KNOWLEDGE-DIRECTED LEARNING[†]

Elliot M. Soloway
Edward M. Riseman

COINS Technical Report 77-6

April 1977

This paper is a revised version of "A Common Sense Approach to Learning," COINS Technical Report 77-2; and it is to appear in the Workshop on Pattern Directed Inference Systems, Hawaii, 1977.

[†]This work was supported by ARI grant DAHC19-76-G-0013.

Abstract

A system embodying a knowledge-directed approach to unsupervised learning is examined in this paper. This approach is based on the premise that knowledge of new situations is acquired and interpreted in terms of the previous knowledge brought to the learning situation. In particular, our system is provided with a general characterization of action-oriented competitive games. This frame of reference is used to construct an interpretation for the patterns of human activity that are observed in games of baseball.

Multiple levels of knowledge and processing are used to proceed through various levels of description of the observed human behavior. Hypothesis Generation shifts the pattern description from observed physical actions such as "catch" and "run" to inferred goals and causal relationships of the players executing those actions. Hypothesis Generalization abstracts generalized classes of events and schemata that represent concepts such as "hit" and "out." Hypothesis Evaluation closes the loop in the learning process by verifying or rejecting the various hypotheses. Knowledge encoded as schemata direct these processes; there are schemata for inferring competitive and cooperative goals and causal relationships of players.

An important aspect of the system is its ability to use acquired knowledge. The multi-level organization facilitates the integration of the new information into the existing knowledge structure. Also, both the initial knowledge and the acquired knowledge are represented uniformly as schemata (production rules). Acquired schemata, then, are available to assist in interpreting and predicting future events. This ability demonstrates the effectiveness of our knowledge-directed approach to learning.

I. Introduction

In this paper we outline the major points of a computer system embodying a knowledge-directed approach to learning. The motivation for this approach comes from our daily experience; it seems that when faced with a learning situation (e.g., understanding sequences of apparently novel events) one does not rely solely on statistical learning techniques. Rather, one uses various levels of knowledge and processing to focus in a highly directed fashion on what is important in the observations. This direction is provided by the predispositions, or frames [1, 2], used to interpret those observations.

In particular, our system is provided with a general characterization of action-oriented competitive games. It uses that frame of reference in order to construct an interpretation for the patterns of human activity in the observed games of baseball. These behavior patterns are described in terms of four attributes: actor, action, location, and time. The goal of the system is to acquire a hierarchical network of schemata and concepts that represent an understanding of the observed activity at various levels of abstraction. The generalized schemata and concepts capture the relationships between the actions of the players and the goals intended by those actions. A key objective of our research is to allow the acquired schemata to aid in the further understanding of the observed patterns of behavior. This learning process requires both general knowledge of the goals and causal relationships in competitive action-oriented games as well as knowledge about particular physical actions.

We have chosen this knowledge-directed approach to learning for several reasons. First, knowledge is required in order to limit the combinatorics inherent in generating and generalizing concepts in such a complex task domain. Whereas many of the problems explored by other rule induction/concept formation systems have included relatively few features, there are thousands of features in the baseball games observed by our system. While data-directed (bottom-up) induction techniques may work on constrained problem domains, their unguided application in our domain would result in an overwhelming explosion of possible generalizations. Alternatively, our strategy is to use knowledge to form hypotheses about subsets of features that are relevant to the domain of interpretation. Though this process introduces other types of problems, it does succeed in significantly reducing the number of features over which generalization must occur. Moreover, the features that are most interesting relative to some domain of interpretation often are not "in" the data. Thus, knowledge must be used to hypothesize and add those important features. For example, the goals and causal relationships of the players in a competitive game are not observed explicitly--rather they must be inferred using a priori knowledge. Finally, since our objective is to have a dynamic system that can use what it learns, this requires that the system must know where to put the acquired information in the existing knowledge structure so that it can be effectively retrieved and utilized.

Our work on knowledge-directed learning is more akin to work on other knowledge-based learning (cf. [8, 14, 15, 26, 27]) and understanding (cf. [3, 4]) systems than to the more formal systems for rule induction/concept formation (cf. [9, 10, 11]). While we use generalization techniques similar to Hayes-Roth [9] and Vere [11], we do so only after we have used knowledge to restrict the number of features over which generalizations will take place. Employing a knowledge-directed approach, Lenat's [27] AM system discovers new mathematical concepts from a large knowledge base of interacting mathematical concepts. Winston [8] developed a system which used heuristic generalization techniques to learn structural descriptions in the blocks world. In both Waterman's [14] system to learn heuristics in power and Sussman's [15] HACKER system to debug programs about manipulating objects in a blocks world, knowledge that has been acquired is subsequently used. Schmidt's [17, 19] work on the inference of peoples' intentions, from observations of their actions, has influenced our approach to this problem in the baseball domain. The importance of causal relationships in understanding connected discourse and connected behavior has been discussed and analyzed by Schank [21] and Rieger [7]. Lesser and Erman's [4, 31] HEARSAY-II system uses a multi-level architecture in a speech understanding task, while we use that architectural approach in our learning system.

Figure 1 illustrates the multiple levels of knowledge, levels of processing, and levels of pattern description exhibited in our system.[1] The Attention Mechanism, described in more detail elsewhere [12, 13], "sees" the games of baseball in terms of symbolic primitives representing actions. The actual continuous activity is frozen into discrete moments and represented by <u>snapshots</u>. Each snapshot encodes all the actions of players occurring at that moment in time. The Attention Mechanism uses the biologically motivated heuristic that one should "attend to change" as an initial technique for filtering irrelevant detail and focusing on interesting information. Later, feedback from Hypothesis Evaluation can redirect the initial attention strategy. The output of this level of processing consists of <u>episodes</u>. These are sequences of activity characterized by a period of low activity (e.g., pitcher holding the ball), followed by high activity (e.g., pitcher throwing the ball, batter hitting, etc.), and concluding with low activity (e.g., the pitcher holding the ball again). An infield single, ground-out, or fly-out would all be examples of episodes.

The role of Hypothesis Generation is to accomplish a shift in description--from patterns of <u>acts</u> of individuals to patterns of <u>goals</u> of individuals executing those acts. Several levels of knowledge are used in this process. The output comprises episodes annotated with hypothesized goals and plans for both the actors and teams.

Hypothesis Generalization uses the annotated episodes to form generalizations of the hypotheses at various levels of abstraction. At one level this process uses both the actions and goals to generate classes of similar episodes, while at another level it focusses only upon the goals. Finally, Hypothesis Evaluation collects various forms of evidence that bear on the validity of the hypotheses. Based upon that evidence, hypotheses are accepted or rejected as truths. Once acquired knowledge is verified and accepted as new truth, it is fed back to Hypothesis Generation and used in the same ways as the a priori knowledge of the system. This acquired knowledge is also used to aid in the evaluation of the validity of unverified hypotheses. Hypotheses in the pool of alternatives that are inconsistent with new constraints are deleted.

The remainder of the paper will discuss the most intersting aspects of the latter three stages of processing. Further details appear in [16].

II. The Selection of Baseball as the Task Domain

An important consideration in the development of AI paradigms and techniques is the choice of task domain. Selection of an action-oriented gaming domain such as baseball might appear to be a frivolous choice. However, we feel that baseball has provided us with a rich set of behaviors in which to explore issues crucial to AI. It is a spatio-temporal world in which there is simultaneous and continuous activity. This activity is generated by human actors having purposes and plans and by inanimate objects obeying physical laws. Thus, the Frame Problem (cf. [22, 30]), modeling simultaneous and continuous activity (cf. [24]), and understanding causality (cf. [7, 21]) must all be considered. Moreover, whether human activity is perceived by reading stories or through actual visual observation, the same underlying processes must be performed--namely, the inference of goals, and plans of the actors (cf. [5, 6, 20]) based upon their actions. Thus, baseball encompasses many of the important issues in the mainstream of AI research.

In this domain we have had to develop the knowledge base for action-oriented competitive games ourselves. Though there has been research into games (cf. [29]), its orientation and detail have not provided the knowledge needed for inferences by our computer program. Thus, ferreting out the requisite general knowledge about action-oriented games (the logical and physical, competitive and cooperative relationships between the players and their goals) has proven to be a challenging enterprise. Nonetheless, the amount of knowledge needed by the system to acquire a basic understanding of the structure of the game--but not necessarily the subtle details--has been quite reasonable.

Regarding the complexity of baseball, concern has been voiced that it would be difficult for a human to acquire an understanding of that game from observations. On the contrary, we believe that many of the local goals of the observed actors often have only a few rather obvious alternative explanations. The complexity arises because it is difficult for the many local hypotheses to be integrated into a global structure by a human observer. Moreover, as the number of examples increases, so might the confusion. Nonetheless, we believe that, with patience, people can and would understand many of the important rudimentary aspects of this game. In any case, a complex domain is a challenging one for state-of-the-art AI research.

III. Hypothesis Generation

The objective of this level of processing is to generate hypotheses about the goals and causal relationships of the actors in the observed activity. In this section we describe the organization, representation and use of knowledge needed to perform this task.

III.1 Act-Schemata: Domain-Independent Knowledge

The first level of knowledge applied to the episodes output by the Attention Mechanism are the Act-Schemata. They contain information needed to understand actions independent of the particular contexts in which they take place. The major aspects of the spatio-temporal world that they capture are illustrated in the Act-Schema for the action THROW (Figure 2). Consider the entry PRIMARY-PHYSICAL-ENABLING-CONDITION of that Act Schema. When executed, this Act-Schema would find the immediately preceding action that created the state permitting the present action to occur. In our example of THROW, such an action would be to catch or to hold the object.

Outcomes of an action can be affected by varying the amount of skill or energy used in its performance. These effects are noted in the Act-Schema under the entry A-DELTA-INCREASE-IN-SKILL-ENERGY-CAN-AFFECT-PERFORMANCE. This information represents the common sense knowledge that it takes, for example, more skill or energy to throw a ball faster or farther (we presume that even a child would possess this knowledge at an early age). We define a difficult act as one that requires a relatively greater amount of energy and/or degree of skill. As we shall see in the next section, in hypothesizing a competitive goal for an actor, it will be important to ascertain whether or not the act he is executing is difficult.

III.2 Causal-Link Schemata: Domain-Dependent Knowledge that Directs Interpretation

The system is initially provided a general characterization of competitive and cooperative goals in action-oriented games. For example, competitive goals occur in situations where one player (team) wants to achieve an action while an opposing player (team) wants to prevent the execution of that action. The system can exploit this knowledge to identify examples of competition within the episodes analyzed by Act-Schemata. Similarly, a characterization of the causal relationships between the goals of actions is provided. For example, a "timing" relationship might be that one player often tries to execute an act before an opposing player executes some other act; this relationship further specifies that if the opposing player has executed his act earlier, the first player is not allowed to achieve his goal.

This characterization of goals and causal relationships is embodied in rules called Causal-Link Schemata (CLS). Figure 3 illustrates the hierarchical organization of CLSs. Implemented as production rules, the CLSs are triggered both by immediate and inferred aspects of the observations. Once activated, the CLSs hypothesize:

(a) a causal relationship between two actions;

(b) goals for each of the actors executing the actions;

and (c) success and failure labels for the goals of the actions.

As a result of applying these rules, the system shifts its description from observations of actions to hypotheses about the goals and causal relationships of the actors executing the actions.

To illustrate how the CLSs are used to infer goals, consider Figure 4. The current Act-Schemata for THROW and SWING-HIT (for "swing a stick, and then hit an object") are applied and their variables are instantiated as indicated. Now the set of CLSs are applied, and the PHYSICAL-COMPETITION CLS is found applicable--one player physically enabled (PHYSICAL-ENABLE ACT-X ACT-Y) an opposing player (OPPOSING-TEAMS ACTOR-X ACTOR-Y) to execute an act, where both acts were inferred to be difficult (DIFFICULT-ACT ACT-Y). An assumption we believe to be reasonable in the context of action-oriented competitive games has been made here: when one executes an action that requires a considerable degree of skill and energy, that act is probably intentional.

Next, we assume that in a competitive interaction, opposing players must have some opportunity to affect the outcome. In our example, this amounts to inferring whether or not the pitcher A1 could have done something (e.g., throw the object faster) that would have decreased the likelihood of the batter B2 being able to execute his act SWING-HIT. By accessing information in the Act-Schemata, the system (CAN-AFFECT-PERFORMANCE ACT-X ACT-Y) makes the inference that the outcome could have been different (e.g., THROW and NOT SWING-HIT). Thus, it decides that the pitcher had an opportunity to prevent the batter's act. We believe that the common-sense information used in these inferences is possessed by most adults, independent of them ever seeing a baseball game. An example of such knowledge is, "an object is harder to hit if it is moving fast."

From these inferences, the system can hypothesize that the pitcher and batter were in a competitive causal relationship (PHYSICAL-COMPETITION ACT-X ACT-Y) and that the goal of the hitter was to execute the act SWING-HIT while the goal of the pitcher was to prevent that event from happening. Combining this hypothesis together with the observed acts, the system can also hypothesize that the <u>pitcher</u> <u>failed</u> and the <u>batter</u> <u>succeeded</u> in achieving their respective goals.

The complete annotation for an infield single is depicted in Figure 5. In this diagram, there are two levels of description: acts and goals. Now, several other levels of description will be abstracted. We define a Plan to be composed of the sequence of subgoals of a player, with an assumption that the intermediate goals are attempted in order to achieve the last competitive-goal in that episode [17]. In Figure 5, for example, the various subgoals of the batter B3 form a plan to achieve the execution of the action ON FIRST-BASE. There is a Cooperative Summary of an Episode for each team. It is defined as the sequence of cooperative interactions of distinct players. We define Competitive Summary of an Episode to mean the set of competitive interactions in an episode; Figure 6 shows three competitive Causal-Link-Schemata which form the Competitive Summary for the infield single of Figure 5. By Final-Competitive Episode Goal we mean the last competitive goal in an episode, e.g., the ON-CATCH event in our example.

As can be seen from the PHYSICAL-COMPETITION SCHEMA of Figure 4, general CLSs test for properties of an interaction rather than for specific actions. However, the specific CLS which they hypothesize do test for specific actions. It is the particular game under observation, then, that determines which CLSs are triggered. Thus, the specific concepts learned are determined by the data presented to the system.

III.3 Acquired Knowledge: How and When It Can Be Used

The system can use its acquired information because it can integrate it into its existing knowledge framework. That structure defined how the new knowledge can be used and suggests when it can be used. The uniform representation of CLSs in our system facilitates the "how," while the multi-level organization facilitates the "when."

All CLSs have the same production rule structure--a pattern to trigger the schema, and an associated hypothesis to be generated when the schema is activated. Moreover, the hypotheses made by general CLSs have a description identical in form to that of a CLS; Figure 6 depicts three acquired specific CLSs. Learned information, then, is in the same format as the knowledge used in the learning process. Thus, the system acquires CLSs specific to baseball. These specific schemata can be used during Hypothesis Generation like the general schemata, without necessitating modification of the system. Also in Figure 6, we see how the acquired schemata are used to recognize an infield single directly. This is done without requiring the intervention of general CLSs.

Control of the acquired schemata is implicit in the level organization of the system. In particular, the system uses knowledge at the level of CLSs whenever it hypothesizes competitive and cooperative goals (and causal relationships) for the players. In particular then, the acquired CLSs can be used when goals of players need to be inferred. In other words, when to apply the CLSs is information passed from general CLSs to the specific CLSs which they spawned.

IV. Hypothesis Generalization

The role of Hypothesis Generalization is to construct classes of similar annotated episodes at various levels of abstraction. This reflects our commonsense intuition that events which appear different at one level of description often are actually quite similar at another level. For example, while the actions (surface structure) in a "walk" and a "single" are quite different, their final goals (deep structure) of getting ON FIRST-BASE are the same. The system does not know a priori what classes should exist in the data. In order to discover them, we again make use of the underlying semantics of the observed activity. We provide the system with a set of features meaningful in the domain of games which serve as the basis for the formation of specific classes. Thus, rather than matching all the features of several episodes to abstract generalizations--a combinatorial nightmare--only various subsets of the features are used in the matching process.

Presently, two levels of classes are produced: classes based on the Competitive Summary of an episode (the competitive goals plus the corresponding actions in specific causal relations) and classes based solely on Final Competitive Goals of episodes. Figure 6 illustrates generalized hypothesized CLSs which represent the class of infield singles based on Competitive Summaries. Figure 7 illustrates the range of classes which the system has discovered and verified; thus far, "hit" and "out" are the highest level concepts learned (see Section VI).

V. Hypothesis Evaluation: Dealing with Errors and Combinatorics

Hypothesis Evaluation closes the loop in the learning process by verifying or rejecting the hypotheses of goals, causal relationships and episode classes produced earlier. Currently, our system is running <u>without</u> the benefit of trainer-feedback regarding these hypotheses.[2] Consequently, this analysis is complicated by errors in the hypotheses output from Hypothesis Generation and the presence of plausible alternative interpretations (i.e., alternative competitive summaries). We have found that the impact of these problems can be alleviated by the assignment of confidence values.

To this end, Hypothesis Evaluation collects evidence on the validity of the hypotheses and modifies the confidences of those hypotheses accordingly. The observation of additional members of some class increases the confidence in the hypothesis of that class. Another factor in assessing confidence is the degree of consistency with other hypotheses of high confidence. However, the major source of evidence is through predictions derivable from the tentative hypotheses. Predictions are passed to the Attention Mechanism where they induce a "perceptual frame" for future matches in the input. The correct prediction of future events is an excellent test of the accuracy of the system's evolving interpretation of its world.

Three types of predictions can be made. The first type exploits the concept of competition. If players are competing with each other, one should expect to see team A succeed when team B fails, and also expect at other times to see team B succeed when team A fails. For example, in the competitive interaction between the pitcher and the batter (Figures 4 and 5), the hypothesis was that the pitcher failed because he did not prevent the batter from hitting the ball. This allows the prediction that eventually the pitcher will succeed with respect to his goal, and hence the batter will fail to hit the ball. The system forms predictions of complimentary outcomes--success predicts failure, failure predicts success--for all hypotheses of competitive interactions. These predictions are implemented as patterns which the Attention Mechanism uses to match against further observations. Whenever a prediction is matched, the confidence in the original hypothesis is increased. The other two types of predicted events are situations that are implausible vis-a-vis current hypotheses. In particular, hypothesized causal relationships which are valid should not be violated; the appearance of an inconsistent event suggests that the hypothesis is either wrong or incomplete. The third type of prediction detects inconsistencies of hypothesized goal classes. Such evidence is used to decrease the confidence of the related hypothesis.

Finally, Hypothesis Evaluation must decide when to elevate hypotheses to the plateau of accepted "truths." At this point our strategy is rather trivial. High confidences are formed for those hypotheses which lead to positive predictions, occur frequently, and are consistent with other hypotheses. Therefore, the hypotheses having the greatest confidence are considered true.

VI. System Implementation and Experimentation

The system components (Attention Mechanism, Hypothesis Generation, Generalization, and Evaluation) are implemented in LISP on a CDC 6600 [25]. A present, for reasons of simplicity and programming ease, all the episodes are first processed through Hypothesis Generation, then through Hypothesis Generalization, and finally through Hypothesis Evaluation and the Attention Mechanism. In total, the system requires approximately 75K of core.

Table 1 provides a sense of the volume of sensory data which is input to the system.[3] For example, on the average there are 43 episodes in an inning. Recall that an episode was a segment of relatively high activity, such as an infield single, or a flyout. However most of the activity in an inning is simply the ball and strike episodes that precede the singles or the outs.

Table 2 presents the results of various levels of analysis on the raw sensory data. The Attention Mechanism (Figure 1) significantly reduces the number of actions in an inning from about 14,000 to 1000 by filtering based on change in activity. Since subsequent pattern matching uses the four features (actor, action, location, time) of an action, we shift to a description of the data in terms of features; in 1000 actions there are 4000 features. Similarly, we can view the hypothesis of a goal and a causal relationship as the addition of 2 new features to the pattern description. On the average, since there are 11 competitive and cooperative hypotheses in an episode, this implies that 22 features are added to the description of an episode. These features reflect the interpretation of the observations via domain knowledge. While there are about 5500 features in a filtered and annotated (interpreted) inning, the generation of classes of generalized episodes and concepts is based on only the 1000 inferred features. Thus, the system uses only a small amount of data at the interpretation levels, although there is a large volume of data at the sensory level.

The classes depicted in Figure 7 were produced from approximately 172 episodes comprising 4 innings of a simplified version of baseball. The system has not learned the schemata for "strike-out" or "walk" since both require an ability to count and an understanding of the changes in the scoreboard. Currently, knowledge is being added that would allow the system to monitor the scoreboard and make hypotheses about the relationships between changes in the markers on the board and the goals and events on the field.

VII. Summary

We have outlined a system that embodies a knowledge-directed approach to unsupervised learning in a complex, real-world task domain. The objective of the system is to construct an interpretation for observed patterns of human activity. Multiple levels of knowledge and processing enable the system to describe patterns of behavior and goals at various levels of abstraction. The generalized classes of episodes and goals which are learned, such as "out" and "hit," are far removed from observations of physical acts such as "run" or "catch." All of the learned concepts are consistent with each other and with the general, a priori domain knowledge.

An important characteristic of any learning process is the ability to use the acquired knowledge in a flexible manner. The information that has been learned by our system is not just a set of isolated rules that represent correlated events. Rather, the new schemata capture the underlying relationships between the players' actions and their inferred goals. Once the schemata are verified, their integration into the a priori knowledge framework of the system is straightforward because the acquired knowledge and the initial general knowledge are represented in a uniform way. Thus, acquired schemata are available to aid in interpreting and predicting future episodes. This ability demonstrates the effectiveness of our knowledge-directed approach to learning.

VII. Bibliography

[1] Minsky, M. (1974) A framework for the representation of knowledge, TR-306, M.I.T. A.I. Lab 1974.

[2] Arbib, M.A. (1975) Two papers on schemas and frames. Univ. of Mass. COINS Technical Report 75C-9, Amherst.

[3] Hanson, A.R. and E.M. Riseman (1976) A progress report on VISIONS: Representation and control in the construction of visual models. COINS Technical Report 76-9, Amherst.

[4] Erman, L.D. and V.R. Lesser (1975) A multi-level organization for problem solving using many diverse, cooperating sources of knowledge, IJCAI-4, U.S.S.R.

[5] Schank, R.C. and R.P. Abelson (1975) Scripts, plans and knowledge. Advanced Papers IJCAI-4, U.S.S.R.

[6] Schmidt, C.F. and N.S. Sridharan (1976) The representation of plans. Dept. of Computer Science Technical Report CBM-TR61, Rutgers Univ., New Brunswick.

[7] Rieger, C. (1973) The common-sense algorithm as a basis for computer models of human memory, inference, belief and contextual language comprehension. Theoretical Issues in Natural Language Processing, June, Cambridge.

[8] Winston, P.H. (1970) Learning structural descriptions from examples. M.I.T. Project MAC Technical Report 76, Cambridge.

[9] Hayes-Roth, F. and J. McDermott (1976) Knowledge acquisition from structural descriptions. Computer Science Technical Report, Carnegie-Mellon University, Pittsburgh.

[10] Michalski, R.S. (1974) Learning by inductive inference, NATO Advanced Study Institute on Computer Oriented Learning Processes, France.

[11] Vere, S.A. (1977) Inductive learning of relational productions. To appear in Proceedings of the Conference on Pattern-Directed Inference, Hawaii.

[12] Soloway, E.M. and E.M. Riseman (1975) Common-sense theory formation: towards understanding baseball. Univ. of Mass COINS Technical Report 75C-5, Amherst.

[13] Soloway, E.M. and E.M. Riseman (1976) Mechanizing the common-sense inference of rules which direct behavior, Proceedings of the AISB Conference, Edinburgh, 1976.

[14] Waterman, D.A. (1970) Generalization techniques for automating the learning of heuristics. Artificial Intelligence 1, 121-170.

[15] Sussman, G.J. (1973) A computational model of skill acquisition. M.I.T. AI Lab Technical Report TR-297, Cambridge.

[16] Soloway, E.M. (1977) Knowledge-directed learning using multiple-levels of pattern description. Ph.D. Dissertation, Dept. of Computer Science, Univ. of Mass., in preparation.

[17] Schmidt, C.F. and J.L. Goodson (1976) Summarizing observed actions: a strutural hypothesis about the nature of summaries. Dept. of Computer Science Technical Report CBM-TR62, Rutgers Univ., New Brunswick.

[18] Davis, R. and J. King (1976) An overview of production systems. Technical Report, Dept. of Computer Science, Stanford University.

[19] Schmidt, C.F. (1976) Understanding human actions: recognizing the plans and motives of other persons, in Cognition and Social Behavior (eds. J. Carroll & J. Payne). Erlbaum Press, in press.

[20] Rumelhart, D.E. (1975) Notes on a schema for stories, in Representation and Understanding (eds. D. Bobrow & A. Collins). New York: Academic Press.

[21] Schank, R.C. (1974) Understanding paragraphs. Instituto per gli Studi Semantici e Cognitivi, Technical Report 5, Castognola, Switzerland.

[22] Sridharan, N.S. (1976) The architecture of BELIEVER, PART II: the frame problem. Dept. of Computer Science Technical Report CBM-TR47, Rutgers Univ., New Brunswick.

[23] Charniak, E. (1972) Toward a model of children's story comprehension. M.I.T. AI Lab Technical Report AI-TR266, Cambridge.

[24] Hendrix, G. (1973) Modeling simultaneous actions and continuous processes. Artificial Intelligence, 4, 145-180.

[25] Konolige, K. (1975) ALISP Users Manual. University Computing Center, University of Massachusetts, Amherst.

[26] Buchanan, B.G., Feigenbaum, E.A. and J. Lederberg (1971) A heuristic programming study of theory formation in science. International Joint Conference on Artificial Intelligence-2, British Computer Society, 40-50.

[27] Lenat, D.B. (1976) AM: An artificial intelligence approach to discovery in mathematics as heuristic search. Ph.D. Dissertation, Computer Science Dept., Stanford Univ., A.I. Lab Memo AIM-286.

[28] Soloway, E.M. and E.M. Riseman (1977) Knowledge-directed learning using multiple levels of pattern description. Univ. of Mass. COINS Technical Report 77-5, Amherst.

[29] Avedon, E. and B. Sutton-Smith (1971) The Study of Games. New York: Wiley and Sons Inc.

[30] McCarthy, J. and P. Hayes (1969) Some philosophical problems from the standpoint of artificial intelligence, in Machine Intelligence 4 (ed. D. Michie). Edinburgh: Edinburgh University Press.

[31] Lesser, V. and L.E. Erman (1977) A retrospective view of the HEARSAY-II Architecture. Technical Report, Dept. of Computer Science, Carnegie-Mellon University.

Footnotes

1 - For a more detailed description and analysis of the multi-level architectured used in our system see [28].

2 - In supervised learning a trainer informs the system of the accuracy of its decisions; this provides the system with a powerful focus of attention and feedback mechanism.

3 - The tables are read in a column per row (column/row) fashion. For example, there are 13 snapshots per episode, on the average.

Figure Captions

Figure 1 - Organization of system

Levels serve to structure the knowledge, processing and pattern description.

Figure 2 - Simplified outline of an Act-Schema

The arguments to the various functions in the schema are patterns, where the "?" indicates an unbound variable and the "$" indicates a pattern-matched variable previously bound by a "?".

Figure 3 - Hierarchical description of domain knowledge

This represents the organization implicit in the domain knowledge. The tip nodes of the tree are the actual independent schemas which have been implemented.

Figure 4 - Moving from observed actions to goals and causal relationships by hypothesizing specific Causal-Link-Schemata

The Act-Schemata add features to the description of the observations that capture an understanding of non-game activity; e.g., the physical-enabling-condition that A1 set up (the ball moving) enabled B2 to execute his act. Then the Causal-Link-Schemata use those features while adding their own competitive game interpretation. The result of triggering a general CLS is the hypothesis of a CLS specific to the observed actions; in this case $ACT-X is bound to the act-pattern (THROW A1 ...) and $ACT-Y is bound to the act-pattern (SWINGHIT B2 ...). The general CLS and the hypothesized CLS have the same production rule structure.

Figure 5 - An infield single episode annotated by hypothesis generation

The thin arrows indicate the work of the Act-Schemata, while the thick arrows indicate the work ot the Causal-Link Schemata. The Cooperative Causal-Link Schemata also make hypotheses about pairs of actions, e.g.,

(#14 THROW A5 SS BALL) - (#22 CATCH A3 FB BALL)

The three compete links relate the following pairs of actions:

(#2 THROW A1) - (#6 SWINGHIT B3)
(#6 SWINGHIT B3) - (#12 CATCH A5)
(#20 ON B3) - (#22 CATCH A3)

Figure 6 - Using generalized versions of acquired CLSs to recognize an infield single

The three inferred CLSs represent the generalized version of the Competitive Plan Summary for the infield single episode depicted in figure 5. For example, the left-most one (PHYSICAL-COMPETITION between ?ACTOR-Z ?ACTOR-Y) is derived from actions #2 and #6 in figure 5. Variables can replace constants in the person, location and time position; this permits the successful matching of similar examples. In the above figure the variable ACTOR-Z is bound to the player A1, ACTOR-Y to B2, ACTOR-X to A5 and ACTOR-W to A3. Also, HP is an abbreviation for homeplate, PM for pitcher's-mound, SS for shortstop and FB for first base.

Figure 7 - Acquired classes of concepts and schemata

Different subsets of features of the annotated description of the observations ([action actor location time goal casual-relation]) at level 6 are used as the basis for finding similarities. One subset may uncover a similarity between two observations while another will not. For example, based on features of the hypothesized Causal-Link Schemata in the competitive plan summary of level 6, infield groundouts and flyouts are not similar at level 7. Based upon only the final competitive goals in the description, infield groundouts and flyouts become similar at level 8.

Table 1 - Unanalyzed sensory data input to the system

There are 26 actions in a snapshot if all the actions and all the markers on the scoreboard are considered. On the average there are 43 episodes/inning. (Read these tables: column per row.)

Table 2 - Filtered and annotated data at higher levels of description

After filtering out non-changing activity at the level of the Attention Mechanism, the average number of actions/snapshot is reduced from 26 to 2. Four features comprise an action: actor, action, location, time. Hypothesis Generation adds new features to the description of the activity by interpreting that activity as an action-oriented game. Each hypothesis adds a goal feature and a causal relationship feature. Since on the average there are 11 such competitive and cooperative hypotheses per episode, 22 features are added per episode. The generation of classes of episodes and concepts is based on these inferred features.

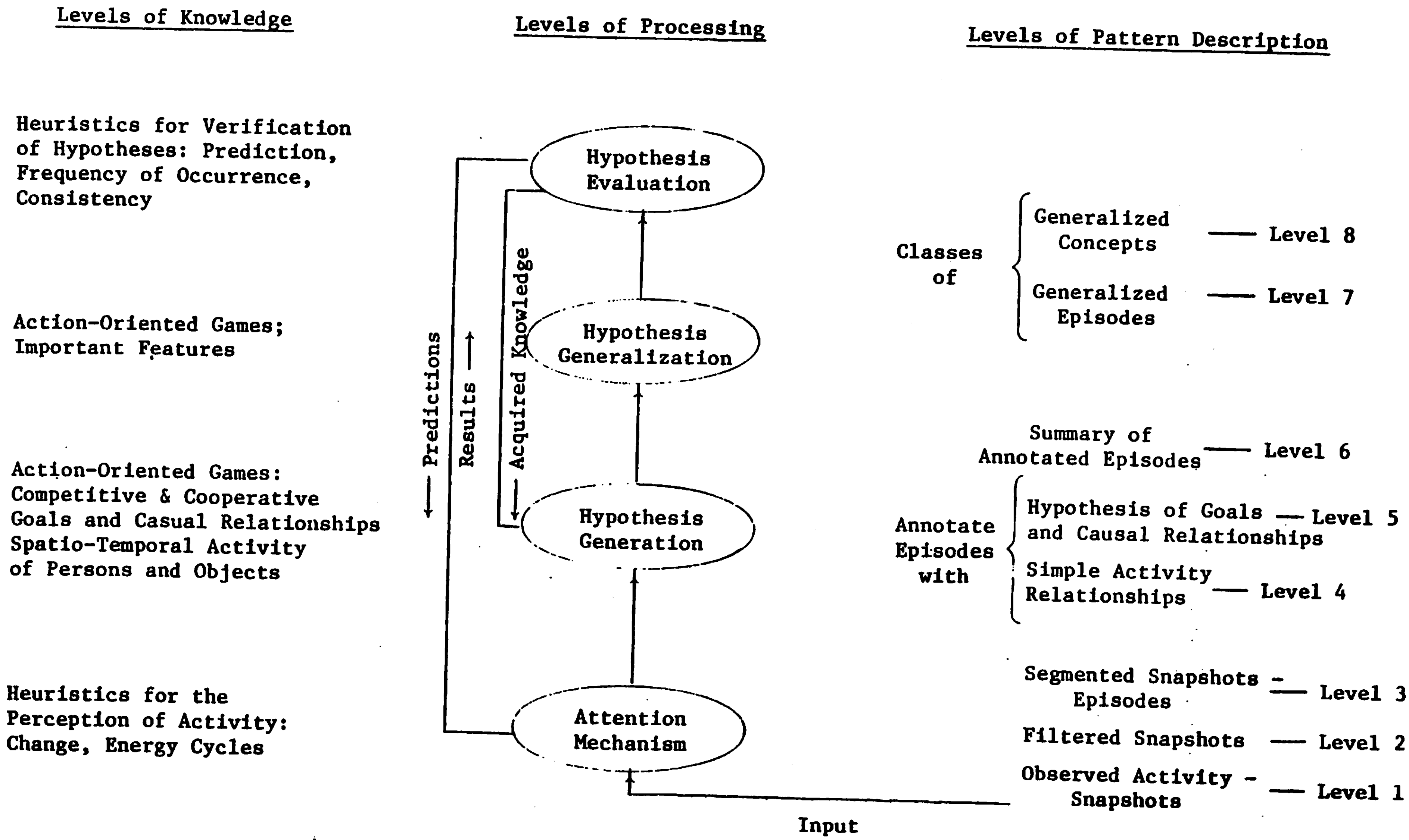
Levels of Knowledge
Levels of Processing
Levels of Pattern Description
Heuristics for Verification of Hypotheses: Prediction, Frequency of Occurrence, Consistency
Action-Oriented Games; Important Features
Action-Oriented Games: Competitive & Cooperative Goals and Casual Relationships Spatio-Temporal Activity of Persons and Objects
Heuristics for the Perception of Activity: Change, Energy Cycles
Hypothesis Evaluation
Hypothesis Generalization
Hypothesis Generation
Attention Mechanism
Predictions
Results
Acquired Knowledge
Input
Classes of
Generalized Concepts
Level 8
Generalized Episodes
Level 7
Summary of Annotated Episodes
Level 6
Annotate Episodes with
Hypothesis of Goals and Causal Relationships
Level 5
Simple Activity Relationships
Level 4
Segmented Snapshots - Episodes
Level 3
Filtered Snapshots
Level 2
Observed Activity - Snapshots
Level 1

Figure 1

```
THROW

    PRIMITIVE-ACT-TYPE:          PROPEL-FROM (?PERSON1 ?OBJECT ?START-LOCATION ?TIME)

    PRIMARY-PHYSICAL-
    ENABLING-CONDITION:          POSSESS     ($PERSON1 $OBJECT $START-LOCATION (BEFORE $TIME))

    PRIMARY-OUTCOME:             ARRIVE-AT   (?PERSON2 $OBJECT ?END-LOCATION  (AFTER  $TIME))

           A-DELTA-INCREASE-IN-SKILL-ENERGY-CAN-AFFECT-PERFORMANCE-BY:

                     (FASTER-MOVING   OBJECT  START-LOCATION)

                     (FARTHER-MOVING   OBJECT  START-LOCATION)
```

Figure 2

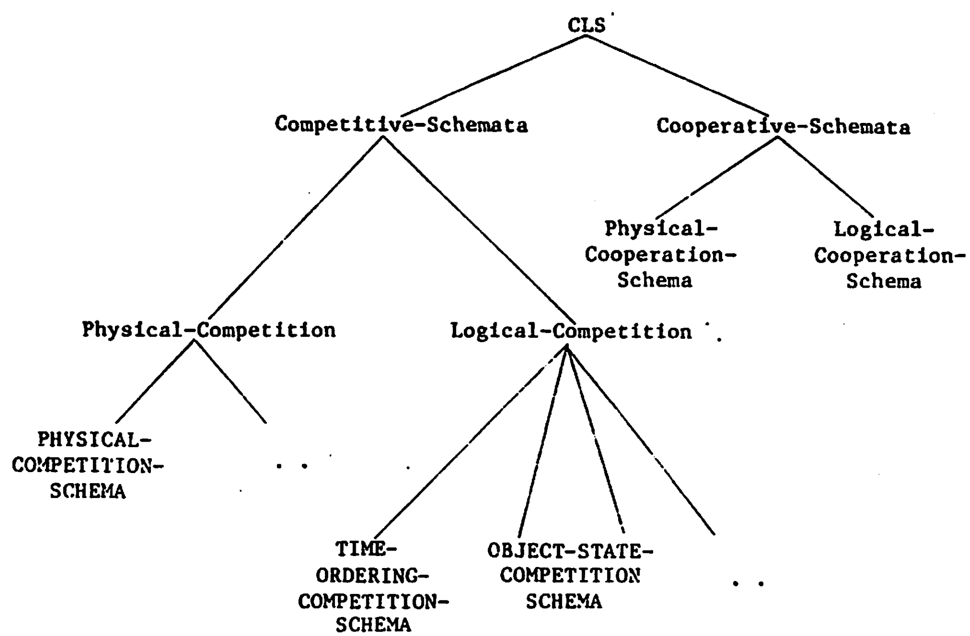

Figure 3

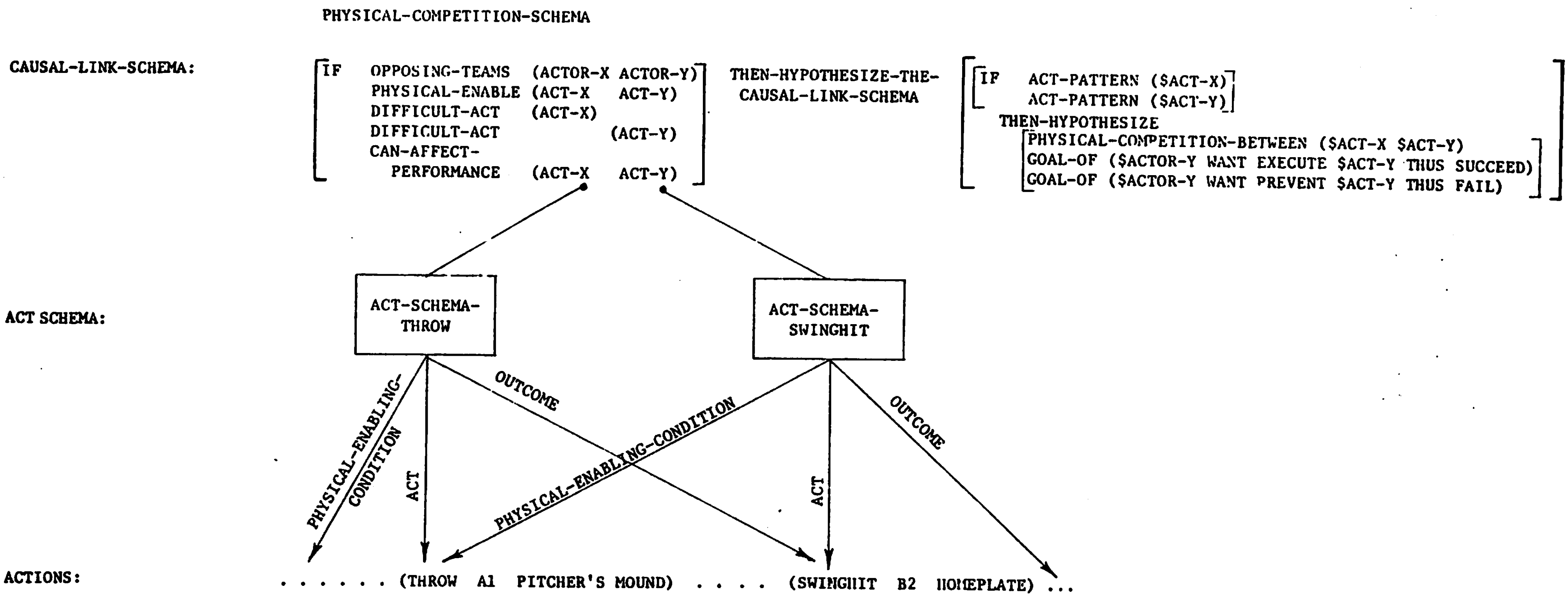
PHYSICAL-COMPETITION-SCHEMA
CAUSAL-LINK-SCHEMA:
IF OPPOSING-TEAMS (ACTOR-X ACTOR-Y)
PHYSICAL-ENABLE (ACT-X ACT-Y)
DIFFICULT-ACT (ACT-X)
DIFFICULT-ACT (ACT-Y)
CAN-AFFECT-PERFORMANCE (ACT-X ACT-Y)
THEN-HYPOTHESIZE-THE-CAUSAL-LINK-SCHEMA
IF ACT-PATTERN ($ACT-X)
ACT-PATTERN ($ACT-Y)
THEN-HYPOTHESIZE
PHYSICAL-COMPETITION-BETWEEN ($ACT-X $ACT-Y)
GOAL-OF ($ACTOR-Y WANT EXECUTE $ACT-Y THUS SUCCEED)
GOAL-OF ($ACTOR-Y WANT PREVENT $ACT-Y THUS FAIL)
ACT SCHEMA:
ACT-SCHEMA-THROW
ACT-SCHEMA-SWINGHIT
PHYSICAL-ENABLING-CONDITION
ACT
OUTCOME
PHYSICAL-ENABLING-CONDITION
ACT
OUTCOME
ACTIONS:
. . . . . . (THROW A1 PITCHER'S MOUND) . . . . (SWINGHIT B2 HOMEPLATE) ...

Figure 4

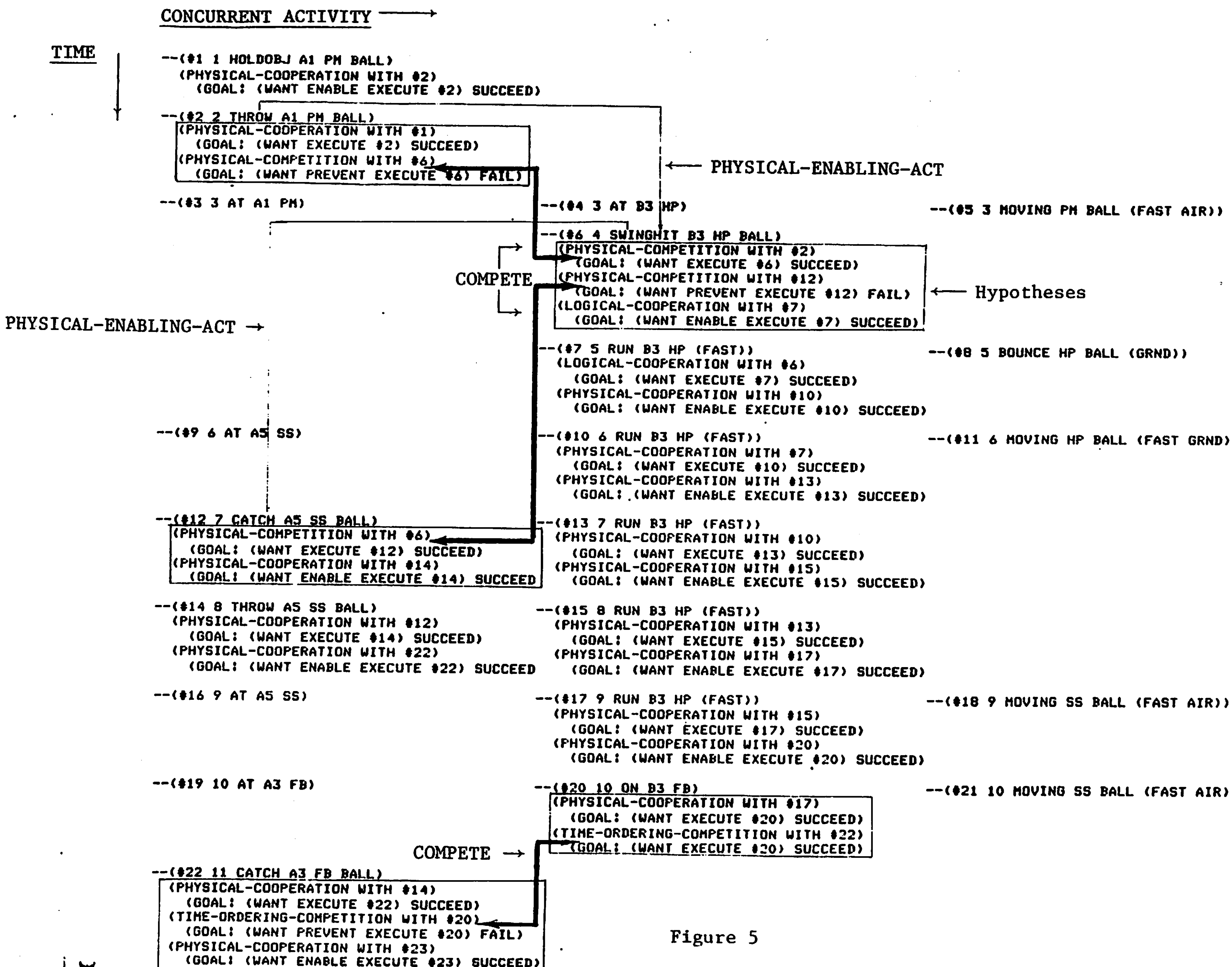
CONCURRENT ACTIVITY
TIME
--(#1 1 HOLDOBJ A1 PM BALL)
(PHYSICAL-COOPERATION WITH #2)
(GOAL: (WANT ENABLE EXECUTE #2) SUCCEED)
--(#2 2 THROW A1 PM BALL)
(PHYSICAL-COOPERATION WITH #1)
(GOAL: (WANT EXECUTE #2) SUCCEED)
(PHYSICAL-COMPETITION WITH #6)
(GOAL: (WANT PREVENT EXECUTE #6) FAIL)
PHYSICAL-ENABLING-ACT
--(#3 3 AT A1 PM)
--(#4 3 AT B3 HP)
--(#5 3 MOVING PM BALL (FAST AIR))
--(#6 4 SWINGHIT B3 HP BALL)
(PHYSICAL-COMPETITION WITH #2)
(GOAL: (WANT EXECUTE #6) SUCCEED)
(PHYSICAL-COMPETITION WITH #12)
(GOAL: (WANT PREVENT EXECUTE #12) FAIL)
(LOGICAL-COOPERATION WITH #7)
(GOAL: (WANT ENABLE EXECUTE #7) SUCCEED)
COMPETE
Hypotheses
PHYSICAL-ENABLING-ACT
--(#7 5 RUN B3 HP (FAST))
(LOGICAL-COOPERATION WITH #6)
(GOAL: (WANT EXECUTE #7) SUCCEED)
(PHYSICAL-COOPERATION WITH #10)
(GOAL: (WANT ENABLE EXECUTE #10) SUCCEED)
--(#8 5 BOUNCE HP BALL (GRND))
--(#9 6 AT A5 SS)
--(#10 6 RUN B3 HP (FAST))
(PHYSICAL-COOPERATION WITH #7)
(GOAL: (WANT EXECUTE #10) SUCCEED)
(PHYSICAL-COOPERATION WITH #13)
(GOAL: (WANT ENABLE EXECUTE #13) SUCCEED)
--(#11 6 MOVING HP BALL (FAST GRND))
--(#12 7 CATCH A5 SS BALL)
(PHYSICAL-COMPETITION WITH #6)
(GOAL: (WANT EXECUTE #12) SUCCEED)
(PHYSICAL-COOPERATION WITH #14)
(GOAL: (WANT ENABLE EXECUTE #14) SUCCEED
--(#13 7 RUN B3 HP (FAST))
(PHYSICAL-COOPERATION WITH #10)
(GOAL: (WANT EXECUTE #13) SUCCEED)
(PHYSICAL-COOPERATION WITH #15)
(GOAL: (WANT ENABLE EXECUTE #15) SUCCEED)
--(#14 8 THROW A5 SS BALL)
(PHYSICAL-COOPERATION WITH #12)
(GOAL: (WANT EXECUTE #14) SUCCEED)
(PHYSICAL-COOPERATION WITH #22)
(GOAL: (WANT ENABLE EXECUTE #22) SUCCEED
--(#15 8 RUN B3 HP (FAST))
(PHYSICAL-COOPERATION WITH #13)
(GOAL: (WANT EXECUTE #15) SUCCEED)
(PHYSICAL-COOPERATION WITH #17)
(GOAL: (WANT ENABLE EXECUTE #17) SUCCEED)
--(#16 9 AT A5 SS)
--(#17 9 RUN B3 HP (FAST))
(PHYSICAL-COOPERATION WITH #15)
(GOAL: (WANT EXECUTE #17) SUCCEED)
(PHYSICAL-COOPERATION WITH #20)
(GOAL: (WANT ENABLE EXECUTE #20) SUCCEED)
--(#18 9 MOVING SS BALL (FAST AIR))
--(#19 10 AT A3 FB)
--(#20 10 ON B3 FB)
(PHYSICAL-COOPERATION WITH #17)
(GOAL: (WANT EXECUTE #20) SUCCEED)
(TIME-ORDERING-COMPETITION WITH #22)
(GOAL: (WANT EXECUTE #20) SUCCEED)
--(#21 10 MOVING SS BALL (FAST AIR))
COMPETE
--(#22 11 CATCH A3 FB BALL)
(PHYSICAL-COOPERATION WITH #14)
(GOAL: (WANT EXECUTE #22) SUCCEED)
(TIME-ORDERING-COMPETITION WITH #20)
(GOAL: (WANT PREVENT EXECUTE #20) FAIL)
(PHYSICAL-COOPERATION WITH #23)
(GOAL: (WANT ENABLE EXECUTE #23) SUCCEED)

Figure 5

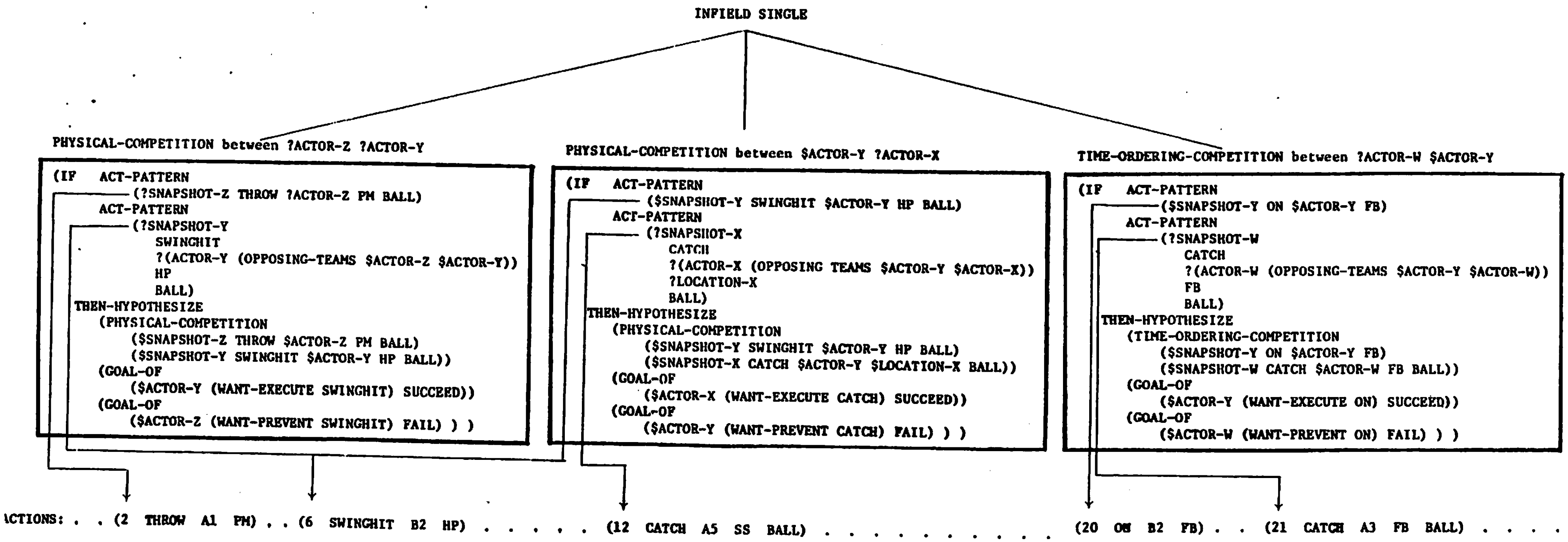
INFIELD SINGLE
PHYSICAL-COMPETITION between ?ACTOR-Z ?ACTOR-Y
(IF ACT-PATTERN
(?SNAPSHOT-Z THROW ?ACTOR-Z PM BALL)
ACT-PATTERN
(?SNAPSHOT-Y
SWINGHIT
?(ACTOR-Y (OPPOSING-TEAMS $ACTOR-Z $ACTOR-Y))
HP
BALL)
THEN-HYPOTHESIZE
(PHYSICAL-COMPETITION
($SNAPSHOT-Z THROW $ACTOR-Z PM BALL)
($SNAPSHOT-Y SWINGHIT $ACTOR-Y HP BALL))
(GOAL-OF
($ACTOR-Y (WANT-EXECUTE SWINGHIT) SUCCEED))
(GOAL-OF
($ACTOR-Z (WANT-PREVENT SWINGHIT) FAIL)))
PHYSICAL-COMPETITION between $ACTOR-Y ?ACTOR-X
(IF ACT-PATTERN
($SNAPSHOT-Y SWINGHIT $ACTOR-Y HP BALL)
ACT-PATTERN
(?SNAPSHOT-X
CATCH
?(ACTOR-X (OPPOSING TEAMS $ACTOR-Y $ACTOR-X))
?LOCATION-X
BALL)
THEN-HYPOTHESIZE
(PHYSICAL-COMPETITION
($SNAPSHOT-Y SWINGHIT $ACTOR-Y HP BALL)
($SNAPSHOT-X CATCH $ACTOR-Y $LOCATION-X BALL))
(GOAL-OF
($ACTOR-X (WANT-EXECUTE CATCH) SUCCEED))
(GOAL-OF
($ACTOR-Y (WANT-PREVENT CATCH) FAIL)))
TIME-ORDERING-COMPETITION between ?ACTOR-W $ACTOR-Y
(IF ACT-PATTERN
($SNAPSHOT-Y ON $ACTOR-Y FB)
ACT-PATTERN
(?SNAPSHOT-W
CATCH
?(ACTOR-W (OPPOSING-TEAMS $ACTOR-Y $ACTOR-W))
FB
BALL)
THEN-HYPOTHESIZE
(TIME-ORDERING-COMPETITION
($SNAPSHOT-Y ON $ACTOR-Y FB)
($SNAPSHOT-W CATCH $ACTOR-W FB BALL))
(GOAL-OF
($ACTOR-Y (WANT-EXECUTE ON) SUCCEED))
(GOAL-OF
($ACTOR-W (WANT-PREVENT ON) FAIL)))
ACTIONS: . . (2 THROW A1 PM) . . (6 SWINGHIT B2 HP) . . . . . . (12 CATCH A5 SS BALL) . . . . (20 ON B2 FB) . . (21 CATCH A3 FB BALL) . . . .

Figure 6

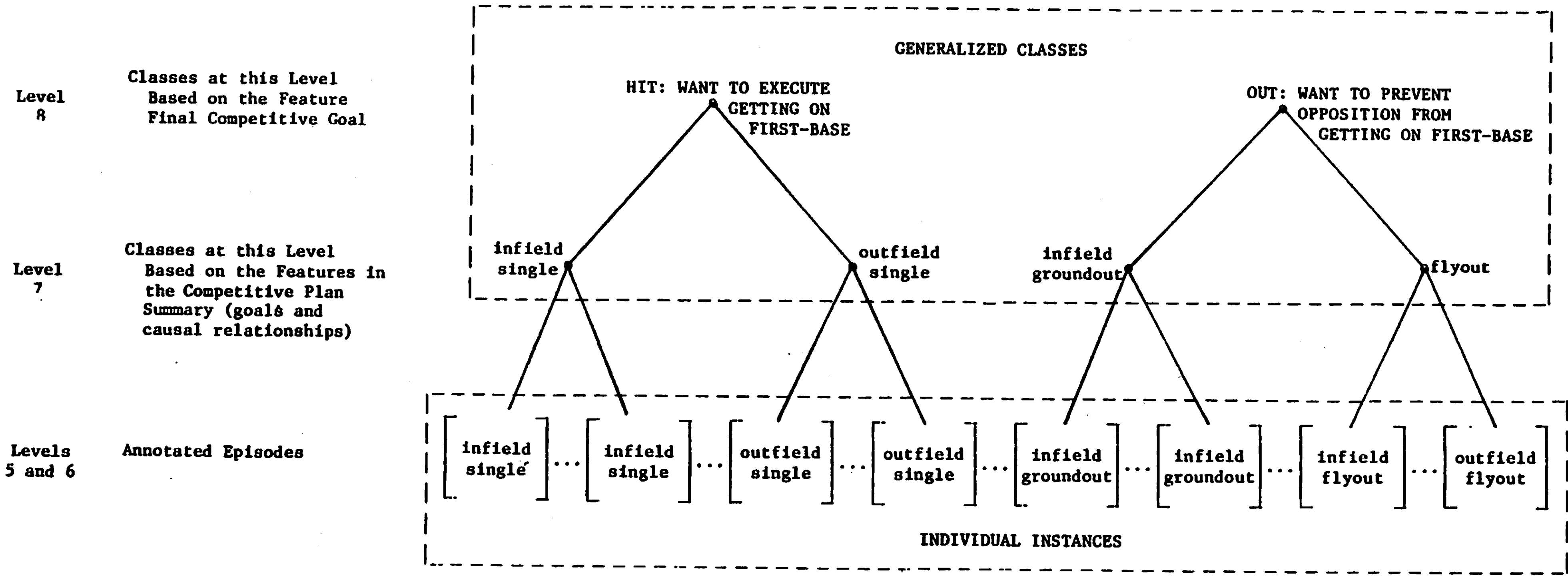
Level 8
Classes at this Level Based on the Feature Final Competitive Goal
Level 7
Classes at this Level Based on the Features in the Competitive Plan Summary (goals and causal relationships)
Levels 5 and 6
Annotated Episodes
GENERALIZED CLASSES
HIT: WANT TO EXECUTE GETTING ON FIRST-BASE
OUT: WANT TO PREVENT OPPOSITION FROM GETTING ON FIRST-BASE
infield single
outfield single
infield groundout
flyout
infield single
infield single
outfield single
outfield single
infield groundout
infield groundout
infield flyout
outfield flyout
INDIVIDUAL INSTANCES

Figure 7

	ACTIONS	SNAPSHOTS	EPISODES
SNAPSHOTS	26	—	—
EPISODES	338	13	—
INNING	14,534	559	43

Table 1

	ACTIONS		FEATURES		ADDITIONAL HYPOTHESIZED FEATURES
	UNFILTERED	FILTERED	UNFILTERED	FILTERED	
SNAPSHOTS	26	2	104	8	0
EPISODES	338	26	1,352	208	22
INNING	14,534	1,118	58,136	4,472	946

Table 2