GENERALIZING EXPLANATIONS
OF NARRATIVES INTO SCHEMATA

BY

RAYMOND JOSEPH MOONEY

B.S., University of Illinois at Urbana-Champaign, 1983

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1985

Urbana, Illinois

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

MAY 1985

WE HEREBY RECOMMEND THAT THE THESIS BY

RAYMOND JOSEPH MOONEY

ENTITLED GENERALIZING EXPLANATIONS OF NARRATIVES INTO SCHEMATA

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

Director of Thesis Research

Head of Department

Committee on Final Examination†

Chairperson

† Required for doctor's degree but not for master's.

O-517

GENERALIZING EXPLANATIONS OF NARRATIVES INTO SCHEMATA

Raymond Joseph Mooney, M.S.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1985
Thesis advisor: Gerald F. DeJong

This thesis describes a natural language system called GENESIS which improves its own performance through learning. The system processes short English narratives and is able to acquire, from a single narrative, a new schema for a stereotypical set of actions. During the understanding process, the system attempts to construct explanations for characters' actions in terms of the goals their actions were meant to achieve. When the system observes that a character in a narrative has achieved an interesting goal in a novel way, it generalizes the set of actions they used to achieve this goal into a new action schema. The generalization process is a knowledge-based analysis of the causal structure of the narrative which removes unnecessary details while maintaining the validity of the causal explanation. The resulting generalized combination of actions is then stored as a new schema in the system's knowledge base. This new schema can then be used by the system to correctly process narratives which were previously beyond its capabilities.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Professor Gerald DeJong, for initiating this line of research and providing a helpful, interesting, and friendly environment in which to work. Thanks are also due to Alberto Segre, Ashwin Ram, and Mark Edel as past members of the natural language learning project, to Paul O'Rorke for interesting discussions on generalization, and to other past and present members of the CSL AI group for providing an interesting and friendly place to work. Last and also least, I would like to thank HAL I and II, the Dolphin and Dandelion on which the implementation was developed.

This research was conducted at the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. Support was provided in part by the Air Force Office of Scientific Research under grant F49620-82-K-0009 and in part by the National Science Foundation under grant NSF-IST-83-17889.

TABLE OF CONTENTS

CHAPTER

CHAPTER 1

INTRODUCTION

Learning without thought is labor lost;
thought without learning is perilous.

Confucius

Artificial Intelligence, and Cognitive Science in general, has made extensive use of a form of knowledge representation known variously as schemata [Chafe75, Rumelhart80], frames [Bobrow, Charniak77, Minsky75], scripts [Cullingford78, DeJong82, Schank77], and MOPs [Dyer83, Schank82]. These structures can be broadly defined as packets of general knowledge about stereotypical objects, situations, events, or actions. In this paper, the term *schema* (a term traceable to [Kant]) will be used to refer to such packets of knowledge about a particular concept.

In particular, AI systems for understanding natural language narratives[1] [Charniak77, Cullingford78, DeJong82, Dyer83, Norvig83, Wilensky78] have relied heavily on schemata, especially schemata for representing stereotypical sequences of actions. However, all of these systems use schemata which were built by the programmer and added to the system by hand. There has been relatively little discussion of how such schemata might be automatically acquired by a system during its normal course of operation. Nevertheless, hand coding all the schemata a natural language processor would need to process a wide range of text would be an extremely arduous if not impossible task. This thesis presents an approach to learning schemata for natural language processing. This approach is embodied in a system called GENESIS (for GENeralizing Explanations of Stories Into Schemata), which acquires new schemata in the normal course of processing narratives.

The research work reported in this thesis was initially motivated by problems encountered in extending the abilities of the FRUMP system [DeJong FRUMP.]. FRUMP was a system which produced summaries of newspaper articles received directly from the UPI news wire. Its approach to understand-

[1] The use of the term "story" in natural language processing has been controversial [Brewer82]. Therefore, in this paper we

ing was to first determine which of its schemata (if any) a particular story matched and then to skim the article looking for events predicted by this schema. With a library of 63 *sketchy scripts* (relatively undetailed schemata for stereotypical actions) FRUMP was able to "understand" a fair percentage of newspaper articles. However, it was soon realized that a system capable of understanding unrestricted text would require on the order of thousands of schemata for typical actions; much too many to construct by hand. Consequently, a realistic natural language system must have a way of augmenting its knowledge of the world by learning new schemata during the normal course of its operation.

In the small amount of currently existing literature and work on learning schemata for natural language processing [Lebowitz80, Schank77, Schank82], it is assumed that such a learning process will require being exposed to numerous examples of a concept and constructing a generalized schema by extracting the common features across these examples. Such an inductive approach to learning has been the subject of much research in machine learning [Michalski83b, Mitchell78, Vere78, Winston70], and has been applied to various domains such as learning concepts in the blocks world [Winston70], and learning rules for diagnosing soybean diseases [Michalski80].

In the work being presented, we take a different approach to learning. This approach, which we call *explanation-based learning* [DeJong85], allows a system to learn from a single example. The approach involves acquiring or modifying a problem solving technique by observing and understanding a sample problem solution. The learning process is driven by an explanation, in terms of the system's model of the world, of why the sample solution solves the problem. The new knowledge acquired from the example can then be applied to conceptually similar future problems. This approach to learning has received a good deal of recent support. In addition to the work of DeJong and his students, explanation-based learning is being applied to solving integration problems [Mitchell83], designing VLSI circuits [Mitchell85], solving linear equations [Silver83] and playing games [Minton84]. Earlier systems which used explanation-based learning are STRIPS [Fikes72] and HACKER [Sussman73] both of which learn plans in the blocks world and a system by Soloway which learns about baseball [Soloway78].

have adopted the term "narrative" to refer to connected text which may lack a plot or other defining aspect of a "story."

The particular type of explanation-based learning which GENESIS uses is called *explanation-based schema acquisition* (ESA). ESA involves generalizing the successful problem solving behavior of an external agent into a new action schema. In addition to the GENESIS system for narrative understanding [DeJong81, Mooney85], there are ESA systems which learn theorem proving techniques [O'Rorke84], assembly tasks in robotics [Segre85], and techniques in physics problem solving [Shavlik85]. GENESIS is a completely implemented system which acquires new schemata in the normal course of processing narratives. After acquiring a new schema, the system is able to correctly process narratives which were previously beyond its capabilities. The remainder of the thesis is a description of the GENESIS system and the methodology underlying its abilities.

CHAPTER 2

OVERVIEW OF THE GENESIS SYSTEM

What exactly is GENESIS?

Dr. Carol Marcus
Star Trek II: The Wrath of Khan

This section presents the overall architecture of the GENESIS narrative processing system and demonstrates its learning ability by showing its input/output behavior on a set of example narratives.

2.1. General System Organization

The general organization of the GENESIS narrative processing system is shown in figure 2-1. First, English input is processed by a parser into a *conceptual representation* (CRep), a case-frame representation which uses some *conceptual dependency primitives* [Schank75] as well as predicates for complex

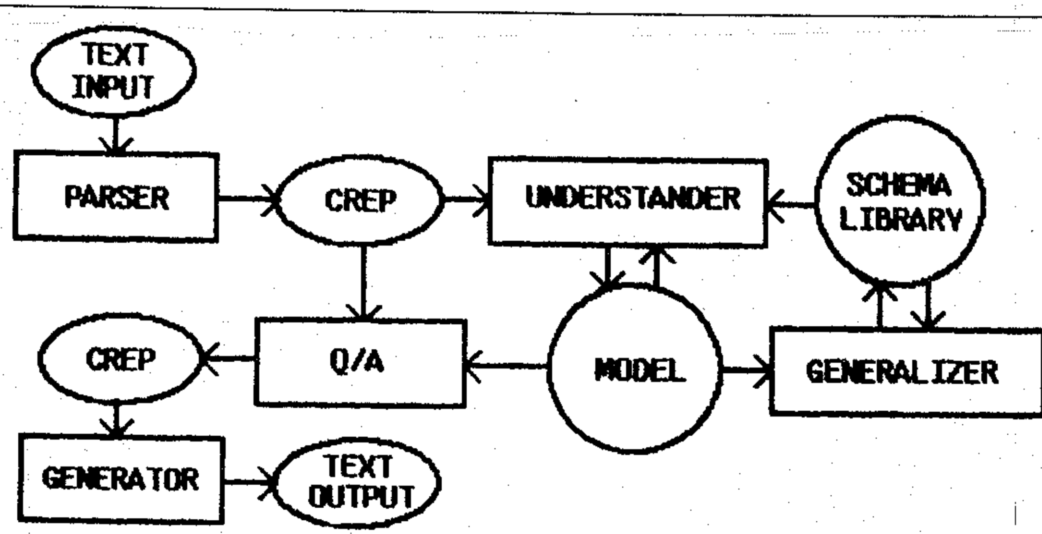


Figure 2-1: General System Organization

schemata.[1] A sample representation of an English sentence is shown in figure 2-2. Currently, we are using an adaptation of Dyer's McDYPAR [Dyer83] (a simplified version of the parser used in the BORIS narrative processor). However, since learning is the focus of our research, we make no claims about parsing and alternative approaches could be used for this task (e.g. [Marcus80, Waltz84]). Since BORIS uses *integrated parsing* [DeJong82, Dyer83] (i.e. the parsing process is directly aided by the system's higher level pragmatic and schematic knowledge), it should be noted that the parser in GENESIS is not well integrated with its schematic knowledge. Although we agree with the importance of integrated parsing, it is not the focus of our present research.

The basic task of the *understander* is to construct a causally complete *explanation* of the characters' actions in a narrative. It does this by building a *model* which contains explicit representations for all the inputs as well as the many inferences that must be made to causally connect them together.

English:
John told Fred if Fred gave him $250000 at Trenos then John would release Mary.

CRep:
```
[MTRANS (ACTOR (PERSON (NAME (JOHN)) (GENDER (MALE))))
        (TO (PERSON (NAME (FRED)) (GENDER (MALE))))
        (OBJECT (IMPLIES [ANTE (ATRANS (ACTOR (PERSON (NAME (FRED))
                                                      (GENDER (MALE))))
                                       (OBJECT (MONEY (TYPE (DOLLAR))
                                                      (AMOUNT 250000))))
                                       (TO (PERSON (NAME (JOHN))
                                                   (GENDER (MALE))))
                                       (AT (LOCATION (OF (RESTAURANT (NAME (TRENOS]
                             (CONSE (SET-FREE (ACTOR (PERSON (NAME (JOHN))
                                                             (GENDER (MALE))))
                                              (OBJECT (PERSON (NAME (MARY))
                                                              (GENDER (FEMALE]
```

Figure 2-2: A Sample Parse

---

[1] It is interesting to note that other recent systems which employ conceptual dependency (CD) [Dyer83, Lytinen84] do not insist that all inputs be parsed completely into CD primitives but allow sentences to be represented by direct reference to higher level schematic constructs such as TELEPHONE and ARREST.

There are four types of causal links connecting assertions in the model of a narrative. These are:

precondition: A link between a state and an action it enables.

effect: A link between an action and a resulting state.

motivation: A link between a volitional action and the beliefs and goals of the actor which motivated him to perform the action.

inference: A link between a state and another state which it implies.

To avoid confusion, such "causal" links between assertions in the model will be called *support* links, since a precondition of an action *supports* the performance of that action but does not *cause* it. The closely related term *data dependency link* [Doyle78], is not used since it is normally reserved for the support of inferences, not for the support of both inferences and actions. Inferring causal connections necessarily employs a large amount of background knowledge which is stored in the schema library (see figure 2-1). The techniques and representations used in the understanding process are similar to those used in past work in narrative understanding [Charniak77, Cullingford78, DeJong82, Dyer83, Wilensky83] and are discussed in chapter 3.

In order to demonstrate the capabilities of the understander, a simple question answering system is used to inspect the model. Since our interests lie in guiding the generalization process through the use of causal relationships, this subsystem is primarily used for accessing the reasons *why* an actor performed a certain action or *why* a particular state exists. This information is easily retrieved by inspecting the support links between the various states and actions in the model. Of course, there are many issues involved in retrieving the most appropriate answers to questions (see [Lehnert78]) which we do not fully confront in this subsystem.

A simple natural language generator for translating replies into English is also included as part of the system. This subsystem is an adaptation of McMUMBLE [Schank81] (a micro version of the generator used by the TAILSPIN narrative generating system). This generator verbalizes conceptual representations by using a particular English construction for each type of action and state known to the system.

If an actor in a narrative achieves an important goal through a novel combination of actions for which the system does not already have a schema, this set of actions undergoes *generalization* into a new schema. The constructed causal combination of actions in the model is generalized as far as possible without breaking any of the connecting support links or violating the well-formedness of individual actions and states. This generalized structure is then stored as a new schema in the library where it is used to facilitate the processing of future narratives. This generalization process is discussed at length in chapter 4 and is the key to the learning technique of explanation-based schema acquisition.

2.2. Example Performance of the GENESIS System

Currently, GENESIS has been tested on two examples. In one example, the system learns a schema for someone kidnapping an individual and holding them for ransom. In the other example, it learns a schema for someone burning his own building to collect the insurance. In both cases, before learning the schema, the system is given a "test" narrative. This narrative is not detailed enough for the system to construct a causally complete explanation for the characters' actions without a schema to fill in missing actions and inferences. Consequently, the system is unable to answer questions which require making certain default inferences about what must have taken place. Next, the system is given a different narrative on the same topic (e.g. kidnapping or arson) which contains actions and information which the previous one lacked. GENESIS is able to construct a causally complete explanation for this narrative and can therefore answer questions about why actors performed particular actions. Since the system recognizes that an important goal (e.g. getting money in both examples) has been achieved using a novel combination of actions, it generalizes these actions into a new schema. Then the system is given the test narrative for a second time. This time, it can use the schema it has just learned to fill in missing information and can therefore answer questions which it was previously unable to answer.

First we will show the performance of the system on the kidnapping example. Before processing the following narratives, GENESIS contains information in its schema library about bargaining, capturing and confining individuals, threatening, and many other concepts; however, it does not have a schema for kidnapping for ransom. First it receives the following narrative:

INPUT: ((Ted is the husband of Alice)
(He won $ 100000 in the lottery)
(Bob imprisoned Alice in his basement)
(Bob got $ 75000 and released Alice))

Processing Story...
Finished processing.

No thematic goals achieved by a novel combination of actions: no generalization.

Ready for questions:

? (Who gave Bob the money)
Answer unknown.

? (Why did Bob lock Alice in his basement)
Cannot find sufficient reason for his action.

? (Why did Bob release Alice)
Cannot find sufficient reason for his action.

? (Why does Ted have $ 100000)
Because Ted won $ 100000 in the lottery.

Notice that in this narrative it is not mentioned how Bob got the money or why Bob imprisoned Alice and then released her. Since the system does not have a schema for kidnapping, it cannot infer the missing information and causally connect the actions in the input. Next, it is given the following narrative.

INPUT: ((Fred is the father of Mary and is a millionaire)
(John approached Mary)
(She was wearing blue jeans)
(John pointed a gun at her and told her he wanted her to get into his car)
(He drove her to his hotel and locked her in his room)
(John called Fred and told him John was holding Mary captive)
(John told Fred if Fred gave him $ 250000 at Trenos then John would release Mary)
(Fred gave him the money and John released Mary))

Processing Story...
Finished processing.

John achieved the thematic goal: John has $ 250000.
Generalizing... Assembling new schema: CAPTURE-BARGAIN.

Ready for questions:

? (Why did John imprison Mary in his room)
So John and Fred could make a bargain in which John released Mary and Fred gave
John 250000 dollars at Trenos.

? (Why did John make the bargain with Fred)
Because John wanted to have 75000 dollars more than he wanted to hold Mary captive.

? (Why did Fred give John the money)
Because John and Fred made a bargain in which John released Mary and Fred gave John 250000 dollars at Trenos.

? (Why did Fred make the bargain with John)
Because Fred wanted Mary to be free more than he wanted to have 250000 dollars.

? (Why did Fred want Mary free)
Because Fred was Mary's father.

Unlike the first narrative, this one is detailed enough to allow GENESIS to causally connect the individual actions. The resulting causal structure is then generalized into a new schema for kidnapping for ransom (which GENESIS calls CAPTURE-BARGAIN). Next, the system is given the first narrative again, and using the schema it has just acquired, it is able to infer the missing information and causally connect the actions. Consequently, it is able to answer the questions which previously it could not answer.

INPUT: ((Ted is the husband of Alice)
(He won \$ 100000 in the lottery)
(Bob imprisoned Alice in his basement)
(Bob got \$ 75000 and released Alice))

Processing Story...
Finished processing.

No thematic goals achieved by a novel sequence of actions: no generalization.
Ready for questions:

? (Who gave Bob the money)
Ted gave Bob 75000 dollars.

? (Why did Bob lock Alice in his basement)
So Bob and Ted could make a bargain in which Bob released Alice and Ted gave Bob 75000 dollars.

? (Why did Bob release Alice)
Because Bob and Ted made a bargain in which Bob released Alice and Ted gave Bob 75000 dollars.

? (Why did Bob make the bargain with Ted)
Because Bob wanted to have 75000 dollars more than he wanted to hold Alice captive.

? (Why did Ted make the bargain with Bob)
Because Ted wanted Alice to be free more than he wanted to have 250000 dollars.

? (Why did Ted want Alice free)

Because Ted was Alice's husband.

The current implementation has also been tested on the following example about arson. Before processing the following narratives, the system has schemata for insuring objects, indemnifying clients for the loss of property, burning objects, and various other concepts; however, it does not have a schema for burning one's own property to collect the insurance money. First GENESIS processes the following test narrative.

```
INPUT: ((John owned a barn)
        (He burned it)
        (He got $ 40000))

Processing Story...
Finished processing.

No thematic goals achieved by a novel combination of actions: no generalization.

Ready for questions:

? (Who gave John the money)
Answer unknown.

? (Why did John burn his barn)
Cannot find sufficient reason for his action.
```

Since this narrative does not explain how John got the money or why he burned the barn, the system cannot causally connect the narrative without a schema to fill in missing information. Next, GENESIS is given the following more detailed narrative.

```
INPUT: ((Stan owned a warehouse)
        (He insured it against fire for $ 100000)
        (Stan burned the warehouse)
        (He called Prudential and told them it was burnt)
        (Prudential paid him $ 100000))

Processing Story...
Finished processing.

John achieved the thematic goal: John has $ 100000.
Generalizing... Assembling new schema: INSURE.OBJECT-BURN-INDEMNIFY

Ready for questions:

? (Why did Stan burn his warehouse)
```

So Stan could call Prudential and tell them that the warehouse was burnt.[1]

? (Why did Stan tell Prudential that the warehouse was burnt)
So Prudential would indemnify Stan 100000 dollars for the loss of the warehouse.

? (Why did Prudential give Stan $ 100000)
Because Stan was insured by Prudential for 100000 dollars if the warehouse was burnt and Prudential believed the warehouse was burnt.

? (Why did Prudential believe the warehouse was burnt)
Because Stan called Prudential and told them that the warehouse was burnt.

? (Why did Stan insure the warehouse)
So Prudential would indemnify Stan 100000 dollars for the loss of the warehouse.

Once again, unlike the test narrative, the second narrative is detailed enough to allow the system to causally connect the characters' actions. The resulting causal structure is then generalized into a new schema for arson-for-insurance. GENESIS is then given the test narrative again and this time it uses its newly acquired schema to fill in the missing information.

INPUT: ((John owned a barn)
(He burned it)
(He got $ 40000))

Processing Story...
Finished processing.

No thematic goals achieved by a novel combination of actions so no generalization.

Ready for questions:

? (Who gave John the money)
The insurance company indemnified John $ 40000 for the loss of the barn.

? (Why did John burn his barn)
So John could contact the insurance company and tell them that the barn was burnt.

? (Why did John tell the insurance company the barn was burnt)
So the insurance company would indemnify John 40000 dollars for the loss of the barn.

? (Why did the company pay John $ 40000)
Because John was insured by the insurance company for $ 40000 if the barn was burnt

[1] The answer to this question may seem a little peculiar. The system considers it an appropriate answer for the following reasons. It has a very naive model of communication in which believing a proposition is a precondition for telling it to someone else and telling someone something results in them believing it. In regards to the narrative at hand, when Stan burns his warehouse it results in him believing it is burnt which enables him to tell the insurance company about it so they will believe it is burnt and indemnify him. The question answerer assumes if an action produces a state which enables another action then the reason the actor performed the first action was to enable the second. Improved heuristics for answering questions could provide a better answer in this case by following several support links to a more appropriate answer.

and the insurance company believed the barn was burnt.

? (Why did the company believe that the barn was burnt)
Because John contacted the insurance company and told them that the barn was burnt.

CHAPTER 3

REPRESENTATION AND UNDERSTANDING IN GENESIS

I wish he would explain his explanation.

Lord Byron
Don Juan

This chapter discusses the representation of GENESIS' knowledge and how this knowledge is used by the system to construct a causally complete explanation of events in a narrative. An effort has been made to keep all of the system's world knowledge in a declarative form in the schema library instead of distributed throughout the system in special purpose procedures. Also, although a planner is not a part of the current system, an effort has been made to use a representation which would be useful for planning as well as understanding. Work on the FAUSTUS system has emphasized the importance of this approach [Norvig83, Wilensky83]. Using a uniform declarative representation makes extending the system's knowledge base a relatively easy task since it simply involves adding new schemata to the library. It is also a necessary requirement for a learning system since the system must be able to automatically construct new representations to add to the system, and this task is obviously much more difficult if knowledge is encoded in special purpose procedures.

3.1. Knowledge Representation

GENESIS' knowledge is represented in a library of schemata: packets of general information about stereotypical objects, situations, and actions. All schemata in the library are arranged in a hierarchical inheritance net under the three major classes of ACTION, STATE, and OBJECT (the highest level class is simply called SCHEMA). The present hierarchy is a tree; however, a tangled hierarchy (in which a class can belong to more than one parent class) will be incorporated in the next version of the system. An instance of a schema is denoted by a CRep with a set of *roles* filled by other schemata. All fillers must eventually bottom out in OBJECT schemata. The parsed representation shown in figure 2-2 is an example of a schema instance. The type of information associated with a schema depends on whether it is an ACTION, STATE, or OBJECT so each of these will be discussed in turn.

ACTION schemata represent dynamic events which change the state of the world. ACTIONs are immediately subdivided into VOLITIONAL-ACTIONs, which require volitional actors (e.g. BARGAIN), and NONVOLITIONAL-ACTIONs which represent physical events and processes (e.g. EARTHQUAKE). ACTIONS include what Schank and Abelson [Schank77] call *scripts* and *plans*. Schank and Abelson make the distinction that scripts are specific and detailed complex actions while plans are more abstract. In GENESIS, actions can be represented at various levels of abstraction in the hierarchy, and in this sense are more like the representation of plans in the NOAH planning system [Sacerdoti75], and Schank's more recent MOPs representation [Schank82]. The following pieces of information are attached to ACTION schemata. In addition to the information attached directly to a particular schema, each ACTION inherits the information attached to ACTIONs above it in the abstraction hierarchy.

Roles:	A set of case roles for this action.
Role Constraints:	Each role is marked with the type of schema which can legally fill it.
Defaults:	Default fillers can be specified for each role.
Preconditions:	States which must be true in order for the action to take place.
Motivations:	States (specifically belief and goal structures) which explain why an actor would perform this action (only for VOLITIONAL-ACTIONs).
Effects:	States which are true after the action is performed.
Terminations:	States which are no longer true after the action is performed. (These are similar to the delete-lists in STRIPs but states are marked as no longer holding instead of being deleted from the model.)
Expansion Schemata:	A set of lower-level states and actions which actually make up this action along with the support relationships between them (similar to the body of a script).
Suggested Schemata:	Larger composite actions which this action may be a part of.
Determining Conditions:[1]	A set of lower-level actions and states which if all present indicate the occurrence of this action.

Below is an example of some of the information attached to the action representing the transfer of possession.

[1] The term *determination* is used by the FAUSTUS system [Norvig83] to refer to the process of adding a suggested frame to the system's representation of a narrative.

ATRANS

```
ROLES:          (?ACTOR ?OBJECT ?TO)

CONSTRAINTS:    ((?ACTOR (CHARACTER)) (?TO (CHARACTER)) (?OBJECT (OBJECT)))

PRECONDITIONS: (POSSESS (SUBJECT ?ACTOR)(OBJECT ?OBJECT))

EFFECTS:        (POSSESS (SUBJECT ?TO)(OBJECT ?OBJECT))

TERMINATIONS:   (POSSESS (SUBJECT ?ACTOR)(OBJECT ?OBJECT))
```

STATEs, on the other hand, represent relatively static situations in the world, such as an individual being someone's father or being in possession of some object. Two STATEs which are particularly important are the psychological states BELIEF and GOAL which represent, respectively, a character believing or desiring the truth of a particular proposition. The following pieces of information are attached to STATE schemata. In addition to the information attached directly to a particular schema, each STATE inherits the information attached to STATEs above it in the abstraction hierarchy.

Roles:	A set of case roles for this state.
Role Constraints:	Each role is marked with the type of schema which can legally fill it.
Defaults:	Default fillers can be specified for each role.
Inferences:	Other states which are reasonable inferences to make from this state.
Achieving Actions:	Actions which can be used to accomplish this state.

Below is an example showing some of the information attached to a particular state.

HUSBAND

```
ROLES:        (?SUBJECT ?OBJECT)

CONSTRAINTS: ([?SUBJECT (PERSON (GENDER (MALE]
               (?OBJECT (PERSON (GENDER (FEMALE]

INFERENCES:   (WIFE (SUBJECT ?OBJECT)(OBJECT ?SUBJECT))
```

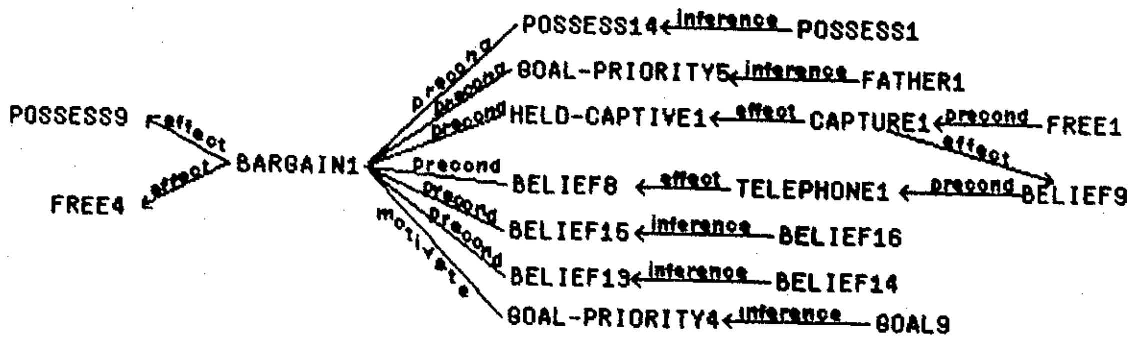
OBJECTs represent types of things in the world. These include PHYSICAL-OBJECTs such as people and cars as well as ABSTRACT-OBJECTs such as names and locations. The information attached to OBJECT schemata varies from class to class. Common examples for physical objects would be defaults for size, shape, and other physical attributes.

3.2. The Understanding Process

Since applying explanation-based schema acquisition depends on having a causal chain of actions to generalize, the "understanding" ability of GENESIS is concentrated on constructing this chain by inferring missing information and causally connecting inputs together. We do not attempt to deal with other important issues which have recently occupied researchers in narrative understanding such as *plot units* [Lehnert82], *thematic abstractions units* [Dyer83], *story points* [Wilensky82], affect [Dyer83], and *integrated parsing* [DeJong82, Dyer83].

To illustrate the result of the understanding process, the constructed causal explanation for the second kidnapping narrative is shown in figure 3-1. It should be noted that the support networks shown in this paper (called *highest-level support networks*) contain only the largest or highest level schemata which were determined to be in the narrative. Most of the representation at the level of the inputs and their connecting inferences is contained in the *expansions* of the CAPTURE and BARGAIN schemata which were activated bottom-up from the inputs.

In accomplishing the task of constructing causal connections, GENESIS, like FAUSTUS [Norvig83, Wilensky83], uses a combination of top-down and bottom-up processing techniques. If a set of inputs in a narrative matches a schema which the system already has, then it uses top-down processing to fill in the *expansion* of this schema with the particular inputs of this narrative, much like a script driven program such as SAM [Cullingford78] or FRUMP [DeJong82]. However, if an action in the narrative is not explained by a known schema, it attempts to connect it to other actions and states in the narrative by searching for existing states which fulfill the preconditions for this action, or by hypothesizing intermediate actions which causally connect it to existing states or actions. In this way, it also operates in a more bottom-up fashion like plan-based understanders such as PAM [Wilensky78].



POSSESS9	John has $250,000.
FREE4	Mary is free.
BARGAIN1	John makes a bargain with Fred in which John releases Mary and Fred gives $250,000 to John.
POSSESS14	Fred has $250,000.
POSSESS1	Fred has millions of dollars.
GOAL-PRIORITY5	Fred wants Mary free more than he wants to have $250,000.
FATHER1	Fred is Mary's father.
HELD-CAPTIVE1	John is holding Mary captive.
CAPTURE1	John captures Mary.
FREE1	Mary is free.
BELIEF8	Fred believes John is holding Mary captive.
TELEPHONE1	John calls Fred and tells him that he is holding Mary captive.
BELIEF9	John believes he is holding Mary captive.
BELIEF15	John believes Fred has $250,000.
BELIEF16	John believes Fred has millions of dollars.
BELIEF13	John believes Fred wants Mary to be free more than he wants to have $250,000.
BELIEF14	John believes Fred is Mary's father.
GOAL-PRIORITY4	John wants to have $250,000 more than he wants to hold Mary captive.
GOAL9	John wants to have $250,000.

Figure 3-1: Highest-level Support Network for Kidnap Narrative #2

3.2.1. Schema Activation and Determination

If a schema-based system is to be able to process a broad range of inputs, it must have access to a large number of schemata. Therefore, in order to avoid repeated searching through the entire database of schemata, it must also have an efficient method for selecting the particular schemata which are applicable to the current input. This process is frequently referred to as *frame selection* or *schema* *activation*. It is a difficult problem and has been the subject of several research projects [Charniak78, Charniak82, DeJong82, Lytinen84, Norvig83]. Below is description of GENESIS' approach to this problem.

When GENESIS processes an input, it adds this schema instance to the model and *activates* instances for all the schemata in the list of *suggested schemata* for the input instance. Activating a schema does not imply that the system immediately infers that this schema is occurring in the narrative, only that it could possibly explain the input. Active schemata then monitor subsequent inputs and check if they match parts of its *expansion* and can therefore be considered part of this active schema. When all the *determining conditions* of an active schema are met, it is *determined* or considered to have occurred in the narrative and is added to model along with the schemata and support relationships given in its expansion. If a determining condition is an action, then it is also considered to have occurred if all of its effects are in the model or if all of its effects can be inferred to exist since they are preconditions of other determining conditions which are already satisfied.

For example, an individual MTRANSing to someone that they will perform some action if that person performs some other action suggests the BARGAIN schema, which represents a successfully completed bargain between two individuals. Consequently, when GENESIS processes the following input from the detailed kidnapping narrative: "John told Fred that John would release Mary if Fred gave him $250,000," a BARGAIN schema representing a completed bargain between John and Fred is suggested. The determination conditions for a BARGAIN are the agreed upon actions of the two participants. In this case these are Fred giving John $250,000 and John releasing Mary. The next input the system processes is: "Fred gave John the money and John released Mary." This satisfies both of the determining conditions of the active BARGAIN schema, so it is determined and added to the model as the explanation of these three actions.

3.2.2. Bottom-up Construction of Support Relationships

When a new schema instance is added to the model (either as the result of an input or an inference on the part of the system), the system first tries to explain it as part of a known schema. However, if the new instance does not suggest any higher-level schemata nor match part of any already active schemata, then GENESIS tries to causally connect it to other actions and states in the model using planning information. Every time the understander establishes a new support link during this process, it annotates it with the information from the schema library used in constructing it. This information is important for maintaining the causal structure of the explanation during generalization. The exact form of the information stored with each support link and how it is used during generalization is discussed in detail in chapter 4.

The first step in integrating a new schema instance into the model is to add any *primary* inferences or effects. The effects and inferences attached to a schema are divided into *primary* and *secondary* categories. Primary ones are used in a forward inferencing fashion while secondary ones are used in a backwards inferencing fashion and only added to the model if they are required by the explanation. For example, a secondary inference of the state FATHER (inherited from more a more abstract state POSITIVE-INTERPERSONAL-THEME) is that a father values the freedom of his child more than the possession of money. One would not want to infer this GOAL-PRIORITY immediately, but this inference should be available if the explanation requires it.

If the new instance is an action, then its preconditions must be *reconciled* with the model. This means that it first searches the model for each precondition and if it finds it, it adds an appropriately labeled support link from it to the new action. If it does not find a precondition, it next attempts to infer it by looking for secondary effects or inferences which match this precondition. If this also fails, it hypothesizes the existence of an action which can be used to achieve this precondition (using the *achieving actions* attached to this state) and attempts to *reconcile* its preconditions with the model.

GENESIS also attempts to find *motivations* for volitional actions. It does this by checking if the action achieves a state which is a goal for the actor, or if it the actor has the specific goals and beliefs marked as possibly motivating this action. Certain types of goals can be automatically inferred if they will motivate a character's action. To determine these goals we use aspects of Schank and Abelson's theory of goals and *themes* [Schank77]. In their view, a goal arises either as a subgoal of a plan to achieve a higher level goal, or as a result of a theme. Schank and Abelson define themes as: "containing the background information upon which we base our prediction that an individual will have a certain goal." Goals which arise from themes are called *thematic goals* while subgoals are called *instrumental goals*.[1] Thematic goals are ones which arise from basic human wants and needs and therefore require no further explanation. They are further classified as either *satisfaction goals* such as satisfying hunger or sexual desires, *achievement goals* such as acquiring money or power, *enjoyment goals* such as wanting to be entertained, and *preservation goals* such as preserving one's health or safety. The system maintains a list of such thematic goals, and if a thematic goal will motivate a character's action it is inferred and a support link is added stating that it motivates the given action.

If the new instance to be incorporated in the model is a state, the system also checks to see if it satisfies a known or inferrable goal for some character. This is to catch any character's goals which are achieved by another character's action. For example, in the detailed arson narrative, Stan's presumed goal of possessing money is not achieved as an immediate effect of one of his own actions, but rather by Prudential's action of reimbursing him for the loss of his warehouse. Any achieved goals which are discovered during this process or during the process of motivating actions are specially marked by the understander. This information is later used by the generalizer to determine whether or not a narrative can be used to construct a new schema (see chapter 4).

To illustrate the process of constructing support relationships, we will examine how the BARGAIN action discussed in the previous section is integrated into the model constructed for the kidnapping narrative. When it is determined that John and Fred have completed a bargain in which John released Mary and Fred gave John $100,000, the action BARGAIN1 (see figure 3-1) is added to the model. Since this action does not suggest any higher level schemata, GENESIS tries to connect it to other information in the narrative using its knowledge about bargaining.

First, it adds the effects of BARGAIN1 to the model. These specify that John has $100,000 and that Mary has regained her freedom. Next it tries to reconcile the preconditions for the bargain. The precon-

[1] Although Schank and Abelson do not actually use the term thematic goal, it is a natural distinction given their analysis.

ditions of a BARGAIN include the preconditions of the two actions which make it up. These are that Fred has $100,000 and that John is holding Mary captive. The first of these is inferred as a secondary inference from the fact that Fred is a millionaire. The second has already been added to the model as an effect of the action CAPTURE1 which was determined from earlier actions in the narrative. See figure 3-1 for the causal structure resulting from this process.

Since John initiated the bargain, he is considered to be its main actor. The information attached to BARGAIN specifies that the motivations for the second actor can be considered preconditions from the main actor's point of view. Therefore, the condition that the second actor wants the effects of the main actor's action more than the state terminated by his own action is treated as a precondition of a BARGAIN. In this case, this precondition is that Fred value Mary's freedom more than $100000. This GOAL-PRIORITY is inferred as a secondary inference from the fact that he is Mary's father (see figure 3-1).

Finally, the system tries to motivate John's original initiation of the bargain. The motivation specified for a BARGAIN is that the main actor want the effects of the other person's action more than the state terminated by his own action. Under the current context, this means that John must want $100000 more than he wants to hold Mary captive. Since the system cannot infer that John has the goal of holding Mary captive, it is willing to infer this GOAL-PRIORITY if it can infer that John does have the goal of possessing $100,000. Since the system knows that wanting money is a thematic goal, it infers this goal and marks it as supporting the GOAL-PRIORITY necessary for motivating John's initiation of the bargain (see figure 3-1).

3.2.3. Avoiding Extensive Search During Understanding

GENESIS's understanding process is built on the assumption that combinatorially explosive search through possible explanations should be avoided. PAM [Wilensky78], on the other hand, is an explanation-based understanding system which searches for an explanation of an action until it exhausts its applicable knowledge. PAM searches for an explanation of a character's action by predicting future actions the character might take and seeing if any of them satisfy a known goal for that character. In terms of the representation presented here, when PAM processes an action, it searches through possible next actions which have preconditions which match the effects of the previous action. For example, consider the following narrative which is normally used as an example of how PAM processes a narrative.

Willa was hungry. She picked up the Michelin guide and got in her car.

When the system processes the action that Willa has picked up the Michelin guide, it considers that the next action she might perform is to read the guide since possessing a book is a precondition for reading it. The effect of reading the Michelin guide is that she will know the location of a restaurant. Since knowing the location of a restaurant is a precondition for going to a restaurant, it considers the next action she might take is to go to a restaurant. Finally, since being at a restaurant is a precondition for eating there, it considers eating at the restaurant to be a possible next action. This action is realized as satisfying her hunger and therefore the chain of actions just discussed is considered to be the explanation for her actions. If at some time the system cannot propose a possible next action due to lack of knowledge of an action which has a precondition which matches the previous action's effect, then it backtracks to the last choice and picks another possible action. This search is continued until the system encounters an action which satisfies a known goal or until it has exhausted its knowledge and cannot predict any more possible future actions.

However, searching through a space of possible future actions is combinatorially explosive. Consequently, such an approach is intractable if a system's knowledge of actions is large, which it obviously must be if it is to be able to understand a wide range of narratives. Consider what a PAM-like system would do if it where used to process the detailed kidnapping narrative. Upon determining that John had captured Mary, it would conduct an exhaustive search for an explanation of this action. Such a system would waste an incredible amount of time exploring possible courses of action which John could take now that he has Mary locked in his room. If the system had a large knowledge base of actions, it would be a long time before it stumbled upon the idea of using the action of releasing Mary as part of a bargain with another person.

Currently, in order to avoid such combinatorially explosive searches, GENESIS does not try to predict future courses of action at all. Suggesting schemata can be seen as predicting possible future actions, but this is a much more restricted process and only occurs when an action suggests a known schema. GENESIS does try to infer possible past actions which could have been used to achieve the preconditions of a known action, but even this process is limited since the *achieving actions* attached to each state are incomplete. GENESIS is probably a little *too* conservative with regards to searching for an explanation. Some searching is probably necessary for understanding narratives such as the PAM example shown above. But clearly, searching for an explanation should not be unconstrained since it quickly becomes intractable.

Since GENESIS does not conduct a complete search for an explanation, it is incapable of "understanding" narratives which have large gaps and do not suggest known schemata. When the first kidnapping narrative is processed without a schema for "kidnapping for ransom," very little of the missing structure can be inferred and the only support links the system can construct are shown in figure 3-2. As a result, it is unable to answer the questions shown earlier.

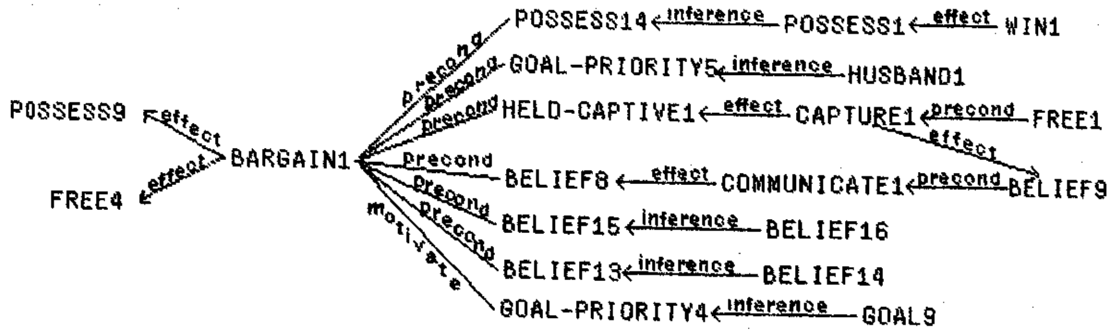
POSSESS3 $\xleftarrow{\text{effect}}$ ATRANS1 POSSESS1 $\xleftarrow{\text{effect}}$ WIN1

FREE2 $\xleftarrow{\text{effect}}$ SET-FREE1 $\xleftarrow{\text{precond}}$ HELD-CAPTIVE1 $\xleftarrow{\text{effect}}$ CAPTURE1 $\xleftarrow{\text{precond}}$ FREE1

POSSESS1	Ted has $100,000.
WIN1	Ted wins $100,000 in the lottery.
POSSESS3	Bob has $75,000.
ATRANS1	Someone gives Bob $75,000.
FREE2	Alice is free.
SET-FREE1	Bob releases Alice.
HELD-CAPTIVE1	Bob is holding Alice captive.
CAPTURE1	Bob captures Alice.
FREE1	Alice is free.

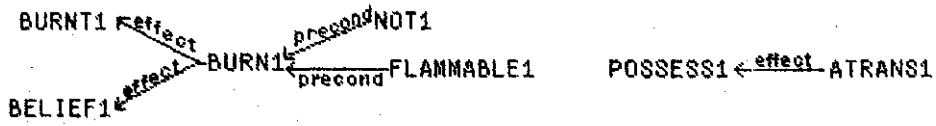
Figure 3-2: Support Network for Kidnap Narrative #1 Before Learning

However, with the same initial knowledge, it is able to understand the second narrative because the gaps and missing information are not too severe. Consequently, the understander is able to construct the support network shown in figure 3-1. As indicated earlier, this structure is then generalized into a new schema. When the first narrative is processed again, Bob's action of imprisoning Alice in his basement *determines* a CAPTURE schema, and this in turn *suggests* the new "kidnapping for ransom" schema. The new schema is then used in a top-down fashion to fill in missing information. It is finally *determined* when both of the effects of the BARGAIN: Alice becoming free again and Ted receiving money, are added to the model. The final support network (the *expansion* of the new schema for this narrative) is shown in figure 3-3. The corresponding causal structures for the arson narratives are shown in figures 3-4, 3-5, and 3-6.



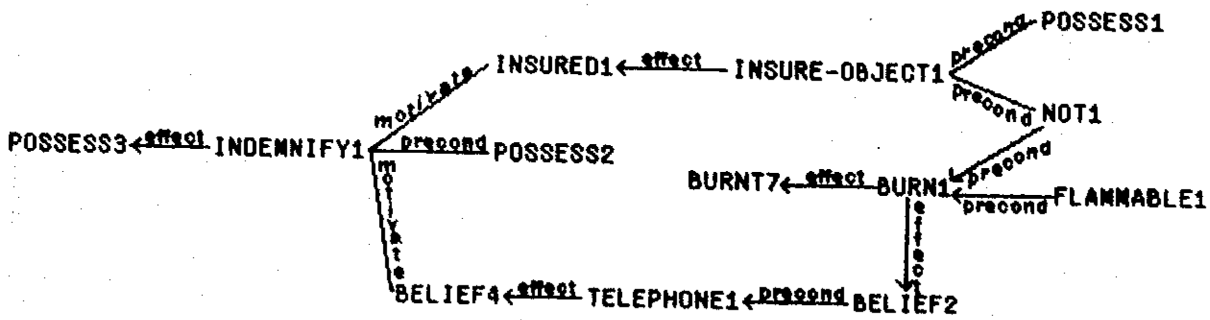
POSSESS9	Bob has $75,000.
FREE4	Alice is free.
BARGAIN1	Bob makes a bargain with Ted in which Bob releases Alice and Ted gives $75,000 to Bob.
POSSESS14	Ted has $75,000.
POSSESS1	Ted has $100,000.
WIN1	Ted wins $100,000 in the lottery.
GOAL-PRIORITY5	Ted wants Alice free more than he wants to have $75,000.
HUSBAND1	Ted is Alice's husband.
HELD-CAPTIVE1	Bob is holding Alice captive.
CAPTURE1	Bob captures Alice.
FREE1	Alice is free.
BELIEF8	Ted believes Bob is holding Alice captive.
COMMUNICATE1	Bob contacts Ted and tells him that he is holding Alice captive.
BELIEF9	Bob believes he is holding Alice captive.
BELIEF15	Bob believes Ted has $75,000.
BELIEF16	Bob believes Ted has $100,000.
BELIEF13	Bob believes Ted wants Alice to be free more than he wants to have $75,000.
BELIEF14	Bob believes Ted is Alice's husband.
GOAL-PRIORITY4	Bob wants to have $75,000 more than he wants to hold Alice captive.
GOAL9	Bob wants to have $75,000.

Figure 3-3: Support Network for Kidnap Narrative #1 After Learning



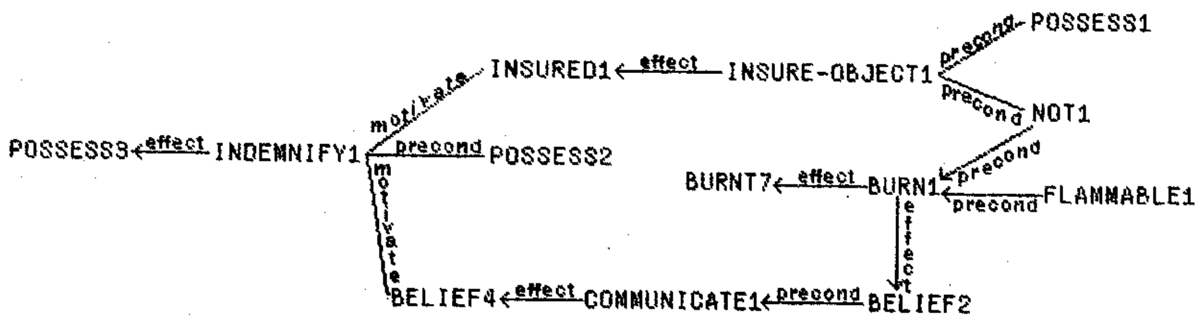
BURNT1	The barn is burnt.
BELIEF1	John believes the barn is burnt.
BURN1	John burned the barn.
NOT1	The barn is not already burnt.
FLAMMABLE1	The barn is able to burn.
POSSESS1	John has $40,000.
ATRANS1	Someone gave John $40,000.

Figure 3-4: Support Network for Arson Narrative #1 Before Learning



POSSESS3	Stan has $100,000.
INDEMNIFY1	Prudential reimburses Stan $100,000 for the loss of the warehouse.
INSURED1	Stan is insured by Prudential for $100,000 if the warehouse is burnt.
INSURE-OBJECT1	Stan insures the warehouse with Prudential for $100,000 in the case that the warehouse is burnt.
POSSESS1	Stan owns the warehouse
NOT1	The warehouse is not already burnt.
POSSESS2	Prudential has $100,000.
BELIEF4	Prudential believes that the warehouse is burnt.
TELEPHONE1	Stan calls Prudential and tells them that the warehouse is burnt.
BELIEF2	Stan believes that the warehouse is burnt.
BURN1	Stan burns the warehouse.
BURNT7	The warehouse is burnt.
FLAMMABLE1	The warehouse is able to burn.

Figure 3-5: Highest-Level Support Network for Arson Narrative #2



POSSESS3	John has $40,000.
INDEMNIFY1	The insurance company reimburses John $40,000 for the loss of the barn.
INSURED1	John is insured by the insurance company for $40,000 if the barn is burnt.
INSURE-OBJECT1	John insures the barn with the insurance company for $40,000 in the case that the barn is burnt.
POSSESS1	John owns the barn
NOT1	The barn is not already burnt.
POSSESS2	The insurance company has $40,000.
BELIEF4	The insurance company believes that the barn is burnt.
COMMUNICATE1	John contacts the insurance company and tells them that the barn is burnt.
BELIEF2	John believes that the barn is burnt.
BURN1	John burns the barn.
BURNT7	The barn is burnt.
FLAMMABLE1	The barn is able to burn.

Figure 3-6: Support Network for Arson Narrative #1 After Learning

CHAPTER 4

THE GENERALIZATION PROCESS

Learning is not attained by chance, it must be sought for with ardor and attended to with diligence.

Abigail Adams

Once a causally complete explanation for a narrative has been constructed, the learning process can begin. A top-level outline of the learning process is shown in figure 4-1. First, GENESIS must decide whether or not an interesting new action schema can be learned from the narrative. This decision process is described in the next section. If it decides that a new schema can be learned from this example, it generalizes the set of actions and states in the narrative which will compose the new schema. Section 4.2 is a description of this process. Finally, GENESIS must package the new schema into a form that can be stored in the schema library and used in the processing of future narratives. This final process is described in section 4.3.

(1) Decide whether or not to generalize a set of actions in the example into a new schema.

(2) Construct a *generalized support network* which represents the generalized causal structure of the set of actions composing the new schema.

(3) Package the schema into a form suitable for the schema library and store it away for use in processing future narratives.

Figure 4-1: Top-level outline of the learning procedure.

4.1. When to Generalize

If every novel combination of actions the system encountered was generalized into a new schema, the system would soon become overloaded with rarely used schemata. Most actions would *activate* a large number of schemata and selecting among these would require an excessive amount of processing time.

The problem of generating large numbers of useless schemata is mentioned in the conclusion of the STRIPS learning paper [Fikes72]. The solution proposed there was to include a mechanism for forgetting old plans. This mechanism would keep statistics on the frequency with which each MACROP is used and discard those which fell below a certain threshold.[1] It is important to note that STRIPS generalized and constructed a MACROP for every sequence of actions which was not subsumed by an already existing MACROP. A better method is to avoid the effort involved in generalizing and saving plans which are unlikely to be helpful in future processing. This approach demands having a set of criteria for judging whether or not a plan is worth generalizing. Of course, this method does not preclude the additional usefulness of a method for forgetting old schemata.

The procedure GENESIS uses to determine when to generalize is outlined in figure 4-2. It uses the following criteria in deciding whether or not to generalize a set of causally connected actions.

Find all achieved goals;

FOR each achieved goal DO
    IF the goal is not achieved using a known schema AND the goal is a thematic goal
    THEN generalize the actions used to achieve this goal into a new schema

Figure 4-2: Outline of the procedure for determining when to generalize.

---

[1] This solution was simply a proposal and the authors state: "We have not, however, experimented with any such mechanisms."

1) Does the combination of actions achieve a goal for a character in the narrative?

2) Does the combination of actions used to achieve this goal fail to match an existing schema?

3) Is the achieved goal a general one which is likely to be encountered again?

Recall that the understander checks each state which is added to the model to see if it fulfills some character's goal. If it does fulfill a goal, the understander marks that goal as being achieved. Consequently, obtaining a list of achieved goals is simply a matter of inspecting the model. Each of these goals is then examined to see if it meets the second two criteria. Each one which meets the remaining criteria results in the generation of a new schema. Therefore, more than one schema could be acquired from a single narrative.

The second criterion is the obvious one of not already possessing a schema for the combination of actions which achieves a goal. This simply involves checking the highest level support for the goal state and making sure it contains a combination of actions by the character who achieved the goal. If the system already had a schema for this case, it would have used it in processing the narrative and the goal state would be supported at the highest level by an instance of this schema instead of by a combination of actions.

The third criterion is the crucial one for insuring that the acquired schema will be a useful one. A novel way of achieving a state which satisfies normal human wants and desires is likely to result in an interesting schema, one which will arise again and again. Recall that *thematic goals* were defined as the highest level goals which motivate a character's action. If an action achieves a thematic goal for its actor, it requires no further explanation. Consequently, a novel combination of actions is considered to be worth generalizing into a new schema if and only if it achieves a thematic goal. Recognizing instances of thematic goals is easy since the understander specially marks these goals as thematic and therefore requiring no further explanation. Recall that the understander is able to do this because the system is given a list of thematic goals including, for example, the goal of possessing money.

In the detailed kidnapping narrative, John achieves the goal of possessing 250,000 dollars. Since the system does not possess a schema for kidnapping for ransom when it processes this narrative, this goal state is supported at the highest level by a combination of actions as shown in figure 3-1. Since wanting to possess money is a thematic goal, the combination of actions he used to achieve this goal meets all three criteria for generalization.

Additional criteria for generalization were discussed in [DeJong83]. These included the following:

4) Are the resources used in the achievement of the goal generally available?

5) Is the new method for achieving the goal at least as efficient as other known schemata for achieving this goal?

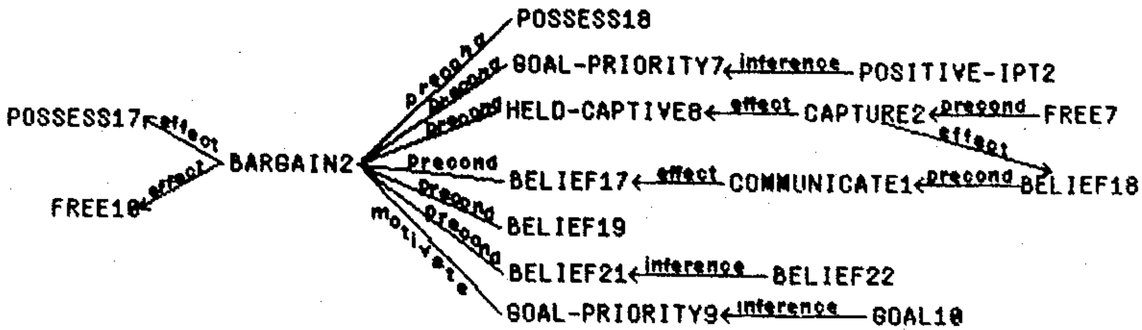
However, we have not yet developed exact methods for determining whether or not a resource is "generally available," or for comparing the efficiency of various plans. Consequently, these criteria are not used in the current implementation.

4.2. Constructing a Generalized Support Network

After deciding to generalize, the system constructs a generalized version of the combination of actions used to achieve the selected thematic goal. This generalized version contains only the information necessary to maintain the causal connectedness and well-formedness of the original network. Details specific to the particular narrative have been removed. The generalized support network constructed from the kidnapping narrative is shown in figure 4-3. The procedure for constructing a generalized support network is outlined in figure 4-4.

4.2.1. Step 1: Isolate the causal structure

The first step in the generalization procedure is to isolate the actions and states which actually contribute to the achievement of the thematic goal. This simply involves extracting the highest level support for the achieved goal state. In the kidnapping example, the achieved state is John possessing \$250,000 and the support for this state is shown in figure 3-1. This step eliminates extraneous information in the narrative which does not contribute to the achievement of the goal, such as the fact in the kidnapping example that Mary was wearing blue jeans. In addition, by only including the highest level action schemata which were determined to be in the narrative, lower level actions which are sub-



POSSESS17	Person4 has Money3.
FREE10	Person5 is free.
BARGAIN2	Person4 makes a bargain with Person8 in which Person4 releases Person5 and Person8 gives Money3 to Person4.
POSSESS18	Person8 has Money3.
GOAL-PRIORITY7	Person8 wants Person5 free more than he wants to have Money3.
POSITIVE-IPT2	There is a positive interpersonal relationship between Person8 and Person5.
HELD-CAPTIVE8	Person4 is holding Person5 captive.
CAPTURE2	Person4 captures Person5.
FREE7	Person5 is free.
BELIEF17	Person8 believes Person4 is holding Person5 captive.
COMMUNICATE1	Person4 contacts Person8 and tells him that he is holding Person5 captive.
BELIEF18	Person4 believes he is holding Person5 captive.
BELIEF19	Person4 believes Person8 has Money3.
BELIEF21	Person4 believes Person8 wants Person5 to be free more than he wants to have Money3.
BELIEF22	Person4 believes there is a positive interpersonal relationship between Person8 and Person5.
GOAL-PRIORITY9	Person4 wants to have Money3 more than he wants to hold Person5 captive.
GOAL10	Person4 wants to have Money3.

Figure 4-3: Generalized Support Network for Kidnap Narrative #2

(1) Isolate the causal structure which contributes to the achievement of the thematic goal.

(2) Produce an over-generalized version of this causal structure. In the remaining steps, this structure will be specialized just enough to be a causally complete explanation of how a character can use the technique illustrated in the narrative to achieve a thematic goal.

(3) Constrain the achieved goal in the generalized causal structure to be a thematic goal.

(4) Impose interschema constraints on the generalized causal structure. These maintain the validity of each causal link in the explanation.

(5) Impose intraschema constraints on the generalized causal structure. These maintain the well-formedness of each schema instance in the explanation.

(6) Combine individual occurences of objects in the generalized causal structure to form a set of conceptual roles for the overall plan.

(7) Remove inappropriate roles from instances in the generalized causal structure. As a result of generalization, schema instances may be left with roles which are no longer relevant. In this step, these roles are removed.

Figure 4-4: Outline of the procedure for producing a generalized support network.

sumed by larger actions are ignored. For example, only the fact that John CAPTUREd Mary is included in the highest level support. The lower level actions which describe exactly how he carried out this task are subsumed by the CAPTURE and are therefore not included. Since the system already possesses knowledge about capturing individuals, it is assumed that it will be able to determine this event from bottom-up clues if it occurs in another narrative. Once a CAPTURE is determined, it can be incorporated as part of higher level actions such as kidnapping for ransom.

4.2.2. Step 2: Produce an over-generalized copy of the causal structure

The overall approach in constructing a generalized support network is to initially generalize as far as possible and then re-introduce only the constraints necessary to maintain the causal connections between schemata and the well-formedness of individual schemata. But first, in order to avoid destroy-ing the original representation of the narrative, a copy is made of the support network extracted in step 1. If the original network itself was generalized, the specifics of the original narrative would be lost and the system would be unable to answer questions about it.

The copy which is made is over-generalized in two respects. First, the class of each schema instance in the network is generalized to SCHEMA, the highest level in the hierarchy. In steps 3, 4, and 5, constraints will be introduced which force instances to be in more specialized classes. The class of each instance will be progressively refined until it is just specific enough to maintain the validity of the explanation. Second, each slot is filled by a new, unique schema instance. In the model generated by the understander, there is a single representation for each individual OBJECT. For example, "John" is represented by the instance PERSON3, and all of John's actions have PERSON3 as their actor. However, a single person or object may fill several different roles which could be filled by different individuals without disrupting the causal structure of the narrative. Therefore, in the initially over-generalized copy, each occurrence of an object is represented by a unique instance of an OBJECT schema. For example, if the model contains:

```
(CAPTURE1 [ACTOR (PERSON3 (NAME (JOHN1))(GENDER (MALE1]
          (OBJECT (PERSON2 (NAME (MARY1))(GENDER (FEMALE1]
```

and

```
(POSSESS9 [SUBJECT (PERSON3 (NAME (JOHN11))(GENDER (MALE1]
          (OBJECT (MONEY1]
```

then in the corresponding locations in the generalized support network are:[1]

```
(SCHEMA12 [ACTOR (SCHEMA13 (NAME (SCHEMA14))(GENDER (SCHEMA15]
          (OBJECT (SCHEMA16 (NAME (SCHEMA17))(GENDER (SCHEMA18]
```

and

```
(SCHEMA19 [SUBJECT (SCHEMA20 (NAME (SCHEMA21))(GENDER (SCHEMA22]
          (OBJECT (SCHEMA23]
```

In steps 3, 4, and 5, constraints will be introduced which force separate instances to be equivalent.

[1]Clearly, unique identifiers such as SCHEMA12, SCHEMA13, etc. are only meaningful to a computer and do not allow the reader to identify which over-generalized instances correspond to which instances in the original model of the narrative. Consequently, in order to make the trace produced by the generalizer readable, the system uses names which make this correspondence clear. Instead of using opaque names such as SCHEMA12 or G0000252, new names are created which match the class of the instance in the

During this process, objects which must be the same in order to maintain the validity of the explanation will be marked as being equivalent and merged into a single instance in step 6.

In regards to the over-generalized version of a schema instance, it may seem odd that roles such as NAME, GENDER, ACTOR, and SUBJECT are retained on instances which are initially only constrained to be SCHEMAs since these roles are not appropriate for all classes of schemata. All the original roles are kept in order to maintain the basic skeleton of the original explanation and are necessary for re-introducing necessary constraints in later steps. Roles which are inappropriate for the final generalized class of an instance are removed in step 7. It should be possible to remove all roles initially and re-introduce them as the necessary constraints demand them; however, we have not yet explored this approach.

An important part of the original support network which is retained in the over-generalized version is the information attached to each support link. As mentioned in the beginning of section 3.2.2, the understander attaches to each support link information which was used in constructing the causal connection it represents. This information will allow us to re-introduce necessary constraints in step 4, and is described in detail in section 4.2.4.

4.2.3. Step 3: Constrain achieved goal to be thematic

The first step involved in re-introducing necessary constraints is to constrain the achieved goal to be a thematic one. This insures that the generalized combination of actions will still achieve an important goal and therefore represent an interesting schema. When the understander infers the existence of a thematic goal, it marks it as being thematic and annotates it with the thematic goal pattern which it matches. Like the information attached to support links, this information is retained in the over-generalized copy and is used to appropriately constrain it. This is done by constraining the goal and goal state in the generalized support network to match the thematic goal pattern. The following is the trace produced by the generalizer when constraining the goal and goal state from the kidnapping example.[2]

model, e.g. CAPTURE2, PERSON7. Although these instances retain their original class names, they are over-generalized versions and are marked as being immediate members of the class SCHEMA. This information should be kept in mind when reading upcoming excerpts from the trace produced by the generalizer.

```
Imposing constraints to maintain thematic quality of the goal:
(GOAL10 [SUBJECT (PERSON53 (NAME (JOHN21))
                           (GENDER (MALE37]
        (OBJECT (POSSESS25 [SUBJECT (PERSON54 (NAME (JOHN22))
                                              (GENDER (MALE38]
                           (OBJECT (MONEY12 (AMOUNT (#250000#9))
                                            (TYPE (DOLLAR11]

Must match thematic goal pattern:
(GOAL (SUBJECT ?SUBJECT)
      (OBJECT (POSSESS (SUBJECT ?SUBJECT)
                       (OBJECT (MONEY]

      GOAL10 constrained to be a GOAL
      POSSESS25 constrained to be a POSSESS
      PERSON54 must be equivalent to PERSON53
      JOHN22 must be equivalent to JOHN21
      MALE38 must be equivalent to MALE37
      MONEY12 constrained to be a MONEY

Imposing thematic constraints on achieved state:
 (POSSESS17 [SUBJECT (PERSON4 (NAME (JOHN2))(GENDER (MALE2]
            (OBJECT (MONEY3 (AMOUNT (#250000#2))(TYPE (DOLLAR2]

      POSSESS17 must be equivalent to POSSESS25
      PERSON4 must be equivalent to PERSON54
      JOHN2 must be equivalent to JOHN22
      MALE2 must be equivalent to MALE38
      MONEY3 must be equivalent to MONEY12
      #250000#2 must be equivalent to #250000#9
      DOLLAR2 must be equivalent to DOLLAR11
```

4.2.4. Step 4: Impose interschema constraints.

Interschema constraints are those which maintain the integrity of connecting support links in the network. We will use the kidnapping narrative to illustrate the process of imposing interschema constraints by showing how the FATHER relationship in its support network is only constrained by the explanation to be a POSITIVE-IPT (for positive interpersonal theme, a superclass of PARENT, SPOUSE, etc.). Every time a support link is added during understanding, it is annotated with the pattern from the schema database used to construct it and the class in the schema hierarchy from which it was inherited. For example, when the system infers the goal priority:

[2] Remember that GOAL10, PERSON53, etc. are simply unique identifiers for instances which are actually in the class SCHEMA until further constrained by the generalizer.

```
(GOAL-PRIORITY [SUBJECT (PERSON (NAME (FRED]
               [GREATER-GOAL (FREE (SUBJECT (PERSON (NAME (MARY]
               [LESSER-GOAL (POSSESS (SUBJECT (PERSON (NAME (FRED))))
                                     (OBJECT (MONEY (TYPE (DOLLAR))
                                                    (AMOUNT (250000]]
```

as a secondary inference from the state:

```
(FATHER (SUBJECT (PERSON (NAME (FRED))))
        (OBJECT (PERSON (NAME (MARY]
```

it annotates it with the pattern from the schema library:

```
(GOAL-PRIORITY (SUBJECT ?SUBJECT)
               (GREATER-GOAL (FREE (SUBJECT ?OBJECT)))
               (LESSER-GOAL (POSSESS (SUBJECT ?SUBJECT)(OBJECT (MONEY]
```

and the fact that this inference was inherited from the schema POSITIVE-IPT.

When the constraints for a support link are imposed, the schema instance corresponding to the source of the information is constrained to be in the class from which the information was inherited, and the schema instance on the opposite end of the link is constrained to match the pattern from the schema library. Which end of a support link is the source of information depends on the type of support link involved. In the case of inference support links like the example, the source of the information is the antecedent (e.g. FATHER) and the opposite end of the link is the consequent (e.g. GOAL-PRIORITY). For effect support links, the source of the information is the action and the opposite end is the state which it supports. For example, if BELIEF7 is supported by an action TELEPHONE1 as an effect, then BELIEF7 is constrained to match the corresponding pattern specifying the effect of a TELEPHONE, and TELEPHONE1 is constrained to be in the class from which this effect was inherited (i.e. COMMUNICATE). For precondition and motivation support links, the source of the information is the action and the opposite end is the state which supports it. For example, if BELIEF8 supports TELEPHONE1 as a precondition, then BELIEF8 is constrained to match the corresponding pattern specifying the precondition of a TELEPHONE, and TELEPHONE1 is constrained to be in the class from where this precondition was inherited (i.e. COMMUNICATE). The following is the trace produced by the generalizer when imposing the interschema constraints for the example inference link.

```
Imposing interschema constraints for:
(GOAL-PRIORITY7 [SUBJECT (PERSON18 (NAME (FRED3))(GENDER (MALE6]
                [GREATER-GOAL (FREE8 (SUBJECT (PERSON19 (NAME (MARY6))
                                                        (GENDER (FEMALE6]
                (LESSER-GOAL (POSSESS20 [SUBJECT (PERSON20 (NAME (FRED7))
                                                           (GENDER (MALE13]
                                        (OBJECT (MONEY7 (AMOUNT (#250000#3))
                                                        (TYPE (DOLLAR6]

secondary inference of:
(FATHER3 [SUBJECT (PERSON21 (NAME (FRED8))(GENDER (MALE15]
         (OBJECT (PERSON22 (NAME (MARY8))(GENDER (FEMALE7]

   FATHER3 constrained to be a POSITIVE-IPT
   PERSON18 must be equivalent to PERSON21
   FRED3 must be equivalent to FRED8
   MALE6 must be equivalent to MALE15
   FREE8 constrained to be a FREE
   PERSON19 must be equivalent to PERSON22
   MARY6 must be equivalent to MARY8
   FEMALE6 must be equivalent to FEMALE7
   POSSESS20 constrained to be a POSSESS
   PERSON20 must be equivalent to PERSON21
   FRED7 must be equivalent to FRED8
   MALE13 must be equivalent to MALE15
   MONEY7 constrained to be a MONEY
```

Thus, the system only imposes the required relationship between the individuals filling the roles of kidnap victim and ransom payer. The fact that there was a specific father-daughter relationship in this particular narrative is recognized as incidental and not crucial in maintaining the validity of the explanation. The necessary requirement is that there be some kind of positive relationship between these two individuals so that the ransom payer will value the freedom of the victim more than the possession of money. The same technique is used to generalize the action TELEPHONE, which represents someone using a telephone to MTRANS information to someone else, to the more abstract action COMMUNICATE, which represents setting up a communication path by some means (e.g. letter, electronic mail etc.) and then performing an MTRANS. This generalization is possible because the important effect of the TELEPHONE action in the original narrative is that the ransom payer believes that the kidnapper has the victim, and this effect is inherited from COMMUNICATE.

4.2.5. Step 5: Impose intraschema constraints

Intraschema constraints concern maintaining the well-formedness of each individual schema instance. This is accomplished by imposing on the filler of each role the appropriate *role constraint* from the schema library. For example, since the role constraints specify that both the SUBJECT and OBJECT of a POSITIVE-IPT state must be PERSONS, the SUBJECT and OBJECT role fillers of an instance of POSITIVE-IPT in the generalized support network are constrained to be PERSONS. Note that the constraints that are used are those which are specified for the current generalized class of the instance, not the class specified in the original narrative. If the original class FATHER was used instead of the generalized class POSITIVE-IPT, the SUBJECT would be required to be MALE, which is obviously not a necessary constraint on the ransom payer in a kidnapping schema. Consequently, interschema constraints must be imposed before intraschema constraints in order to determine the appropriate class of actions and states in the generalized support network. Once this is done, the intraschema constraints can be imposed on the schemata filling the roles of these actions and states and then intraschema constraints can be recursively imposed on the role fillers of these schemata. For example, what follows is a trace of the generalizer imposing the intraschema constraints on the POSITIVE-IPT state in the generalized support network for the kidnapping narrative.[3]

```
Imposing intraschema constraints for:
(FATHER3 [SUBJECT (PERSON21 (NAME (FRED8))(GENDER (MALE15]
           (OBJECT (PERSON22 (NAME (MARY8))(GENDER (FEMALE7]

    PERSON21 constrained to be a PERSON
    FRED8 constrained to be a NAME
    MALE15 constrained to be a GENDER
    PERSON22 constrained to be a PERSON
    MARY8 constrained to be a NAME
    FEMALE7 constrained to be a GENDER
```

[3]Remember that FATHER3 is just a unique identifier for an instance which is currently in the class POSITIVE-IPT (after being specialized from the class SCHEMA by the interschema constraint just discussed). Also, the instances PERSON21 and PERSON22 are originally in the class SCHEMA as a result of over-generalization and are just now being constrained to be PERSONS. Instances are eventually renamed at the end of generalization to match their resulting class.

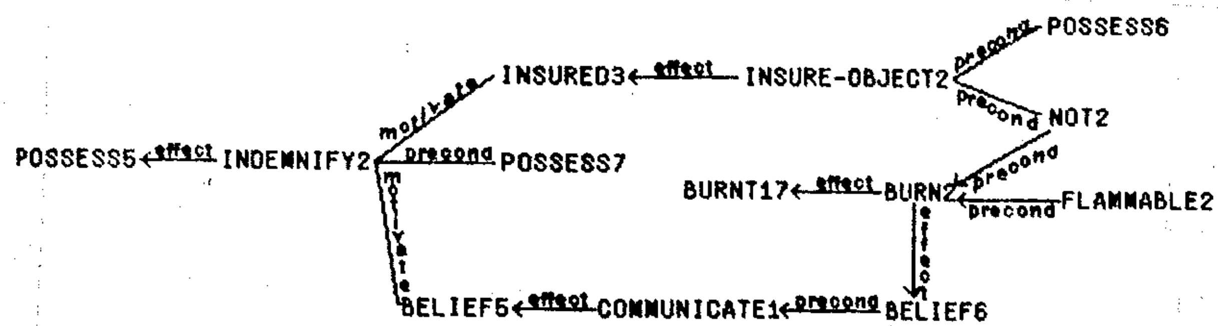
4.2.6. Step 6: Combine equivalent instances

In this next to last step, each equivalence class of OBJECT instances is replaced by a single instance. Constraints on the set of equivalent instances are merged so that the resulting single instance is at least as specific as any individual occurrence in the generalized support network resulting from steps 1 through 5. For example, if one occurrence of an instance has been constrained by the explanation to be an INANIMATE-OBJECT of color RED, and an equivalent instance has been constrained to be a BLOCK of any color, then the resulting merged instance will be a BLOCK of color RED.

In the kidnapping example, this step merges together all of the individual occurrences of the kidnapper, the victim, the ransom payer, and the ransom money and creates a single OBJECT instance for each of these roles. These are respectively the objects PERSON8, PERSON5, PERSON4, and MONEY3 which occur in the final generalized support network shown in figure 4-3. In the arson example, single objects are created for the arsonist, the insurance company, the burned object, and the insurance money. These are easily distinguished in the final generalized support network for the arson example shown in figure 4-5. In both of the currently running examples, each of the important objects in the original narrative happens to fill a single conceptual role in the overall schema. Consequently, the explanation requires that all the individual occurrences be equivalent. Each original object in the narrative maps into a single object in the final generalized support network. Therefore, the current examples do not adequately illustrate the importance of separating out the individual occurrences of an object in a narrative and eventually creating a distinct object for each conceptual role in the final generalized schema.

4.2.7. Step 7: Remove inappropriate roles

The final step in the construction of a generalized support network is removing roles which are not relevant to the final generalized class of a schema instance. For example, consider an instance which was a PERSON with NAME and GENDER roles in the original narrative but is only constrained by the explanation to be a PHYSICAL-OBJECT. Since all PHYSICAL-OBJECTs do not have names and genders, these roles should be removed from the final generalized representation.



POSSESS5	Person2 has Money2.
INDEMNIFY2	Insurance-Company2 reimburses Person2 Money2 for the loss of Physical-Object2.
INSURED3	Person2 is insured by Insurance-Company2 for Money2 if Physical-Object2 is burnt.
INSURE-OBJECT2	Person2 insures Physical-Object2 with Insurance-Company2 for Money2 in the case that Physical-Object2 is burnt.
POSSESS6	Person2 owns Physical-Object2
NOT2	Physical-Object2 is not already burnt.
POSSESS7	Insurance-Company2 has Money2.
BELIEF5	Insurance-Company2 believes that Physical-Object2 is burnt.
COMMUNICATE1	Person2 contacts Insurance-Company2 and tells them that Physical-Object2 is burnt.
BELIEF6	Person2 believes that Physical-Object2 is burnt.
BURN2	Person2 burns Physical-Object2.
BURNT17	Physical-Object2 is burnt.
FLAMMABLE2	Physical-Object2 is able to burn.

Figure 4-5: Generalized Support Network for Arson Narrative #2

Finding such inappropriate roles is easy since no interschema or intraschema constraints will have been imposed on their fillers. Consequently, their fillers will still be in the class SCHEMA as a result of the initial over-generalization in step 2. The fillers of all appropriate roles will be at least constrained by an intraschema constraint for that role, and will therefore be in a more specialized class than SCHEMA. Removing all roles whose fillers are only constrained to be SCHEMAs will therefore remove all of the inappropriate roles and leave appropriate ones untouched.

This process is not well illustrated in the examples since the cases in which the class of an instance is generalized (such as the COMMUNICATE and POSITIVE-IPT examples) do not generate any inappropriate roles. Also, as noted in the discussion of step 2, there is probably a way of avoiding this step by removing all roles in the initial over-generalization and re-introducing only the roles necessary to make the explanation complete and well-formed. However, we have not as yet explored this approach to constructing a generalized support network.

4.3. Packaging into a Form for the Schema Library

The final step in acquiring a new action schema is separating the generalized support network into preconditions, effects, expansion schemata, etc., which can be added to the schema library. Following is a discussion of how this information is extracted.

4.3.1. Roles and role constraints

The subject of the achieved thematic goal becomes the actor of the new schema. New roles are created for each remaining OBJECT which appears in more than one place in the generalized support network or is more constrained than required by the intraschema constraints of the one action or state in which it does appear. The reason for omitting objects which appear in only one unconstrained place in the generalized explanation is that such an object cannot be a crucial object in the overall schema since it is completely local and is unconstrained by the rest of the explanation. An example of such an OBJECT in the kidnapping example is the location where the money is delivered. It appears only as part of the completed BARGAIN schema and the only constraint on it is that it be a LOCATION, which is an intraschema constraint required for the BARGAIN to be well-formed. However, roles are created for the kidnapper, victim, ransom payer, and ransom money since each of these appears in several places in the generalized support network.

Each role of the new schema is given the constraint of matching its corresponding OBJECT instance in the generalized support network. For example, the constraint for the kidnap victim is that it be a PERSON and the constraint for the ransom money is that it be MONEY. The actual instances in the generalized support network have the additional constraints that their role fillers be well-formed, that is that names be NAMEs, genders be GENDERs etc., but these constraints are not added explicitly since they are implied by the constraint on the class of the object. If for some reason the explanation requires that a role filler of an object be more constrained than required for well-formedness, then this constraint *is* specified in the role constraints for the new schema.

4.3.2. Preconditions, motivations, effects, and terminations

The preconditions for the new schema are the states which are leaves of the generalized support network but not motivations of actions by the main actor. In the kidnapping example, the ransom payer possessing the ransom money is a precondition.

The motivations for the new schema are the states which are leaves of the generalized support network and motivations of actions by the main actor. In the example, the kidnapper wanting to have money is a motivation.

The effects of the schema are the effects of all actions within the generalized support network which are not terminated by other internal actions. In the example, the kidnapper possessing the ransom money is an effect. The victim being held captive is not an effect since this state is terminated by the successfully completed bargain.

Finally, the terminations of the schema are the states which are terminated by an action within the generalized support network but not produced by another internal action. In the example, the ransom payer no longer possessing the ransom money is a termination. The victim no longer being held captive is not a termination since this state is originally the effect of the CAPTURE action within the generalized support.

4.3.3. Suggesting schemata

The schema for each action within the generalized support network is marked as suggesting the new composite action. In the example, the schema CAPTURE has the new kidnap-for-ransom schema added to its list of suggested schemata. If an action in the generalized support network has constraints on its role fillers, then these must also be met before the new schema is suggested. For example, not just any BARGAIN will suggest the new kidnapping schema. It will only be suggested if it is a BARGAIN in which a person is released in exchange for money.

In some situations, this approach of having every sub-action suggest the new schema is too liberal. In the arson example, this approach results in suggesting the new arson-for-insurance schema every time someone buys fire insurance. There should be a general way of determining which sub-actions should suggest a new schema and which ones should not. A better approach would be for a sub-action to suggest the new schema only if it achieves a state which is not normally a goal state. This means that the actor must have some other reason for performing this action, and perhaps it is because it is part of the new schema which does achieve a normal goal. Since buying fire insurance achieves a normal goal of protecting oneself against financial loss, it should not suggest the arson-for-insurance schema. This would make the system more like PAM in that once it has one explanation for an action it would not try to find another. A planned revision of the system will incorporate this modification.

4.3.4. Determining conditions and expansion schemata

The determining conditions of the new schema are all the actions within the generalized support network. Consequently, in the kidnapping example, the determining conditions are:

1) The kidnapper captures the victim.
2) The kidnapper communicates to the ransom payer that he has the victim.
3) The kidnapper and ransom payer complete a bargain in which the kidnapper releases the victim in exchange for the ransom payer giving him money.

The interior of the generalized support network (i.e. all the states and actions except the preconditions, motivations, effects and terminations along with their interconnecting support links), is stored away as the expansion of the new schema. If the learned schema is ever determined in the context of another narrative, this structure (instantiated with the current role fillers) will be added to the explanation in the model.

It is instructive to trace through how this approach allows the system to determine the appropriate learned schemata in the two "test" narratives. In the kidnapping example, Bob locking Alice in his basement determines a CAPTURE schema. This schema in turn suggests the learned schema CAPTURE-BARGAIN. Recall that a determining condition is considered to be satisfied if either it is added to the model (in which case it was either directly in the input or was itself suggested and determined bottom-up), or if all of its effects can be inferred to exist. An effect is inferred to exist if it is in the model or if it is a precondition of another determining condition which has already been satisfied. After the system receives the inputs that Alice has been released and that Bob has received some money, both of the effects of the BARGAIN determining condition are met and the bargain is inferred to have taken place. Since a precondition for the bargain is that the ransom payer believe that the kidnapper has the victim, and since this is the only effect of the determining condition that the kidnapper has communicated this fact to him, this final determining condition is satisfied and an instance of the learned schema is added to the model along with the causal structure in its expansion.

In the test narrative for the arson example, the determining conditions for the learned schema are:

1) The actor insures a building against fire.
2) The actor burns the building.
3) The actor communicates to the insurance company that his building is burnt.
4) The insurance company reimburses him for the loss.

When John burns his building, the learned schema is suggested and the second determining condition is satisfied. Next, John getting money matches the effect of an insurance company indemnifying the actor of the suggested arson-for-insurance schema, so the 4th determination condition is satisfied. Since the preconditions of the INDEMNIFY are that the actor have fire insurance on the building and that the insurance company believes it is burnt, the effects of both the INSURE and the COMMUNICATE are inferred to exist and the remaining two determining conditions are satisfied.

This approach to determining a learned schema is sometimes too liberal. For example, if in a narrative an insured building is burnt but it is not mentioned who or what caused the fire, the system would infer that the owner burned it in order to collect the insurance (since all the effects of this suggested action exist). There are at least two possible solutions to this problem. First, if both the insuring and indemnifying actions were kept from suggesting the learned schema (using the approach discussed in the last section), then arson-for-insurance would never even be suggested in such a situation. Second, using arson-for-insurance might also be avoided in this case if the system had another schema in which the burnt state was caused by a non-volitional, accidental fire and also had some way of preferring this explanation to the volitional one. In any case, it is not surprising that constructing appropriate suggestions and determination conditions has lead to problems. Selecting the appropriate schema to use in a particular situation has always been a difficult problem in building understanding systems [Charniak82, DeJong82, Lytinen84, Norvig83], and the problem is even harder when the system must construct its own selection criteria for learned schemata. This is obviously an important problem for future research.

CHAPTER 5

COMPARISON TO OTHER WORK IN LEARNING

Much learning does not teach understanding.

Heraclitus

Most research in machine learning [Langley81, Michalski83a, Michalski83b, Mitchell78, Vere78, Winston70] has involved constructing a generalized description of a concept by extracting the common features across a number of examples. Such correlational approaches to learning can be applied with little or no domain knowledge; however, they require a large and representative set of examples and cannot take advantage of causal knowledge about a domain. For example, if a correlational system was used to learn about kidnapping and in all the examples it was given the victim happened to be wearing blue jeans, it might consider wearing blue jeans to be an integral part of kidnapping. On the other hand, an explanation-based system requires detailed causal knowledge about a domain, and can use this knowledge to generalize the causal explanation of a single problem solving example into a generalized description of a combination of actions and the objects which must participate in it. Of course, many concepts are not determined soley by causal and functional constraints and in many situations causal information is unavailable or inappropriate. Consequently, explanation-based learning is not always applicable.

Before discussing other research in explanation-based learning, a correlational system which should be specifically mentioned is IPP [Lebowitz80, Schank82] since, like GENESIS, it is a natural language processor. IPP processes news articles and uses a correlational approach to learn specializations of existing schemata. For example, the system already has a schema for kidnapping and after processing several stories which describe kidnappings in Italy carried out by the terrorist group the Red Brigades, it creates a specialized schema for kidnappings in Italy in which the Red Brigades is the default kidnapper. Later, if it encounters an article describing a kidnapping in Italy in which the kidnappers are not mentioned, it assumes the Red Brigades is the responsible party. Besides using a correlational instead of an explanation-based approach to learning, IPP constructs specializations of existing schemata instead of learning new schemata for compositions of actions which achieve important goals.

The explanation-based generalization process used by GENESIS can be viewed as using standard rules of generalization such as those described in [Michalski83b], but applying them in specific cases for principled reasons motivated by the structure of the explanation. Extracting the causal structure of an example and ignoring the rest of it is an example of applying what Michalski calls the *dropping condition rule*. States and actions which do not contribute to the achievement of the goal are dropped from the description of the schema being constructed. Generalizing up the hierarchy to the class where critical effects or inferences are inherited is an example of what Michalski calls *climbing a generalization tree*. Finally, creating unique OBJECT instances for each role filler and eventually collecting together equivalent instances to form a set of roles for the resulting schema is an example of what Michalski calls *changing constants to variables*. The crucial difference is that in an explanation-based system such as GENESIS, these generalization rules are applied in specific circumstances where it is known that applying them will not destroy the causal structure of a plan or a chain of inferences. This allows an explanation-based system to learn useful concepts from a single example. In addition, it is guaranteed that the resulting generalized set of actions can be used to achieve a certain goal state, assuming, of course, that the system's model of the world implicit in its knowledge of inferences and actions is accurate.

Of all previous learning systems, the generalization process used by GENESIS is most similar to that used by STRIPS to generalize planning sequences into new MACROPs [Fikes72]. Like STRIPS, GENESIS first produces an over-generalized copy of a causally connected set of actions in which each occurrence of an object is represented by a distinct parameter. It then specializes this structure so that it represents a valid causally connected plan of actions. This is done by constraining the over-generalized copy to match patterns from its knowledge base of inferences and the effects and preconditions of actions. However, unlike STRIPS, GENESIS generalizes actions and states as well as objects and locations. As described above, it is able to generalize the state FATHER to POSITIVE-IPT and the action TELEPHONE to COMMUNICATE. This is possible because all actions and states are arranged in an abstraction hierarchy and inherit effects, preconditions, inferences, etc. from schemata above them in the hierarchy. In addition, GENESIS generalizes the temporal order of independent actions. This is because it uses a dependency graph to represent the structure of the overall plan, and can therefore represent partially ordered actions. In contrast, STRIPS uses a triangle table which requires a total ordering on the set of actions.

Recent work on the LEX system which learns heuristics for solving integration problems [Mitchell83] has also emphasized an explanation-based approach to learning. By analyzing the explanation of how an application of an operator contributed to the solution of a single problem, LEX-2 is able to refine the heuristics for when to apply this operator. However, LEX-2, unlike GENESIS and STRIPS, does not construct compositions of operators (schemata or MACROPs). LEX-2 only generalizes the preconditions which an expression must satisfy in order for the operator in question to lead to a solution in the current example. Constructing a schema, however, would result in a method for completely solving similar problems and obviate the need for searching for a sequence of operators in these cases.

GENESIS differs from both STRIPS and LEX in that example problem solutions are not generated by the system solving problems itself but rather by observing and explaining the behavior of another intelligent agent. The learning approach GENESIS uses is *learning from observation* [Carbonell83], since it can be exposed to arbitrary behavior (within the scope of actions it currently possesses) and recognize on its own which examples can be generalized into useful new schemata. Although learning from the behavior of an internal planner makes the system more autonomous, it also limits what the system can learn. Learning from external behavior allows a system to learn problem solving methods which it could not generate itself. Since searching for a problem solution is combinatorially explosive, learning from external behavior is often a more feasible approach.

Recent work by Winston on learning if-then rules from precedents and exercises is an explanation-based approach to learning which uses examples explicitly provided by a teacher. Winston has used this technique in two different systems. One learns principles of character behavior and disease effects from text describing a particular case [Winston82], and another learns a physical description of an object from a functional definition and an example [Winston83]. Since the domain of the first system is more like that of GENESIS, only it will be discussed in more detail. This system is given a simple causally complete summary of Shakespeare's Macbeth and is then given an exercise of showing that a weak noble with a greedy wife may want to be king. It matches the representation of the exercise with the Macbeth precedent and copies the causal structure of the precedent onto the exercise. From this new causal structure it constructs a generalized rule which states that a weak noble married to a greedy lady will want to be king.

There are several important aspects which distinguish Winston's work from the work presented here. First, the "Macbeth system" does not need to do any explanation-driven understanding since all actions and causal connections are given to the system directly in the input text. Winston's system for learning physical descriptions, however, does construct an explanation for why a specific example meets a set of functional specifications. Second, his systems do not learn schemata for typical actions, but rather rules of inference for making certain conclusions (e.g. that a person will want to be king or that a particular object is a cup). Finally, it is not clear from the presentation that his generalization process is completely explanation-based. As in GENESIS, relations which do not enter into the causal structure are ignored. However, it is not clear how the causal structure itself is generalized, although it appears to be crucially effected by the statement of the exercise. In the Macbeth example, if the exercise involves proving something about a noble, then the generated rule is given in terms of a noble, and if the exercise involves proving something about a man, then the rule is given in terms of a man. If one were to use the explanation to relax constraints in the Macbeth example, it would involve using background knowledge, such as the constraint that normally only nobles are possible heirs to a throne, and that normally people can influence their spouses. Using these constraints, one can generalize appropriately without the need for a guiding exercise.

An explanation-based system which both learns from external behavior and constructs schemata for successful combinations of operators is Silver's LP system [Silver83]. LP learns techniques for solving algebraic equations by analyzing sample solutions produced by an external agent. Also, the latest work by Mitchell and his students on learning VLSI design techniques [Mitchell85] learns from the behavior of a human engineer and generalizes a single circuit design into a technique for implementing a particular functional unit. There are three other ESA systems besides GENESIS which learn problem solving techniques in different domains. MA (Mathematician's Apprentice) [O'Rorke84] is a system which learns schemata for proving theorems in logic from example proofs generated during interactive problem solving with a human expert. There is system in robotics [Segre85] which learns a general plan for assembling a given functional unit from a single assembly example performed by an external agent. Finally, there is an ESA system which learns methods for solving problems in classical mechanics from a single specific solution using a particular technique [Shavlik85].

One aspect which distinguishes GENESIS from all the other explanation-based learning systems is that GENESIS uses the schemata it learns to increase its ability to understand external behavior rather than to solve problems on its own. Unlike GENESIS, the other learning systems do not use their schemata to fill in missing steps in the observed behavior of external agents. However, applying learned schemata to understanding requires constructing criteria for when to apply them during understanding, which, as discussed in section 4.3, is a difficult problem. Correspondingly, GENESIS currently does not use learned schemata to construct its own plans; however, the representation of learned actions is appropriate for planning as well as understanding.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

and this is only the beginning of what they will do;...

Genesis 11:6

GENESIS has been a useful system for exploring explanation-based learning of schemata. It has demonstrated the feasibility of improving the performance of a narrative processor through knowledge-based generalization of a single example of a new concept. Although GENESIS was constructed with the kidnapping example in mind, an effort was made to keep it a general narrative processing system only constrained by the representation of actions discussed in chapter 3. Consequently, the arson example was easily handled by adding information about insurance and the volitional burning of objects. Future work will demonstrate the applicability of the approach to other areas besides illegal methods for obtaining money. The method should also be applicable to learning plans for achieving other important goals such as satisfying hunger (e.g. fishing), preserving health (e.g. vaccination), or satisfying sexual desires (e.g. using an intoxicant to lessen a potential partner's inhibition).

However, as many other AI researchers have noted, the primary contribution of constructing an AI system is pointing out problems and providing directions for future research. Consequently, listed below are some problems which require further research.

(1) One goal of future work is to further formalize the representation and attempt the construction of a completely domain independent generalizer which could be used to construct schemata for combinations of actions or inferences in widely different domains. It appears that most of the work in explanation-based learning of stereotypical combinations of actions or inferences [Fikes72, Mitchell85, O'Rorke84, Winston83] uses similar techniques, and it would be interesting to see if a single generalizer could handle examples in all of these domains.

(2) GENESIS' generalization procedure was driven by engineering concerns more than by concerns about psychological validity. Although the general notion of explanation-based generalization

seems psychologically plausible since people are capable of learning a general plan from a single example; it seems unlikely that the process involves over-generalizing and re-constraining. Consequently, we would like to explore generalization procedures which, unlike STRIPS and GENESIS, can generalize directly to the correct level without having to initially over-generalize. The latest implementation of MA [O'Rorke84], an ESA system for theorem proving, uses a procedure which generalizes directly to the correct level. However, it requires maintaining (in the original causal structure) distinct representations for every occurrence of an object and equality relationships between separate representations of the same object.

(3) Currently, GENESIS waits until it finishes processing a narrative before it attempts to generalize a combination of actions which were used to achieve an important goal. A more realistic approach is to generalize as soon as the system has an explanation for how an important goal was achieved. This allows for *within trial learning* [Laird84, O'Rorke84], in which the system uses what it learns from one part of a task to improve its performance in another part of the same task. Unfortunately, learning cannot simply be triggered by the achievement of a thematic goal, since the system may know that a goal has been achieved before it has a complete explanation for *how* it was achieved. For example, consider a narrative in which an heiress inherits an estate due to the death of her rich uncle but the narrative does not reveal until later that she murdered her uncle. Consequently, allowing for within trial learning will require monitoring achieved thematic goals and invoking the generalizer when an explanation is finally constructed for how they were achieved.

(4) When constructing a new schema, the current system traces back through the complete set of actions which a character used to achieve a thematic goal. However, this approach does not always result in a useful schema. For example, consider the following narrative:

> John's uncle owned a warehouse which was insured against fire for $100,000. John gave his uncle coffee laced with arsenic. His uncle died and John inherited his warehouse. He tried to sell the warehouse but couldn't. He burned the warehouse and called Prudential and told them it was burnt. Prudential paid him $100,000.

With the current generalizer, the schema constructed for this narrative would include generalizations of all of John's actions which support the achievement of his goal, including his act of killing his uncle. Clearly, there should be a way of "pruning" the support for an achieved state before constructing a schema. However, determining where to prune is not a trivial problem. For example, one would not want to prune the support in the kidnapping example so that having someone locked in your room is a precondition for using the schema. The problem is that although the effect of a new schema is constrained to be a state which satisfies a normal goal, its preconditions are not similarly constrained. In addition to the notion of end states which are normally desired, there should be a notion of preconditions which are normally met in the world. Then a schema would be constructed for that set of actions which achieves a thematic goal and starts at normal preconditions. Owning a building would be considered a normal precondition, while having someone locked up in your room would not.

(5) Learning better conditions for when to apply a new schema during understanding is another area for future research. The improvements suggested in sections 4.3.3 and 4.3.4 (e.g. having only otherwise unmotivated actions suggest higher-level schemata) need to be implemented. However, these improvements will probably not solve all the problems with suggesting a learned schema in inappropriate situations or failing to suggest it in appropriate situations.

(6) Another interesting problem involves generalizing to higher levels. For example, the kidnapping narrative could feasibly be generalized into a schema for holding any object for ransom, and the arson example could be generalized into a schema for insurance fraud. However, one would not want to eliminate acquiring the schemata which are currently learned from these narratives since they provide more detailed predictions and correspond to situations which people recognize as stereotypical. Rather, it would be better to generalize to several levels and save each resulting schema at a different level of abstraction in the schema hierarchy.

(7) When English expressions exist for new schemata, it should be possible to acquire these from context. This would complement past work on learning new linguistic expressions for already known concepts [Granger77, Selfridge81].

(8) Schemata already in the schema library (as a result of learning or programming) should be open to *refinement* by future inputs to the system. For example, GENESIS has a naive view of bargaining in which both participants faithfully carry out their end of the bargain. If the system were to encounter a situation in which one participant failed to complete his end of the bargain (such as a kidnapping in which the kidnapper got the ransom money and then killed the victim), it should be able to refine its BARGAIN schema to account for this anomaly. Since the need for refinement indicates a problem with the system's model of the world, it will necessarily be *failure-driven*, i.e. initiated by the violation of one of the system's expectations. However, instead of refining a schema based on correlational evidence as in Schank's work on dynamic memory [Schank82], it should be possible to refine a schema based on an explanation of *why* the expectation failure occurred. [Rajamoney85] describes recent work on a system which refines its world model after an expectation failure by constructing an explanation for the failure and performing experiments to isolate the part of the explanation which is at fault.

(9) Many schemata which are useful for narrative processing rely on social conventions in addition to causal explanations (e.g. a wedding ceremony, a trip to a restaurant or supermarket). Such schemata do not lend themselves to explanation-based learning from a single example. However, causal well-formedness at least restricts the possible form such schemata can take. Consequently, it should be possible to use explanations to guide correlational learning of schemata which involve social conventions. Combining explanation-based and correlational approaches to learning in such situations is obviously an important area for future research.

In conclusion, generalizing explanations of narratives has proven to be a successful approach to learning schemata for natural language processing. Of course, many problems, such as those listed above, remain to be solved and require further research. We intend to continue our work on learning schemata for natural language processing, and hopefully progress can be made on some of these important problems.

REFERENCES

[Bobrow] D. G. Bobrow and T. W. Winograd, "An Overview of KRL, A Knowledge Representation Language," *Cognitive Science 1*, 3-46.

[Brewer82] W. F. Brewer, "Plan Understanding, Narrative Comprehension, and Story Schemas," *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, PA, August 1982, 262-264.

[Carbonell83] J. G. Carbonell, R. S. Michalski and T. M. Mitchell, "An Overview of Machine Learning," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (ed.), Tioga, Palo Alto, 1983, pp. 331-363.

[Chafe75] W. Chafe, "Some Thoughts on Schemata," *Theoretical Issues in Natural Language Processing 1*, Cambridge, MA, 1975, 89-91.

[Charniak77] E. Charniak, "MS. MALAPROP, A Language Comprehension System," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1977.

[Charniak78] E. Charniak, "With Spoon in Hand this Must be the Eating Frame," *Theoretical Issues in Natural Language Processing 2*, Urbana, IL, 1978.

[Charniak82] E. Charniak, "Context Recognition in Language Comprehension," in *Strategies for Natural Language Processing*, M. H. Ringle (ed.), Lawrence Erlbaum and Associates, Hillsdale, NJ , 1982, 435-454.

[Cullingford78] R. E. Cullingford, "Script Application: Computer Understanding of Newspaper Stories," Technical Report 116, Department of Computer Science, Yale University, New Haven, CT, January 1978.

[DeJong81] G. DeJong, "Generalizations Based on Explanations," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C. , Canada, 1981, 67-60.

[DeJong82] G. F. DeJong, "An Overview of the FRUMP System," in *Strategies for Natural Language Processing*, W. Lehnert and M. Ringle (ed.), Lawrence Erlbaum and Associates, Hillsdale, NJ , 1982.

[DeJong83] G. DeJong, "An Approach to Learning from Observation," *Proceedings of the 1983 International Machine Learning Workshop*, Urbana, IL (April 1983)., 1983. Also appears as Working Paper 45, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

[DeJong85] G. DeJong, R. Mooney, P. O'Rorke, S. Rajamoney, A. M. Segre and J. Shavlik, "A Review of Explanation-Based Learning," Technical Report in preparation, Coordinated Science Laboratory, Urbana, IL, 1985.

[Doyle78] J. Doyle, "Truth Maintenance Systems for Problem Solving," Technical Report Technical Report-419, MIT Artificial Intelligence Lab, Cambridge, MA, 1978.

[Dyer83] M. Dyer, *In Depth Understanding*, MIT Press, Cambridge, MA, 1983.

[Fikes72] R. E. Fikes, P. E. Hart and N. J. Nilsson, "Learning and Executing Generalized Robot Plans," *Artificial Intelligence 3*, (1972), 251-288.

[Granger77] R. H. Granger, "FOUL-UP: A Program that Figures Out Meanings of Words from Context," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA, 1977, 172-178.

[Kant] E. Kant, *Critique of pure reason*, Macmillan, London, 1963 (originally published 1787).

[Laird84] J. Laird, P. Rosenbloom and A. Newell, "Towards Chunking as a General Learning Mechanism," *Proceedings of the National Conference on Artificial Intelligence*, Austin, TX, August 1984, pp. 188-192.

[Langley81] P. Langley, "Data-driven discovery of physical laws," *Cognitive Science*, 1981, pp. 31-54.

[Lebowitz80] M. Lebowitz, "Generalization and Memory in an Integrated Understanding System," Technical Report 186, Department of Computer Science, Yale University, New Haven, CT, 1980. Also appears as Ph.D. Dissertation..

[Lehnert78] W. Lehnert, *The Process of Question Answering*, Lawrence Erlbaum and Associates, Hillsdale, NJ, 1978.

[Lehnert82] W. G. Lehnert, "Plot Units: A Narrative Summarization Strategy," in *Strategies for Natural Language Processing*, M. H. Ringle (ed.), Lawrence Erlbaum and Associates, Hillsdale, NJ, 1982, 375-414.

[Lytinen84] S. L. Lytinen, "Frame Selection in Parsing," *Proceedings of the National Conference on Artificial Intelligence*, Austin, TX, 1984.

[Marcus80] M. P. Marcus, *A Theory of Syntactic Recognition for Natural Language*, MIT Press, Cambridge, MA, 1980.

[Michalski80] R. S. Michalski and R. L. Chilausky, "Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis," *Policy Analysis and Information Systems 4*, 2 (June 1980), pp. 125-160.

[Michalski83a] R. S. Michalski and R. E. Stepp, "Learning from Observation: Conceptual Clustering," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (ed.), Tioga, Palo Alto, 1983, pp. 331-363.

[Michalski83b] R. S. Michalski, "A Theory and Methodology of Inductive Learning," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (ed.), Tioga Publishing Company, Palo Alto, CA, 1983, 83-134.

[Minsky75] M. L. Minsky, "A Framework for Representing Knowledge," in *The Psychology of Computer Vision*, P. Winston (ed.), McGraw-Hill, New York, NY, 1975, 211-277.

[Minton84] S. Minton, "Constraint-based Generalization," *Proceedings of the National Conference on Artificial Intelligence*, Austin, TX, August 1984, 251-254.

[Mitchell78] T. M. Mitchell, "Version Spaces: An Approach to Concept Learning," STAN-CS-78-711, Stanford University, Palo Alto, CA, 1978.

[Mitchell83] T. Mitchell, "Learning and Problem Solving," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, 1983, 1139-1151.

[Mitchell85] T. M. Mitchell, S. Mahadevan and L. I. Steinberg, "LEAP: A Learning Apprentice for VLSI Design," LCSR-Technical Report-64, Department of Computer Science Rutgers University, New Brunswick N.J., January 1985.

[Mooney85] R. Mooney and G. DeJong, "Learning Schemata for Natural Language Processing," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, 1985.

[Norvig83] P. Norvig, "Frame Activated Inferences in a Story Understanding Program," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 1983, 624-626.

[O'Rorke84] P. O'Rorke, "Generalization for Explanation-based Schema Acquisition," *Proceedings of the National Conference on Artificial Intelligence*, Austin, TX, August 1984.

[Rajamoney85] S. Rajamoney, G. DeJong and B. Faltings, "Towards a Model of Conceptual Knowledge Acquisition through Directed Experimentation," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, 1985.

[Rumelhart80] D. Rumelhart, "Schemata: the Building Blocks of Cognition," in *Theoretical Issues in Reading Comprehension*, W. Brewer (ed.), Lawrence Erlbaum and Associates, Hillsdale, NJ, 1980, 33-58.

[Sacerdoti75] E. D. Sacerdoti, "A Structure for Plans and Behavior," Technical Report 109, SRI Artificial Intelligence Center, Menlo Park, CA, 1975.

[Schank75] R. C. Schank, *Conceptual Information Processing*, North-Holland/American Elsevier, Amsterdam, 1975.

[Schank77] R. Schank and R. Abelson, *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*, Lawrence Erlbaum and Associates, Hillsdale, NJ, 1977.

[Schank81] R. C. Schank and C. Riesbeck, *Inside Computer Understanding*, Lawrence Erlbaum and Associates, Hillsdale, NJ, 1981.

[Schank82] R. C. Schank, *Dynamic Memory*, Cambridge University Press, Cambridge, 1982.

[Segre85] A. M. Segre and G. F. DeJong, "Explanation Based Manipulator Learning: Acquisition of Planning Ability Through Observation," *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 1985.

[Selfridge81] M. Selfridge, "A Computer Model of Child Language Acquisition," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C. , Canada, August 1981, 446-451.

[Shavlik85] J. Shavlik, "Learning about Momentum Conservation," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, 1985.

[Silver83] B. Silver, "Learning Equation Solving Methods from Worked Examples," *Proceedings of the 1983 International Machine Learning Workshop*, Urbana, IL, June 1983, 99-104.

[Soloway78] E. Soloway, "Learning = Interpretation + Generalization: A Case Study in Knowledge-Directed Learning," COINS Technical Report 78-13, University of Massachusetts, Amherst, MA, 1978.

[Sussman73] G. J. Sussman, "A Computational Model of Skill Acquisition," Technical Report Artificial Intelligence-T-297, MIT Artificial Intelligence Lab, Cambridge, MA, 1973.

[Vere78] S. A. Vere, "Inductive learning of relational productions," in *Pattern Directed Inference Systems*, D. A. Waterman and F. Hayes-Roth (ed.), Academic Press, New York, 1978.

[Waltz84] D. L. Waltz and J. B. Pollack, "Phenomenologically Plausible Parsing," *Proceedings of the National Conference on Artificial Intelligence*, Urbana, IL., 1984. Also appears as Working Paper 49, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

[Wilensky78] R. Wilensky, "Understanding Goal-Based Stories," Technical Report 140, Department of Computer Science, Yale University, New Haven, CT, September 1978. Also appears as Ph.D. Dissertation..

[Wilensky82] R. Wilensky, "Points: A Theory of the Structure of Stories in Memory," in *Strategies for Natural Language Processing*, W. Lehnert & M. Ringle (ed.), Lawrence Erlbaum and Associates, Hillsdale, NJ, 1982, pp. 345-73.

[Wilensky83] R. W. Wilensky, *Planning and Understanding: A Computational Approach to Human Reasoning*, Advanced Book Program, Addison-Wesley, Reading, MA, 1983.

[Winston70] P. H. Winston, "Learning Structural Descriptions from Examples," AI Technical Report-231, MIT Artificial Intelligence Lab, Cambridge, MA, 1970.

[Winston82] P. H. Winston, "Learning New Principles From Precedents and Exercises," *Artificial Intelligence 19*, (1982), 321-350.

[Winston83] P. H. Winston, T. O. Binford and M. Lowry, "Learning Physical Descriptions from Functional Definitions, Examples, and Precedents," A.I. Memo No.679, MIT, Cambridge, MA, January 1983.