in cognitive psychology and AI--Newell [1] and Simon [2], Norman, et al., [3], and Winograd [4]. Similarly, much literature in the philosophy of science [5,6] and inductive logic [7,8,9] deals with theory formation and the nature of explanation. These discussions are relevant, but the offerings are too general or too abstract. With these omissions then, let us proceed to the major work in machine learning.

Samuel's [10] checkers playing program is the classic example of computerized learning. Evaluation of a particular board was based on a linear function of a set of parameters which were the features of the game. During play, weights of these parameters were adjusted to reflect the "goodness and badness" of those features. Samuel achieved a great deal of success with this scheme in the checkers domain. However, by and large it has failed to generalize to more complex domains, e.g., chess. From a cognitive point of view, even if such a scheme were to work, what could we learn from it? Minsky [11] calls numeric learning schemes "evaluations" as opposed to "summaries." He argues that an intelligent entity ought to reflect on the considerations that formed that number, i.e., its history. Non-symbolic learning retains little information that would permit further analysis. The impact of Minsky's comments are clear: in order to gain insight into higher level cognitive learning, we should explore symbolic learning systems.

Fikes, Hart, Nilsson [12] have developed extensions to their problem-solving system STRIPS [13] for a robot living in a laboratory environment. The system is allowed to generalize and store the plans-of-action produced by STRIPS. A generalized plan can then be used by STRIPS under circumstances different than when it was first learned. The results of this system are impressive. However, they are due at least in part to a judicious choice of a constrained domain. The number of parameters in the plans is relative small, so syntactic generalization techniques appear to work. In a less constrained problem environment, semantic criteria will be needed to process efficiently and effectively.

Winston [14] made a contribution to symbolic learning. The operational goal of his system was to learn the meaning of various objects (arches and other physical structures) after having been presented with some examples. The objects were represented internally as graph structures, with the nodes and links having conceptual labels (supported by, left-of, etc.). A teacher was required to present a good well-ordered training set of examples and non-examples (labelled as "near miss").

Winston's system generalized its descriptions of an object by finding differences in the structural and physical descriptions of a series of examples. In order to keep the difference finding routine stable, the system was given an ordering on the importance of the differences. Though successful in the blocks domain, the sophistication and extensibility of Winston's techniques is quite another question. For example, the concepts that Winston's system learned (arch, pyramid) were only 1 level removed from the primitive descriptors (left-of, supported-by). Second, if the descriptions become at all complex, trying to <u>syntactically</u> match subgraph isomorphisms is doomed to failure. Third, requiring the teacher to lead the system "inch by inch" in forming and modifying concepts seems like too severe a constraint for all types of learning. We will be suggesting an approach and techniques which would remedy some of these drawbacks.

A system which extends Winston's work is the HACKER program of Sussman [15]. The goal of this system is to learn to build structures in the blocks world, e.g., towers, stacks, etc. It does this by "writing and debugging" computer programs that perform the required tasks. The system comes equipped with some primitive subroutines and extensive knowledge about programming. When a program is generated from canned subroutines, little "bugs" in the logic are apt to appear. Using its given knowledge about bugs, HACKER makes a patch to correct the errant program. Like Winston, Sussman's system requires that a teacher present a well-ordered training set of problems to enable HACKER to build the "right" programs first. Unlike Winston, Sussman's system can often discover for itself when it has failed. The performance of HACKER was intended to model the acquisition of a skill by an apprentice or novice. Sussman's system is promising and if the constraints on the "teacher" were relaxed, it would make the system even better.

The work done on Meta-Dendral [16,17] by Buchanan, et al., addresses itself to some of the similar issues our system will be facing. The goal of their system is to discover theories of mass spectroscopy which the performance program DENDRAL [18] uses. That is, DENDRAL uses a set of rules to output a molecular structure inferred from mass spectrograph input of the test samples. On the other hand, Meta-DENDRAL will try and induce those very rules.

Meta-DENDRAL first accepts as input mass spectograph data and the molecular structure of a sample chemical. Next, it generates hypotheses that might explain the data using heuristics to prune the huge search space. Plausible hypotheses are generalized and the system attempts to verify their validity. Finally, the discovered rules are integrated into the existing theory.

The philosophical basis for this system is discussed in an excellent paper by Churchman and Buchanan [19]. The authors readily admit and document these open questions in their work--which are essentially the open questions in the philosophy of science. Do the given pruning heuristics provide too much direction for the system, thereby begging the question of real discovery and theory formation? What is the actual data to which the system must attend, i.e., does the representation provide too much structure and reflect the biases of the authors? How are hypotheses generalized and verified? What is the criteria for similarity? These are precisely the issues we are facing in in our problem domain. One added difficulty these researchers have is that their problem domain, organic chemistry, is a rather esoteric area and the techniques developed may not be readily transportable to other more every-day real-world domains. Nonetheless, this is a most promising and sophisticated piece of research.

Uhr [26] stresses the importance of learning in a recent overview of a system being developed which integrates perception, deduction, action, and thought. In his wholistic system, multi-model sensory input passes through a network where it is acted upon in order to recognize and understand what is "out there." Additional cognitive transforms triggered by this initial process decide upon the actions to be taken by the system in response to this input.

These decisions are mediated not only by the external demands of the world, but also by the internal desires/states of the system. Because of the complexities involved in dealing with such diverse problems, the number of transformations needed by the system is quite high. He argues that only by hypothesizing and evaluating the utility of new transformations can the system develop truly powerful capabilities. The structures that Uhr deals with point directions for interesting research; however, they are too simple to properly deal with the approximated real world complexities and details we wish to consider.

Finally, there has been much work done on sequence extrapolation problems [e.g., 27, 28]. Even if all problems in induction are logically equivalent to a sequence extrapolation as Solomonoff claims [29], in practice, the techniques used to deal with the latter problem cannot span the chasm of real world induction problems. Furthermore, though some of the work attempts to model the way humans might perform sequence extrapolation tasks, we have great difficulty in seeing how these models can be extended to handle the more complex cognitive processes necessary for dealing with the richness of real-world learning problems.

# III. REPRESENTATION OF THE WORLD

Essentially, there are two major levels of semantic information that we must give the machine. The first level is an appropriate set of primitive descriptors that parameterize the significant dimensions of an action-oriented game world. The second level of semantic information is the context information. Baseball is in the context of competitive-conflict situations.

## III.1 The Primitive Descriptor Units--the Physical World

Input to the system is supplied by a program that simulates the continuous game of baseball by breaking it up into discrete time intervals, called snapshots. Each snapshot consists of a set of descriptors depicting the state of the world at that instant in time. A descriptor is a 4-tuple that captures 4 essential dimensions of such a miniworld--action, actor, location, and time. Figure 2 lists the set of primitive action predicates that we use in our system. They are the natural ones for describing an action oriented game, e.g., RUNNING, THROWING, CATCHING, etc. Predicates like INNING, BATBOX, etc., merely represent the various boxes on the score board. Note that the machine does not understand the semantics of an INNING; that is what it is supposed to learn. Other problem domains would require an augmented set, or even a different set. For example, the set of primitive action predicates for the traffic-corner world would include AT, ON, WALK, etc., but would also include CAR-MOVE, BICYCLE-MOVE, LIGHT-COLOR, etc.

An example sequence of snapshots is illustrated in Figure 3. Each predicate is labelled with its snapshot number to represent its time of occurrence. In addition, each snapshot will have a parameter to indicate the number of time units that have passed since the occurrence of the previous snapshot, i.e., the

CATCH(PLAYER,OBJECT,LOCATION)

THROW(PLAYER,OBJECT,LOCATION)

WALK(PLAYER,STARTING-LOCATION)

GRNDMOVING(OBJECT,STARTING-LOCATION)

AIRMOVING(OBJECT,STARTING-LOCATION)

INNING(INNING-NUMBER)

BATBOX(PLAYER-NAME,STRIKES,BALLS)

TEAMBOX(TEAM NAME,OUTS,RUNS-THIS-INNING,TOTAL-RUNS)

RUN(PLAYER,STARTING-LOCATION)

SWINGHIT(PLAYER,OBJECT,LOCATION)

SWINGMISS(PLAYER,OBJECT,LOCATION)

AT(PLAYER,LOCATION)

ON(PLAYER,LOCATION)

$\left.\begin{array}{l}\text{FAST}\\ \text{SLOW}\end{array}\right\}$ used as prefixes to modify actions

Note: The system does not understand the "baseball" meaning of any of the primitives. In particular, it understands only that INNING, BATBOX, and TEAMBOX are scoreboards. It does not know which events correlate with the counts and does not understand the concept behind INNING.

Figure 2. Listing of Primitive Descriptor Units

COMPLETE SNAPSHOTS:

| TIME: 14 | 15 | 16 |
|---|---|---|
| AT(A1,BALL,PM) | THROW(A1,BALL,PM) | AT(A1,BALL,PM) |
| AT(A2,HP) | AT(A2,HP) | AT(A2,HP) |
| AT(A3,FB) | AT(A3,FB) | AT(A3,FB) |
| . | . | . |
| . | . | . |
| . | . | . |
| AT(A9,RF) | AT(A9,RF) | AT(A9,RF) |
| AT(B1,HP) | AT(B1,HP) | AT(B1,HP) |
| AT(B2,DUGOUTB) | AT(B2,DUGOUTB) | AT(B2,DUGOUTB) |
| AT(B3,DUGOUTB) | AT(B3,DUGOUTB) | AT(B3,DUGOUTB) |
| . | . | . |
| . | . | . |
| . | . | . |
| AT(B9,DUGOUTB) | AT(B9,DUGOUTB) | AT(B9,DUGOUTB) |
| INNING(1) | INNING(1) | INNING(1) |

Figure 3a. Partial raw, prefiltered snapshots

---

REDUCED SNAPSHOTS:

| TIME: 14 | 15 | 16 |
|---|---|---|
| AT(A1,BALL,PM) | THROW(A1,BALL,PM) | AT(A1,BALL,PM) |

Figure 3b. Remaining primitive descriptor units after snapshots are filtered by attention box.

time interval of the action predicates in that snap. As activity increases, the snapshot production increases, and therefore the waiting time between snapshots decreases. A "typical" baseball game has about 6000 snapshots in it. A short 1/2 inning may take less than 100 snapshots while a long one may take upwards of 400-500 snapshots.

Questions can legitimately be raised at this point concerning the primitive descriptor units. We shall raise and attempt to answer the most serious ones.

Question: Why choose this particular set of primitive actions? For example, why not include the BEER-MAN-HAWKING-BREW or CLOUDS-MOVING? Why choose those 4 dimensions of the world to include in a unit? Why not encode the height, weight, hair color, etc. of the various team members?

Answer: There are 3 arguments to support our choice of the primitive features listed in Figure 2. First, we know from our own experience that we often perceive new situations in an "appropriate" way. If we didn't then we would have great difficulty in understanding the new situation. We bring a wealth of experience and knowledge to filter out the activity that probably is not relevant to our particular goals. Our system is initially tuned to activities on the field and the proper numeric markers. We believe most people roughly approach the game with this focus. Animals in general have a predisposition to roughly attend to the relevant features of a situation. In an action-oriented game setting, we probably wouldn't initially perceive the beer man or the clouds moving as something relevant to the game. However, when attempts at comprehension of a situation fail, then we might try to incorporate the features passed over initially.

Second, even though we have limited the number of features, the combinations of the existing features is still quite great. The interesting question of whether the system can discover the relationships between the features still remains.

Third, lest we be accused of finessing the problem of features completely, we mention here that the attention algorithms that we shall present could learn to habituate to non-essential elements in the domain. Our system should be able to handle many types of noise.

Question: Why this particular level of description? Why not describe the actions in terms of arm movements, muscle fiber actions, etc?

Answer: This question is answered, in part, in the previous paragraphs. Furthermore, were we to perceive baseball in terms of limb movements and not in the "higher" level units such as walk, run, etc., we would have more difficulty in learning to understand the game. This is not to say that we wouldn't or couldn't build the structure starting at this level. Rather, it would probably take much longer and also confuse different levels at which the world is perceived. Along these lines, Kilmer [25] is working on a "play-directed" system that can learn to build low level action primitives up into higher level units, in the way a child or chimpanzee might.

One could make the argument that a physical object is described by characteristics of the molecules and atoms in it. A description of the world must start at some level; the interesting aspect is the relative level of the concepts formed to the primitives by which they are expressed.

Therefore, for the reasons given above, we do not feel that our exclusion of features or our choice of a particular level of description is crucial to the model of learning that we are developing.

The system does not initially understand that, for example, a

SWINGHIT(B1,BALL,HOMEPLATE) AIRMOVING(BALL,HOMEPLATE)CATCH(A1,BALL,LEFTFIELD)

embedded in a sequence of snapshots is an out--that is what the system is to learn. However, we do give the system common sense semantic information about the primitives; without this information, neither you nor I would be able to make any sense of the various actions. For example, Figure 4 illustrates a portion of the information in the semantic data base of primitives. Under the action THROW, we list the common sense level goals which are associated with it: (1) to simply execute the act of throwing, (2) to propel an object (probably inanimate) away from the thrower, (3) to propel an object towards a particular destination (a person or place). Also associated with each action is semantic information about the skill and energy needed to perform that action, e.g., to merely propel an object away from oneself requires a minimum of skill or energy; to propel an object at a high rate of speed towards a specific spot requires a

**THROW(X,Y,Z)**

where x is an object usually animate
y is an object usually inanimate
z is a location

**GOAL:** to propel an object y,
(away from, or towards)
(a person or place)

**PHYSICAL ENABLING CONDITIONS:**
x must possess y

**ENERGY/SKILL RELATIONSHIPS:**
1) generally, Energy = HIGH and skill = MEDIUM
2) required energy to be expended increases with speed of object

⋮

**SWINGHIT(X,Y,Z)**

where x is usually animate
y is usually inanimate
z is a location

**GOAL:** to propel an object via another object.

**PHYS.ENAB.COND.:**
y must be moving, or y must be suspended, or . . .

**ENERGY/SKILL RELATIONSHIPS:**
1) generally, Energy = HIGH and Skill = HIGH
2) but, if speed of object increases, usually skill required to hit increases

⋮

**RUN(X,Z)**

⋮

**CATCH(X,Y,Z)**

⋮

**GENERAL RELATIONSHIPS:**

1) SPEED(human running) vs. SPEED(ball thrown) vs. SPEED(ball hit)

or equivalently:

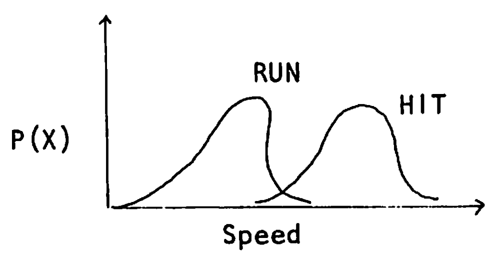


Figure 4. Common-sense facts about the primitives.

high degree of skill and energy; to catch an object moving in the air at a high rate of speed requires little energy but great skill. This knowledge* allows us to make reasonable conjectures about the goals of the players in everyday life and will be heavily utilized in our environment of continual physical activity.

To summarize, then, the semantic information initially given to the machine about the primitive action predicates is at a non-technical, common sense level and is necessary for the understanding of the basics of the game. It is the kind of knowledge that a 4-5 year old child would certainly have.

## III.2 Context: The Logical World

The appropriate context in which to interpret and understand the game of baseball is that of a competitive-conflict situation. Therefore, just as humans have a "theory" of this context, we must give our system a similar theory that captures the essentials of competitive conflict and gaming. We supply this theory in terms of "axioms," or more precisely, conventions. The system will attempt to interpret its observations in terms of this context. If we gave the machine a set of "axioms" describing a religious event, it would try to interpret the events on the field as rituals in that worship ceremony. Why not? This is simply an implementation of Assertion 4 (cf., Introduction); there is a superstructure or context in which events are interpreted.

---

*It is augmented by spatial information and knowledge about the relative physical relationships between the actions (e.g., distribution of the relative speed of objects).

The semantic information associated with the primitive descriptor units captures the common-sense descriptions of those actions and the physical laws which they obey. The contextual information captures the logical features of an action-oriented gaming situation; the kinds of goals and arbitrary relationships that might be important and desirable (e.g., scoring, winning). This latter knowledge, though higher level and game-independent, must interface with the lower level knowledge about the primitive actions to enable the system to interpret those actions. This interpretation permits the system to discover the conventions that structure the varied sequences of those actions.

Let us examine several of the logical conventions. Consider Assertion 2 in Figure 5. Certainly, this rule plays a central part in the understanding of competitive-conflict situations. If one didn't understand Assertion 2, one would have a hard time understanding any game! Next, consider assertions 7 and 11. One can roughly measure the competitive level or degree to which a goal is being pursued in terms of the energy and skill expended in that effort. Actions that most everyone can perform, such as walking, are not usually what is at stake in a game. Rather, what is important are the actions and goals that require skill and/or energy. Assertions 7 and 11 capture this powerful concept. For example, we shall use assertion 7 to help parse the continuous sequences of actions into: (1) segments that probably contain conflict and should be analyzed, and (2) ritual segments which are probably not essential and whose analysis can therefore be deferred; e.g., the man walking from the dugout to homeplate is a ritual activity and not part of the "game."

Assertion 10 allows the machine to possibly perceive an instance of one of the arbitrary rules of baseball, e.g., if a man is on first base and he wants to get to homeplate, he must first run to second base and then to third.

Figure 5. A Representative Set of the Logical Conventions Describing a Competitive-Conflict Situation.

1. The goal of an action is a) the execution of that action itself, b) the facilitation of other actions, c) the achievement of some state of the world, or d) some combination of the above.

2. "Opposing goals" means (1) team A is trying to achieve goal x and Team B is trying to achieve goal x, or (2) Team A is trying to prevent Team B from achieving goal x, or (3) both.

3. In action-oriented games, a competitive-conflict situation is one in which opposing teams have opposing goals.

4. The opposing goals are resolved in a competitive-conflict situation.

5. To resolve a competitive conflict situation is to succeed or fail at the intended goals.

6. To succeed is to achieve a goal or prevent an opposing goal. To fail is not to achieve a goal or not prevent an opposing goal.

7. A competitive conflict situation in an action-oriented world is often indicated by a sequence of high energy/skill actions, executed by opposing teams in a relatively short time interval.

8. A team is a group of people who act in concer (in sequences of possibly concurrent actions) to achieve a common goal.

9. In a sequence of actions performed by a team or an individual, it is often the case that each act facilitates the next.

10. If there is an obviously more efficient method of achieving a goal than that observed, then the observed method was probably done to satisfy some logical constraints imposed by the rules of the game.

11. If a high level of energy and/or skill is required of a physical action, then the actor must be trying to achieve a goal by that action. The desirability of a goal is often proportional to the level of energy/skill required to perform that action.

12. The Differential Analysis Procedure may be used in order to substantiate the hypothesized goals of an observed action
    a. vary the skill/energy required to perform that physical action by $\pm\Delta$
    b. note the effect of $\Delta$ change on the hypothesized goals of the action.
    c. Any changes induced by a positive $\Delta$ of skill/energy is often desirable with respect to the individual goals; changes due to negative $\Delta$ are often undesirable.
    d. If there is no change on the hypothesized goal induced by a $\pm\Delta$, then possibly this was not the intended goal.

13. The Locus of Activity (Closure) Analysis Procedure may be used in order to substantiate the existence of a relationship between some successive and/or concurrent actions.

    a. Divergence and/or convergence of activity in space often implies the goals of the activities may be related.

    b. Activity that begins and/or ends at the same point in time often implies the goals of the actions are related.

14. The Dependence of Convergent Activity Analysis Procedure may be used in order to discover/substantiate the precise dependency relationship between the individual actions.

    a. Apply differential analysis procedure to individual actions.

    b. If this implies that individuals on opposing teams have opposing goals, then more confidence is given to those hypothesized goals.

    c. Also, the differential analysis may provide sufficient information to permit the labelling of success or failure on the achievement of those individual goals.

in terms of simple energy expenditure, that is quite wasteful--better he should just run back to home plate! Therefore, expending all that extra energy suggests that this is required by the logical rules of the game. Another arbitrary causal relationship, which we shall analyze later (cf. the example), is embodied in the ground out versus the ground single. That is, these situations capture the concept that time relationships in games (and in most aspects of life) are important; getting the ball to first base before the runner is the deciding factor in these events. We are presently attempting to develop a better understanding of the general issues underlying these kinds of situations.

There are other still more abstract logical concepts involved in a competitive-conflict situation. For example, there is the notion that players have opportunities to succeed or fail at achieving major goals. Or, there is a scoreboard that keeps count of "interesting" actions. With what are these counts correlated? When the system has built up a set of plausible hypotheses about subgoals such as hitting the ball, or running and stopping at first base, it will try and integrate these abstract concepts with the intermediate level hypotheses that have been constructed. We could allow hypotheses about such concepts to be formed early in the observation. However, they probably would amount to a large number of weak conjectures which would have to be discarded. A conscious strategy of patience appears reasonable in these situations. This hierarchical learning process attempts to capture what studies [20] of the learning process in children have suggested: children seem to learn in stages, and they bring in "new principles" as they develop.

The particular list of conventions depicted in Figure 5 is only a representative set, and may not be complete. This is not the crucial issue. What is important is our use of conceptual semantic information as the framework in which a system can work. We will be able to experiment with various sets of conventions to answer questions like: Is there a fact that must be present? Can the system still function with a "wrong" convention? Will the system get swamped by a highly redundant set of conventions? Care must be taken not to restrict the set of conventions too much; otherwise, we may have in effect begged the question of learning.

## III.3. Implementation

How are these axioms, rules, conventions, represented in the computer? The constraint that our system performs common-sense human-like reasoning with symbolic (non-numeric) knowledge rules out the more traditional theorem proving and statistical approaches. A more appropriate representation is suggested by two current artificial intelligence/brain theory models [11,21]: axioms can be viewed as active procedures or computer programs. These "packets of knowledge" can "think"--they can look at the input, call in other packets, and generate chains of inferences and decisions.

The procedures have embedded in them more than just the content of the axioms. Associated with each procedure is meta-information that indicates in a rough way when a procedure is to be invoked, to what it can be applied, what parts of the axiom are essential and non-essential, what to do in case of trouble, where to go next, etc. The meta-information attempts to give the system an understanding of how to <u>use</u> its knowledge. This is a very important facet of intelligence and an essential feature of a mechanical learning system--as first noted by McCarthy [22] and thereafter often used to apologize for the lack of research into mechanized learning systems, e.g., [23]. Figure 6 gives a sampling of the meta-information that is on the property list of Rule 7.

| | |
|---|---|
| Apply when: | apply this axiom in initial phase of analysis. |
| Apply to what: | apply to the highest skill/energy action of an involvement chain. |
| If succeed, try next: | check rule 9 (Figure 5) for applicability<br>check rule 8 (Figure 5) for applicability |
| If fail, try next: | this may be a "defective" competitive-conflict situation, one in which a player failed to act. Check this out by applying rule 12. |

Figure 6. Meta-information for rule 7 (from Figure 5)

## IV. THE ATTENTION BOX

Just as in animals and humans, our system's attention box plays many important roles. First, in order to prevent the higher level processes of the system from being swamped by input "sensations," it must filter and structure the input data. Ethological and psychological studies suggest that animals and humans attend to change and habituate to constancy. Using this notion, the attention box is "programmed" to look at each snapshot and keep only those descriptors in the snapshot that are changing in time. More precisely, in order to decide whether to keep unit k in snapshot n, one must consider snapshots n-1 and n+1. If unit k does not appear in both adjoining snapshots, then keep unit k, otherwise filter it out. This process of filtering reduces the data to a manageable level; from 21 descriptor units per snapshot to approximately 3-4 descriptors per snapshot. See Figure 3b for an example of prefiltered raw data and the resultant filtered data. The system further structures this data in accordance with rule 7 mentioned previously. It segments the temporal sequence of activity on the basis of high energy actions.

But, this filtering and structuring is not enough. The system should pass on to the higher centers only the more important and interesting action segments. Therefore, the attention box notices subsequences of high energy actions that are <u>repetitive</u>. High activity actions that repeat must be important to the structure of the game. It seems reasonable to try and analyze those types of events first, before analyzing the truly novel events. This requires that some syntactic generalization be performed across one or more of the parameters of the descriptor unit. This allows different sequences to be "matched" for similarity. For example, to even see if unit-A and unit-B match, the system must immediately generalize across the time parameter; in addition, it will also be able to apply a person, place, or action generalization operator. Figure 7 gives examples of units generalized under various combinations of operations. A more detailed discussion of generalization techniques will be given in the section on hypothesis generalization. At this point, we need only notice that with these syntactic operations the attention box can now discover repetitive subsequences of activities. During system start-up, this ability will help the system to focus in and analyze those events that seem to be important.

Finally, there are feedback pathways from the higher centers to the attention box that allow it to be biased to "look for particular events." What the system attends to becomes, at least in part, a function of what it wants or expects to see; the higher centers can partially "reprogram" the attention box. Also, we can view the attention box as having a short-term memory (STM). Computational and memory considerations limit the number of specific events which the attention box can be instructed to look for. We are developing heuristics that can guide the decay and percolation of information in the STM.

## V. THEORY FORMATION: HYPOTHESIS FORMATION, GENERALIZATION, AND INTEGRATION

The combinatorics of this type of problem dictate that only reasonable alternatives be generated during any phase of the theory formation process which includes the following: hypothesis formation, hypothesis generalization, and hypothesis integration. Therefore, the system must have some guidelines for evaluating and directing its investigations. One traditional way to achieve this is to provide the learning system with a teacher who rewards the system when it makes a good choice (hypothesis). It is usually argued that the teacher must present a "good" training set, otherwise the

Original descriptor unit: THROW(A1, FROM-HP, BALL)

Generalization of descriptor unit

| | |
|---|---|
| Person Operator: | THROW(ANY-PERSON, FROM-HP, BALL) |
| Place Operator: | THROW(A1, FROM-ANYPLACE,BALL) |
| Person and Place Operators: | THROW(ANY-PERSON, FROM-ANYPLACE, BALL) |

NOTE: There is implicit generalization over time and the system will only generalize over action as a last resort.

Figure 7. Syntactic generalization operations

system will flounder. In effect, the semantic evaluation of those systems is embodied in the teacher--he performs the critical task of evaluating the system's hypotheses, and choosing a proper set of training events. In real life, however, it is often the case that animals and humans learn without teachers, hence the dictum "experience is the best teacher." Our goal is to explore the degree to which a computer can semantically generate and evaluate its own conjectures--i.e., organize its own "thoughts."

## V.1. Hypothesis Formation

Psychological and computational considerations dictate that the approach to forming good hypotheses is to initially generate plausible ones, as opposed to generating lots of them and filtering later. The conceptual semantic knowledge of the system provides the structure and direction for this to be done. Recall that this knowledge is implemented as procedures with appropriately annotated meta-information. These procedures look at the input data (the sequences of action in baseball) and try to discover the goals of the players performing those actions. The result of this interpretation process will be called goal trees. For those segments passed to it by the attention box, this subsystem will form alternative hypotheses about the goals of each player in the segment and output them as a goal tree. When a player is involved in a sequence of actions, a set of goals interact and limit the possible alternatives on the goal tree for that player. Since the players themselves are interacting with each other's goals, even more information is available to the "packets of knowledge" to further structure and limit the goal trees.

The knowledge procedures are active entities, but not all are applicable to all the data. Therefore, there are pre-conditions (expressed in the meta-information) which must be satisfied before a knowledge packet is permitted to add its particular information to the decision. The procedures communicate with each other, cooperating and competing, in their quest for good hypotheses. The output of this distributed knowledge network are the goal trees. An example of this process will be given shortly.

## V.2. Hypothesis Generalization and Integration

The tasks of this subsystem are threefold: (1) under some generalization(s) to collapse similar goal trees down into a generalized goal tree, thereby abstracting the essential features; (2) to verify the validity of the proposed generalized hypothesis; (3) to integrate the generalized hypothesis into the knowledge base.

Actually applying the generalization operators is relatively straightforward (cf. the attention box discussion). Generalizing over a parameter indicates a "don't care" or "match a whole class" in that dimension. However, before doing generalization, the system must first find some promising candidates (goal trees). Simply generalizing all the goal trees would cause another combinatoric explosion. The system does find likely prospects by using a novel twist of the hypothesize and test paradigm--the hypothesize and wait paradigm. It waits and screens the output from hypothesis formation. Then, using heuristics for similarity of syntactic sequences as well as hypothesized goals, it picks out some interesting goal trees to generalize. In other words, the system can decide what to work on; it organizes its own "thoughts," much as we would.

Once generalized, the validity of such a hypothesis must be tested.

Verificatior evidence is gained by waiting for more instances of the generalization and using the generalization to predict new events. Positive confirming instances can only lend confidence to the proposed generalization, whereas one negative instance can discredit it. The natural step then, is to bias the attention box to be on the lookout for such specific events. Should a negative instance be noticed, the system might then "debug" its proposed generalization, e.g., put constraints on an over-generalization, relax restrictions on an under-generalization, or throw the generalization completely away. Notice however, that deciding when in fact something counts as a negative instance is a very subtle process.

Finally, the system must integrate the generalized hypotheses into the knowledge base. Just because a generalized hypothesis nicely accounts for the data, it may still not "fit" into the set of existing theories. What precisely constitutes fitness--consistency, fuzzy consistency, simplicity, etc.,--is still an open question. However, once a generalized hypothesis has been integrated into the knowledge base, the system can draw on it in order to build up higher level units. For example, the system no longer must "think" in terms of a [batter hitting a ball-running-remaining on first base], but rather in terms of a "single."

## VI. OUTPUT

When all is said and done, what theory has the system formed? What has the system really learned? From the introduction, we recall that the system is forming a rich fabric of interwoven conventions and constraints that structure the alternative events that could occur in a game. This fabric has been arrived at by use of world knowledge and world observations. For example, the machine will eventually decide that "when a batter hits the ball he may run most probably towards 1st base, and in addition, he may also continue to run towards the other bases. Bases seem to be special places in baseball. Getting to first base is a desirable goal and seems to be a necessary precondition to scoring."

Let us examine one thin slice from that network, (Figure 8). The level of the primitive descriptor units is significantly lower than the level of the desired learned concepts. The system will hopefully come up with "primitives relative to a level," e.g., X1 in Figure 8. The "levels" are ill-defined, and certainly the set of relative primitives will not be minimal--but pretty close, if the system is not to generate a unique description for each sequence of events! These new primitives will facilitate a hierarchical construction of concepts. As the system evolves, it develops structures that enable it to understand more complex and disparate sequences of events. Information will be made accessible by hanging off each node in the network the inference chain that explains the basis of that concept.

Finally, when can we say that the system has learned to understand baseball? Is it fair to ask it to know about a balk, or a hit-and-run, or even the infield fly rule? We have trouble with these concepts--yet we can still play and understand the game! When the machine has learned about hits, runs, outs, and innings, then we can honestly say, "The computer understands the game of baseball."

## VII. EXAMPLE

Let us follow an example through the attention box, hypothesis forma-

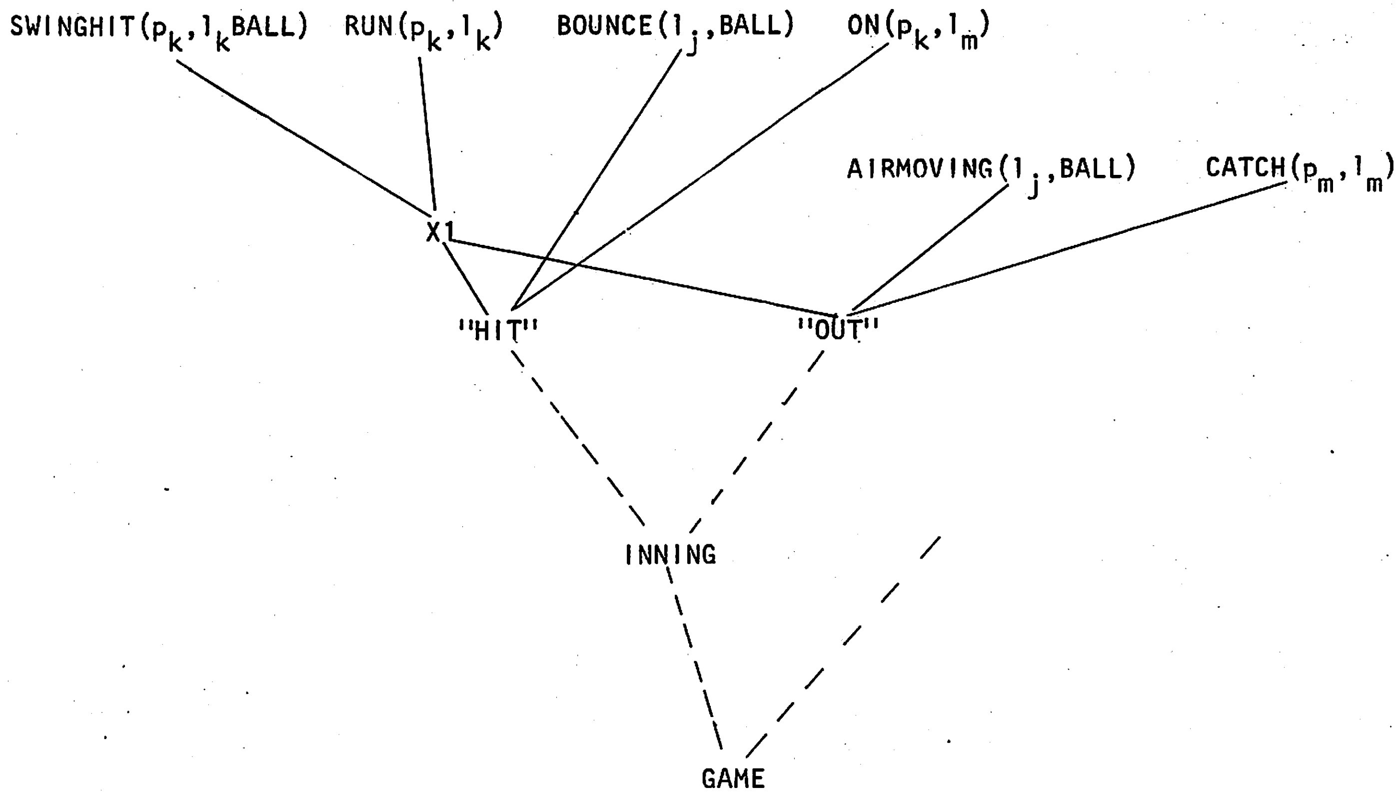


Figure 8. A Slice of the Hierarchically Structured Learned Conventions

tion, and hypothesis generalization and integration processes. Though it will be brief and highly stylized, it should convey a sense of the expected functioning of our system.

As we have said, the attention box can mark off high activity segments. By generalizing over the person and time parameters, it would discover that the following segment recurs often: the pitcher throwing a ball, a batter hitting the ball and running, etc.

Passing this particular segment on, the hypothesis formation module will attempt to build a goal tree for the pitcher and a separate one for the batter. Recall that this is done by applying the knowledge procedures to the data, if the preconditions on the procedures are satisfied. First, a MAKE-GOAL-TREE procedure for the pitcher is created, Figure 9. Now, rule 7 of Figure 5 finds itself applicable. Is this a conflict situation? With a high degree of probability, rule 7 reports yes, because both its preconditions appear to be satisfied: (1) THROWing a ball at a high rate of speed does require a high degree of energy; and (2) an opposition player, B1 the batter, is standing near the path of the oncoming object. What now? Since there is little more information, and since the high activity sequence continues, the system will prudently suspend processing and wait to see what happens.

Going on to Figure 10 where B1 hits the ball, we see Rule 7 active again. After Rule 7 decides that this is a competitive conflict situation (details omitted), Rule 2 returns a list of possible goals that B1 might be trying to achieve, and it also returns the physical enabling conditions associated with each goal. A physical enabling condition (PEC) is an event that must have taken place before the current event. Without going into details, the act of hitting an

ANALYZE: ((AT,THROW,AT) A1,PM)

CALL MAKE-GOAL-TREE(A1)

↓

RULE 5

IS THIS A CONFLICT SITUATION?

A) Is the degree of energy/ skill required high?

--- YES

B) Is time duration short?

--- YES

C) Are opposing teams involved or in the vicinity of involvement?

--- YES, B1

⋮

↓

SUSPEND

Figure 9. Goal tree analysis for the pitcher

Analyze: ((AT,SWINGHIT) B1,HP)

CALL MAKE-GOAL-TREE(B1)

RULE 7

Yes, this is a competitive-conflict situation

RULE 2

ACHIEVE WHICH GOALS?

$GOAL_1$

EXECUTE SWING AND HIT OBJECT

$GOAL_2$

PROPEL OBJECT TOWARD SOMEBODY

$GOAL_3$

PROPEL OBJECT TOWARD SOMEPLACE

PREVENT WHICH GOALS?

SUSPEND

WERE THERE ANY PHYSICAL ENB. CONDITIONS?

Yes, A1 threw the BALL that B1 hit

RULE 12

IF A1 THREW BALL WITH MORE ENERGY/ SKILL, THEN B1 WOULD REQUIRE MORE ENERGY/SKILL TO SWINGHIT

NOTE: PASS MESSAGE TO MAKE-GOAL(A1) "GOAL OF A1 IS TO PREVENT $GOAL_1$ OF B1-MAYBE"

ASSERT $GOAL_1$

FIGURE 10. Goal Tree Analysis for the Batter.

inanimate moving object (specified by $GOAL_1$) leads the system to examine the PEC of how the object was caused to move. <u>Who</u> caused it to move? Aha! A1 threw a ball which B1 hit. Meta-information in Rule causes it to pass a message to Rule 12, the energy/skill analysis axiom, because now we have a case where opposing teams are linked via a physical enabling condition.

Rule 12 is based on the belief that a person's goals are indicated, at least in part, by the amount of energy and/or skill expended in pursuit of the goal. In its analysis, Rule 12 first applies an energy/skill <u>differential</u> to the actions of A1 and B1, and notices the effect on performance. If A1 used <u>more</u> energy, what would happen? He would throw the ball faster. The system knows that the difficulty of hitting a moving object increases as the rate of speed of the object increases. Therefore, if A1 threw the ball faster, B1 would have much greater difficulty in hitting it. The system also knows that throwing objects very fast is a very high skill activity that not many people are capable of. From all of this, the system can infer that one goal of A1 was to <u>prevent</u> B1 from hitting the ball, but alas, A1 failed in that difficult goal. Finally, one more piece of supportive evidence. The system knows that hitting a moving ball requires a great deal of skill and that not everyone can do it. Therefore, since it saw B1 do it, it assumes that B1 did, in fact, <u>want</u> to execute that difficult action. Clearly, if B1 wanted to <u>not hit</u> the ball, it would have been very easy. Rule 12 now returns to Rule 2, with the above inferences. Rule 2 now has evidence to assert, as one hypothesis, that the $GOAL_1$ of B1 was to execute the act of hitting. There would be a similar analysis for the evaluation of the other goals. Note, in generating the goal for B1, we discovered a possible goal for A1: namely, to <u>prevent</u> B1 from hitting the ball. A message to this effect could then be passed to the MAKE-GOAL-TREE routine for A1.

Finally, let us suppose that the above scenario was an infield ground single. Also assume that the system has a goal tree for an infield ground out. The hypothesis generalizer would choose to work on these two scenarios because (1) they do recur several times, and (2) both their sequences of primitive descriptors (surface structure) and their goal trees (deep structure) are similar, although the goal trees differ in some significant aspects. It would then go on to discover the syntactic features that characterize the semantic differences. After integrating these generalized hypotheses into the knowledge base, the system might then be able to build up a more general concept of "out" or "single."

## VIII. CONCLUSION

The system described in this research will address a number of basic issues involved in the cognitive processes of learning and forming theories. The knowledge structure, the organization of logical processes, the ability to form hypotheses and conjectures, and the development of a system whose common-sense reasoning can be analyzed are important facets of our research. In particular, the following are some of the issues that this research will explore:

(a) Learning conceptual structures in real-world spatio-temporal domains; the mechanisms involved allow the structuring of complex information from a limited set of unordered examples.

(b) Learning exhibited by knowledge-based systems which do not rely upon statistical or numerical techniques; we will examine the organization of the semantic knowledge, the techniques for retrieving and applying this knowledge, and the inference processes for forming interesting hypotheses.

(c) Mechanisms for focussing attention; this will involve an examination of processes for focussing upon interesting input information on the basis of both 1) the innate "syntactic" cues; and 2) feedback from higher level cognitive structures which can bias attention.

(d) Hypothesis formation; the system will attempt to generate plausible hypotheses by allowing small "packets" of knowledge to infer the goals of people from observations of their actions; in general, semantics of the problem domain will structure and reduce the possible interpretations.

(e) Hypothesis generalization; the logical processes necessary to collapse down similar hypotheses into the more general case, thereby accounting for a larger class of inputs; syntactic and semantic measures of similarity are being developed.

(f) Verifying and integrating hypotheses into a theory; a set of generalized hypotheses must fit together into a consistent "theory" to explain a complex set of examples; by our "hypothesize-and-wait" paradigm, the system can evaluate its understanding of the environment by predicting conceptual structures; in our problem domain, this involves properly anticipating activity of the players.

## ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of and the many helpful discussions with W. Kilmer, M. Arbib, and C. Schmidt.

## REFERENCES


[1] A. Newell, "Remarks on the Relationship between Artificial Intelligence and Cognitive Psychology." In Banerji and Mesarwic, Eds. Theoretical Approaches to Non-Numerical Problem Solving. Springer-Verlag, Berlin, 1970.

[2] A. Newell and H. Simon, Human Problem Solving, Engelwood Cliffs, Prentice Hall, 1972.

[3] D. E. Rumelhart, R. H. Lindsay, and P. A. Norman, "A Process Model for Long Term Memory," In Tulving and Donaldson, Organization and Memory. Academic Press, New York, 1972.

[4] T. Winograd, Understanding Natural Language. Academic Press, New York, 1972.

[5] K. R. Popper, The Logic of Scientific Discovery, Basic Books, New York, 1959.

[6] C. G. Hempel, Philosophy of Natural Science, Prentice Hall, New York, 1966.

[7] B. Meltzer, "The Semantics of Induction and the Possibility of Complete Systems of Inductive Inference," Artificial Intelligence 1, 1970.

[8] C. G. Morgan, "Hypothesis Generation by Machine," Artificial Intelligence 2, 1971.

[9] G. D. Plotkin, "A Further Note on Inductive Generalization," Machine Intelligence 6, Meltzer, Michie, eds., American Elsevier, New York, 1971.

[10] A. Samuel, "Some Studies in Machine Learning Using the Game of CHECKERS," IBM J. Res. Dev., 3, 1959. Reprinted in Computers and Thought E. Feigenbaum and J. Feldman, eds., McGraw-Hill, New York, 1963.

[11] M. Minsky, "Frame-Systems: A Framework for Representation of Knowledge," M.I.T., Artificial Intelligence Lab, AI Memo 306, June, 1974, Cambridge, MA.

[12] R. E. Fikes, P. E. Hart, N. J. Nilsson, "Learning and Executing Generalized Robot Plans," Artificial Intelligence, 3, 1972.

[13] R. E. Fikes and N. J. Nilsson, "STRIPS: A new approach to the application of theorem proving to problem solving," Artificial Intelligence 2, 1971.

[14] P. M. Winston, "Learning Structural Descriptions from Examples," Tech Report AI TR-231, M.I.T. AI Lab., Cambridge, Mass., 1970.

[15] G. J. Sussman, "A Computational Model of Skill Acquisition," Tech. Note AI TR-297, M.I.T., AI Lab, Cambridge, MA 1973.

[16] B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg, "A Heuristic Programming Study of Theory Formation in Science," In Proc. of IJCAI2, London, 1971.

[17] B. G. Buchanan and N. S. Sridharan, "Analysis of Behavior of Chemical Molecules: Rule Formation on Non-Homogeneous Classes of Objects," in Advanced Papers IJCAI3, Stanford, 1973.

[18] B. G. Buchanan and J. Lederberg, "The Heuristic DENDRAL Program for Explaining Empirical Data," In Proc. of IFIP Congress 71, Ljubljana, Yugoslavia, 1971. Also Stanford AI Project Memo Ma 141.

[19] C. W. Churchman and B. G. Buchanan, "On the Design of Inductive Systems: Some Philosophical Problems," Brit. J. Phil. Sci., 20, 1969.

[20] H. G. Furth, Piaget and Knowledge, Prentice-Hall, 1969.

[21] M. A. Arbib, "Artificial Intelligence and Brain Theory: Unities and Diversities," to appear in Annals of Biomedical Engineering, Lyman, Yates, Eds., Academic Press, New York, 1975.

[22] J. McCarthy, "Programs with Common Sense," In Mechanization of Thought Processes, Vol. 1, Proc. Symp., Nat. Phys. Lab., London, 1958. Reprinted in Semantic Information Processing, M. Minsky, ed. MIT Press, Cambridge, Mass., 1968.

[23] N. J. Nilsson, "Artificial Intelligence," IFIP Congress 74, Stockholm, Sweden 1974. Also SRI AI Tech Note 89.

[24] J. Stelzer, E. Soloway, R. Seidel, "Authoring in an Intelligent CAI Environment," Forthcoming, 1975.

[25] W. Kilmer, "Play-Directed Behavior," Present research, Computer and Information Science, University of Massachusetts, Amherst, 1975.

[26] L. H. Uhr, "A Wholistic Cognitive System (SEER-2) for Integrated Perception, Action, and Thought." Third Milwaukee Symposium on Automatic Computation and Control, Milwaukee, Wisconsin, 1975.

[27] M. Kochen, "An Algorithm for Forming Hypotheses About Simple Functions," Third Milwaukee Syposium on Automatic Computation and Control, Milwaukee, Wisconsin, 1975.

[28] C. L. Hedrick, "A Computer Program to Learn Production Systems Using a Semantic Net," Pittsburgh: Carnegie-Mellon University, Ph.D. Thesis, 1975.

[29] R. J. Solomonoff, "Some Recent Work in Artificial Intelligence," Proceedings of IEEE, Vol. 54, No. 12, Dec. 1966.

[30] K. Melville, "Play's the Thing," The Sciences, N.Y. Academy of Sciences, N.Y., Jan./Feb., 1974.