ARI Research Note 95-28

Integrating Analogy With Rules and Explanations

Greg Nelson, Paul Thagard, and Susan Hardy
University of California at Los Angeles

for

Contracting Officer's Representative
George Lawton

Research and Advanced Concepts Office
Michael Drillings, Acting Chief

June 1995

ARMY RESEARCH INSTITUTE

United States Army
Research Institute for the Behavioral and Social Sciences

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Director

Research accomplished under contract
for the Department of the Army

University of California

Technical review by

Michael Drillings

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1995, June	Final Sep 89 - Aug 92

4. TITLE AND SUBTITLE

Integrating Analogy With Rules and Explanations

5. FUNDING NUMBERS

MDA903-89-K-0179
61102B
74F
2901
C03

6. AUTHOR(S)

Nelson, Greg; Thagard, Paul; and Hardy, Susan (UCLA)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of California, Los Angeles
Department of Psychology
Los Angeles, CA 90024

8. PERFORMING ORGANIZATION REPORT NUMBER

--

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Institute for the Behavioral and Social Sciences
ATTN: PERI-BR
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARI Research Note 95-28

11. SUPPLEMENTARY NOTES

Contracting Officer's Representative, George Lawton.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release;
distribution is unlimited.

12b. DISTRIBUTION CODE

--

13. ABSTRACT ***(Maximum 200 words)***

This paper presents a new integrated artificial-intelligence model, Connecting Analogies With Rules and Explanations (CARE), which combines analogy, rule use, and the assessment of explanatory coherence. The model is applied to complex problem-solving and decision-making tasks in a variety of domains.

14. SUBJECT TERMS

Analogy · Explanation
Rules Artificial intelligence

15. NUMBER OF PAGES

48

16. PRICE CODE

--

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

INTEGRATING ANALOGY WITH RULES AND EXPLANATIONS

CONTENTS

To appear in K.J. Holyoak & J.A. Barnden(Eds.), Advances in connectionist and neural computation theory, Vol.2: Analogical connections. Norwood, NJ: Ablex.

INTEGRATING ANALOGY WITH RULES AND EXPLANATIONS

Greg Nelson, Paul Thagard, and Susan Hardy

1. INTRODUCTION

In the past decade, analogy has been one of the most progressive research areas in cognitive science. Previously, there had been isolated investigations in philosophy, psychology, and artificial intelligence, but the 1980s brought substantial work on many aspects of analogy, particularly on how two analogs can be mapped to each other and on how analogs can be retrieved from memory. Case-based reasoning, which is analogy in workaday clothes with a restriction to single domains, became an active research area in artificial intelligence.

There are, however, important unresolved issues concerning the role of analogy in human cognition. One of the most pressing concerns the relation of analogy to other central cognitive processes. How, for example, is analogical problem-solving related to rule-based problem solving in which chains of rules are used in quasi-deductive fashion to accomplish goals? One extreme view, implied by some of the advocates of case-based reasoning, is that there is no such thing as rule-based reasoning. At the other extreme, there is the view that analogy is of peripheral interest, at most a minor module to be added onto a rule-based system which handles basic cognitive operations. In between, there is the view that analogy and rule-based reasoning should be viewed as integrated aspects of a general cognitive system.

How rule-based reasoning can be integrated with analogical reasoning depends in large part on what computational mechanisms are seen as crucial to retrieving analogs: spreading activation, indexing, or parallel constraint satisfaction. Analogical retrieval mechanisms using spreading activation have been combined with production systems in models like PI (Thagard 1988, Holyoak and Thagard 1989b), PUPS (Anderson and Thompson 1989), and EUREKA (Jones and Langley 1991). Case-based reasoning systems retrieve analogs by a direct computation of similarity between problems and stored cases, often paying special attention to indexing by goals and failures.

This research was supported by contract MDA903-89-K-0179 from the Basic Research Office of the Army Research Institute for the Behavioral and Social Sciences, and conducted at the Princeton University Cognitive Science Laboratory. We thank David Gochfeld for discussions and programming that helped guide us toward the current model, and Dmitry Gorenburgov for developing the Gorbachev example. For helpful comments on a previous draft, we are grateful to John Barnden, Keith Holyoak, and Heather Pfeiffer.

Systems that combine case-based reasoning with rule-based reasoning include CASEY (Koton 1988) and CABARET (Rissland and Skalak 1989, Skalak and Rissland 1990). A third method of retrieving analogs uses parallel constraint satisfaction implemented using localist connectionist techniques (Thagard, Holyoak, Nelson, and Gochfeld 1990; Holyoak and Thagard 1989a; Thagard, Cohen, and Holyoak 1989). Analogs can be retrieved from memory on the basis of parallel satisfaction of a set of semantic, structural, and pragmatic constraints that are effectively implemented in a connectionist network. In this paper we present the CARE model, which uses these techniques to accomplish both analogical and rule-based reasoning.

Rule-based reasoning is traditionally implemented in production systems and logic programming, but we shall construe it in terms of parallel constraint satisfaction. We believe that analogy is not a module operating separately from and external to rule-based processing. Rather, rule-based processing can be viewed as another process of parallel constraint satisfaction, with deep affinities to analogical reasoning. Investigating those affinities provides clues to how it might be possible to develop a fully integrated cognitive system that seamlessly embraces both rule-based and analogical reasoning. We will describe an implemented system called CARE, for "Connecting Analogies with Rules and Explanations," that illustrates a novel kind of rule-based processing complementary with our previous connectionist work on analogy.

Additional motivation for applying parallel constraint satisfaction to rule-based reasoning comes from its successful application to another important area of high-level cognition, the evaluation of explanatory hypotheses. Thagard (1989, 1992) has developed a theory of explanatory coherence that is implemented in a connectionist program called ECHO and applied to numerous cases of reasoning in science and everyday life. Explanation and the evaluation of hypotheses are processes intimately connected to analogy and rule-based reasoning.

2. RELATIONS BETWEEN ANALOGIES, RULES, AND EXPLANATIONS

The ultimate goal of cognitive science is the development of a unified theory that embraces the full range of human information processing from vision to reasoning to language. The current goal of our research program (in collaboration with Keith Holyoak) is much more restricted. We wish to develop a local unified theory that ties together three important factors in high-level cognition: analogy, rule-based reasoning, and explanation including hypothesis generation and evaluation. Other important cognitive functions, such as vision, language, and learning would ideally be integrated into more complex versions of the cognitive system we have designed, extending it to constitute a full cognitive architecture. For now, it is a difficult enough task just to model the interrelations (summarized in figure 1) of the three kinds of thinking we have chosen as our focus.

Insert Figure 1 about here.

First consider how analogical and rule-based reasoning are tied to each other. Analogy tends to be a useful alternative to rule-based reasoning, since rules are often too rigid to suggest a solution to all problems. Analogy offers a more flexible way of using past rule-based solutions to solve problems where no exact rule-based solution is available. Thus analogy can extend a rule-based system to allow inferences and solutions that the rules alone would never have produced. Conversely, rules offer useful ways of elaborating and adapting analogies for successful solutions. Two cases may not even seem to be related to each other until rule-based inferences have clarified the similarities between them. Even after the similarities have been detected, additional inference of the sort carried out by rule-based systems can help to adapt one analog for use in accomplishing some task involving the other. Thus rule-based reasoning can help with analogical thinking as well as vice-versa.

Analogical thinking interacts with the formation and evaluation of explanations in several ways. Analogies can suggest explanatory hypotheses, generating an explanation of some puzzling fact viewed as similar to something already understood. Analogy can also be one of the factors relevant to selecting which of a number of competing explanations provides the best explanation of the facts. In the other direction, explanatory goals help to shape the ways in which analogies are retrieved, mapped, and transferred.

Rule-based reasoning and hypothesis evaluation are even more intimately connected. Though not all explanation is of the deductive sort approximated by rule-based systems, it is often natural to explain a puzzling fact by showing how it can be derived from known facts by a set of known rules. Sometimes the derivation cannot be made simply on the basis of what is known, and facts or rules must be hypothesized to make the derivation work. Evaluation of hypotheses in terms of their explanatory coherence with other facts and hypotheses must then be carried out. Thus rule-based systems are relevant to hypothesis evaluation because they can provide the hypotheses to be evaluated. Going in the other direction, hypothesis evaluation can be crucial to the operation of rule-based systems, since it can determine whether a hypothesis can be viewed as accepted and therefore capable of figuring in new deductions. An integrated explanation system would be continuously forming and evaluating hypotheses and using them in subsequent inferences.

3. CONSTRAINT SATISFACTION: WHY LOCALIST CONNECTIONISM?

CARE is an attempt to integrate, extend, and revise previous computational models of analogy and hypothesis evaluation: ACME (Holyoak and Thagard 1989a), ARCS (Thagard, Holyoak, Nelson, and Gochfeld 1990), and ECHO (Thagard 1989). As in those programs, the fundamental approach is constraint satisfaction using localist connectionist networks, extended to model rule-based reasoning and analogical transfer. Three questions naturally arise: why the connectionist approach, why use local rather than distributed representations, and what is the relevance of constraint satisfaction to rule-based reasoning?

Various methods for constraint satisfaction have been developed using techniques as diverse as logic programming and mathematical programming. We find the connectionist approach attractive for several reasons. Reasoning problems of the sort we are interested in can be viewed as optimization problems of a very complicated sort: given what you now know and the various inference methods you possess, infer the most reasonable and effective set of conclusions. Unlike a typical optimization problem in mathematical programming, it is not feasible to describe a function precisely stating the variables that are being optimized. Even if they could be described, such functions for cognitive processes would be nonlinear, since the soft constraints on reasoning involve complex trade-offs. ACME, ARCS, and ECHO use input about analogs and explanations to produce networks that yield answers based on satisfaction of multiple constraints. Other techniques for constraint satisfaction might be workable, but do not appear to be either so naturally applicable or so computationally efficient.

Some cognitive tasks have been insightfully investigated using connectionist models with distributed representations, in which concepts or hypotheses are represented by patterns of activation across numerous units rather than by individual units. This approach has computational advantages and neural plausibility, but no full-fledged distributed system for rule-based reasoning has yet been developed, although some progress has been made toward understanding how systems employing distributed representations can be given the capacity to do such reasoning (see for example Ajjanagadde and Shastri 1989, Barnden 1991, Pollack 1990, Smolensky 1990, Touretzky and Hinton 1988, van Gelder 1990). We view local representations as approximations to kinds of distributed representation that remain to be understood, just as distributed representations as produced by current backpropagation methods are obviously only weak approximations to neurological structures. Like Hendler (1991), Lehnert (1991), Eskridge (this volume), and Kokinov (this volume), we have developed a hybrid model in order to exploit new connectionist insights without abandoning valuable ideas from traditional AI.

4. RULES AND ANALOGIES

Rules and analogs have some important similarities. Both must be retrieved from long-term memory in order to be of any use. Rules and analogs must go through a binding or mapping process in which the correct match between elements of the rule or analog and the problem being solved is sought out. When a good match is found, rules are fired and analog components are transferred, generating new plausible inferences to contribute toward problem solutions.

The largest differences between these two types of information seem to be matters of degree of complexity and degree of specificity. The conditions of a rule usually only have a few propositions and most rule-based systems require a complete match of these against a data base of facts. Inferences from rules are relatively sound. Analogs usually contain many more propositions than rules, and their greater structural complexity makes them relevant in fewer situations than rules. However, it is less critical to match the entirety of an analog, because analogies often include many uninteresting details that do not fit the current problem, and sometimes only a part of the analogy may be relevant. Because of this flexibility, inferences drawn through analogical transfer may be less reliable, but more insightful. Rules and analogies may therefore be viewed as the ends of a spectrum rather than as the basis of unrelated problem-solving strategies. Our CARE system is intended in part to test the plausibility of this hypothesis. The major innovation in CARE is the use of connectionist methods to model rule-based reasoning. In section 6.2, we will describe how inferences using rules can be naturally and flexibly accomplished using connectionist techniques.

In rule retrieval as in analog retrieval, the primary goal is to find within a general knowledge base an element that may be applicable to a particular case. While some efficient search mechanisms have been developed for rule retrieval (Buchanan and Shortliffe 1984, Forgy 1982), they rely on powerful indexing and a great deal of computation, rather than on psychologically plausible theories. Our theory suggests that principles used for analog retrieval such as those in our ARCS retrieval model should be applicable to rules as well. In particular, semantic similarity should have a strong influence on retrieval, but should be combined with information about structural similarity and pragmatic relevance.

5. CARE KNOWLEDGE BASES

CARE, a program implemented in Common LISP, uses two different kinds of knowledge representation. Analogs, rules, and problem descriptions are represented as grouped sets of propositions using an extended version of predicate calculus. A problem to be solved is represented by a list of predicate calculus propositions; the predicate of each proposition is a semantic concept. Semantic concepts are represented as frame-like structures, with slots describing their semantic relationships to other concepts (such as synonymy, antonymy, and part-whole relations). Like ARCS, CARE's retrieval mechanisms use these semantic relationships to select rules and analogs which are semantically similar to the problem. This method helps to restrict the potential candidates for retrieval in a psychologically plausible way, yet allows more flexibility than matching techniques that require identical predicates.

The central element of our representation of rules and analogs is the *proposition*, a unit containing information comparable to that in a very short sentence. Each proposition consists of a *predicate*, a list of *arguments*, a *acceptance value*, and a *proposition name*. The predicate represents a concept, found in the semantic database, that describes the relationship between the arguments. It can be a simple concept like **DOG** or a complex relation like **COLLECT-FROM.** The arguments represent the objects being related, and may refer to specific objects like **NEW-YORK-CITY** or objects considered local to a given structure, such as **OBJ-DOG.** Arguments may be either constants or variables, with variables indicated by the use of **?** or **%** as the first character. Arguments in problem descriptions and analogs are usually constants, while in rules they are usually variables. The acceptance value is either **true, false,** or **unknown**, and represents the degree to which the problem solver believes a fact. These values play an important role in the inferential network described in section 6.2, as do the proposition names.

The database of rules and analogs in CARE, like the analog database in our previous models, consists of sets of propositions grouped together and given a single name. Within these collections, smaller functional groupings exist: for rules, they are conditions and actions; for analogs, they may include a start state and a goal field. As rules are defined in our system, multiple conditions must all be satisfied for a rule to apply, and if a rule that has multiple actions is applied, all of the actions are used. Analogs do not need to be fully mapped in order to apply, and only those parts which are considered pragmatically relevant are transferred.

Table 1 shows the input to CARE to create a simple rule that if something is a tree, it is likely to be tall. The variable **?x** may match to any argument of a predicate that is semantically related to trees or to tallness. Rules may be used in different sorts of inferences. The most familiar is forward inference in accord with the logical principle of *modus ponens*: from the fact that something is a tree, we may infer that it is tall. The rule may also be used backwards in accord with the principle of *modus tollens*: if something is not tall, then it is not a tree. In addition to making inferences, rules are used in problem solving to identify subgoals that seek out possible inferential paths from the goals of the problem back to the starting conditions. CARE implements two kinds of subgoaling that we call *accomplishing* and *preventing*. Given the goal to show that an object is tall, we can establish the accomplishing subgoal of showing that it is a tree, since with the rule that trees are tall we could then accomplish the inference that it is tall. On the other hand, if the goal is to show that an object is not tall, we want to avoid showing that it is a tree, since otherwise we could use the rule to infer that it is tall. Hence CARE establishes a preventing subgoal of showing that the object is not a tree.

Insert Table 1 about here.

Problems in CARE are complex structures with three fields, for data, starting conditions, and goals. Propositions in the data and start field both represent information taken to be true at the start of problem solving, with the difference that data propositions are ones likely to remain true. For retrieval, the propositions in the problem are matched against propositions in analogs and rules. Table 2 presents a small problem along with two rules relevant to its solution. Arthur is a young man who wishes to drink a love potion possessed by the magician who made it. The two rules say that if you have a liquid and wish to drink it, you drink it, and, if you do not have a liquid, even if you wish to drink it, you do not drink it. Each rule contains five propositions: four conditions and one action. From the second rule, Arthur can infer that he cannot drink the potion, but from the first, he generates the subgoal of having it. With more rules, this might lead to him to purchase or steal the potion. In the rules in table 2, the variable **?x** stands for the owner or non-owner of the liquid, **?y** stands for the liquid, and **%have-c3** stands for the proposition about the owner drinking the liquid. The **%** sign distinguishes variables representing propositions from variables that can match to anything.

Insert Table 2 about here.

6. THE CONSTRAINT NETWORKS

So far, CARE looks like a traditional rule-based system. In order to perform retrieval and inference, however, CARE uses two separate connectionist networks that we call the *comparison* network and the *inference* network. The first network is used for comparing the current problem against analogs stored in memory and amalgamates the functions of retrieval and mapping performed by our previous programs ARCS and ACME. A very different constraint network is needed to make inferences and to perform subgoaling.

Each network consists of *units* with real-valued *links* between them. Each unit has an activation between -1.0 and +1.0 that represents the degree of acceptability of what the unit stands for. In the comparison network, the units stand for hypotheses concerning correspondences between the current problem and stored analogs. In the inference network, the units represent propositions derived from rules or analogical transfer, activation of -1.0 interpreted as full rejection, 1.0 as complete acceptance, and 0.0 as acceptance value unknown. In the comparison network, all links are symmetrical, as in ARCS, ACME and ECHO. But CARE's inference network employs asymmetric links to capture the directional nature of rule-based inference. Both networks reach their conclusions by means of the relaxation algorithm described in section 6.3.

6.1. The Comparison Network

The comparison network, like the networks of ARCS and ACME, consists of units that represent hypotheses about potential mappings between the target problem and potential analogs. Each unit has a complex name that represents the different predicates, arguments, and propositions being matched. For example, it might contain a unit **DOG=DOG** representing the hypothesis that a dog in the target problem should be matched to a dog in an analog. This unit would compete with such units as **DOG=CAT** but would be favored by stronger semantic similarity. Units also exist that represent the match between the target problem and the source analog as a whole. The activations of these units reflect the *degree* of retrieval of a particular analog, and only analogs that reach a retrieval threshold actually undergo a complete mapping with the problem.

Inhibitory links with negative weights are used to discourage mappings that are not one-to-one. Excitatory links with positive weights are used to connect hypotheses that fit well together, such as ones concerning synonyms or structurally coherent elements. Although the density of links make comparison networks difficult to display graphically, the algorithm for generating them is quite simple. The comparison network acts like an ARCS retrieval network, only mapping propositions whose predicates have some semantic similarity. For each pair of propositions that are put into correspondence with each other, the unit representing the proposition mapping is connected to a unit representing the predicate mapping, to each unit representing argument mappings, and to a unit representing the overall mapping between source and target. The predicate mappings units are also connected to each of the argument mapping units. Much more detailed descriptions of how such networks are created is provided in papers on ACME and ARCS (Holyoak and Thagard 1989a; Thagard, Holyoak, Nelson, and Gochfeld 1990). When a unit representing a correspondence between the problem to be solved and some analog reaches a threshold, more units are created to produce a full ACME-like mapping between the problem and analog. After this network settles, transfer, described in section 7.3, can occur.

6.2. The Inference Network

The inference network implements the most novel aspect of CARE: rule-based inference is construed in terms of parallel satisfaction of multiple constraints. In its simplest form, rule-based reasoning appears straightforwardly deductive. A rule is a general statement that says that if certain conditions are met, then an action follows. From the rule $P \& Q \& R \rightarrow S$ and the facts P, Q, and R, we can infer S. But human are never simply deductive in this way. We may well have another rule $T \& U \rightarrow \neg S$ which, with the facts T and U would license the conclusion that S is false, contradicting the original inference. The two competing rules must then be understood in terms of uncertainty and the conflict between the conclusions S and $\neg S$ resolved as a problems of satisfying the constraints provided by the same rules. In general, we may not be able to infer Q from P and $P \rightarrow Q$, since there may be other reasons for rejecting Q (Harman 1973). Even when there are no conflicts between possible inferences, rule-based reasoning must be constrained in ways that contribute toward the accomplishment of the inferential task at hand. In problem solving, the purpose of inference is to determine how to accomplish a set of goals; in explanation, the purpose is to produce a chain of reasoning that explains a puzzling fact. The inference network in CARE is used to implement logical constraints on what can be consistently inferred as well as pragmatic constraints concerning what is worth inferring. It also makes possible a graceful kind of nonmonotonic reasoning, in which additions of new information can lead to the retraction of what had been previously accepted.

In the inference network, units represent propositions that are either taken from the problem description or from rules and analogs that are used in the course of problem solving. Unlike standard connectionist networks, the inference network is dynamic, in that new units and links are added to it in the course of problem solving. Some links between units are established by virtue of special predicates in the problem description that describe relations between propositions: **cause**, **if**, and **conjoin-event**. For example, if there is a causal connection from proposition1 to proposition2, then CARE places a link from the unit representing proposition1 to the unit representing proposition2. Other links are established by virtue of rules that are retrieved because their propositions are semantically similar to propositions in the problem description. For example, there will be links from the propositions in the conditions of the rule to propositions in the actions of the rule, once the variables in these propositions have been bound to objects in the problem description. Although we would prefer a connectionist method, CARE currently does variable binding using a standard resolution technique.

In section 5, we described four functions of rules, two involving the performance of inferences in accord with the principles *modus ponens* and *modus tollens*, and two involving subgoaling to either accomplish or prevent desired inferences. CARE uses links between units to enable a rule to fulfill these functions if its conditions or actions have been fully matched. Figure 2 displays the part of the inference network created to deal with the forward inference function of the rule that trees are tall. Here **TREE1** is the unit representing the first proposition of the rule that employs the predicate **TREE**, and **TALL2** is the unit representing the proposition in the action of the rule representing the proposition with the predicate **TALL**. The link from **TREE1** to **TALL2** is asymmetric, since *modus ponens* only licenses forward inference.

Insert Figure 2 about here.

The plus sign on the link between the units indicates a novel aspect of the design of CARE's inference network. Unlike the links in the comparison network, the links in the activation network are *activation dependent*: whether a link transmits activation from one unit to another depends on whether the activation of the transmitting unit is positive or negative. A *positively qualified* link is one that functions only if the transmitting unit has positive activation, while a *negatively qualified* link functions only if the transmitting unit has negative activation. How a link is qualified is independent of whether it is excitatory or inhibitory. The link in figure 2 is both excitatory and positively qualified and has the effect that if **TREE1** has positive activation, then **TALL2** will tend to get positive activation. If this link were not qualified, then whenever **TREE1**'s activation dropped below 0, it would tend to deactivate **TALL2**, producing the fallacious inference that if something is not a tree then it is not tall (see the algorithm for updating activation described in section 6.3).

Backwards inferences in accord with the logical principle of *modus tollens* are implemented in CARE by means of links that are excitatory and negatively qualified. Figure 3 is figure 2 with an additional excitatory link from **TALL2** to **TREE1**. This link is negatively qualified, since its function is ensure that if the activation of **TALL2** drops below 0, so will the activation of **TREE1**. According to the rule that trees are tall, if something is not tall, we have some grounds for inferring that it is not a tree.

Insert Figure 3 about here.

Activation dependent links can be inhibitory when the condition of a rule has the acceptance value false. For example, the rule "If X is not tall, then X is short," generates the forward inference link shown in figure 4 which is inhibitory and negatively qualified. This link has the effect that if the activation of unit **TALL2** drops below 0 (i.e., an object is not tall), then the negative activation combined with the negative weight on the link will tend to produce positive activation in unit **SHORT3** (i.e., the object is short). Other combinations of positive and negative weights, and positive and negative activation qualifications, are needed to deal with different combinations of truth and falsity in the conditions and actions.

Insert Figure 4 about here.

Further complications are necessary to deal with rules with multiple conditions. A rule such as "If X is a tree and X is alive, then X is green." should only be used in a forward inference based on *modus ponens* if both the conditions hold, i.e. if an object is both a tree and alive. Hence a link from a unit representing that something is a tree to a unit representing that it is green needs to be dependent not only on the unit for tree but also on the unit for alive. Thus links in CARE's inference network can be dependent on the activation of more than unit. Figure 5a displays the part of the inference network created to make forward inferences using the rule that live trees are green. The curved lines are not links, but instead indicate additional qualifications. The link from **TREE1** to **GREEN6** is excitatory but is allowed to have an effect only if both **TREE1** and **ALIVE5** have positive activation. A slightly more complicated case involving a negative condition is shown in figure 5b, which shows the links created for forward inference with the rule "If X flies and X is not feathered, X is a bat." We want to use this rule to infer that something is a bat only if we know both that it flies and that it is not feathered. Hence the excitatory link from **FLIES7** affects **BAT9** only if **FLIES7** has positive activation and **FEATHERED8** has negative activation. Similarly, the inhibitory link from **FEATHERED8** to **BAT9** operates just in case **FEATHERED8** has negative activation (yielding a positive effect on **BAT9**) and **FLIES7** has positive activation. Making the effects of links dependent on the activation of units does not contradict the principle of connectionism that computations should be local and parallel, since such dependencies arise only between units representing propositions occurring in the same rule; units representing those units can easily have local access to each other's activation values, just as do units that have excitatory and inhibitory links between them.

Insert Figure 5 about here.

The full set of possibilities for rules with various combinations of true and false conditions and inferences in accord with both *modus ponens* and *modus tollens* is too long to detail; CARE automatically creates the appropriate links for these cases. Two or more rules can affect the inference of a conclusion, either by all supporting it, by all attacking it, or by a mixture of support and attack. To take a famous example from the literature on nonmonotonic reasoning, consider the inference whether Dick is a pacifist given that he is both a Republican and a Quaker. The rules that Quakers generally are pacifists and that Republicans generally are not will tend to lead to different inferences. Figure 6 shows the network that CARE would create to deal with this case, with the unit representing Dick being a Quaker exciting the unit representing his being a pacifist, which is inhibited by the unit representing his being a Republican. Which conclusion CARE reaches in such cases depends on the comparative activation of the units that provide excitation and inhibition, as well as on the weights on the links, which can vary depending on how reliable the rules are known to be. If the rule that Quakers are pacificists is less reliable than the rule that Republicans are not, the excitatory link in figure 6 will be weaker than the inhibitory link, so pacificism will not be inferred. Thus the inference network in CARE performs many of the functions of a conventional AI truth-maintenance system while also allowing non-monotonic and probabilistic reasoning.

Insert Figure 6 about here.

The subgoaling section of the CARE inference network acts as a rule-based planning mechanism. In the same ways that inferences are made from the data and starting conditions of a problem, subgoals are made from the goals, using the information in the rules to determine possible solution paths. Activation is passed through this network starting at units representing the top-level goals of the system and filtering down to the lowest level subgoal units. These goal units interact in the same ways as other proposition units, with links connecting them created by the accomplishing and preventing subgoaling principles. Possible ''plans'' are evaluated by the settling of this network. Plans which would cause other goals to fail will be rejected because the other goal inhibits them. Viable options are selected because their acceptance does not conflict with other goals. Some goals generated by the planning system actually generate inferences. When the problem solver realizes that it has a goal to do something for which all of the preconditions are satisfied, it can act upon this. This prevents the system from taking action before a plan is fully formed.

6.3. Relaxing the Constraint Networks

Comparison and inference networks reach conclusions by updating activation until all units reach asymptotic activation levels. Settling is done incrementally, with each unit's activation being updated based on its links with other units. The change in activation passed along each link is dependent on the activation of the unit the link comes from (the *input unit*), the weight on the link, and, if the link is activation dependent, on whether the input unit and other relevant units are positive or negative. The algorithms used for updating activation are based on the ones proposed by Grossberg (1978). (We have modified them slightly because our activations are not always positive.) The activation level of unit j on cycle $t+1$ is given by:

$$a_j(t+1) = a_j(t)(1-\theta) + enet_j(max - a_j(t)) + inet_j(a_j(t) - min). \quad (1)$$

The inputs $enet_j$ (the net excitatory input) and $inet_j$ (the net inhibitory input, a negative number), are determined by the equations:

$$enet_j = \sum_i w_{ij} o_i(t) \text{ for } w_{ij} o_i > 0; \text{ and} \quad (2)$$

$$inet_j = \sum_i w_{ij} o_i(t) \text{ for } w_{ij} o_i < 0. \quad (3)$$

In each of these equations, $o_i(t)$ is the output of unit i on cycle t, set by:

$$o_i(t) = \max(a_i(t), 0). \quad (4)$$

The parameters *min* (normally -1) and *max* (normally +1) in equation (1) determine the minimum and maximum activation of any given unit. θ is a decay parameter that determines the rate at which the activation will decay to zero in the absence of external input. CARE's comparison network uses a decay value of .1, as was used in ACME and ARCS. But the decay parameter for the inference network is set at 0, since otherwise the system would cease to believe things or forget its goals for temporal reasons alone.

Within connectionism, CARE's inference network mechanism is similar to that proposed by Ballard (1986), who shows how parallel logical inference can be treated as an energy minimization problem. His method, however, is a connectionist version of theorem proving by resolution, in which the negation of a theorem is "resolved" with the knowledge base, and if a contradiction results, the theorem is considered proven. Our method is more akin to so called natural deduction theorem provers, which work from a set of premises and use known inferential relations to work forward to the theorem to be proven.

Our inference model is also similar in some ways to the work of Shastri (1988), who describes semantic classification by "is-a" hierarchies using localist networks with units that may be either "enabled" or "disabled". We perform the same function with our activation dependent links, but the types of reasoning we can do are not limited to questions of category membership. Shastri limited types of relationships considered to ensure that the system could operate in constant time. Our system takes more time for complex problems, but still settles in a reasonable number of cycles.

7. TEST CASES

We have developed several applications to test whether CARE successfully can perform rule-based and analogical reasoning. The first models Juliet's decision in Shakespeare's play to drink a special potion that will make others believe she has died. This case shows the ability of CARE to do rule-based reasoning in problem solving. The second application models the explanation by a leading Sovietologist of why Mikhail Gorbachev appointed Eduard Shevardnadze as foreign minister and shows that CARE can use rules to explain as well as to solve problems. The third case, modeling Duncker's familiar radiation problem, illustrates CARE's capacity for analogical reasoning, including retrieval and transfer.

7.1. Solving Juliet's Problem

In the fourth act of Shakespeare's *Romeo and Juliet*, Juliet has secretly married Romeo, but her father is pressing her to marry Paris. She must decide how to achieve several goals:

1) She does not want to commit the sin of bigamy.
2) She does not want her father to be angry with her.
3) She wants to be with Romeo.

Juliet knows that:

1) She is alive.
2) Her father wants her to marry Paris.
3) She has a potion that will make her stop breathing.

CARE models how Juliet uses this information and a set of rules to find a solution to her problem. English versions of the rules are presented in table 3.

Insert Table 3 about here.

CARE recognizes that to be with Romeo, Juliet cannot be dead. Meanwhile, it determines that she can prevent her father from getting angry at her, by appearing to be dead. There are two ways to appear dead: being dead (by killing oneself) or not breathing. Juliet knows she has a potion that will stop her breathing without killing her. Taking this route, she can convince her father that she is dead while not precluding being with Romeo. Now her father will not ask her to marry Paris, and so she will not become a sinner.

The networks created include 16 comparison units for the predicates that appear in the various rules, and 60 inference units representing 42 facts and 18 goals. The hierarchical structure of the goal network is shown in figure 7, in which the labels on the units are names for propositions whose full predicate calculus representation is given in table 4. Here some of the types of interacting subgoals can be seen. The units in the networks created by CARE have 458 links between them, of which 406 are activation dependent in the way described in section 6.2. Settling the networks completely requires 116 cycles. CARE concludes that by drinking the potion Juliet finds an adequate solution for her problem since her goals have been achieved.

Insert Figure 7 about here.

Insert Table 4 about here.

7.2. Explaining Gorbachev's Decision

In 1985, Mikhail Gorbachev replaced Andrei Gromyko with Eduard Shevardnadze as foreign minister. This was surprising, since Gromyko had held the post for thirty years, and Shevardnadze had no foreign policy experience. CARE has been used to model the explanation of Gorbachev's action by noted Sovietologist Jerry Hough (1990). According to Hough, Gorbachev wanted to replace Gromyko because of his resistance to change, but avoided firing him by promoting him to a ceremonial post. Shevardnadze's lack of foreign policy experience was actually attractive to Gorbachev, who knew that Shevardnadze was committed to reform, not to the foreign policy establishment.

CARE's model of Gorbachev's decision uses 21 rules describing Soviet politics, stating, for example, that a leader can make appointments and that only one person can occupy a position. CARE infers that because Gromyko has been in power for a long time, he opposes reform; but Gromyko's experience make it appropriate to promote him to chair the Presidium of the Supreme Soviet. CARE also infers that Shevardnadze was named the minister because he wanted reform and lacked experience. CARE used 107 units with 867 links in its inference network and derived the desired propositions within 100 cycles of activation, although the network did not completely settle until 564 cycles.

Our simulation of Gorbachev explained his action by applying rules in deductive fashion; it did not have to conjecture any hypotheses in order to perform the explanation. CARE has the capacity to generate explanatory hypotheses, both by chaining rules backward and by analogy, but this capacity has so far been tested only on very small cases.

7.3. The Duncker Radiation Problem

The radiation problem of Duncker (1945) has been widely used in psychological experiments (Gick and Holyoak 1980, 1983; Holyoak and Koh 1987). It concerns how to use an X-ray to destroy a tumor without harming the patient's tissue that surrounds it. Since beams of a high enough intensity to kill the tumor will also destroy the healthy tissue and kill the patient, simply shooting the X-ray beams at the tumor will not solve the problem. Instead, a convergence solution should be used: multiple beams weak enough not to harm the healthy tissue should be aimed to meet at the tumor where their combined strength will be great enough to destroy it. One analog to the problem, discussed by Holyoak and Koh (1987) concerns using a laser beam to fuse the broken filament in a lightbulb without destroying the bulb. A convergence solution of the tumor problem or the lightbulb problem provides a great aid to solving the other problem.

CARE is given a solution to the tumor problem and is presented with the lightbulb problem to solve. CARE's goal is to figure out how to use a laser to fuse the filament without breaking the glass. As in the tumor problem, simply aiming the laser at the filament will not work, since a beam strong enough to fuse the filament will break the glass. The predicate calculus representations of the laser problem and the tumor analog are given in tables 5 and 6.

Insert Tables 5 and 6 about here.

In order to solve the lightbulb problem, CARE retrieves and maps the tumor problem as an analog and transfers the convergence solution. First it creates a comparison network with 371 units and 10,187 links that places the elements of the stored tumor problem in correspondence with the elements of the lightbulb problem. In addition, an inference network with 25 units is created. 18 cycles of updating the activation of units determines the relevance of the tumor problem to the lightbulb problem, and the analogs are then mapped to each other. At cycle 91, the networks settle and the convergence solution is transferred from the tumor problem to the lightbulb problem. English versions of the new propositions created by transfer are shown in table 7. These propositions are added to the inference network to see whether they yield a solution, and after a total of 451 cycles of updating the network CARE finds that all goals of the tumor problem have been accomplished so that the problem is solved.

Insert Table 7 about here.

We will now explain the transfer process in greater detail. CARE's transfer mechanism operates on the source analog propositions which represent possible clues to solving the target problem. After a mapping is established between the source and target, CARE identifies propositions in the source that do not correspond to anything in the target. For example, the predicate **converge-on** in the solved tumor problem does not map to anything in the lightbulb problem. CARE does not, however, transfer all such information from the source to the target, since much of the source may be irrelevant to the target. Rather, it determines which unmapped elements are related to the system's goals, using the inference network. A proposition is goal related if any of its arguments map well to any of the arguments in the goals of the current problem. The predicate **converge-on** is goal related because its arguments representing the rays and the tumor map well to the beams and filament that are part of the statement of the goals of the lightbulb: we want the beams to fuse the filament. In this case, the goal relevance is immediately obvious, but in more complex cases CARE traces back causal chains of goal related propositions in the source, using the connectives **if, cause,** and **conjoin-event**. All goal-related source propositions without correspondences are transferred.

Transferring a proposition requires reconstituting it in a form appropriate for the target problem, using the mappings identified by the comparison network. To transfer the proposition from the tumor problem, (**converge-on (obj-ray obj-tumor)**), CARE needs to find a map for the predicate and each of the arguments. The comparison network has units with high activation representing maps between ray from the tumor problem and beam from the lightbulb problem, and between tumor and filament. The corresponding elements are therefore substituted. But since no unit representing a mapping for **converge-on** has high activation, the predicate is transferred as is, resulting in the new proposition (**converge-on (obj-beam obj-filament)**).

Once a new proposition is created, a unit representing it is added to the inference network, along with links resulting from **if, cause,** and **conjoin-event** statements in the source and target. Once in the inference network, propositions transferred from a source analog are indistinguishable from propositions generated by other mechanisms. Because new inferences are made whenever units are added to the inference network, the transferred solutions will be modified automatically (debugged) if there are rules or other analogs that recognize obvious problems. Transferred propositions may be partial at first, but further inference brought about by settling the expanded inference network can bring about a solution.

The transfer mechanism in CARE is a version of what Holyoak, Novick and Melz (this volume) call copying with substitution and generation (CWSG), although CARE differs from their extension of ACME in that CARE's use of a goal hierarchy (derived from rules) enables it to select elements for transfer on pragmatic grounds. This constrains transfer in a more natural way, so that the source analog need not be cut up into compartments delineating the starting conditions, goal, and solution fields. In most cases of analogical problem solving, particularly across domains, it seems unlikely that these categories will be known in advance. In fact, it is quite conceivable that what acts as the solution for one problem might be the analog of the starting conditions for another.

Some of the limitations of ACME described by Hofstadter and Mitchell (this volume) have been addressed by CARE. Like their Copycat model, CARE's problem descriptions are not static structures, but can be modified and extended by the actions of a variety of inference rules. Unlike Copycat, however, CARE also addresses the question of analog retrieval and how it may be integrated with analogical mapping. Copycat does not model retrieval, instead focusing strongly on the rerepresentation by perceptual mechanisms of a source and target analog that are provided. Since these perceptual mechanisms cannot operate until after a potential analog has been retrieved, it is not clear from Copycat how retrieval and mapping might interact. CARE reflects psychological evidence that mapping and retrieval do interact and that the same pressures that affect mapping also affect retrieval.

8. CONCLUSION

In sum, CARE goes beyond our previous connectionist work on analogy in important ways. It shows how the retrieval and mapping of analogs can occur in the context of a rule-based problem solver. Most importantly, it shows that rule-based reasoning can be understood in terms of parallel constraint satisfaction implemented using a novel kind of localist connectionist network. CARE also extends our previous models in that it does pragmatically guided analogical transfer as well as mapping and retrieval.

Nevertheless, the integration of analogy and rule-based reasoning accomplished in CARE could be taken farther. We conjecture that matching of parts of rules against problem descriptions could be performed by mechanisms like those used for matching of analogs in the comparison network. Then determination of what rules to fire could be performed simultaneously with analog retrieval instead of by the traditional AI unification method that CARE currently uses. In addition, we would like explanatory hypotheses that are formed by CARE using rules and analogs to be evaluated for their coherence as is done by the program ECHO. Such evaluation would require integration or adaptation of the networks used by ECHO so that they could become part of or enhance CARE's inference network. Finally, since the simulation examples that we have used so far in CARE's development are quite small, larger data bases should be constructed to provide more stringent tests of CARE's ability to do rule-based and analogical reasoning using parallel constraint satisfaction using connectionist networks.

REFERENCES

Ajjanagadde, V., and Shastri L, (1989). Efficient inference with multi-place predicates and variables in a connectionist system. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum, 396-403.

Anderson, J., and Thompson, R. (1989). Use of analogy in a production system architecture. In S. Vosniadou and A. Ortony (Eds.), *Similarity and analogical reasoning*. Cambridge: Cambridge University Press. 267-297.

Ballard, D. (1986). Parallel logical inference and energy minimization. *Proceedings of the Fifth National Conference on Artificial Intelligence*. Cambridge, MA: MIT Press, 840-847.

Barnden, J. (1991). Encoding complex symbolic data structures with some unusual connectionist techniques. In J. Barnden and J. Pollack (Eds.), *Advances in connectionist and neural computation theory*, Vol. 1: *High level connectionist models*. Norwood, NJ: Ablex, 180-240.

Buchanan, B., and Shortliffe, E. (Eds.), (1984). *Rule-based expert systems*. Reading, MA: Addison Wesley.

Duncker, K. (1945). On problem solving. *Psychological Monographs*, 58(No. 270).

Eskridge, T. (this volume). A hybrid model of analogical reasoning.

Forgy, C. (1982). RETE: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence, 19*, 17-37.

Gick, M. L., and Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology, 12*, 306- 355.

Gick, M. L., and Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology, 15*, 1-38.

Grossberg, S. (1978). A theory of visual coding, memory, and development. In E. Leeuwenberg and J. Buffart (Eds.), *Formal theories of visual perception*. New York: Wiley.

Harman, G. (1973). *Thought*. Princeton: Princeton University Press.

Hendler, J. (1991). Developing hybrid symbolic/connectionist models. In J. Barnden and J. Pollack (Eds.), *Advances in connectionist and neural computation theory*, Vol. 1: *High level connectionist models*. Norwood, NJ: Ablex, 165-179.

Holyoak, K., and Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory and Cognition*, *15*, 332-340.

Holyoak, K., Novick, L., and Melz, E., (this volume). Component processes in analogical transfer: Mapping, pattern completion, and adaptation.

Holyoak, K. and Thagard, P. (1989a) Analogical mapping by constraint satisfaction. *Cognitive Science, 13,* 295-355.

Holyoak, K., and Thagard, P. (1989b). A computational model of analogical problem solving. In S. Vosniadou and A. Ortony (Eds.), *Similarity and analogical reasoning*. Cambridge: Cambridge University Press. 242-266.

Hough, J. (1990). *Russia and the west: Gorbachev and the politics of reform.* 2nd edition. New York: Simon & Schuster.

Jones, R. and Langley, P. (1991). An integrated model of retrieval and problem solving. Unpublished manuscript, University of Pittsburgh, Department of Computer Science.

Kokinov, B. (this volume). A hybrid model of reasoning by analogy.

Koton, P. (1988). Reasoning about evidence in causal explanations. *Proceedings of the Seventh National Conference on Artificial Intelligence*. Cambridge, MA: MIT Press, 256-261.

Lehnert, W. (1991). Symbolic/subsymbolic sentence analysis: Exploiting the best of both worlds. In J. Barnden and J. Pollack (Eds.), *Advances in connectionist and neural computation theory,* Vol. 1: *High level connectionist models*. Norwood, NJ: Ablex, 135-164.

Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence, 46*, 77-105.

Rissland, E., and Skalak, D. (1989). Combining case-based and rule-based reasoning: A heuristic approach. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. San Mateo: Morgan Kaufmann: 524-530.

Shastri, L. (1988). A connectionist approach to knowledge representation and limited inference. *Cognitive Science, 12*, 331-392.

Skalak, D., and Rissland, E. (1990). Inductive learning in a mixed paradigm setting. *Proceedings of the Eighth National Conference on Artificial Intelligence*. Cambridge, MA: MIT Press, 840-847.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence, 46*, 159-217.

Thagard, P. (1988). *Computational philosophy of science*. Cambridge, MA: MIT Press/Bradford Books.

Thagard, P. (1989). Explanatory coherence. *Behavioral and Brain Sciences, 12*, 435-467.

Thagard, P. (1992). *Conceptual revolutions*. Princeton University Press, in press.

Thagard, P., Cohen, D., and Holyoak, K., (1989) Chemical analogies: Two kinds of explanation. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. San Mateo: Morgan Kaufmann, 819-824.

Thagard, P., Holyoak, K., Nelson, G., and Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46*, 259-310.

Touretzky, D., and Hinton, G. (1988) A distributed production system. *Cognitive Science, 12*, 423-466.

van Gelder, T. (1990). Compositionality: a connectionist variation on a classical theme. *Cognitive Science, 14*, 355-384.

Figure Captions

Figure 1. Relations among analogy, rule-based reasoning, and hypothesis formation.

Figure 2. Component of the inference network created using the rule that trees are tall to license an inference in accord with *modus ponens*. The arrow indicates a unidirectional excitatory link. The plus sign signifies that the unit TREE1 has an effect only if it has positive activation.

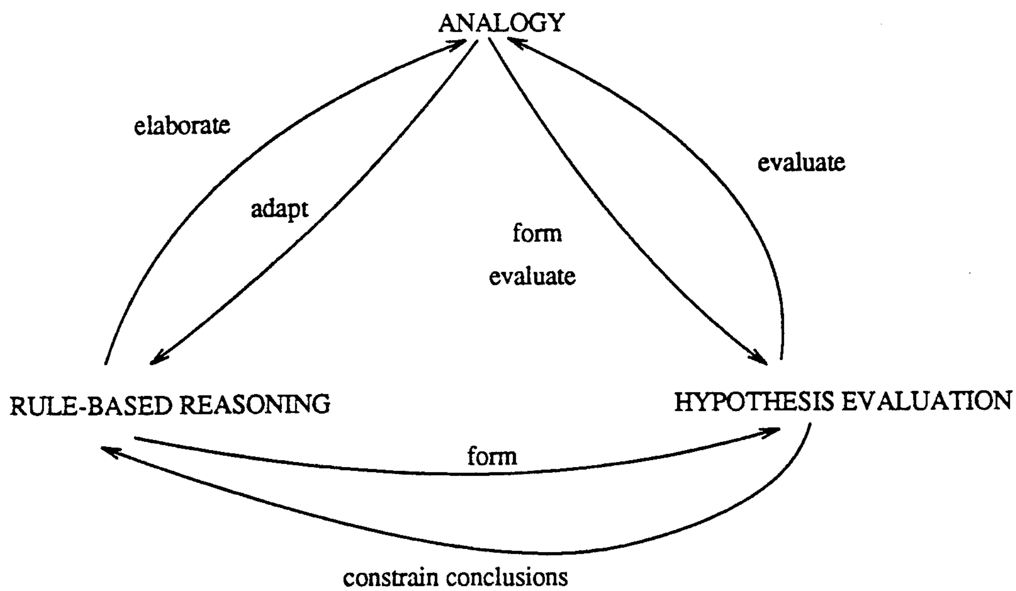
Figure 3. Inference network enhanced to license an inference in accord with *modus tollens*. The minus sign signifies that the unit TALL2 has an effect only if it has negative activation.

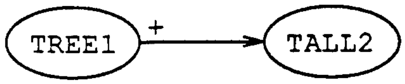
Figure 4. Implementation of the rule that if something is not tall then it is short. The dotted line indicates an inhibitory link. The minus sign signifies that the unit TALL2 has an effect only if it has negative activation.

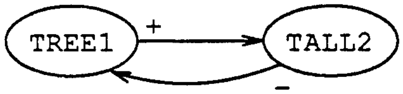
Figure 5. Rules with multiple conditions. See text for explanation.

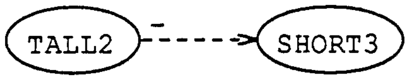
Figure 6. Inference network for inferring whether Nixon is a pacifist. The plus signs signifies that the marked units have an effect only if they have positive activation. The solid line indicates and excitatory link, while the dotted line indicates an inhibitory link.

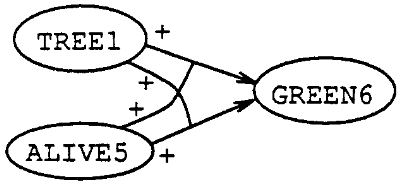
Figure 7. The hierarchical goal structure of Juliet's problem. The top level goals are those connected to the PRAGMATIC unit, which keeps these active. The goals generated by the system are ones further down the tree. The links show various goals interactions: incompatible plans of action, ways of accomplishing goals, and ways to prevent undesirable things from happening.


ANALOGY
elaborate
adapt
form
evaluate
evaluate
RULE-BASED REASONING
HYPOTHESIS EVALUATION
form
constrain conclusions

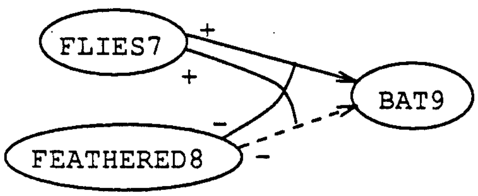

TREE1
+
TALL2


TREE1
+
TALL2
-

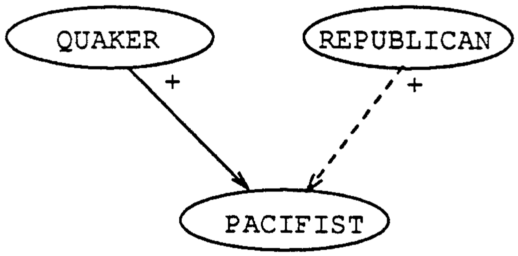

TALL2
-
SHORT3



(a)



(b)


QUAKER
REPUBLICAN
+
+
PACIFIST

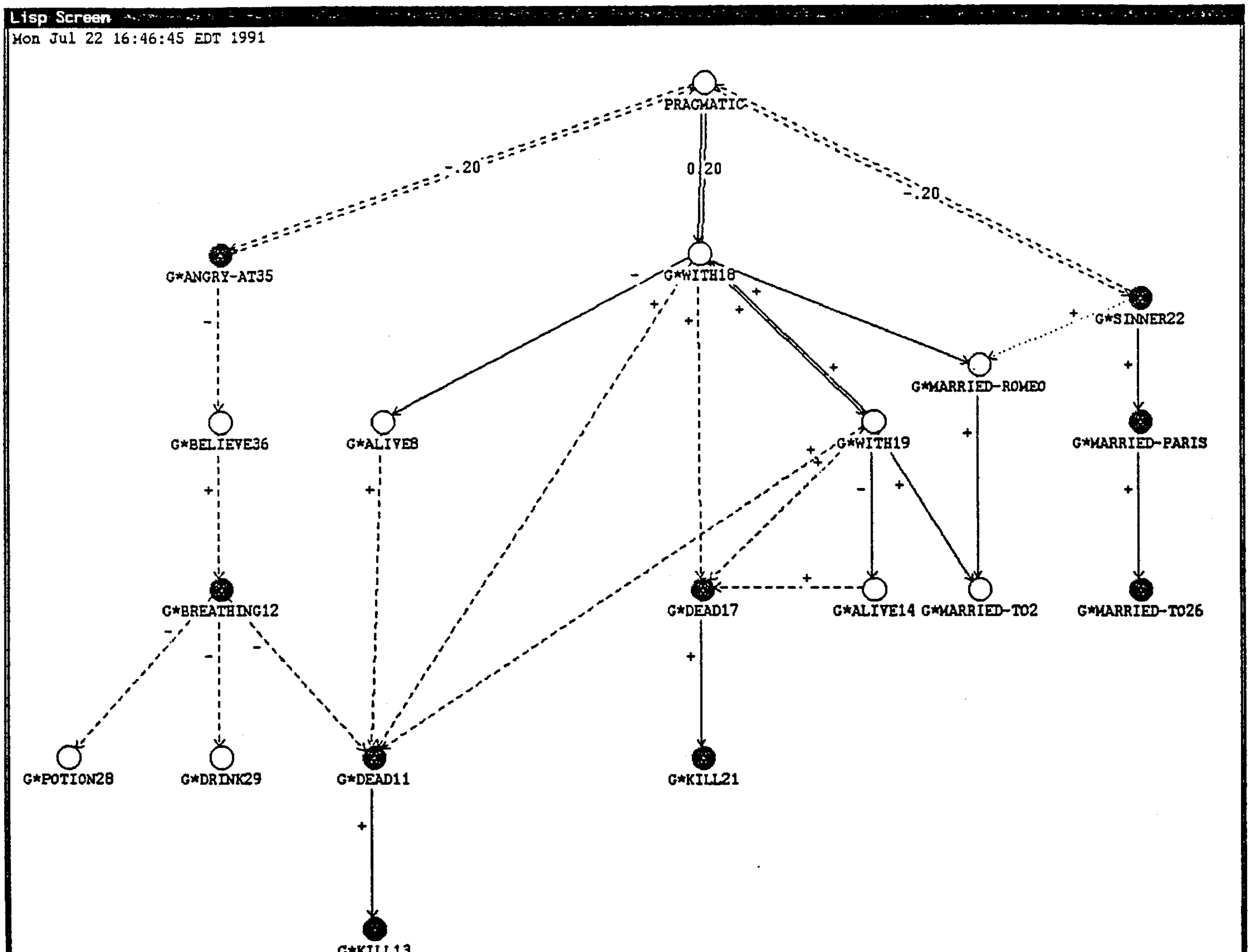

Lisp Screen
Mon Jul 22 16:46:45 EDT 1991
PRAGMATIC
-.20
0
.20
-.20
G*ANGRY-AT35
G*WITH18
G*SINNER22
G*MARRIED-ROMEO
G*BELIEVE36
G*ALIVE8
G*WITH19
G*MARRIED-PARIS
G*BREATHING12
G*DEAD17
G*ALIVE14
G*MARRIED-TO2
G*MARRIED-TO26
G*POTION28
G*DRINK29
G*DEAD11
G*KILL21
G*KILL13

Table 1: A simple rule

```
(mrule 'trees-are-tall 'causal
       '((tree (?x) true is-a-tree))
       '((tall (?x) true then-is-tall))
)
```

Table 2: Arthur's rules

```
(make-problem 'arthurs-problem
        '((liquid (obj-potion) true)
          (love-potion (obj-potion) true)
          (magician (merlin) true))
        '((have (merlin obj-potion) true)
          (have (arthur obj-potion) false)
          (drink (arthur obj-potion) unknown))
        '((drink (arthur obj-potion) true))
)

(mrule 'have-liquid-can-drink 'causal
        '((have (?x ?y) true have-c1)
          (liquid (?y) true have-c2)
          (drink (?x ?y) unknown %have-c3)
          (desire (?x (%have-c3 true)) true have-c4))
        '((drink (?x ?y) true %have-c3))
)

(mrule 'dont-have-liquid-cant-drink 'causal
        '((have (?x ?y) false dont-have-c1)
          (liquid (?y) true dont-have-c2)
          (drink (?x ?y) unknown %dont-have-c3)
          (desire (?x (%dont-have-c3 true)) true dont-have-c4))
        '((drink (?x ?y) false %dont-have-c3))
)
```

Table 3: Rules for Juliet's Dilemma

Semantic rules:
- If you are dead, you are not alive.
- If you are not dead, you are alive.
- If x is with y, y is with x.
- If x is married to y, y is married to x.
- A potion is a liquid.

General rules:
- If x is capable of performing an action that x wants, x will perform the action.
- If x kills x, x is dead.
- If x is dead, x doesn't breathe.
- If y believes x is not breathing, y believes x is dead.
- If y believes x is dead, y believes x can't do anything.
- If y believes x is dead, y will not be angry at x.
- If x is not alive, x can't be with anyone else.
- If x has a liquid, x is both capable of drinking it, and capable of not drinking it.
- If x is alive and x and y are people, and x and y are not already married and x is not equal to y, x is both capable of marrying y, and capable of not marrying y.
- If x and y are married and neither is dead, they are with each other.
- If you are married to two different people, you are a sinner.
- If x drinks the potion, x will stop breathing.

Table 4: Conclusions in Juliet's Dilemma

Mon Jul 22 17:11:12 EDT 1991

Description	Proposition name	Confidence
(MARRIED-TO (JULIET ROMEO))	MARRIED-ROMEO	0.99000
(MARRIED-TO (ROMEO JULIET))	MARRIED-TO2	0.98413
(FATHER-OF (CAPULET JULIET))	FATHER-OF3	0.99000
(PERSON (JULIET))	PERSON4	0.99000
(PERSON (ROMEO))	PERSON5	0.99000
(PERSON (PARIS))	PERSON6	0.99000
(PERSON (CAPULET))	PERSON7	0.99000
(ALIVE (JULIET))	ALIVE8	0.99000
(CAN-ACHIEVE (JULIET MARRIED-ROMEO+))	CAN-ACHIEVE9	0.00000
(CAN-ACHIEVE (JULIET MARRIED-ROMEO-))	CAN-ACHIEVE10	0.00000
(DEAD (JULIET))	DEAD11	-0.98413
(BREATHING (JULIET))	BREATHING12	-0.98982
(KILL (JULIET JULIET))	KILL13	-0.97353
(ALIVE (ROMEO))	ALIVE14	0.99000
(CAN-ACHIEVE (ROMEO MARRIED-TO2+))	CAN-ACHIEVE15	0.00000
(CAN-ACHIEVE (ROMEO MARRIED-TO2-))	CAN-ACHIEVE16	0.00000
(DEAD (ROMEO))	DEAD17	-0.98413
(WITH (JULIET ROMEO))	WITH18	0.99000
(WITH (ROMEO JULIET))	WITH19	0.99000
(BREATHING (ROMEO))	BREATHING20	0.00000
(KILL (ROMEO ROMEO))	KILL21	-0.97353
(SINNER (JULIET))	SINNER22	-0.25000
(MARRIED-TO (JULIET PARIS))	MARRIED-PARIS	-0.99000
(CAN-ACHIEVE (JULIET MARRIED-PARIS+))	CAN-ACHIEVE24	0.99000
(CAN-ACHIEVE (JULIET MARRIED-PARIS-))	CAN-ACHIEVE25	0.99000
(MARRIED-TO (PARIS JULIET))	MARRIED-TO26	-0.98413
(DESIRE (CAPULET MARRIED-PARIS+))	DESIRE27	0.99000
(POTION (POTION[PROBLEM-JULIETS-DILEMMA]))	POTION28	0.99000
(DRINK (JULIET POTION[PROBLEM-JULIETS-DILEMMA]))	DRINK29	0.98271
(DRINK (ROMEO POTION[PROBLEM-JULIETS-DILEMMA]))	DRINK30	0.00000
(LIQUID (POTION[PROBLEM-JULIETS-DILEMMA]))	LIQUID31	0.98413
(HAVE (JULIET POTION[PROBLEM-JULIETS-DILEMMA]))	HAVE32	0.99000
(CAN-ACHIEVE (JULIET DRINK29+))	CAN-ACHIEVE33	0.98992
(CAN-ACHIEVE (JULIET DRINK29-))	CAN-ACHIEVE34	0.98992
(ANGRY-AT (CAPULET JULIET))	ANGRY-AT35	-0.95077
(BELIEVE (CAPULET DEAD11+))	BELIEVE36	0.97349
(BELIEVE (CAPULET CAN-ACHIEVE10-))	BELIEVE37	0.95077
(BELIEVE (CAPULET CAN-ACHIEVE9-))	BELIEVE38	0.95077
(BELIEVE (CAPULET CAN-ACHIEVE25-))	BELIEVE39	0.95077
(BELIEVE (CAPULET CAN-ACHIEVE24-))	BELIEVE40	0.95077
(BELIEVE (CAPULET CAN-ACHIEVE34-))	BELIEVE41	0.95077
(BELIEVE (CAPULET CAN-ACHIEVE33-))	BELIEVE42	0.95077

Table 5: Representation of Laser Problem

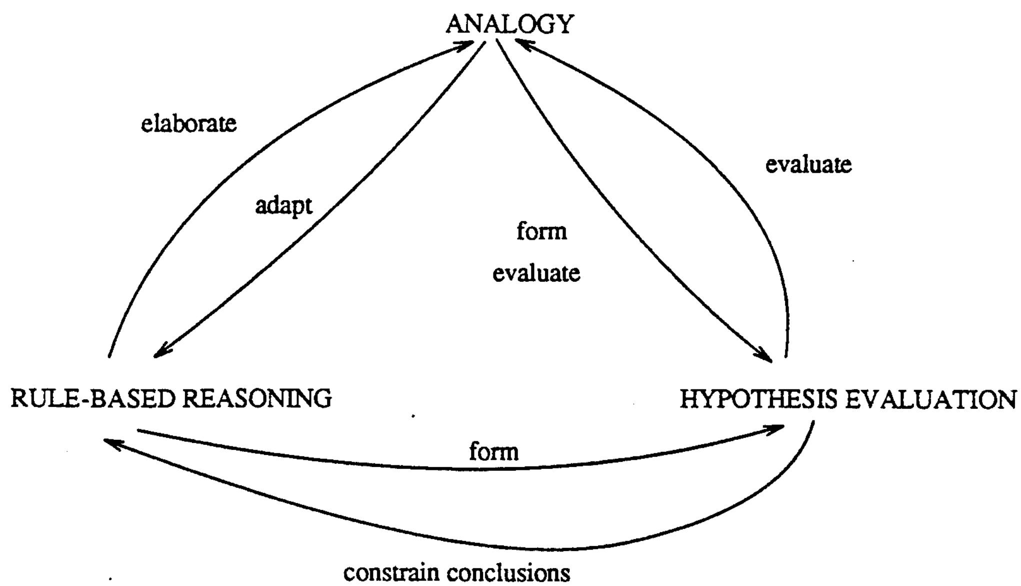
```
(make-problem 'laser-fragile
  '()
  '((filament (obj-filament) true lf-1)
    (broken (obj-filament) true lf-2)
    (glass (obj-glass) true lf-3)
    (useless (obj-bulb) true lf-s1)
    (have (obj-bulb obj-filament) true lf-s2)
    (conjoin-event ((lf-2 true) (lf-s2 true)) true lf-s3)
    (cause ((lf-s3 true) (lf-s1 true)) true lf-s4)
    (surround (obj-glass obj-filament) true lf-4)
    (beam (obj-beam) true lf-5)
    (laser (obj-laser) true lf-6)
    (produce (obj-laser obj-beam) true lf-7)
    (strength (obj-beam obj-beam-strength) true tp-s6)
    (variable (obj-beam-strength) true tp-s7)
    (occurs-at ((lf-8 true) obj-filament) true lf-s8)
    (occurs-at ((lf-9 true) obj-glass) true lf-s9)
    (cause ((lf-8 true) (lf-2 false)) true lf-s10)
    (cause ((lf-s3 false) (lf-s1 false)) true lf-s11)
    (strong (obj-beam) unknown lf-s12)
    (cause ((lf-s12 true) (lf-8 true)) true lf-s13)
    (cause ((lf-s12 true) (lf-9 true)) true lf-s13)
   )
   '((fuse (obj-beam obj-filament) true lf-8)
     (break (obj-beam obj-glass) false lf-9)
     (useless (obj-bulb) false lf-s5)
    )
)
```

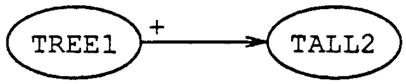
Table 6: Representation of the Tumor Problem Analog

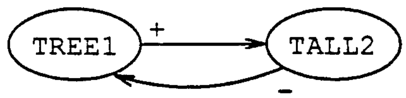
```
(manalog 'tumor
        '(start ((tumor (obj-tumor) true tp-1)
                 (malignant (obj-tumor) true tp-2)
                 (tissue (obj-tissue) true tp-3)
                 (ill (obj-patient) true tp-s1)
                 (have (obj-patient obj-tumor) true tp-s2)
                 (conjoin-event ((tp-2 true) (tp-s2 true)) true tp-s3)
                 (cause ((tp-s3 true) (tp-s1 true)) true tp-s4)
                 (surround (obj-tissue obj-tumor) true tp-4)
                 (ray (obj-ray) true tp-5)
                 (ray-source (obj-ray-source) true tp-6)
                 (produce (obj-ray-source obj-ray) true tp-7)
                 (strength (obj-ray obj-ray-strength) true tp-s6)
                 (variable (obj-ray-strength) true tp-s7)
                 (strong (obj-ray) unknown tp-s12)
                 (cause ((tp-s12 true) (tp-8 true)) true tp-s13)
                 (cause ((tp-s12 true) (tp-9 true)) true tp-s14)
                 (converge-on (obj-ray obj-tumor) true tp-sol1)
                 (weak (obj-ray) true tp-sol2)
                 (cause ((tp-sol1 true) (tp-8 true)) true tp-sol3)
                 (cause ((tp-sol2 true) (tp-9 false)) true tp-sol4)
                 (occurs-at ((tp-8 true) obj-tumor) true tp-s8)
                 (occurs-at ((tp-9 true) obj-tissue) true tp-s9)
                 (cause ((tp-8 true) (tp-2 false)) true tp-s10)
                 (cause ((tp-s3 false) (tp-s1 false)) true tp-s11)
                )
         )
        '(goal ((destroy (obj-ray obj-tumor) true tp-8)
                (destroy (obj-ray obj-tissue) false tp-9)
                (ill (obj-patient) false tp-s5)
               )
         )
)
```

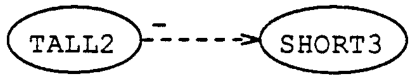
Table 7: Propositions Transferred from Tumor to Laser Problem

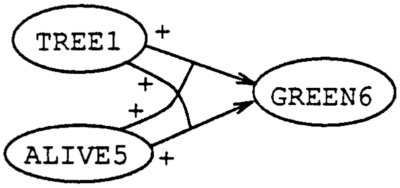
The beams are weak.
The beams converge on the tumor.
Because the beams converge, they fuse the filament.
Because the beams are weak, they do not break the glass.


ANALOGY
elaborate
adapt
form
evaluate
evaluate
RULE-BASED REASONING
HYPOTHESIS EVALUATION
form
constrain conclusions

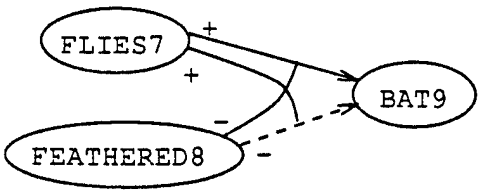

TREE1
+
TALL2


TREE1
+
TALL2
-

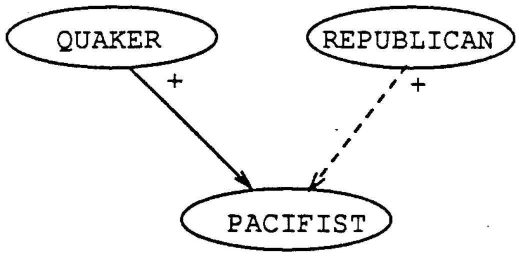

TALL2
-
SHORT3

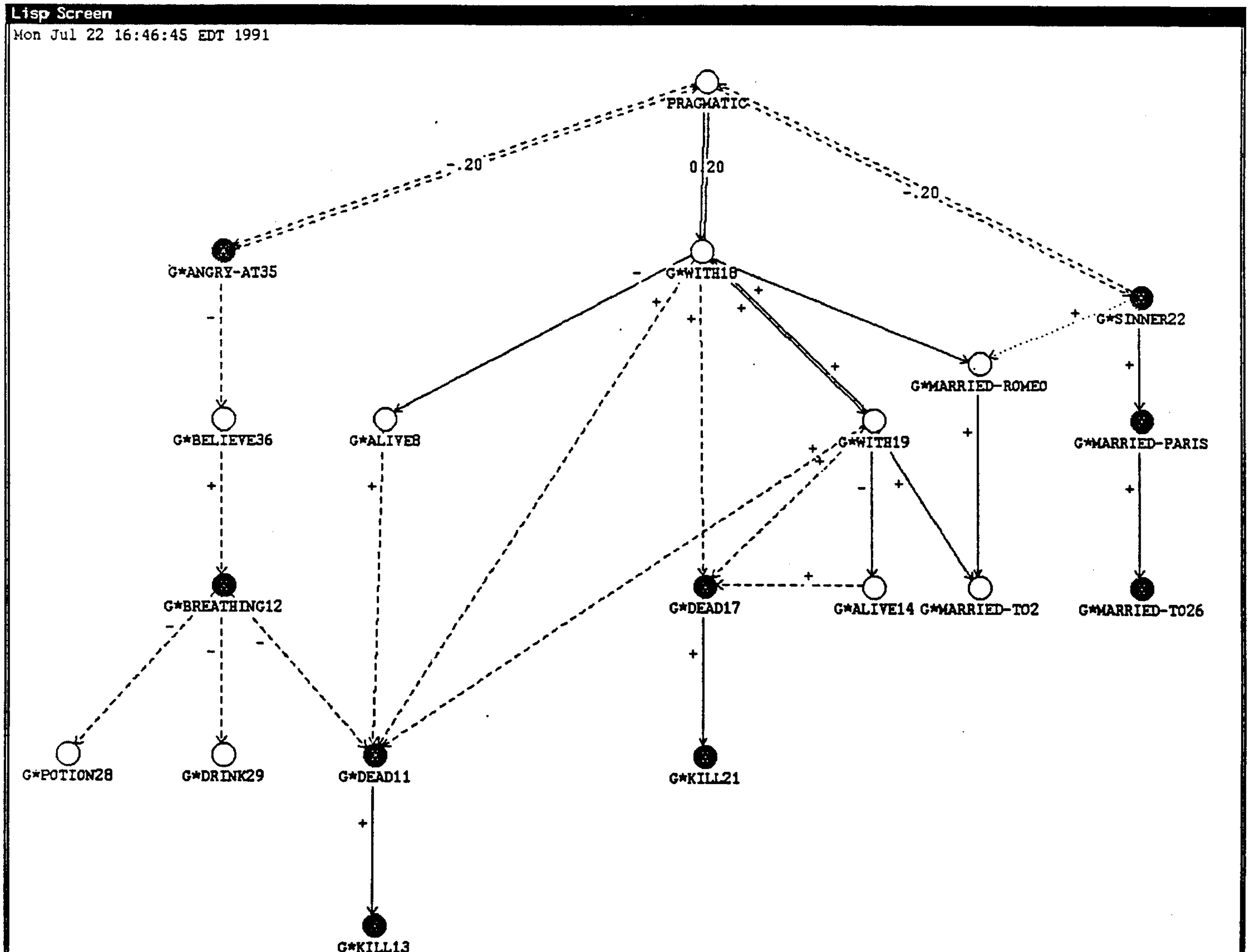


(a)



(b)


QUAKER
REPUBLICAN
+
+
PACIFIST


Lisp Screen
Mon Jul 22 16:46:45 EDT 1991
PRAGMATIC
-.20
0.20
-.20
G*ANGRY-AT35
G*WITH18
G*SINNER22
G*MARRIED-ROMEO
G*BELIEVE36
G*ALIVE8
G*WITH19
G*MARRIED-PARIS
G*BREATHING12
G*DEAD17
G*ALIVE14
G*MARRIED-TO2
G*MARRIED-TO26
G*POTION28
G*DRINK29
G*DEAD11
G*KILL21
G*KILL13