A First-Order Horn-Clause Abductive System and Its Use in Plan Recognition and Diagnosis

Hwee Tou Ng
Raymond J. Mooney
Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712
htng@cs.utexas.edu, mooney@cs.utexas.edu

June 21, 1992

Abstract

A diverse set of intelligent activities, including natural language understanding and diagnosis, requires the ability to construct explanations for observed phenomena. In this paper, we view explanation as *abduction*, where an abductive explanation is a consistent set of assumptions which, together with background knowledge, logically entails a set of observations. We have successfully built a domain-independent system, ACCEL, in which knowledge about a variety of domains is uniformly encoded in first-order Horn-clause axioms. A general-purpose abduction algorithm, AAA, efficiently constructs explanations in the various domains by caching partial explanations to avoid redundant work. Empirical results show that caching of partial explanations can achieve more than an order of magnitude speedup in run time. We have applied our abductive system to two general tasks: plan recognition in text understanding, and diagnosis of medical diseases, logic circuits, and dynamic systems. The results indicate that ACCEL is a general-purpose system capable of plan recognition and diagnosis, yet efficient enough to be of practical utility.

Keywords: abduction; plan recognition; diagnosis.

1 Introduction

Finding explanations for events and actions is an important aspect of general intelligent behavior. A diverse set of intelligent activities, including natural language understanding, diagnosis, scientific theory formation, and image interpretation, requires the ability to construct explanations for the phenomena observed. For instance, in text understanding, a reader infers the high-level goals and plans of the characters in a text in order to explain the events and actions described in the text. In dialog understanding, a participant infers the goals and plans of other participants based on the utterances exchanged in a conversation. This kind of inference is known as *plan recognition*, and it is an important component of text and dialog understanding [3].

Similarly, in medical diagnosis, based on the observed symptoms of a patient, a physician infers the possible diseases that may explain the symptoms. In physical device diagnosis, based on the observed misbehavior of a physical device, a diagnostician infers the possible faults that may explain the misbehavior. In image interpretation, based on a two-dimensional image, a vision system infers the objects present in the scene that may explain the image.

In this paper, we view explanation as *abduction*. The philosopher C. S. Peirce [54] defined *abduction* as the process of finding the best explanation for a set of observations; i.e. inferring cause from effect. The standard logical formalization of abduction within artificial intelligence (AI) defines an abductive explanation as a consistent set of assumptions which, together with background knowledge, logically entails a set of observations [10]. Abduction has been proposed as a unifying formalism for explanation in a variety of tasks including natural language understanding, diagnosis, scientific theory formation, and image interpretation [10].

Formulating the generation of explanatory hypotheses as abduction has some advantages over the traditional "expert system" approach. In an expert system, heuristic rules of the form $e \rightarrow h$ are used to encode the fact that some evidence e may suggest hypothesis h. A separate inference engine deduces the set of possible hypotheses that are "implied" by the evidence. Conflict resolution strategies are employed to decide which rules should be fired first and hence to determine the most plausible hypotheses. Such an approach requires reversing the causal links between hypothesis and evidence. Also, control information about what to deduce is mixed with declarative knowledge about the relationship between hypothesis and evidence. This is in contrast to abduction, which encodes the relevant knowledge in its most natural form as "hypothesis $h \rightarrow$ evidence e". Abduction also relies on a separate evaluation criterion such as simplicity to determine which of the candidate hypotheses best explain some evidence. Abduction is therefore a more natural and declarative approach to modeling the generation of explanatory hypotheses.

While it has been realized for quite some time within AI that abduction is a general model for explanation, there have been no empirical explorations into the practical feasibility of such a general abductive approach to explanation. Many important questions remain unexplored. For example, is it possible to have a general-purpose yet efficient algorithm that can be used for making useful abductive inference in all the various domains? Do we need special-purpose control heuristics separately tailored for each domain? Do the criteria

for selecting the best explanations vary according to the domain? How difficult is it to encode the knowledge necessary for constructing explanations in the various domains?

To address these important issues, we have successfully built a domain-independent system called ACCEL (Abductive Construction of Causal Explanations in Logic). In our system, knowledge about a variety of domains is uniformly encoded in first-order Horn-clause axioms. A general-purpose abduction algorithm, AAA (ATMS-based Abduction Algorithm), efficiently constructs explanations in these domains. We have applied our abductive system to two general tasks: plan recognition in text understanding, and diagnosis of medical diseases, logic circuits, and dynamic systems. We believe our approach represents a good trade-off between generality and efficiency — ACCEL is a general-purpose system capable of performing all of the above tasks, yet efficient enough to be of practical utility. In this paper, we will present extensive empirical results demonstrating the efficacy and efficiency of our system in performing the above tasks.

Previous abduction algorithms and systems, when compared to ACCEL, are either too restrictive, too inefficient, or both. Although the ATMS algorithm of [16] has been proven to be a general abduction algorithm for propositional Horn-clause theories [39], many interesting abduction tasks require the expressibility of first-order predicate logic. For example, the tasks of plan recognition in narrative texts, as well as abductive diagnosis of logic circuits and continuous dynamic systems, require that the domain theory be expressed in first-order predicate logic. Furthermore, in first-order logic, the important operation of unifying assumptions (factoring) becomes relevant. Frequently, simple and coherent explanations can only be constructed by unifying initially distinct assumptions so that the resulting combined assumption explains several observations [60; 66]. This important problem does not arise in the propositional case.

On the other hand, the general-purpose first-order abduction algorithm proposed in [66] tends to perform a great deal of redundant work in that partial explanations are not cached and shared among multiple explanations. The ATMS algorithm, though it caches and reuses partial explanations in order to avoid redundant work, has not been extended to perform general first-order abduction. Also, the ATMS algorithm exhaustively computes all minimal explanations, which is computationally very expensive for large problems. Even in the propositional case, computing all minimal explanations is a provably exponential problem [40; 65]. This indicates that resorting to heuristic search to find the best explanations is the most reasonable approach to building a practical abductive system.

Another important issue in abduction concerns the evaluation of the quality of explanations, i.e., what are the distinguishing features of a good explanation, and how can evaluation metrics be formulated so as to select and keep only the good explanations among the exponentially large number of explanations. Simplicity of explanations, defined as making the least number of assumptions in an abductive explanation, is a widely used metric to select the best explanations [61; 5; 32]. In Section 3, we will give convincing evidence that simplicity is inadequate as an evaluation metric for explanations in text understanding.

Our algorithm AAA overcomes both the generality and efficiency problems in that it is an abduction algorithm for first-order Horn-clauses, and it uses ATMS-style caching to avoid redundant work. In Section 6, we will present empirical results which demonstrate

that caching of partial explanations can achieve more than an order of magnitude speedup in run time. The AAA algorithm also incorporates a form of heuristic beam search in order to limit the computational efforts expended in finding the best explanations. Explanations are ranked according to their evaluation metric values and only the best explanations within the beam width of the beam search algorithm are kept.

Although ACCEL provides a more declarative approach to the generation of explanatory hypotheses than a traditional expert system, it is often necessary that axioms be formulated carefully so that the system will perform the desired task correctly and efficiently. As in traditional logic programming, it is frequently insufficient to just "state the correct knowledge" and expect the desired answers to be inferred. Appropriate programming methodologies must be developed so that a user knows how to axiomatize a problem to correctly and efficiently compute the desired answers [58]. This is also true in "abductive logic programming". By successfully applying ACCEL to the tasks of plan recognition and diagnosis, we have demonstrated via many examples *how* a general abductive system can be used to achieve these tasks.

We now give a brief overview of the various domains on which ACCEL has been tested:

1. Plan recognition: We define a novel evaluation criterion, called *explanatory coherence*, and give empirical results demonstrating that coherence is a better evaluation metric than simplicity in plan recognition. We also give supporting evidence that our system is sufficiently general to be able to handle similar plan recognition problems not known to the system developer in advance.

2. Set covering diagnosis: We prove that, given the appropriate form of axioms, ACCEL computes the same diagnoses as those of the set-covering method of Reggia [61; 55]. We also present empirical results demonstrating the efficiency of ACCEL at diagnosing 50 real-world patient cases using a sizable knowledge base with over six hundred rules.

3. Model-based diagnosis: We use abduction to perform model-based diagnosis, which concerns inferring faults from first principles given knowledge about the correct structure and behavior of a system. The approach is applied to diagnosing logic circuits (a full adder) and dynamic systems (a proportional temperature controller and the water balance system of the human kidney). Empirical results are presented illustrating the capability of ACCEL in abductive diagnosis.

The rest of this paper is organized as follows. Section 2 gives a formal definition of the abduction problem that we are addressing, and describes the AAA algorithm. Section 3 concerns abduction in the plan recognition domain. Section 4 concerns the use of general abduction to achieve diagnosis based on the set covering method. Section 5 presents ACCEL's abductive approach to model-based diagnosis. Section 6 presents empirical results on the speedup obtained through the use of caching in the AAA algorithm. Section 7 presents related work. Section 8 discusses future work. Section 9 gives the conclusion.

2 Problem Definition and Algorithms

2.1 Problem Definition

The abduction problem that we are addressing can be defined as follows.
Given:

- A set of universally quantified first-order Horn-clause axioms T (the domain theory), where an axiom is either of the form
  $C(v_1, \ldots, v_k) \leftarrow P_1(v_1, \ldots, v_k) \wedge \ldots \wedge P_r(v_1, \ldots, v_k)$ (a rule), or
  $F(v_1, \ldots, v_k)$ (a fact)
- An existentially quantified conjunction O of atoms (the input atoms) of the form
  $\exists v_1, \ldots, v_k \; O_1(v_1, \ldots, v_k) \wedge \ldots \wedge O_m(v_1, \ldots, v_k)$

Find:
All *explanations* with *minimal* (w.r.t. *variant-subset*) sets of assumptions.
We define *explanation*, *variant-subset*, and *minimality* as follows.

Definition 2.1 *Let A (the assumptions) be an existentially quantified conjunction of atoms of the form*

$$\exists v_1, \ldots, v_k \; A_1(v_1, \ldots, v_k) \wedge \ldots \wedge A_n(v_1, \ldots, v_k)$$

where $n \geq 0, A \cup T \models O$, and $A \cup T$ is consistent. An assumption set A (together with its corresponding proof) is referred to as an explanation *(or an* abductive proof*) of the input atoms.*

We will write A as the set $\{A_1, \ldots, A_n\}$ with the understanding that all variables in the set are existentially quantified and that the set denotes a conjunction.

Definition 2.2 *An assumption set A is a* variant-subset *of another assumption set B if there is a renaming substitution σ such that $A\sigma \subseteq B$.*

For example, $A = \{p(X, b)\}$ is a variant-subset of $B = \{p(Y, b), q(Y)\}$ with the renaming substitution $\sigma = \{X/Y\}$.[1]

Definition 2.3 *A set of explanations S is* minimal *if there is no explanation in S whose assumption set is a variant-subset of the assumption set of another explanation in S.*

Since the definition of abduction requires consistency of the assumed atoms with the domain theory, the abduction problem is in general undecidable. In our implemented system ACCEL, consistency checking is accomplished in two ways:

[1] In this paper, we denote variables by uppercase letters, and constants by lowercase letters.

1. Using a pre-determined list of *nogoods*, where a nogood is a set of assumptions $\{A_1(v_1, \ldots, v_k), \ldots, A_n(v_1, \ldots, v_k)\}$ such that

$$\forall v_1, \ldots, v_k \; A_1(v_1, \ldots, v_k) \wedge \ldots \wedge A_n(v_1, \ldots, v_k) \rightarrow false.$$

   Consistency checking ensures that an assumed set of atoms is not subsumed by any nogoods (i.e., no instance of a nogood is a subset of an assumed set of atoms);

2. Using procedural code to check for inconsistency of assumptions (for efficiency reasons).

2.2 The SAA Algorithm

2.2.1 Definition and Algorithm

Stickel has proposed an algorithm for computing the set of all first-order Horn-clause abductive proofs [66]. His algorithm, which we will call SAA (Stickel's Abduction Algorithm), operates by applying inference rules to generate goal clauses. The initial goal clause is the input atoms $O_1, \ldots, O_m$. Each atom in a goal clause can be marked with one of *proved*, *assumed*, or *unsolved*. All atoms in the initial goal clause are marked as unsolved. A final goal clause must consist entirely of proved or assumed atoms.

Let G be a goal clause $Q_1, \ldots, Q_n$, where the leftmost unsolved atom is Q_i. The algorithm SAA repeatedly applies the following inference rules to goal clauses G with unsolved atoms:

- *Resolution with a fact.* If Q_i and a fact F are unifiable with a most general unifier (mgu) σ, the goal clause $Q_1\sigma, \ldots, Q_n\sigma$ can be derived, where $Q_i\sigma$ is marked as proved.

- *Resolution with a rule.* Let $C \leftarrow P_1 \wedge \ldots \wedge P_r$ be a rule where Q_i and C are unifiable with a mgu σ. Then the goal clause

$$Q_1\sigma, \ldots, Q_{i-1}\sigma, P_1\sigma, \ldots, P_r\sigma, Q_i\sigma, \ldots, Q_n\sigma$$

  can be derived, where $Q_i\sigma$ is marked as proved and each $P_k\sigma$ is marked as unsolved.

- *Making an assumption.* If Q_i is *assumable*, then $Q_1, \ldots, Q_n$ can be derived with Q_i marked as assumed. By *assumable*, we mean that Q_i has been designated as an atom that the algorithm is allowed to assume. The algorithm also checks that all assumptions made are consistent with the domain theory.

- *Factoring with a proved or assumed atom.* If Q_j and Q_i $(j < i)$ are unifiable with a mgu σ, the goal clause

$$Q_1\sigma, \ldots, Q_{i-1}\sigma, Q_{i+1}\sigma, \ldots, Q_n\sigma$$

  can be derived.

2.2.2 Problems with the SAA Algorithm

The SAA algorithm as described above suffers from two problems.

1. Combinatorial Explosion

   It has been shown that, even in the propositional case, computing all minimal (w.r.t. subset) explanations is provably exponential [40; 65], since in the worst case, the number of minimal explanations is exponentially large. The SAA algorithm computes all first order Horn-clause abductive explanations and therefore it is also at least an exponential algorithm. (Actually, since the SAA "algorithm" includes consistency checking of the assumptions, it may not even terminate in general.)

   However, in practice, what is needed is only the best explanation, or the best few explanations. To avoid combinatorially explosive computation, [66; 30] proposed the use of a cost metric to rank and heuristically search the more promising explanations first. Each input atom is assigned a cost of assuming that input atom. The antecedents $A_1, \ldots, A_n$ of every rule R in the knowledge base are assigned relative costs $C_1, \ldots, C_n$ so that when the algorithm backward-chains on a subgoal G using a rule R, the cost of each new antecedent subgoal A_i is $cost(G) \times C_i / \sum_{i=1}^{n} C_i$. The best explanation has assumptions with the least cumulative cost.

   Simplicity, defined as making the minimum number of assumptions in an explanation (known as the *minimum* explanation), is another commonly used metric to select the best explanations [61; 5; 32]. However, previous work has shown that finding the minimum abductive explanation is NP-hard [61; 62; 2; 4].

   In this paper, we used a form of beam search to overcome the computational intractability problem. Evaluation metrics including a coherence metric and a simplicity metric are used to determine the quality of an abductive proof, and a limited list of the best abductive proofs are maintained during the search.

2. Redundant Inference

   Even with the use of heuristic search to restrict the computation expended in finding the good explanations, the SAA algorithm can still perform a great deal of redundant work in that partial explanations are not cached and shared among multiple explanations. To see why this is the case, consider the two examples shown in Figures 1 and 2.

   In the first example, after backward-chaining on the rule $a \leftarrow b \wedge c \wedge d$, the SAA algorithm can either make b an assumption, or backward-chain on the rule $b \leftarrow e \wedge f$. This introduces two partial abductive proofs, both have the identical subgoals c and d. The SAA algorithm will then duplicate the same inferences in expanding the subgoals c and d in the two partial proofs. Since the proof tree rooted at the subgoals c and d can be arbitrarily deep, substantial effort will be wasted duplicating inferences.

   In the second example, each time a subgoal (like a, b, and e) is expanded by backward-chaining, two partial proofs are generated, and inferences will be duplicated across

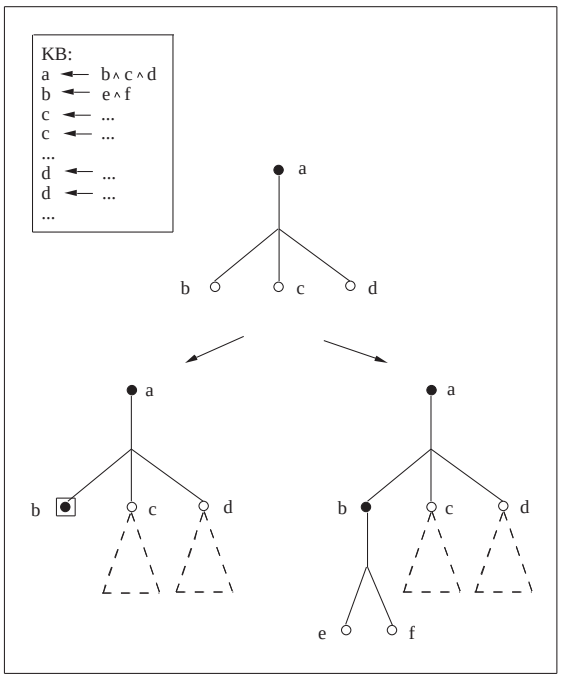

Figure 1: Duplicating inference: example 1.

the two successor partial proofs. For instance, the rule $b \leftarrow e \wedge f$ is applied twice, the rule $e \leftarrow h \wedge i$ is applied four times, etc.

Note that this problem of duplicating inference arises in deductive theorem proving too. However, we believe that duplicating inference poses a more serious problem in abduction because multiple abductive proofs must usually be pursued in the search for a best explanation, whereas in deduction, we are usually interested in a single deductive proof. The need for multiple abductive proofs tends to result in more duplicate inferences being made. Also, note that the situation in example 1 arises each time an assumption is made, and since making assumptions occurs frequently in abduction, duplicating inferences almost always arise in abduction. In Section 6, we present empirical results showing that avoiding duplicate inferences can achieve more than an order of magnitude speedup.

2.2.3 Variant-subsets

The explanations generated by the SAA algorithm may include some that are variant-subsets of another. For instance, given the following axioms:

$inst(G, going) \leftarrow inst(S, shopping) \wedge go\text{-}step(S, G)$.

$goer(G, P) \leftarrow inst(S, shopping) \wedge go\text{-}step(S, G) \wedge shopper(S, P)$.

and the input atoms $inst(go1, going)$ and $goer(go1, john1)$, we can derive the explanation F with assumptions $A_F = \{inst(X, shopping), go\text{-}step(X, go1), inst(Y, shopping), go\text{-}step(Y, go1), shopper(Y, john1)\}$ by backward-chaining on the two axioms. Applying the

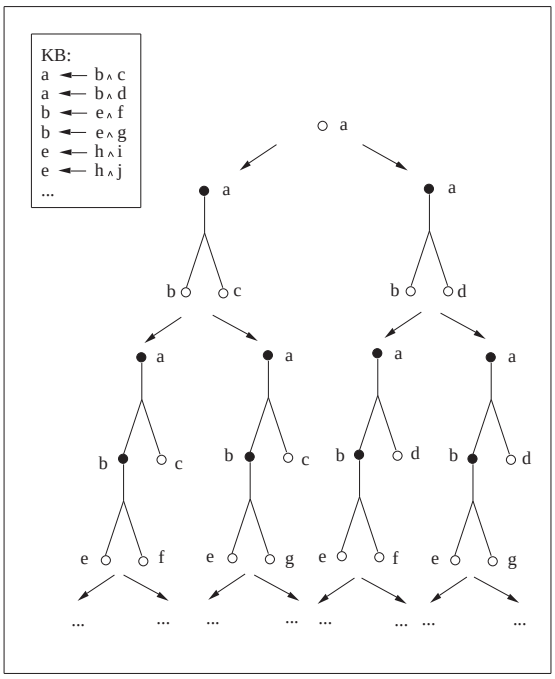

Figure 2: Duplicating inference: example 2.

factoring operation, we can obtain another explanation E with assumptions $A_E = \{inst(X, shopping), go\text{-}step(X, go1), shopper(X, john1)\}$. But note that although $A_E \not\subseteq A_F$, $A_E\sigma \subset A_F$ with the renaming substitution $\sigma = \{X/Y\}$.

Note that the variant-subset relation is a special case of subsumption. A subsumes B if $A\sigma \subseteq B$ for some substitution σ. However, for A to be a variant-subset of B, the substitution σ must be a renaming substitution.

Since explanations that are variant-supersets of other explanations are essentially redundant, they need to be eliminated. Unfortunately, it can be readily shown that determining variant-subset relation is an NP-complete problem by reduction from directed subgraph isomorphism, a known NP-complete problem [25, page 202]. A directed graph $G = \langle V, E\rangle$ is transformed into an assumption set A as follows: for every vertex $v \in V$, add the assumption $N(X_v)$ to A, where X_v is a variable; for every directed edge $e = (v_1, v_2) \in E$, add the assumption $E(X_{v_1}, X_{v_2})$ to A, where X_{v_1}, X_{v_2} are variables. That is, $|A| = |V| + |E|$. It follows that G_1 is isomorphic to a subgraph of G_2 if and only if A_1 is a variant-subset of A_2. Hence, determining variant-subsets introduces yet another source of computational complexity when finding the minimal explanations in a first-order Horn-clause theory.

2.3 The AAA Algorithm

We now present the abduction algorithm used in ACCEL, called AAA (ATMS-based Abduction Algorithm). This algorithm is much like the SAA algorithm, except that the abductive proofs for a subgoal G are cached and reused when the subgoal G or an instance of G is

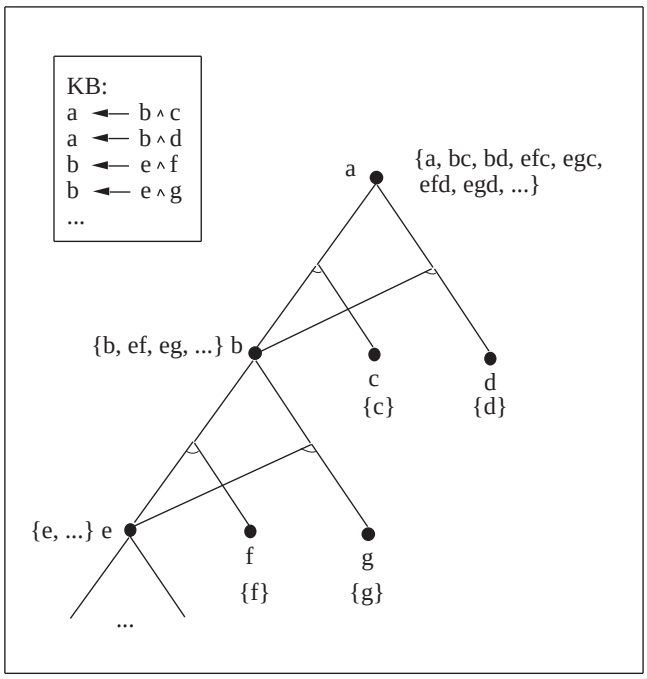

Figure 3: Caching and sharing inference steps.

encountered subsequently in the search for the best abductive proofs.

To illustrate the idea of proof caching in AAA, consider the example shown in Figure 3. In the AAA algorithm, each of the rules $b \leftarrow e \wedge f$, $b \leftarrow e \wedge g$, $e \leftarrow h \wedge i$, etc., will be applied only once. Associated with each proved subgoal, we store all the abductive proofs of that subgoal. Figure 3 shows the abductive proofs associated with each subgoal in finding the abductive proofs for a in the AAA algorithm.

When the subgoal b is first encountered as an antecedent in the rule $a \leftarrow b \wedge c$, its abductive proofs are constructed by backward-chaining on the rules $b \leftarrow e \wedge f$, $b \leftarrow e \wedge g$, etc., in the knowledge base in a depth-first search order. When all the abductive proofs of b are found, they are associated with b and stored in a cache. Subsequently, backward-chaining using the second rule $a \leftarrow b \wedge d$ will encounter the subgoal b again. At this time, all the previously found abductive proofs of b will be reused without recomputing them again.

2.3.1 Definition and Algorithm

The use of proof caching is very much in the style of an Assumption-based Truth Maintenance System (ATMS) [16], which caches and reuses partial explanations to avoid redundant work. The ATMS is a general facility for managing logical relationships among propositional formulas. It maintains multiple contexts at once and is particularly suitable for problem solving that involves constructing and comparing multiple explanations. In the ATMS, each problem solving datum is associated with a *node*. A node in the ATMS can be further designated as an *assumption*. Nodes are related via *justifications*. A justification is

a propositional Horn-clause of the form $A_1 \wedge \ldots \wedge A_n \rightarrow C$, where each A_i is an antecedent node and C is the consequent node. An *environment* is a set of assumptions. Associated with each node is a set of environments called its *label*. The difference between AAA and the ATMS is that AAA constructs first-order Horn-clause proofs while the ATMS only deals with propositional Horn-clauses. Also, AAA is a backward-chaining algorithm while the traditional ATMS is forward-chaining.

We now give a formal description of the AAA algorithm. We will use similar terminology as in the ATMS.

Definition 2.4 *Let* $E = \langle A, \sigma \rangle$, *where* A *is a set of assumptions* $\{A_1(v_1, \ldots, v_k), \ldots, A_n(v_1, \ldots, v_k)\}$ *to be interpreted as an existentially quantified conjunction of assumptions, and* σ *is a substitution. We say that an atom* G *has an* environment E *iff* $A \cup T \models G\sigma$ *and* $A \cup T$ *is consistent.*

Note that a substitution σ is included as part of an environment E. This allows us to know directly from an environment $E = \langle A, \sigma \rangle$ associated with an atom G which instance of G is provable from the assumptions A.

Definition 2.5 *The label of an atom* G *(denoted* $label(G)$*) is a set of environments* $\{E_1, \ldots, E_n\}$, *to be interpreted as a disjunction of environments* $E_1 \vee \ldots \vee E_n$, *where* $E_i = \langle A_i, \sigma_i \rangle$, *such that*

- *(Soundness)* $A_i \cup T \models G\sigma_i$;
- *(Consistency)* $A_i \cup T$ *is consistent;*
- *(Completeness) For any consistent set of assumptions* A *where* $A \cup T \models G$, A *is subsumed by some* A_i; *and*
- *(Minimality) No* A_i *is a variant-subset of some other* A_j.

The label of an atom is thus the set of all minimal (w.r.t. variant-subset) explanations of the atom G. Note that the set of explanations in a label is minimal w.r.t. to variant-subset but not subsumption. This is because a set of assumptions A that is obtained by factoring another set of assumptions B is such that A is subsumed by B (since $A = B\sigma$ for some substitution σ), but we do not want to remove A from the label since factoring frequently results in better explanations.

Without loss of generality, we can assume that the task of abduction is to find all minimal explanations of an atom $O(v_1, \ldots, v_k)$, since all explanations of $O_1(v_1, \ldots, v_k) \wedge \ldots \wedge O_m(v_1, \ldots, v_k)$ is the same as all explanations of $O(v_1, \ldots, v_k)$ once we add the rule

$$O(v_1, \ldots, v_k) \leftarrow O_1(v_1, \ldots, v_k) \wedge \ldots \wedge O_m(v_1, \ldots, v_k)$$

to the domain theory.

The AAA algorithm is presented in Tables 1–2. The top level procedure is *compute-label*(G, D), which will compute and return all possible abductive proofs of depth D or less

of the atom G. In order to limit the search to the promising explanations, the algorithm will only maintain at most β_{intra} number of best explanations for each subgoal encountered in the search, where the quality of an explanation is determined by some evaluation metric (such as coherence or simplicity). When $\beta_{intra} = \infty$, all possible abductive proofs of depth D are computed. β_{intra} is thus the beam width of the heuristic beam search used to limit the computational efforts expended in finding the best explanations. We used beam search instead of best-first search since best-first search requires maintaining the complete list of partial explanations and so is too memory-consuming. If we let $\beta_{intra} = 1$, the beam search algorithm becomes a hill-climbing algorithm.

The AAA algorithm presented in this paper supersedes a previous version reported in [48; 47] which is incomplete. Specifically, the previous version misses explanations that are obtained when, during resolution of a subgoal with a fact or the consequent of a rule, the most general unifier is such that some variables in the subgoal are instantiated.

2.3.2 Indexing and Cache Lookup

The label of a subgoal, once computed, is stored in the cache indexed under the subgoal. The cache indexing scheme implemented in ACCEL is discrimination tree indexing, as described in [50].

When AAA queries the cache for some subgoal G, if there exists in the cache some previous subgoal G' that is an alphabetic variant of G or is more general than G (i.e., $G = G'\sigma$ for some substitution σ), then the appropriate subset of G''s label, suitably renamed, will be returned as G's label. The procedure for cache lookup is given in Table 2.

A more aggressive caching scheme may generalize the abductive proofs of a subgoal G as much as possible at the end of executing *compute-label*(G, D) so as to obtain generalized abductive proofs for a more general subgoal G' [68]. This has the advantage of potentially avoiding more duplicate inferences, but at the expense of incurring more work to perform the generalization at cache insertion time. The AAA algorithm implemented in ACCEL does *not* perform such generalization before storing a subgoal and its proofs in the cache, as it is unclear if there are any overall net savings in doing so.

The indexing of facts and rules that can potentially unify with a subgoal is also accomplished via discrimination tree indexing. See [50] for more details.

2.3.3 An Illustrative Example

To illustrate the working of the AAA algorithm, consider the following domain theory:

$p(X) \leftarrow q(X, Y) \wedge r(Y, X)$

$q(a, Z) \leftarrow s(a, Z)$

$r(U, a) \leftarrow s(U, a) \wedge t(U)$

$r(V, b) \leftarrow s(V, b) \wedge t(V)$

For this example, we let all atoms be assumable. Suppose the input atom is $p(X)$ and we want to find all abductive explanations of $p(X)$. The procedure *compute-label*$(p(X), \infty)$ first assumes $p(X)$ and adds the environment

$A_1 : \langle \{p(X)\}, \{\} \rangle$

compute-label(G,D)
if cache-lookup(G,D) succeeds then return
label(G) $\leftarrow \emptyset$
if G is assumable then
  label(G) $\leftarrow \{\langle\{G\},\{\}\rangle\}$
for each fact F unifiable with G
    rename the variables in F
    $\sigma \leftarrow$ mgu(F, G)
    label(G) $\leftarrow$ label(G) $\cup\{\langle\emptyset,\sigma\rangle\}$
backward-chain(G,D)
store (G,D,label(G)) in the cache

backward-chain(G,D)
if $D = 0$ then return
for each rule $C \leftarrow P_1 \wedge \ldots \wedge P_r$
    rename the variables in the rule
    $\sigma \leftarrow$ unify(C,G)
    if $\sigma \neq$ fail then
      $P_1 \leftarrow P_1\sigma, \ldots, P_r \leftarrow P_r\sigma$
      compute-label($P_1, D-1$)
      partial-envs $\leftarrow \emptyset$
      for each environment $\langle A, \lambda\rangle \in$ label(P_1)
          partial-envs $\leftarrow$ partial-envs $\cup\{\langle A, \sigma\lambda\rangle\}$
      for each $P_i \in \{P_2, \ldots, P_r\}$
          partial-envs $\leftarrow$ cross-product(partial-envs,P_i,D)
          variant-subset-minimize(partial-envs)
          if |partial-envs| $> \beta_{intra}$ then
            sort partial-envs by the evaluation metric
            truncate the size of partial-envs to β_{intra}
      label(G) $\leftarrow$ label(G) $\cup$ partial-envs
variant-subset-minimize(label(G))
if |label(G)| $> \beta_{intra}$ then
  sort label(G) by the evaluation metric
  truncate the size of label(G) to β_{intra}

Table 1: The AAA Algorithm.

cross-product(partial-envs,P,D)
old-partial-envs $\leftarrow$ partial-envs
partial-envs $\leftarrow \emptyset$
for each $E_1 = \langle A_1, \sigma_1 \rangle \in$ old-partial-envs
    compute-label($P\sigma_1, D-1$)
    for each $E_2 = \langle A_2, \sigma_2 \rangle \in$ label($P\sigma_1$)
        $E \leftarrow \langle A_1\sigma_2 \cup A_2, \sigma_1\sigma_2 \rangle$
        if E is consistent then
            partial-envs $\leftarrow$ partial-envs $\cup$ factoring(E,$A_1\sigma_2$,A_2)
return(partial-envs)

factoring(E,A_1,A_2)
$\sigma \leftarrow$ substitution of E
factors $\leftarrow \{E\}$
for each $a_1 \in A_1$
    for each $a_2 \in A_2$
        $\sigma' \leftarrow$ unify(a_1,a_2)
        if $\sigma' \neq$ fail then
            $E' \leftarrow \langle A_1\sigma' \cup A_2\sigma', \sigma\sigma' \rangle$
            if E' is consistent then
                factors $\leftarrow$ factors $\cup$ factoring($E', A_1\sigma', A_2\sigma'$)
return(factors)

cache-lookup(G,D)
if (G',D',label(G')) exists in the cache such that
  depth $D' \geq D$ and
  $G = G'\sigma'$ for some substitution σ' then
  label(G) $\leftarrow \emptyset$
  for each $E = \langle A, \sigma \rangle \in$ label(G')
      $H \leftarrow G'\sigma$
      rename H and A
      $\lambda \leftarrow$ mgu(H, G)
      if $\lambda \neq$ fail then
        label(G) $\leftarrow$ label(G) $\cup \{\langle A\lambda, \lambda \rangle\}$
  return(label(G))
else
  return(fail)

Table 2: The AAA Algorithm.

to $label(p(X))$. $backward\text{-}chain(p(x), \infty)$ is then executed and the rule $p(X) \leftarrow q(X,Y) \wedge r(Y,X)$ is considered. AAA first makes the recursive call $compute\text{-}label(q(X,Y), \infty)$ to compute all abductive explanations of the first antecedent of the rule. Two abductive explanations are returned: $E_1 = \langle\{q(X,Y)\}, \{\}\rangle$ and $E_2 = \langle\{s(a,Z)\}, \{X/a, Y/Z\}\rangle$, corresponding to assuming $q(X,Y)$ and backward-chaining on the second rule $q(a,Z) \leftarrow s(a,Z)$.

Next, AAA considers the second antecedent $r(Y,X)$ of the first rule. It calls $cross\text{-}product(\{E_1, E_2\}, r(Y,X), \infty)$ to find the cross product of the labels of $q(X,Y)$ and $r(Y,X)$. The procedure *cross-product* steps through the environments of the label of $q(X,Y)$ one at a time, and for each environment E_i, it makes a recursive call to compute the label of $r(Y,X)$ instantiated under the substitution of E_i. Since the substitution of $E_1 = \langle\{q(X,Y)\}, \{\}\rangle$ is empty, the recursive call $compute\text{-}label(r(Y,X), \infty)$ is made and it returns three abductive explanations: $F_1 = \langle\{r(Y,X)\}, \{\}\rangle$, $F_2 = \langle\{S(U,a), t(U)\}, \{Y/U, X/a\}\rangle$, and $F_3 = \langle\{S(V,b), t(V)\}, \{Y/V, X/b\}\rangle$. The three explanations correspond to making $r(Y,X)$ an assumption, and backward-chaining on the third and fourth rules. Taking the union of the appropriately instantiated environment E_1 and each of the environments F_1, F_2, F_3 yields three additional abductive explanations for the label of $p(X)$:

$A_2 : \langle\{q(X,Y), r(Y,X)\}, \{\}\rangle$

$A_3 : \langle\{q(a,U), s(U,a), t(U)\}, \{Y/U, X/a\}\rangle$

$A_4 : \langle\{q(b,V), s(V,b), t(V)\}, \{Y/V, X/b\}\rangle$

The second environment E_2 of the label of $q(X,Y)$ has the substitution $\{X/a, Y/Z\}$, so another recursive call $compute\text{-}label(r(Z,a), \infty)$ is made. However, since the more general subgoal $r(Y,X)$ has been encountered previously and its abductive proofs are cached, AAA reuses the appropriate subset of the label of $r(Y,X)$ in the cache. The environments returned from the call $compute\text{-}label(r(Z,a), \infty)$ are (after renaming) $G_1 = \langle\{r(Y',a)\}, \{Z/Y', X'/a\}\rangle$ and $G_2 = \langle\{s(U',a), t(U')\}, \{Z/U'\}\rangle$. Taking the union of the appropriately instantiated environment E_2 and the environments G_1 and G_2 yields two additional abductive explanations for the label of $p(X)$:

$A_5 : \langle\{s(a,Y'), r(Y',a)\}, \{X/a, Y/Y', Z/Y', X'/a\}\rangle$

$A_6 : \langle\{s(a,U'), s(U',a), t(U')\}, \{X/a, Y/U', Z/U'\}\rangle$

Finally, factoring of the assumptions $s(a,U')$ and $s(U',a)$ in A_6 yields an additional abductive explanation for the label of $p(X)$:

$A_7 : \langle\{s(a,a), t(a)\}, \{X/a, Y/a, Z/a, U'/a\}\rangle$

The label of $p(X)$ computed is $\{A_1, \ldots, A_7\}$.

2.3.4 Enhancements to AAA

The AAA algorithm presented above is actually a simplification of the one implemented in ACCEL. For the sake of clarity in exposition, we have omitted some less essential details of the algorithm in Tables 1–2. In order to be more useful and efficient, the AAA algorithm has a few additional parameters (in addition to D and β_{intra}) so that the algorithm can be specialized to execute more efficiently in each of the different domains. These parameters are:

- β_{inter}: This parameter controls the number of explanations kept after processing each of the input atoms in the conjunction $O_1 \wedge \ldots \wedge O_m$ given to ACCEL. It is always the case that $\beta_{inter} \leq \beta_{intra}$, since β_{intra} determines the number of explanations kept at every subgoal, and so the number of explanations kept at an input atom can be no more than β_{intra}.

- Factoring: This is a parameter to control if factoring should be performed. Factoring is essential for plan recognition and set-covering-based diagnosis, but it is turned off for model-based diagnosis.

- Variant-superset: Since eliminating variant-supersets is an expensive operation, there is a parameter to control whether it should be used. Eliminating variant-supersets is necessary for plan recognition, but it is turned off for model-based diagnosis. It is specialized to removing (simple) supersets for set-covering-based diagnosis, since the axioms for set-covering-based diagnosis are propositional.

- Evaluation metric: Each domain has its own explanation evaluation metric to determine the quality of a given explanation. In the plan recognition domain, coherence is a better evaluation metric, whereas in the diagnosis domains, the simplicity metric suffices.

- Assumable predicates: The atoms that are assumable vary according to the domain. In the plan recognition domain, all atoms are assumable. In the diagnosis domain, only atoms corresponding to diseases or behavioral modes (normality, fault modes, abnormality) are assumable. Abduction in which the assumable atoms are restricted to a pre-determined set of predicates is known as *predicate specific abduction* [66].

A complete list of efficiency enhancements to AAA is given in [44].

3 Plan Recognition

Given a logical representation of the literal meaning of a narrative text in terms of an existentially quantified conjunction of input atoms, ACCEL infers an "embellished" interpretation by constructing an abductive proof in which a set of higher-level plans is assumed that logically entail the characters' observed actions. An abductive proof is considered an interpretation of the input sentences. We do not focus on the parsing aspect of natural language understanding, and ACCEL does not accept natural language input. Instead, we assume the existence of some appropriate parser that translates a given set of input sentences into an existentially quantified conjunction of input atoms.

3.1 Explanatory Coherence

3.1.1 Motivation

In previous research on abduction for text understanding and plan recognition, simplicity has been proposed as a metric for selecting the best explanation. For instance, in [5],

the best interpretation is one that maximizes $E - A$, where E = the number of explained observations, and A = the number of assumptions made. The work of Kautz explicitly incorporates the assumption of minimizing the number of top-level events in deducing the plan that an agent is pursuing [32].

Though an important factor, the simplicity criterion is not sufficient by itself to select the best explanation. In the area of language understanding, we argue that some notion of explanatory coherence is more important in deciding which explanation is the best. Consider the sentences: "Mary had a heart attack. John is depressed." The sentences translate into the conjunction of the following atoms: name(m,mary), has(m,h), heart-attack(h), name(j,john), and depressed(j). A knowledge base of axioms relevant to these input atoms are:

depressed(X) $\leftarrow$ like(X,Y) $\wedge$ bad(condition(Y)) $\wedge$ irreplaceable(Y)
depressed(X) $\leftarrow$ pessimist(X)
bad(condition(X)) $\leftarrow$ has(X,Y) $\wedge$ illness(Y)
illness(X) $\leftarrow$ heart-attack(X)

Based on the above axioms, there are two possible interpretations of these sentences, as shown in Figure 4. Suppose the simplicity metric is defined as the inverse of the number of assumptions made, where every leaf node in the proof graph counts as an assumption, including input atoms that are not explained (the assumptions in Figure 4 are underlined). Relying on this simplicity metric results in selecting the interpretation that John is depressed because he is a pessimist, someone who always feels gloomy about life (Figure 4b). This is in contrast to our preferred interpretation of the sentences — John is depressed because John likes Mary and Mary had a heart attack (Figure 4a).

Note that varying the definition of simplicity somewhat will not help here. For instance, using the simplicity criterion of [63] based on subset minimality does not work well for this example — it is indifferent towards both interpretations, instead of choosing the preferred one. If we decide not to count input atoms as assumptions, then the preferred interpretation still makes more assumptions (four) compared to only one assumption in the other interpretation. Charniak's simplicity metric of $E - A$ also will not work, since both explanations would explain exactly one input atom, that John is depressed.

Intuitively, it seems that the first interpretation (Figure 4a) is better because the input atoms are connected more "coherently" than in the second interpretation (Figure 4b). We manage to connect "John is depressed" with "Mary had a heart attack" in the first interpretation, whereas in the second interpretation, they are totally unrelated. This is the intuitive notion of what we mean by *explanatory coherence*, i.e., how well the various parts of the input sentences are "tied together" in the interpretation.

That sentences in a natural language text are connected in a coherent way is reflected in the well known "Grice's conversational maxims" [28], which are principles governing the production of natural language utterances, such as "be relevant", "be informative", etc. Although the notion that natural language text is coherently structured has long been recognized by researchers in natural language processing (see [3]), our work is the first to incorporate the notion of coherence in the context of evaluating an abductive explanation.

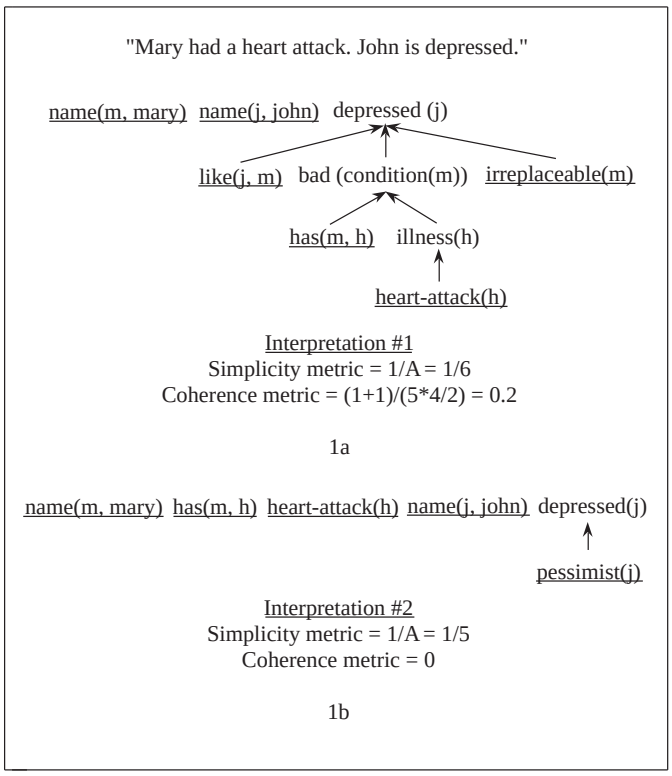

Figure 4: The importance of explanatory coherence.

3.1.2 Definition

We would like to formulate our coherence metric so as to possess several desirable properties. In particular, explanations with more connections between any pair of input atoms should have higher coherence metric values. Also, a coherence metric with values lying in a unit range 0–1 will facilitate the comparison of explanations. We have developed a formal characterization of what we mean by explanatory coherence in the form of a coherence metric satisfying these properties.

Definition 3.1 *The coherence metric C is defined as follows:*

$$\mathcal{C} = \frac{\sum_{1 \le i < j \le l} N_{i,j}}{l(l-1)/2}$$

where

l = *the total number of input atoms;*

$N_{i,j}$ = *1 if there is some node n in the proof graph such that there is a (possibly empty) sequence of directed edges from n to n_i and a (possibly empty) sequence of directed edges from n to n_j, where n_i and n_j are input atoms. Otherwise, $N_{i,j} = 0$.*

The numerator of this metric is the total number of pairs of input atoms that are connected. The denominator of the metric scales the sum according to the size of the explanation so that the final metric value falls between 0 and 1. Note that the definition of coherence given in this paper is a slight modification of the one given in [46; 45]. The new definition remedies the anomaly reported in [51] of occasionally preferring spurious interpretations of greater depths.

To illustrate the computation of the coherence metric, consider the explanation in Figure 4a. Let n_1 = (name m mary), n_2 = (name j john), n_3 = (depressed j), n_4 = (has m h), and n_5 = (heart-attack h). The total number of input atoms $l = 5$. In this explanation, $N_{3,4} = 1$, since there is a node n_4 such that there is a directed path from n_4 to n_3 and also a directed path from n_4 to n_4 (the trivial empty path). Similarly, $N_{3,5} = 1$. All other $N_{i,j} = 0$. This results in the coherence metric $\mathcal{C} = 0.2$, as shown in Figure 4a.

Some advantages of our coherence metric include:

1. Coherent explanations are often simple explanations. This is because in a coherent explanation, propositions tend to be more tightly connected together. This increases the likelihood of assumptions being unified, and leads to a reduction in the number of assumptions made and thus a simpler explanation.

2. Compared to the simplicity metric, the coherence metric is less vulnerable to changes in the underlying representation of the knowledge base. It is relatively easy to encode the axioms in a knowledge base in a slightly different way so as to change the number of assumptions made in an explanation. However, connections between propositions are less dependent (relatively speaking) on such changes. For example, suppose we change the axioms in the given example slightly so that as long as a person likes something that is in a bad condition (but not necessarily irreplaceable), then that person is depressed. Also, suppose one has to be poor as well as a pessimist to be depressed. Given this modified set of axioms, the first interpretation now requires only five assumptions, while the second interpretation requires six. So all of a sudden, the first interpretation becomes the simpler explanation of the two. However, the coherence metric values of both interpretations remain unchanged.

3. Evaluating explanations based on coherence also nicely resolves a problem in abduction, that of deciding on the appropriate level of specificity of explanations. Previous approaches fall into several categories: most specific abduction, least specific abduction, cost-based (weighted) abduction, and predicate specific abduction. In most specific abduction, the assumptions made must be *basic*, i.e., they cannot be "intermediate" assumptions that are themselves provable by assuming some other (more basic) assumptions [13]. In least specific abduction, the only allowable assumptions are the input atoms [66]. In cost-based abduction, costs (or weights) are assigned to the antecedents of backward-chaining rules in order to influence the decision on whether to backward-chain on a rule [30]. In predicate specific abduction, the assumptions made must have predicates from a pre-determined set of predicates.

   However, none of the above approaches is completely satisfactory. Least specific abduction is too restrictive since frequently, assumptions other than the input atoms

must be made, such as those to be inferred by a reader. Most specific abduction is also too rigid since it is not always the case that we want to explain everything in terms of every available cause, since the causes that explain different input atoms may be completely unrelated to one another. Cost-based abduction could presumably arrive at the correct explanation given the "appropriate" set of costs, but it is unclear how the costs can be assigned in general to work on all problems. Predicate specific abduction is not suitable for text understanding since the assumptions made by a reader in text understanding are not restricted to a fixed set of predicates.

In our approach, the desired specificity of an explanation is one which maximizes coherence. That is, we backward-chain on rules to prove the subgoals in an explanation only if doing so increases its overall coherence, and thus we make assumptions just specific enough to connect the input atoms. Coherence has been successfully used to determine the appropriate level of specificity of explanations for the 50 narrative texts processed by ACCEL. Hence, we believe our coherence-based approach is better than the alternative approaches in determining the specificity of explanations.

Finally, we want to point out that it is *not* our belief that simplicity is completely irrelevant to the selection of explanations. (In fact, in the diagnosis domain, we rely on simplicity as our evaluation metric.) Rather, we consider explanatory coherence to be a *more* important criterion in selecting good explanations in text understanding and plan recognition. As such, we evaluate explanations in the plan recognition domain based on their coherence. When there is a tie between the coherence metric values of two explanations, we then rely on the simplicity metric to break the tie, where the simplicity metric is defined here as $1/A$ (A = the total number of assumptions made in an explanation). Our empirical results to be presented later in this section confirm that coherence is indeed a better measure in the plan recognition domain.

3.1.3 Computation

The coherence metric as defined above can be efficiently computed. We assume that the proof graph contains no cycles, since circular justification is not considered a good trait of an explanation. Using a standard depth-first graph search algorithm [1], it can be readily shown that $\mathcal{C}$ can be computed in time $O(l^2 \cdot N + l \cdot e)$, where l = the total number of input atoms, N = the total number of nodes in the proof graph, and e = the total number of directed edges in the proof graph. See [44] for more details on computing the coherence metric.

3.2 Finding Coherent Explanations

As mentioned in Section 2, finding the simplest abductive explanation has been shown to be NP-hard [61; 62; 2; 4]. Unfortunately, finding the most coherent explanation is also NP-hard. We present a proof that a specialized instance of our problem of finding the most coherent explanation is NP-hard. The specialized optimization problem is: finding

the maximally coherent explanation that satisfies simple contradiction restrictions in a two-level, propositional abduction model.

We will denote the coherence value of an explanation E as $\mathcal{C}(E)$. We show that the corresponding decision problem is NP-complete.

3.2.1 An NP-Completeness Proof

Definition 3.2 MAXIMALLY COHERENT EXPLANATION (MCE)
INSTANCE : A set O of observations, a set A of assumptions, a relation $M \subseteq A \times O$ (where $\langle a, o \rangle \in M$ denotes $a \rightarrow o$), a collection C of subsets of A (where each subset of A in C is taken to mean that the conjunction of the assumptions in the subset is contradictory), and a positive real number $K < 1$. Define an explanation E to be a graph $\langle O \cup A', M' \rangle$ with nodes $O \cup A'$ and edges M' such that $A' \subseteq A$, $M' \subseteq M$, $\{a | \langle a, o \rangle \in M'\} = A'$, and for every $C_i \in C$, $C_i \not\subseteq A'$. (The last condition ensures that A' is consistent.)
QUESTION : Is there an explanation $E = \langle O \cup A', M' \rangle$ such that $\mathcal{C}(E) \geq K$?

Theorem 3.1 *The* MCE *problem is NP-complete.*

Proof

It is clear that MCE is in NP. A nondeterministic algorithm for it need only guess some graph $\langle O \cup A', M' \rangle$ and check to see whether the graph constitutes an explanation and whether $\mathcal{C}(E) \geq K$. This can be easily done in (nondeterministic) polynomial time.

To satisfy the second requirement of NP-completeness, we will reduce the known NP-complete problem HITTING SET [25] to MCE. The HITTING SET problem is :

INSTANCE : A collection D of subsets of a set S, and a positive integer L.
QUESTION : Does S contain a *hitting set* for D of size L or less, that is, a subset $S' \subseteq S$ with $|S'| \leq L$ and such that S' contains at least one element from each subset in D?

Given an instance of the HITTING SET problem $\langle S, D, L \rangle$, where $|S| = n$, we construct an instance of the MCE problem as follows:

$O = \{o_1, o_2, \ldots, o_{2n-1}, o_{2n}\}$
$A = S = \{a_1, a_2, \ldots, a_n\}$
$M = \{\langle a_1, o_1 \rangle, \langle a_1, o_2 \rangle, \langle a_2, o_3 \rangle, \langle a_2, o_4 \rangle, \ldots, \langle a_n, o_{2n-1} \rangle, \langle a_n, o_{2n} \rangle\}$
$C = D$
$K = \frac{n-L}{2n(2n-1)/2}$

Clearly, the construction of such an instance takes deterministic polynomial time. It remains to prove that D has a hitting set H of size L or less if and only if there is an explanation $E = \langle O \cup A', M' \rangle$ such that $\mathcal{C}(E) \geq K$.

($\Rightarrow$) Suppose D has a hitting set H where $|H| \leq L$. Let $E = \langle O \cup A', M' \rangle$ where $A' = S - H = \{a_{i_1}, a_{i_2}, \ldots, a_{i_{n-|H|}}\}$, $M' = \{\langle a_{i_1}, o_{2i_1-1} \rangle, \langle a_{i_1}, o_{2i_1} \rangle, \langle a_{i_2}, o_{2i_2-1} \rangle, \langle a_{i_2}, o_{2i_2} \rangle, \ldots\}$. That is, there are two edges connecting every assumption in A' to its two corresponding observations in the explanation. Then

$$\mathcal{C}(E) = \frac{n - |H|}{2n(2n-1)/2}$$

Since in addition $K = \frac{n-L}{2n(2n-1)/2}$ and $|H| \leq L$, it follows that $\mathcal{C}(E) \geq K$.

To prove that for each $C_i \in C$, $C_i \not\subseteq A'$, assume otherwise. Then for some $C_i \in C$, $C_i \subseteq A'$. Since $H \cap A' = \emptyset$, this implies $H \cap C_i = \emptyset$, contradicting the fact that H is a hitting set for D $(= C)$.

($\Leftarrow$) Suppose there is an explanation $E = \langle O \cup A', M' \rangle$ such that $\mathcal{C}(E) \geq K$.

$$\mathcal{C}(E) = \frac{|A'|}{2n(2n-1)/2}$$

Since $\mathcal{C}(E) \geq K = \frac{n-L}{2n(2n-1)/2}$, it follows that $|A'| \geq n - L$.

Let $H = S - A'$. Then $|H| = n - |A'| \leq L$. To prove that H is a hitting set for D, assume otherwise. That is, there is a subset $D_i \in D$ such that $D_i \cap H = \emptyset$. Since in addition $A' = S - H$, and $D_i \subseteq S$, it follows that $D_i \subseteq A'$. Since $D_i \in C$, this implies that the set of assumptions A' is inconsistent, which contradicts that fact that E is an explanation. □

3.2.2 Heuristic Search

Since our abduction problem of finding the most coherent explanation contains as a special case the abduction model formalized in the proof, our problem is clearly computationally intractable. This justifies the use of heuristic beam search in the AAA algorithm to compute the (approximately) best explanations, where the best explanations are those with the highest coherence metric, and ties are broken based on the simplicity metric of $1/A$ (A is the number of assumptions made).

In the plan recognition domain, consistency checking ensures that an object cannot be of two non-compatible sorts. This is accomplished using a subsort-supersort hierarchy in the knowledge base that explicitly gives the various sort relationship, like a gun is a weapon, a bus is a vehicle, etc. Consistency checking also enforces temporal constraints, such as the first substep of a plan precedes its second substep, a plan cannot contain itself as a substep, etc. Such consistency checking is achieved via procedural code. See [44] for more details.

3.3 Empirical Results

Evaluating natural language processing systems is becoming an increasingly important issue. For instance, there is ongoing work to evaluate NLP systems that perform information extraction from unconstrained texts [38]. We have completed an evaluation of ACCEL on a test suite of 25 examples taken from Robert Goldman's PhD thesis [27]. We chose this set of examples to test our system since we are aware of no other pre-existing set of test data for plan recognition, and it also facilitates comparison between different approaches.

The knowledge base was initially constructed so as to handle this set of 25 examples. In order to test for generality, Ray Mooney came up with another 25 test examples unbeknown

to the knowledge base builder (Hwee Tou Ng). The intent is that these additional examples will test for other novel combinations and sequences of actions that the knowledge base constructed for the initial 25 examples in principle should be able to handle. We will call the first set of 25 examples the *training* examples, and the second set of 25 examples the *test* examples. Note that our evaluation methodology is similar to that of Goldman, except that we tested ACCEL on a *different* set of 25 test examples whereas Goldman tested his system's ability to pair up an additional set of 25 *similar* examples to the initial 25 examples. Hence, our evaluation criterion is tougher.

Examples of the 50 narrative texts processed by ACCEL include: "Bill went to the liquor-store. He pointed a gun at the owner."; "Bill took a bus to a restaurant. He drank a milkshake. He pointed a gun at the owner. He got some money from him."; "Fred got a gun. He went to the restaurant. He packed a suitcase."; etc. The knowledge base axioms are formulated such that higher-level plans (like shopping and robbing) together with appropriate role-filler assumptions (like someone is the shopper of a shopping plan or the robber of a robbing plan) imply the input atoms representing the observed actions (like going to a store and pointing a gun). Below are some examples of the knowledge base axioms:

$$\begin{aligned} \mathit{inst}(G, \mathit{going}) &\leftarrow \mathit{inst}(S, \mathit{shopping}) \wedge \mathit{go\text{-}step}(S, G) \\ \mathit{goer}(G, P) &\leftarrow \mathit{inst}(S, \mathit{shopping}) \wedge \mathit{go\text{-}step}(S, G) \wedge \\ &\quad \mathit{shopper}(S, P) \\ \mathit{dest\text{-}go}(G, P) &\leftarrow \mathit{inst}(S, \mathit{shopping}) \wedge \mathit{go\text{-}step}(S, G) \wedge \\ &\quad \mathit{store}(S, P) \end{aligned}$$

The first axiom asserts that if S is a shopping event and the go-step of S is G, then G is a going event; the second axiom asserts that if S is a shopping event, the go-step of S is G, and the shopper of S is P, then the goer of G (i.e., the agent of the going event G) is P; and so on.

The plans in the knowledge base include shopping, robbing, restaurant dining, traveling in a vehicle (bus, taxi, or plane), partying, and jogging. Each of these plans in turn has subplans, and some of the plans contain recursive subplans. For instance, traveling by plane includes the subplan of traveling (in some vehicle) to the airport to catch a plane. For each example, a set of input atoms representing the sentences is given to ACCEL. To give a sense of the size of our examples and the knowledge base used, there is a total of 107 KB rules, 45 assumption-nogoods, and 70 taxonomy-sort symbols. Every taxonomy-sort symbol p will add an axiom (in addition to the 107 KB rules) of the form $\mathit{inst}(X, p) \rightarrow \mathit{inst}(X, \mathit{supersort\text{-}of\text{-}p})$. The average number and maximum number of input atoms per example are 12.6 and 26 respectively. The knowledge base and the 50 examples are included in [44].

For each example, the correct explanation was determined based on the authors' intuition before running the example. To measure the quality of an explanation computed by ACCEL, we compared it to the correct explanation and recorded three error rates: the recall error rate R = the number of missing assumptions divided by the number of assumptions in the correct explanation, the precision error rate P = the number of excess assumptions

Example type	Coherence			Simplicity		
	R	P	O	R	P	O
Training	0.2%	0%	0.1%	26%	25%	25%
Test	2%	2%	2%	39%	38%	38%
All	1.1%	1%	1%	32%	31%	32%

Table 3: Empirical results comparing coherence and simplicity.

divided by the number of assumptions in the computed explanation, and the overall error rate O = the average of the recall and precision error rates. (We used similar quality measures and terminology as in [38].) If more than one best explanations are computed for an example, we take the error rates for the example to be the average of the error rates over all the best explanations.

We set the beam widths of AAA at $\beta_{inter} = 10$ and $\beta_{intra} = 30$ when processing the set of 50 examples. We ran ACCEL on the 50 examples using two different evaluation metrics: the coherence metric (breaking ties based on simplicity) and the simplicity metric. The empirical results are summarized in Table 3, which shows the average recall (R), precision (P), and overall (O) error rates for the training examples, test examples, and all examples. The average run time per example is 1.83 minutes on a Sun Sparc 2 workstation.

The empirical results demonstrate that ACCEL can efficiently process these narrative texts, and it is sufficiently general to be able to handle similar plan recognition problems not known to the system developer in advance. Furthermore, coherence consistently performs better than simplicity on the examples tested.

3.4 Comparison with the Probabilistic Approach

Charniak and Goldman [9; 8] have adopted the Bayesian probabilistic approach to plan recognition and text understanding. In this approach, an explanation is selected based on the conditional probabilities of the abduced events given the observations stated in the input text. As mentioned earlier, the first 25 training examples used in our system were taken from Goldman's thesis and these examples have been successfully processed based on finding the most probable explanation. Note that ACCEL achieved results similar to Goldman's system on the 25 training examples. Almost all the best explanations are found even though our knowledge base does not contain any probabilistic or likelihood information.

This may seem surprising. In the probabilistic approach, the primary purpose of *a priori* probabilities is to select a most likely explanation when there are otherwise multiple competing explanations. For instance, in the sentence "John went to the supermarket.", a higher *a priori* probability of someone shopping at the supermarket as compared to robbing the supermarket enables the supermarket shopping interpretation to be selected over the supermarket robbing interpretation. In our system, we achieve an analogous effect by having an axiom in the knowledge base that explains supermarket in terms of supermarket shopping, but the knowledge base does *not* have the corresponding axiom for "supermarket robbing" that explains supermarket. That is, we have the following axiom in the knowledge base:

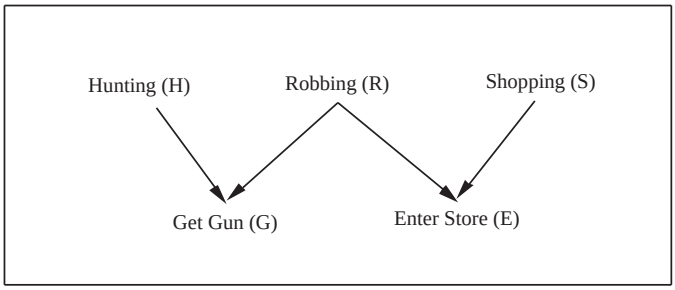

Figure 5: The Bayesian network for "John got a gun. He entered the grocery store."

$inst(S, smarket\text{-}shopping) \wedge store(S, P) \rightarrow inst(P, smarket)$
but *not*
$inst(S, smarket\text{-}robbing) \wedge store(S, P) \rightarrow inst(P, smarket)$
This is justified since supermarket shopping is a commonly occurring plan, but not supermarket robbing. (In fact, the knowledge base does not have the high-level plan supermarket-robbing. Only robbing is present as a high-level plan in the knowledge base.) It then follows that supermarket shopping is a more coherent interpretation of "John went to the supermarket." since supermarket shopping explains and thus connects both supermarket and the going action, whereas robbing only explains the going action but not supermarket.

With this style of axiomatization and the use of the coherence metric, ACCEL is able to select the correct explanation without resorting to the use of numeric probabilities. In essence, what is achieved by numeric probabilities in the probabilistic approach is accomplished by the judicious use of logical axioms. This is in contrast to the probabilistic approach which critically depends on knowledge about the numerous prior and posterior probabilities of the nodes in a Bayesian network constructed from the input sentences. In practice, such knowledge may not always be available in the required form. For example, in the probabilistic approach, in order to understand the sentences "John got a rope. He killed himself.", one needs to know the prior probability of a hanging event, the prior probability of an entity being a rope, etc, which in turn necessitates making the assumptions that there are 10^{20} things in the world, out of which there are 10^{9} ropes, 10^{15} get events, 10^{3} hangings, etc [9]. Engineering an appropriate set of probabilities is a major weakness of the probabilistic approach to text understanding.

In addition, the most probable interpretation selected depends quite critically on the specific values assigned to the various probabilities, and reasonable probability values may result in the wrong interpretation being selected. For example, consider the sentences "John got a gun. He entered the grocery store." Figure 5 shows a simple Bayesian network constructed for these sentences.

Suppose we adopt the following reasonable estimates for the various prior and posterior probabilities: $P(h) = 10^{-3}$; $P(r) = 10^{-5}$; $P(s) = 10^{-2}$; $P(g \mid h, r) = 0.95$; $P(g \mid h, \overline{r}) = P(g \mid \overline{h}, r) = 0.90$; $P(g \mid \overline{h}, \overline{r}) = 0.01$; $P(e \mid r, s) = 0.95$; $P(e \mid r, \overline{s}) = P(e \mid \overline{r}, s) = 0.90$; and $P(e \mid \overline{r}, \overline{s}) = 0.01$. Note that under these probability estimates, shopping is more probable than either hunting or robbing *a priori*. From these estimates, we can compute

$P(r \mid g,e) = 0.038$, $P(h \mid g,e) = 0.080$, $P(s \mid g,e) = 0.459$, and $P(h,s \mid g,e) = 0.038$. Since the conditional probability of shopping given the observations get-gun and enter-store is the highest, the chosen interpretation is that John is shopping in the store! Even though the chosen probability estimates are quite reasonable, the preferred interpretation that John is robbing the grocery store is not selected. Furthermore, suppose all the above prior and posterior probabilities remain the same except that $P(r) = 1.222 * 10^{-4}$. Then $P(r \mid g,e) \approx P(s \mid g,e) \approx 0.3249$, and for $P(r) > 1.222 * 10^{-4}$, $P(r \mid g,e)$ becomes the highest of all the conditional probabilities. This suggests that the selected interpretation is quite sensitive to slight variation in the estimated subjective probabilities. Note that the correct interpretation that John is robbing the store will be selected using our coherence metric, since the robbing interpretation has a positive coherence value compared to the zero coherence of hunting or shopping.

Besides the problem of engineering the numerous prior and posterior probabilities of the nodes in a Bayesian network, the probabilistic approach does not take into account the importance of text coherence. Selecting an interpretation based solely on the probability of propositions about the situation being described is ignoring the fact that these propositions are adjacent sentences in a natural language text, not just random facts observed in the world.

Cost-based abduction is another scheme proposed by [30] to select an interpretation based on the cost of an abductive proof. However, as shown in [11], cost-based abduction can be given a probabilistic semantics. Therefore, cost-based abduction can be regarded as a kind of probabilistic approach, and it suffers from the same problems.

In summary, our method yields the correct interpretations without the heavy machinery of the probabilistic approach, and consistently produces more accurate interpretations than a metric based on simplicity. The approach generalized well to novel test examples. Our empirical results indicate that maximizing connections between observations is an important property of a good explanation in plan recognition.

4 Diagnosis Based on Set Covering

4.1 Generalized Set Covering

Over the past decade, Reggia and his colleagues have developed an increasingly sophisticated theory of diagnosis, the Generalized Set Covering (GSC) model, and applied the theory primarily to medical disease diagnosis [55].

Definition 4.1 *The basic diagnostic problem in the GSC model is defined by four sets:* (D, M, C, M^+)

D: *A finite set of potential disorders*

M: *A finite set of potential manifestations (symptoms)*

$C \subseteq D \times M$: *A causation relation where* $(d, m) \in C$ *means "d may cause m"*

$M^+ \subseteq M$: *The set of observed manifestations for the current case*

$E \subseteq D$ is called a cover *of M^+ iff for each $m \in M^+$ there exists $d \in E$ such that $(d, m) \in C$. A cover is said to be* minimum *if its cardinality is the smallest among all covers and* irredundant *(*minimal*) if none of its proper subsets is also a cover.*

Depending on the domain, one may consider all minimum or all minimal covers of the observed symptoms as the best diagnoses.

4.2 Set-Covering-Based Diagnosis as Abduction

We can map a GSC diagnostic problem into an abduction problem in ACCEL as follows: Let the domain theory T be the set of axioms $\{d \rightarrow m | (d, m) \in C\}$, and let the input atoms $O = \bigwedge_{m \in M^+} m$. We use predicate specific abduction such that only atoms $d \in D$ are assumable.

Theorem 4.1 *The set of covers of GSC = the set of explanations in* ACCEL.

Proof Let $E = \{d_1, \ldots, d_r\}, M^+ = \{m_1, \ldots, m_s\}$.
($\Rightarrow$) Let E be a cover of M^+.
Note that $d_1 \wedge \ldots \wedge d_r \wedge T$ is consistent.
Since E is a cover of M^+, for each $m_j \in M^+$, there exists $d_i \in E$ such that $(d_i, m_j) \in C$. That is,

$$d_i \wedge T \models m_j$$

It follows that $d_1 \wedge \ldots \wedge d_r \wedge T \models m_j$ for each $m_j \in M^+$. Hence,

$$d_1 \wedge \ldots \wedge d_r \wedge T \models m_1 \wedge \ldots \wedge m_s$$

Therefore, E is an explanation for $m_1 \wedge \ldots \wedge m_s$.
($\Leftarrow$) Let E be an explanation for $m_1 \wedge \ldots \wedge m_s$. That is,

$$d_1 \wedge \ldots \wedge d_r \wedge T \models m_1 \wedge \ldots \wedge m_s$$

Since $T = \{d \rightarrow m | (d, m) \in C\}$, it follows that for each $m_j \in M^+$, there exists $d_i \in E$ such that $(d_i, m_j) \in C$. (For if there is no such $d_i \in E$, then $d_1 \wedge \ldots \wedge d_r \wedge T \wedge \neg m_j$ is consistent, contradicting $d_1 \wedge \ldots \wedge d_r \wedge T \models m_1 \wedge \ldots \wedge m_s$.)
Hence E is a cover of M^+. □

It follows from this theorem that the set of all *minimal* covers of GSC is identical to the set of all *minimal* explanations in ACCEL.[2]

Since the logical abduction approach is based on a more expressive representation language, it can accommodate more naturally "causal chaining" [55], incompatible disorders, and symptoms caused by combinations of disorders. Causal chaining can be achieved in logical abduction by allowing backward-chaining of depth greater than one. Incompatible disorders can be enforced through consistency checking by adding nogoods $d_1 \wedge d_2 \rightarrow false$.

[2] Actually, that all minimal covers of GSC are all minimal explanations in abduction also follows as a corollary of two published theorems, Theorem 7.1 in [63] and Theorem 4.2 in [57].

A symptom s that is caused by multiple, simultaneous disorders $d_1, \ldots, d_n$ can be encoded as $d_1 \wedge \ldots \wedge d_n \rightarrow s$.

A standard concern with the logical approach is that a disorder may not always cause all of its manifestations. In this case, the axiom $d \rightarrow m$ is too strong since assuming d would be inconsistent if $\neg m$ is observed. As described in [57], this problem is easily handled by making $d \rightarrow m$ a potential assumption rather than an axiom when the symptom is not deterministic. If assumptions are required to be atomic (as in ACCEL), then one can achieve the same effect by adding an extra unique antecedent to rules for nondeterministic symptoms, $d \wedge a \rightarrow m$, where a represents the assumption that d actually causes m in the current case.[3]

4.3 Empirical Results

Since GSC diagnostic problems can be nicely represented as abduction problems, the remaining question is whether a general logic-based abductive system can solve such problems efficiently. Furthermore, because the GSC diagnostic problem is NP-hard [62], the issue then becomes whether a logical abductive system can solve real problems in reasonable time and is competitive with existing set-covering algorithms. To address this issue, we tested ACCEL on the medical problem studied in [73], which involves determining the areas of the brain that were damaged in a stroke. There are a total of 25 brain areas (e.g. right frontal lobe) whose damage can explain 37 basic symptom types (e.g. impaired gag reflex). The knowledge base is quite large, consisting of 648 rules of the form: $d \rightarrow m$. We were only able to obtain 50 of the original 100 cases from the authors of the initial study, each consisting of an average of 8.56 symptoms.

ACCEL efficiently computed all of the minimal (w.r.t. subset) explanations in an average of 2.4 seconds per case on a Sun Sparc 2 workstation. Unfortunately, we could not compare this result to that obtained in the original study, since no information on run time was provided. However, the empirical results strongly suggest that a general abductive system can solve real diagnostic problems in reasonable time.

Since abduction computes the same explanations as set covering when given the same evaluation criteria, ACCEL should replicate the accuracy results of the original study. As discussed in the original study, minimality is too unrestrictive to produce useful results (ACCEL returned an average of 26.6 minimal diagnoses per case). With minimum cardinality, ACCEL produced an average of only 4.6 diagnoses per case. In 44% of the cases, one of these diagnoses matches the expert's exactly; and in another 46% of the cases, one of the system's diagnoses was a subset or superset of the expert's (called a "close match" in [73]). The remaining 10% of the cases have a diagnosis that either partially matches the expert's (2%) or all of the diagnoses are totally wrong (8%). These results are slightly better than those reported in the original study: 6.5 diagnoses/case with 40% exact, 38% close, 5% partial, 17% wrong. This is presumably due to the fact that our results are based on only 50 of the original 100 cases. Two other evaluation metrics reported in the original

[3] If minimum covers are desired, then the extra assumptions should not count as contributing to the size of the cover.

study, most-probable and minimum-collapsed, performed even better. In [73], it is claimed that, although there have been no direct comparisons, the results from any of the covering metrics appear more promising than those obtained from standard rule-based approaches to this problem.

In summary, the results presented in this section demonstrate that our general-purpose logic-based abductive system can effectively represent and efficiently solve large realistic problems suitable for set-covering methods. Consequently, the desirability of the existing special-purpose approach for such problems is lessened. The logical approach is more general and flexible, yet capable of efficiently solving problems in this more restrictive class.

5 Model-Based Diagnosis via Abduction

5.1 Introduction

ACCEL also performs model-based diagnosis, which concerns inferring faults from first principles given knowledge about the correct structure and behavior of a system. The model-based approach to diagnosis has some advantages over the associational, heuristic rule-based approach of conventional expert systems. The model-based approach is compositional in that it lets us define models for a library of basic components, and it works on all systems composed from those components. The system designer can focus on getting the component models right, leading to more robust and sound diagnostic systems. The potential is also better for verification of the underlying knowledge base.

Much research in model-based diagnosis has taken the *consistency-based* approach and has been applied primarily to devices with static, persistent states such as combinational logic circuits [15; 17; 63; 18]. In the consistency-based approach, a diagnosis is a set of normality and abnormality assumptions about device components that are *consistent* with the observations and the system description. This is in contrast to the *abductive* approach of diagnosis used in ACCEL, where normality and abnormality assumptions about device components together with the system description must *imply* or *explain* the observations.

Poole has proved that the consistency-based and abductive approaches are equivalent for propositional theories [57], and Konolige has extended the conditions under which equivalence holds to general first-order causal theories allowing for correlations, uncertainty, and acyclicity in the causal structure [33].[4] In view of such formal equivalence results, issues such as ease of representation and computational efficiency are most important. Our empirical results suggest that a number of diagnostic problems, ranging from combinational logic circuits to continuous dynamic systems such as a proportional temperature controller and the water balance system of the human kidney, can be effectively represented and efficiently diagnosed using an abductive approach.

Research in model-based diagnosis can also be classified according to whether information about fault models is utilized in diagnosis. The *normality-based* approach of [63;

[4] Abduction appears to be better in some cases, as Konolige has reported that "the utility of the consistency based method is open to question", since in explanatory diagnostic tasks, "the answers it produces may have elements that are not relevant to a causal explanation" [33, page 257].

17] does not utilize fault models and any misbehavior differing from the correct functioning of a device can be diagnosed. However, the lack of fault models may result in hypothesizing implausible faults [18; 69]. On the other hand, the work of [21] is *fault-based* in that the fault models are *a priori* determined and given to the diagnostic system. Hence, unanticipated faults are not detected. ACCEL combines both normality-based and fault-based diagnosis in that information about fault models is used in diagnosis and any deviation from the correct behavior can be diagnosed. The diagnostic systems Sherlock [18] and GDE+ [69] have similar capability.

In the model-based diagnosis domain, ACCEL uses predicate specific abduction, where the assumable atoms include component behavioral mode assumptions of three types: (1) a component is normal; (2) a component is in some known fault mode; or (3) a component is abnormal (but not necessarily in any known fault mode). Other assumable atoms are "auxiliary" assumptions including assumptions that the input values of a device are as given, and in dynamic system diagnosis, that some qualitative magnitude is positive/negative, that two qualitative values obey some corresponding value constraint, etc. (More details about these auxiliary assumptions will be provided later.) Explanations in this domain are evaluated based on simplicity, where the best explanation is one with the least number of components that are not normal, which include components that are in some known fault mode and those that are not. Normality assumptions and auxiliary assumptions are "free" and do not affect the simplicity metric of an explanation. If two explanations have the same number of components that are not normal, then the one with the most number of components that are in some known fault mode is preferred.

5.2 Diagnosing Logic Circuits

5.2.1 Representation

In this section, we describe how the abductive approach of ACCEL is used to diagnose a full adder which is representative of standard, combinational logic circuits. Figure 6(a) shows a full adder which consists of 2 exclusive-or gates (x1, x2), two and gates (a1, a2), and one or gate (o1). We assume that each gate has 4 behavioral modes: normal (the output bit reflects the correct gate behavior at all times), stuck-at-0 (the output bit is stuck at 0 regardless of the input bits), stuck-at-1 (the output bit is stuck at 1 regardless of the input bits), and abnormal (the behavior of the gate is unconstrained).

The knowledge base axiom that describes the correct behavior of an exclusive-or gate is:

$$out(X,W,T) \quad \leftarrow \quad xorg(X) \wedge in1(X,U,T) \wedge in2(X,V,T) \wedge norm(X) \wedge xor(U,V,W)$$

The axiom asserts that if X is an exclusive-or gate ($xorg(X)$), the first input of X is U at time T ($in1(X,U,T)$), the second input of X is V at time T ($in2(X,V,T)$), X is normal ($norm(X)$), and the exclusive-or of U and V is W ($xor(U,V,W)$), then the output of X is W at time T ($out(X,W,T)$). In addition we have the facts $xor(0,0,0)$, $xor(0,1,1)$, $xor(1,0,1)$, and $xor(1,1,0)$. The axioms for and gates and or gates are similar.

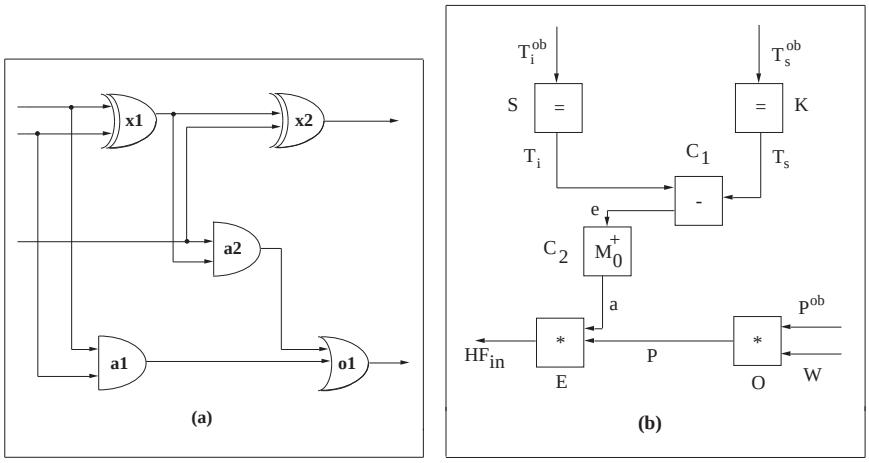

Figure 6: (a) Full adder; (b) Temperature controller.

The following axiom describes the fault mode *stuck-at*-0 for all gates:

$$out(X, 0, T) \leftarrow in1(X, U, T) \wedge in2(X, V, T) \wedge stuck\text{-}at\text{-}0(X)$$

The axiom for the fault mode *stuck-at*-1 is similar. Note that when a gate is assumed to be abnormal, no prediction can be made about its output bit. However, abduction requires that the observations be proved from the component behavioral mode assumptions (including the abnormality assumptions). To overcome this problem, we employ a technique used by Poole to "parameterize" the abnormality assumption as follows [59]:

$$out(X, W, T) \leftarrow in1(X, U, T) \wedge in2(X, V, T) \wedge ab(X, U, V, W, T)$$

The antecedent $ab(X, U, V, W, T)$ in the rule is to be interpreted as "X is abnormal in such a way that at time T, given input bits U and V, its output bit is W". Note that for any input bits U and V, and any output bit W, the above axiom always allows us to assume that the component is abnormal by making the assumption $ab(X, U, V, W, T)$. This axiom achieves our objective of being able to prove the output observations from the parameterized abnormality assumption $ab(X, U, V, W, T)$.

So far, the axioms given are not specific to the full adder; they are used to model the behavior of exclusive-or gates, and gates, and or gates. We also need axioms that specify the connections among the gates in the given adder, such as

$$in1(a1, X, T) \leftarrow in1(x1, X, T)$$

as well as facts that identify the five components: $xorg(x1), xorg(x2)$, etc. Furthermore, in order to allow backward-chaining to terminate at the terminal input values of the full adder (these terminal input values cannot be further explained in terms of the other gate values), we need the axiom

$$in1(x1, X, T) \leftarrow given\text{-}in1(x1, X, T)$$

and two other similar axioms for the second input of $x1$ and the first input of $a2$. We let *given-in*1(...) (and *given-in*2(...)) be assumable. They are the auxiliary assumptions, and do not affect the simplicity metric of an explanation.

5.2.2 Empirical Results

To assess the performance of ACCEL, we randomly generated 10 scenarios by assuming that the various behavioral modes of each gate occur with the following probabilities: *norm* 65%, *stuck-at*-0 15%, *stuck-at*-1 15%, and *ab* 5%. Each of the 10 scenarios that was actually generated had one or two gates that were faulty, and the scenarios included some where a gate was abnormal (*ab*). For each scenario, we gave ACCEL I/O tuples where the input-output bits of the adder differed from those of a correctly functioning adder. (By an I/O tuple, we mean a particular combination of input and output values of the full adder.) For each I/O tuple, we first gave the three input bits and the two output bits of the adder, and then the output bits of the three gates x1, a1, and a2, in that order. For each scenario, we stopped as soon as the best diagnosis found by ACCEL is the correct diagnosis. We recorded the number of I/O tuples needed to converge on the correct diagnosis for each scenario. On a Sun Sparc 2 workstation, ACCEL took an average of 17 seconds to identify the correct diagnosis for the 10 scenarios tested. The average number of I/O tuples needed before the correct diagnosis was found is 2.1.

In summary, the abductive approach of diagnosis used in ACCEL can effectively represent and efficiently diagnose logic circuits of the kind previously diagnosed by the consistency-based approach.

5.3 Diagnosing Dynamic Systems

5.3.1 Representation

Much research in model-based diagnosis has focused on diagnosing static, discrete devices like logic circuits. However, many devices and biological systems are continuous and dynamic and require reasoning about changes in behavior over time. Although there has been a great deal of research on modeling and simulating such systems [35; 23], there have been few attempts to apply general, model-based diagnostic methods to them. The work of [43; 42] attempts to address this deficiency by diagnosing dynamic systems using the consistency-based approach. In this section, we present an abductive approach to diagnosing continuous, dynamic systems.

We adopt the representation of continuous dynamic systems used in the work of Kuipers' qualitative simulation (QSIM) [35]. The continuously changing behavior of a dynamic system over time is represented as a sequence of qualitative states, where a qualitative state consists of the qualitative values of the variables of the system. A qualitative value has two components: a qualitative magnitude (*qmag*) and a qualitative direction (*qdir*). A qualitative magnitude can either be a landmark value or an open interval between two landmark values, where a landmark value is a value of special significance that a variable takes on at some point in time. A qualitative direction can be one of increasing (*inc*), decreasing (*dec*), or steady (*std*).

The behavior of each dynamic system is governed by a set of qualitative constraints. The qualitative constraints on the temperature controller (Figure 6(b)) include (each constraint is preceded by a name identifying that constraint) $S : T_i^{ob} = T_i$, $K : T_s^{ob} = T_s$, $C_1 : T_s - T_i = e$, $C_2 : m_0^+(e) = a$, $O : P^{ob} \cdot W = P$, and $E : a \cdot P = HF_{in}$. The $m_0^+(e) = a$ constraint asserts that there is a strictly monotonically increasing function between e and a. However, the exact form of this monotonic function is unspecified. This accounts for the qualitative nature of the constraint. The purpose of this device is to control the temperature T_i^{ob} in the room, so that if the device is connected to a power source with power P^{ob}, the power switch is turned on (represented as $W = on$), and the temperature T_s^{ob} set by the temperature control knob differs from the temperature T_i^{ob} in the room, heat flow HF_{in} (in the form of hot air or cold air, depending on the direction of temperature difference) will be generated. Furthermore, the amount of heat flow generated is proportional to the temperature difference $T_s^{ob} - T_i^{ob}$.

We have successfully represented QSIM's knowledge about the various qualitative constraints $(=, -, \cdot, /, d/dt, m_0^+)$ in Horn-clause axioms in a way suitable for logic-based abductive diagnosis. Since these Horn-clause axioms encode general knowledge about QSIM constraints, they are needed in the diagnosis of every dynamic system. These axioms encode the various qualitative constraints by defining a "holds.constraint-type" predicate for each type of qualitative constraint. For example, one of the 9 axioms that encode the m_0^+ constraint is:

$$holds.m_0^+(F, G, M1, inc, M2, inc) \leftarrow pos(M1) \wedge pos(M2) \wedge corr\text{-}mag.m_0^+(F, G, M1, M2)$$

$holds.m_0^+(F, G, M1, D1, M2, D2)$ asserts that $m_0^+(F) = G$ holds with the qualitative value of the variable $F = \langle M1, D1 \rangle$ and the qualitative value of the variable $G = \langle M2, D2 \rangle$. $pos(M1)$ ($neg(M1)$) asserts that the qualitative magnitude $M1$ is positive (negative). $corr\text{-}mag.m_0^+(F, G, M1, M2)$ asserts that $m_0^+(F) = G$ holds with the qualitative magnitude of $F = M1$ and the qualitative magnitude of $G = M2$. In QSIM, $(M1, M2)$ are referred to as corresponding values. The 9 axioms for the m_0^+ constraint cover all the distinct possibilities in which $m_0^+(F) = G$ holds since the qualitative magnitude of F can be positive, negative, or zero, and its qualitative direction can be *inc*, *std*, or *dec*. The other "holds.constraint-type" predicates, $holds.-$, $holds.*$, $holds./$, and $holds.d/dt$, are defined by 39, 97, 70, and 9 axioms, respectively. The axioms for $holds.*(F, G, H, M1, D1, M2, D2, M3, D3)$ ensure that, among other things, the first-order derivative constraint $F \cdot G' + F' \cdot G = H'$ is obeyed. The exact axioms for all the qualitative constraints are listed in [44].

Besides the axioms that encode general QSIM constraints, there are also Horn-clause axioms that encode knowledge about a specific dynamic system. We assume in this paper that a dynamic system malfunctions because of one or more violated constraints, and that the task of mapping from violated constraints to the affected components is done by some other module external to ACCEL. The following axioms describe the normal behavior:

$$\begin{aligned} qval(ti, M1, D1, T) &\leftarrow norm(s) \wedge qval(ti\text{-}ob, M1, D1, T) \\ qval(e, M3, D3, T) &\leftarrow norm(c1) \wedge qval(ts, M1, D1, T) \wedge qval(ti, M2, D2, T) \wedge \\ &\quad\ holds.-(ts, ti, e, M1, D1, M2, D2, M3, D3) \end{aligned}$$

$qval(ti, M1, D1, T)$ asserts that the qualitative value of the variable ti is $\langle M1, D1\rangle$ at time (qualitative state) T. The first axiom asserts that if constraint s is normal, and the qualitative value of *ti-ob* is $\langle M1, D1\rangle$ at time T, then the qualitative value of ti is also $\langle M1, D1\rangle$ at time T. This encodes the equality constraint between the variables *ti-ob* and ti. The second axiom asserts that if constraint $c1$ is normal, the qualitative value of ts is $\langle M1, D1\rangle$ at time T, the qualitative value of ti is $\langle M2, D2\rangle$ at time T, and $ts - ti = e$ holds with $ts = \langle M1, D1\rangle, ti = \langle M2, D2\rangle, e = \langle M3, D3\rangle$, then the qualitative value of e is $\langle M3, D3\rangle$ at time T. Similar axioms encode the other constraints.

Note that atoms with the predicate $qval$ are not assumable. As such, in order to allow backward-chaining to terminate at the terminal input values of a dynamic device (these terminal input values cannot be further explained), we also need the axiom

$$qval(\textit{ti-ob}, M1, D1, T) \leftarrow \textit{given-qval}(\textit{ti-ob}, M1, D1, T)$$

and three other similar axioms for *ts-ob*, *p-ob*, and w. We let $\textit{given-qval}(\ldots)$ be assumable. They are part of the "auxiliary" assumptions in an abductive explanation.

Note the directionality in which one qualitative value is explained in terms of other qualitative values. Since abductive diagnosis requires that the input observations (which consists of the qualitative values of the variables of a dynamic system) be *proved*, the axioms are formulated in such a way that the output values (e.g., $qval(hfin, \ldots)$) of a dynamic system can be proved from normality assumptions (e.g., $norm(s)$), fault mode assumptions, and auxiliary assumptions about the input values (e.g., $\textit{given-qval}(\textit{ti-ob}, \ldots)$) and the qualitative magnitudes and corresponding values of the variables (these are introduced when ACCEL attempts to prove the holds.constraint-type atoms).

For the temperature controller, we also assume that the components corresponding to the various constraints exhibit the following fault modes: stuck-at-0-std (S, K, C_1, C_2, O, E), stuck-at-roomtemp-std (S), stuck-at-1st-in (C_1, O), and stuck-at-2nd-in (C_1). Under the fault mode stuck-at-0-std (stuck-at-roomtemp-std), the output of a component is $\langle 0, std\rangle$ ($\langle \textit{room-temp}, std\rangle$) regardless of the input values. Under the fault mode stuck-at-1st-in (stuck-at-2nd-in), the output of a component is stuck at its first (second) input. One Horn-clause axiom is used to encode one fault mode, as follows:

$$\begin{aligned} qval(ti, 0, std, T) &\leftarrow \textit{stuck-at-0-std}(s) \wedge qval(\textit{ti-ob}, M1, D1, T) \\ qval(e, M1, D1, T) &\leftarrow \textit{stuck-at-1st-in}(c1) \wedge qval(ts, M1, D1, T) \wedge qval(ti, M2, D2, T) \end{aligned}$$

The Horn-clause axioms in ACCEL that represent the qualitative constraints capture the knowledge that QSIM uses to propagate qualitative values across constraints in order to complete the qualitative values of variables in a qualitative state. In ACCEL, such knowledge is used for the purpose of diagnosis. However, since the knowledge is now encoded declaratively, it can also be used for simulation purpose by a forward-chaining inference procedure. In fact, QSIM can be viewed as a special-purpose theorem prover for predicting the behavior of dynamic systems described by qualitative constraints. However, not all of QSIM's knowledge in simulation has been captured in ACCEL. Specifically, knowledge of state transition that QSIM uses to generate the next qualitative state(s) from an initial qualitative state is not encoded in ACCEL, since such knowledge is not needed in diagnosis.

5.3.2 Empirical Results

We randomly generated 10 scenarios for the temperature controller where each scenario contains one to two faults and in which no heat flow was generated into the room. For each scenario, we gave the input atoms representing the qualitative values of the variables in the following order: $T_s^{ob}, T_i^{ob}, P^{ob}, W, HF_{in}$ at the initial qualitative state (t_1); $T_s^{ob}, T_i^{ob}, P^{ob}, W, HF_{in}$ at the next distinguished time-point qualitative state (t_2); and the intermediate variables T_s, T_i, e, a, P at state t_2.

In 9 out of the 10 scenarios, ACCEL found the correct diagnosis as its best diagnosis. The one scenario in which ACCEL failed to find the best diagnosis has two faults $\{$*stuck-at*-0-*std*$(c1)$, *stuck-at*-0-*std*$(c2)\}$. In this case, the best diagnosis that ACCEL found after processing all the intermediate variables is $\{$*stuck-at*-0-*std*$(c1)\}$. This is as it should be, since when $c1$ is stuck at $\langle 0, std\rangle$, the correct behavior of $c2$ if it is normal is to output $a = \langle 0, std\rangle$ at all times, which is indistinguishable from the behavior of $c2$ if it is in the fault mode *stuck-at*-0-*std*. That $c2$ is in fact faulty would be detected when $c1$ is replaced by a normal, working component and the controller is still found to be malfunctioning. Overall, the average run time per scenario is 4.24 minutes, and the average number of measurements of intermediate variables needed to arrive at the correct diagnosis is 4.4.

We also tested ACCEL on 10 faulty scenarios of the kidney water balance system, a QSIM model of which is given in [34; 36]. The system has 7 qualitative constraints and 10 qualitative variables. Two of the scenarios tested correspond to the disorders Diabetes Insipidus and the Syndrome of Inappropriate Secretion of Anti-Diuretic Hormone (SIADH), which are disorders found in real patients. ACCEL found the correct diagnosis as its best diagnosis in all the 10 scenarios. The average run time per scenario is 6.98 minutes, and the average number of measurements of intermediate variables needed to arrive at the correct diagnosis is 3.7.

The empirical results indicate that ACCEL is capable of efficiently diagnosing dynamic systems of the kind modeled by QSIM, and that first-order Horn-clause axioms can effectively represent the qualitative constraints of such systems for the purpose of efficient model-based diagnosis.

6 The Utility of Caching

To substantiate our claim that caching is indeed very important in improving the efficiency of abduction, we ran ACCEL with and without caching on a randomly selected set of problems from each domain. We would have run ACCEL on all the problems except that some of them took too long (more than one hour) to run without caching, and frequently did not successfully run to completion before overflowing memory.

For each problem selected, we ran AAA in its normal caching mode as well as in non-caching mode. If caching is not done, each time when *compute-label* returns all the abductive proofs of a subgoal, these proofs are not cached, so that the next time when the same subgoal is encountered again, its abductive proofs are recomputed. Table 4 shows the performance figures comparing the caching and non-caching versions. In the table, run time is the elapsed time (in minutes) on a Sun Sparc 2 workstation running Lucid

Problem	Time (min)			Inference count		
	No-cache	Cache	speedup	No-cache	Cache	ratio
train1	22.19	3.10	7.16	1418	138	10.27
train8	0.43	0.24	1.79	118	45	2.62
train13	11.42	2.02	5.65	919	120	7.66
test9	>25.04	2.96	>8.46	>1401	127	>11.03
test19	>26.78	4.17	>6.42	>1561	209	>7.47
brain5	0.16	0.07	2.29	2198	142	15.48
brain10	0.07	0.03	2.33	1004	63	15.94
brain42	0.05	0.02	2.50	783	100	7.83
adder1	43.77	0.51	85.82	157210	749	209.89
adder4	3.81	0.08	47.63	30225	237	127.53
adder10	4.79	0.12	39.92	38110	255	149.45
tc1	>72.53	4.70	>15.43	>17766	442	>40.19
tc4	>68.08	4.35	>15.65	>17766	415	>42.81
tc8	>68.22	4.26	>16.01	>17766	385	>46.15
kidney1	>61.24	7.86	>7.79	>8661	470	>18.42
kidney5	>65.26	7.89	>8.27	>8577	433	>19.81
Average			17.07			45.78

Table 4: Empirical results comparing caching and non-caching performance

Common Lisp; and inference count is the number of times that *compute-label* is called (i.e., the number of logical inferences). Problems denoted by train# and test# are the training and test examples in the plan recognition domain; problems denoted by brain# are the examples in the set covering diagnosis domain; and problems denoted by adder#, tc#, and kidney# are the examples for adder, temperature controller, and kidney water balance system, respectively. If ACCEL ran out of memory space and did not completely solve a problem, its run time and inference count figures are preceded by a >.

The empirical results indicate that caching can achieve speedup over an order of magnitude, and similarly for the inference count ratio. The exact efficiency improvement varies according to the domain, with the adder problems showing the best improvement and the set covering problems the least. They clearly show that caching is needed to implement an efficient abduction algorithm.

Previous systems have used caching-like mechanisms to improve their efficiency. For example, the SLD-resolution theorem prover of [22] caches successes and failures to avoid repeated proof efforts. However, he only deals with deductive theorem proving whereas ACCEL deals with abductive inference. Empirical results on the use of caching has sometimes produced conflicting evidence as to its usefulness. Although Elkan achieved good results with the use of caching in [22], he also reported that Stickel had independently discovered and implemented a caching scheme similar to his, but that the results Stickel obtained were unfavorable to caching on the class of theorems Stickel investigated at the time.

As mentioned in Section 2, we believe that duplicating inference poses a more serious

problem in abduction because multiple abductive proofs must usually be pursued in the search for a best explanation, whereas in deduction, we are usually interested in a single deductive proof. The need for multiple abductive proofs tends to result in more duplicate inferences being made, since the multiple abductive proofs maintained tend to share many identical subgoals. In [68], Stickel has also expressed similar opinions of the "strong motivation" to "eliminate search-space redundancy" for abduction since "the presence of an additional inference rule that allows literals to be either assumed or proved makes the search space for abduction even larger than that for deduction". Our empirical results confirm that caching is indeed very effective in improving the efficiency of abduction.

7 Related Work

There is a great deal of research related to abduction, plan recognition, and diagnosis. Compared to ACCEL, previous general abduction algorithms and systems are more restrictive, less efficient, and not as well tested on real problems. On the other hand, related research in the areas of plan recognition and diagnosis are more domain specific and not based on a general, unifying formalism.

7.1 General Theory and Algorithms

Pople was the first researcher to explore abductive reasoning in AI [60], although he was mainly concerned with using abduction to perform disease diagnosis. Charniak and McDermott proposed abduction as a general model for explanation, and recognized that many diverse AI tasks, including natural language understanding, diagnosis, and image interpretation, can be elegantly modeled as abduction [10]. Our work takes this hypothesis one step further and demonstrates via an implemented system that general and efficient abduction for the tasks of plan recognition and diagnosis is indeed possible.

The SAA algorithm does not perform caching of partial explanations and therefore duplicates inferences. To address this problem, Stickel has proposed a method to formulate a goal-directed, backward-chaining algorithm "metatheoretically" for execution by a forward-chaining reasoning system such as hyperresolution [68]. Subsumption checks in the forward-chaining system ensure that duplicate inferences are not made, and the goal-directedness of the backward-chaining algorithm can also be preserved. Our AAA algorithm achieves analogous effects of goal-directedness and non-duplicating inference in a direct way, via backward-chaining and caching.

Ginsberg has implemented a first-order ATMS using a multi-valued-logic theorem prover, MVL [26]. Compared to ACCEL, MVL is a more general theorem prover for full first-order predicate logic and it is capable of many kinds of reasoning including default reasoning, circumscription, temporal reasoning, and probabilistic reasoning. However, his implementation of the first-order ATMS does not cache previously computed partial explanations. This is in marked contrast to the (propositional) ATMS of de Kleer, in which caching and sharing of explanations are the distinguishing features. Hence, Ginsberg's system is an "ATMS" only in the sense that it is an algorithm that computes all possible proofs

(explanations). In addition, his system has not been tested on large problems.

Kautz has developed a formal theory of plan recognition based on first-order predicate logic [31; 32]. In his theory, an event hierarchy captures isa relationships between events (abstraction hierarchy) as well as part-of relationships of events and their components (decomposition hierarchy). One major difference between Kautz's theory and our work is that his theory models plan recognition as non-monotonic deduction rather than abduction. The difference is similar to that between consistency-based diagnosis and abductive diagnosis. His theory also makes the assumption that as few top-level events occur as possible, which is a form of the simplicity criterion, whereas ACCEL relies on coherence to select explanations in plan recognition.

The computational complexity of several abductive problems has been formally analyzed. It has been shown that, even in the propositional case, computing all minimal explanations is provably exponential [40; 65], since in the worst case, the number of minimal explanations is exponentially large. Reggia et al. have shown that finding parsimonious (i.e., minimum) explanations in the GSC model is NP-hard [62]. Bylander et al. have investigated the complexity of various classes of abduction [4] and have shown that unless some very restrictive conditions are satisfied, abduction is computationally intractable.

Note that the GSC model and the various classes of abduction studied by Reggia et al. and Bylander et al. only concern propositional abduction in which abductive proofs are restricted to be of depth one. Similarly, Levesque's characterization of abduction is in terms of propositional beliefs [39]. However, our abduction model is more general in that it allows first-order Horn clause axioms with variables.

To limit the computational efforts expended in the ATMS, Forbus and de Kleer introduced a "focusing" technique in which only relevant environments in an ATMS are maintained and propagated [24]. Dressler and Farquhar used a similar focusing mechanism in their model-based diagnostic system COCO to achieve efficient diagnosis of logic circuits [19]. Such focusing techniques achieve pruning effects similar to our use of heuristic beam search in the AAA algorithm. Focusing eliminates environments that are not implied by some focus environments, while our heuristic beam search eliminates environments based on their evaluation metric.

Poole's implemented system, Theorist, is a general default and abductive reasoning system [58]. Compared to ACCEL, Theorist also deals with default reasoning, and it handles full first-order predicate logic. However, the hypotheses (i.e., assumptions) that Theorist can make must be given to the system *a priori*, while all atoms are assumable in ACCEL (in the plan recognition domain). In addition, Theorist is not concerned with efficient inference and does not use caching to avoid redundant work, nor has it been tested on large problems.

7.2 Plan Recognition and Natural Language Understanding

7.2.1 Abductive Approaches

Several research efforts have adopted an abductive approach to text understanding. In [7], it is shown that noun-phrase reference determination can be achieved by an abductive unification procedure that allows for unifying two entities if it is consistent to do so. Hobbs

et al. have used abduction to solve the four local pragmatics problems of text understanding: reference resolution, compound nominal interpretation, syntactic ambiguity resolution, and metonymy resolution [30]. They argued that the abductive approach provided an elegant and thorough integration of syntax, semantics, and pragmatics, by combining the idea of interpretation as abduction and that of parsing as deduction.

The work reported here differs from those of [7] and [30] in that unlike their emphasis on mostly linguistic issues like noun-phrase reference determination and syntactic ambiguity resolution, ACCEL is concerned with recognizing characters' plans in a narrative text. The work of [5] also dealt with plan recognition, but evaluated explanations based on their simplicity, as opposed to our coherence metric. In addition, unlike their use of marker passing to restrict the search for explanations, we used a form of beam search to efficiently construct explanations.

7.2.2 Probabilistic Approaches

Inferring cause from effect is an inherently uncertain process — it is only plausible inference. Statistics and probability theory is the established discipline of study that deals with uncertainty. Within AI, Pearl has done extensive work on probabilistic reasoning [53]. Resolving ambiguity in natural language understanding can be formulated as reasoning under uncertainty, which is the approach adopted by Charniak and Goldman [9; 8; 27]. However, as explained in Section 3.4, selecting interpretations based solely on probability fails to capture the importance of text coherence.

7.2.3 Non-Abductive Approaches

Two early approaches to narrative understanding are script-based and plan-based understanding. In the script-based approach used by SAM [14], knowledge of stereotypical events are used to guide the understanding process. In the plan-based approach used by PAM [74; 75], knowledge about the actions, plans and goals of characters are used to connect the observed states and actions to their high level plans and goals. The realization that a complete understanding of narratives requires knowledge of events, plans and goals characterizes these early approaches [64].

The research of Norvig involves the use of marker passing mechanism to make inferences from narratives [49]. The knowledge base is structured in a semantic net, and inferences are made through the collisions of markers at nodes in the semantic net. The weakness of this approach is that in a sizable knowledge base, the spreading of markers can still lead to many possible path collisions even when constrained by the predetermined set of allowable inference paths. Furthermore, the semantics of these predetermined regular-expression-style inference paths is unclear and the paths appear to be created solely for the convenience of constraining marker movement to make the inferences desired.

One shortcoming of these non-abductive approaches is that the underlying inference processes tend to be rather *ad hoc* and not based on any general, logical foundation. A logic-based approach offers an expressive representation language — first-order predicate

calculus, with a clear, well-understood semantics. Inferences made in plan recognition acquire a firm semantic foundation when they are modeled as abduction.

7.3 Diagnosis

The GSC model of Reggia et al. is essentially a propositional abduction model in which abductive proofs are restricted to be of depth one [61; 62]. Allemang et al. have used a similar abduction model and an approximate algorithm to compute parsimonious diagnoses in a system that performs antibody identification in the domain of red blood cell typing [2]. ACCEL is more general in that it deals with first-order Horn clauses, and the explanations constructed can be of any depth.

Cox and Pietrzykowski have developed a general abductive inference procedure for computing fundamental causes of any observation stated as a first-order predicate calculus sentence [12; 13]. Their theory of abduction falls under the category of most specific abduction. However, their inference procedure does not utilize caching to improve efficiency, and it has been tested only on diagnostic problems in logic circuits.

Model-based diagnosis has recently been a very active area of research in AI [63; 17; 18; 69; 29]. This body of related work has been discussed in Section 5 when model-based diagnosis was introduced.

7.4 Abduction in Other Domains

Thagard has independently proposed a computational theory of explanatory coherence and applied it to the evaluation of scientific theories [72]. However, his theory of explanatory coherence consists of seven principles — symmetry, explanation, analogy, data priority, contradiction, acceptability, and system coherence. Independent criteria like simplicity and connectedness have been collapsed into one measure which he termed "explanatory coherence".

O'Rorke et al. have modeled scientific theory formation as abduction [52]. They illustrated how some of Lavoisier's key insights during the Chemical Revolution can be viewed as examples of theory formation by abduction. Their system differs from ACCEL in that it is a theory revision system designed to make changes to the rules in the underlying domain theory, while ACCEL assumes that its domain theory is correct.

8 Future Work

Future research issues can be broadly classified into three areas: representation and algorithms, natural language understanding, and diagnosis.

8.1 Representation and Algorithms

The axioms allowed in ACCEL are restricted to first-order Horn-clause axioms for efficiency reasons, since linear resolution with Horn-clauses is in general more efficient than binary

resolution with general clauses. However, the need for full first-order predicate logic representation, and hence, a general full first-order abduction algorithm may arise in the future. Stickel has already showed how his upside-down meta-interpretation method for abduction can be extended to deal with non-Horn clauses [68].

The efficiency of ACCEL can be further improved by compiling the Horn-clauses in the knowledge base, in the same way that the efficiency of a deductive theorem prover can be greatly improved via clause compilation [67; 50].

One shortcoming of the AAA algorithm is that the queue of best partial explanations maintained may become empty at some point in computing the abductive proofs. This can occur if the beam width β_{intra} is not sufficiently large and all best partial explanations become inconsistent after adding a new input atom. A better approach would have the capability of recovering from an empty beam of explanations by revising the assumptions in an inconsistent explanation in order to resolve the contradiction detected and arrive at a consistent explanation. The work of Subramanian and Mooney addresses this issue [70; 71]. Their multistrategy learning system, BRACE, combines abduction and theory revision to incorporate observations into a domain theory

8.2 Natural Language Understanding

ACCEL is currently only able to deal with the plan recognition aspect of text understanding. As mentioned in the related work section, abductive reasoning can also model noun-phrase reference determination [7] and syntactic ambiguity resolution [30]. ACCEL needs to be extended to include parsing of input sentences, and resolving lexical and syntactic ambiguity.

It is often the case that input atoms are too specific and cannot be directly deduced from abductive assumptions. This problem has been reported in [6; 45]. For instance, in the sentences "John went to the supermarket. He bought some milk.", assuming that John was shopping at the supermarket only allows us to derive that he would buy some food, but not necessarily milk. This problem can be overcome by generalizing explanations to include abductive proofs of *logical consequences* of the input atoms. The difficulty in this generalized definition of abductive explanations is to determine what are the relevant and interesting consequences to derive and explain, since deriving all possible consequences is clearly intractable. An example of methods to control forward inferences is that developed for automated knowledge integration [41].

The plan recognition knowledge in ACCEL primarily encodes stereotypical knowledge as in a script-based system like SAM [14]. As such, ACCEL is not capable of handling novel plans, which involve actions not explicitly defined as part of a common plan, yet these actions are causally related and accomplish some high-level goal of an agent. In addition, ACCEL currently fails to handle common substeps that are shared by multiple plans, for instance, that a going action is part of both the supermarket-shopping and robbing plans. Overloading of actions to simultaneously achieve multiple goals is known to be a common occurrence [56]. Furthermore, ACCEL is currently restricted to abducing high-level plans from observed actions, but not observed states. Hence, it will fail to abduce, for instance, that an agent has the restaurant-dining plan when told that he is hungry. We can of course write additional axioms to assert that a high-level plan implies an observed state, but to

do so correctly, we must also assert when a state holds relative to the time of occurrence of plans and actions. For example, a person is only hungry before dining at a restaurant but not after. In other words, to properly handle the abduction of high-level plans from observed states, ACCEL must be able to reason more generally about time, and the preconditions and effects of actions in terms of the states that an action enables or disables. Extending ACCEL to correctly handle these deficiencies is an important area for future research.

Currently, explanations in the plan recognition domain are evaluated solely on coherence. Future work needs to integrate likelihood information as an part of the evaluation criterion, perhaps as a measure secondary to coherence.

8.3 Diagnosis

The work of [57; 59; 33] has revealed some interesting relationships between consistency-based and abductive diagnosis, which are two major paradigms in model-based diagnosis. To what extent do the two approaches coincide and differ, especially in practical terms such as ease of representation and diagnostic efficiency, remains to be investigated.

Our research does not focus on gathering additional measurements to further differentiate and narrow the diagnostic candidates. Intelligently selecting which component output and which additional sensor values to measure is important in achieving efficient diagnosis. For example, in GDE [17], additional measurements is gathered via a method that minimizes the expected entropy of candidate probabilities.

The use of quantitative information in qualitative simulation can greatly reduce the ambiguity of the qualitative behavior of a dynamic system [37]. The added precision allows better differential diagnosis [20]. The ability to monitor a dynamic system over time and performs diagnosis in real time as the device operates is also important [20]. Extending ACCEL to deal with quantitative knowledge and device monitoring are important future research issues.

As mentioned earlier, normality-based diagnosis is more flexible but may generate implausible diagnoses, while fault-based diagnosis requires explicit knowledge of fault models. To overcome the limitation of explicitly knowing all fault models in advance, we can instead develop a method to automatically acquire fault models over time. This can be accomplished by generalizing the common input-output behavior patterns as summarized by the parameterized abnormality assumptions of a component (e.g., $ab(X, U, V, W, T)$). Such a learning module would improve the diagnostic accuracy of ACCEL by recognizing the common fault modes of a device component.

9 Conclusion

Finding explanations for observed phenomena underlies a diverse set of intelligent activities, including natural language understanding, diagnosis, scientific theory formation, and image interpretation. The ubiquity of explanation underscores its importance as a research topic in artificial intelligence. In this paper, we view explanation as logical abduction, which serves as a unifying formalism for explanation.

This paper has made several important contributions:

1. We have demonstrated the practical feasibility of a general abductive approach to explanation by successfully building a domain-independent system, ACCEL, that is general enough to perform both plan recognition and diagnosis, yet efficient enough to be of practical utility. We support this claim by extensively evaluating the system on 50 narrative texts in the plan recognition domain, on 50 real patient cases in the set covering diagnosis domain, and on 10 model-based diagnosis scenarios each on an adder, a temperature controller, and the water balance system of the human kidney. Except for the adder circuit, each of the knowledge bases contains hundreds of Horn-clause rules.

2. We have developed a novel evaluation criterion, explanatory coherence, to evaluate the quality of explanations in the plan recognition domain. We present empirical results indicating that our coherence metric outperforms the simplicity metric in selecting the best explanation in the plan recognition domain. Our coherence-based approach performs as well as the probabilistic approach of plan recognition, but without the need to engineer numerous prior and posterior probabilities.

3. We present empirical evidence showing that caching of previously computed explanations is critical to the efficiency of an abduction algorithm. Specifically, speedup of more than an order of magnitude has been obtained on our test problems.

In summary, this paper has demonstrated via an implemented system that general and efficient abduction for the tasks of plan recognition and diagnosis is indeed possible, and the future holds much promise for such a general abductive approach to explanation.

10 Acknowledgments

This research was supported by a University of Texas MCD Fellowship, an IBM Graduate Fellowship, and by the NASA Ames Research Center under grant NCC-2-429. Computing equipment used was donated by Texas Instruments. We are indebted to Dr. Stanley Tuhrim of Mount Sinai School of Medicine, New York, and Dr. James Reggia of the University of Maryland at College Park who kindly provided us the knowledge base on brain damage and the 50 patient test cases. Thanks also to Adam Farquhar for kindly allowing us to use his ATMS code on which an earlier version of ACCEL was built, and for discussing technical details of the ATMS. Discussions with Siddarth Subramanian on diagnosis have been helpful.

References

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, MA, 1974.

[2] Dean Allemang, Michael C. Tanner, Tom Bylander, and John R. Josephson. Computational complexity of hypothesis assembly. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1112–1117, Milan, Italy, 1987.

[3] James F. Allen. *Natural Language Understanding*. Benjamin/Cummings, Menlo Park, CA, 1987.

[4] Tom Bylander, Dean Allemang, Michael C. Tanner, and John R. Josephson. Some results concerning the computational complexity of abduction. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 44–54, Toronto, Ontario, Canada, 1989.

[5] Eugene Charniak. A neat theory of marker passing. In *Proceedings of the National Conference on Artificial Intelligence*, pages 584–588, Philadelphia, PA, 1986.

[6] Eugene Charniak. Logic and explanation. *Computational Intelligence*, 3:172–174, 1987.

[7] Eugene Charniak. Motivation analysis, abductive unification, and nonmonotonic equality. *Artificial Intelligence*, 34:275–295, 1988.

[8] Eugene Charniak and Robert Goldman. A probabilistic model of plan recognition. In *Proceedings of the National Conference on Artificial Intelligence*, pages 160–165, Anaheim, CA, 1991.

[9] Eugene Charniak and Robert P. Goldman. A semantics for probabilistic quantifier-free first-order languages, with particular application to story understanding. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, 1989.

[10] Eugene Charniak and Drew McDermott. *Introduction to Artificial Intelligence*. Addison Wesley, Reading, MA, 1985.

[11] Eugene Charniak and Solomon E. Shimony. Probabilistic semantics for cost based abduction. In *Proceedings of the National Conference on Artificial Intelligence*, pages 106–111, Boston, MA, 1990.

[12] P. T. Cox and T. Pietrzykowski. Causes for events: Their computation and applications. In *Proceedings of the Eighth International Conference on Automated Deduction*, pages 608–621. Springer-Verlag, 1986. Lecture Notes in Computer Science 230.

[13] P. T. Cox and T. Pietrzykowski. General diagnosis by abductive inference. In *Proceedings of the 1987 Symposium on Logic Programming*, pages 183–189, 1987.

[14] Richard E. Cullingford. Script application: Computer understanding of newspaper stories. Technical Report 116, Department of Computer Science, Yale University, New Haven, CT, January 1978.

[15] Randall Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410, 1984.

[16] Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[17] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[18] Johan de Kleer and Brian C. Williams. Diagnosis with behavioral modes. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1324–1330, Detroit, MI, 1989.

[19] Oskar Dressler and Adam Farquhar. Putting the problem solver back in the driver's seat: Contextual control of the ATMS. In *Proceedings of the Second Model-Based Reasoning Workshop*, Boston, MA, 1990.

[20] Daniel Dvorak. *Monitoring and Diagnosis of Continuous Dynamic Systems Using Semiquantitative Simulation.* PhD thesis, Department of Computer Sciences, University of Texas at Austin, Austin, TX, May 1992.

[21] Daniel Dvorak and Benjamin J. Kuipers. Model-based monitoring of dynamic systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1238–1243, Detroit, MI, 1989.

[22] Charles Elkan. Conspiracy numbers and caching for searching and/or trees and theorem-proving. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 341–346, Detroit, MI, 1989.

[23] Kenneth D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.

[24] Kenneth D. Forbus and Johan de Kleer. Focusing the ATMS. In *Proceedings of the National Conference on Artificial Intelligence*, pages 193–198, St. Paul, Minnesota, 1988.

[25] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, San Francisco, CA, 1979.

[26] Matthew L. Ginsberg. A circumscriptive theorem prover. *Artificial Intelligence*, 39:209–230, 1989.

[27] Robert P. Goldman. *A Probabilistic Approach to Language Understanding.* PhD thesis, Department of Computer Science, Brown University, Providence, RI, December 1990. Technical Report CS-90-34.

[28] H. P. Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics 3: Speech Acts*, pages 41–58. Academic Press, New York, 1975.

[29] Walter Hamscher. XDE: Diagnosing devices with hierarchic structure and known component failure modes. In *Proceedings of the Sixth IEEE Conference on Artificial Intelligence Applications*, pages 48–54, Santa Barbara, CA, 1990.

[30] Jerry R. Hobbs, Mark E. Stickel, Paul Martin, and Douglas Edwards. Interpretation as abduction. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 95–103, Buffalo, New York, 1988.

[31] Henry A. Kautz. *A Formal Theory of Plan Recognition.* PhD thesis, Department of Computer Science, University of Rochester, Rochester, NY, May 1987. Technical Report 215.

[32] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *Proceedings of the National Conference on Artificial Intelligence*, pages 32–37, Philadelphia, PA, 1986.

[33] Kurt Konolige. Abduction versus closure in causal theories. *Artificial Intelligence*, 53:255–272, 1992.

[34] Benjamin J. Kuipers. Qualitative simulation in medical physiology: A progress report. Technical Report MIT/LCS/TM-280, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 1985.

[35] Benjamin J. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289–338, 1986.

[36] Benjamin J. Kuipers. Qualitative reasoning: Modeling and simulation with incomplete knowledge. Book draft, May 1991.

[37] Benjamin J. Kuipers and Daniel Berleant. Using incomplete quantitative knowledge in qualitative reasoning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 324–329, Saint Paul, Minnesota, 1988.

[38] Wendy Lehnert and Beth Sundheim. A performance evaluation of text-analysis technologies. *AI Magazine*, 12(3):81–94, 1991.

[39] Hector J. Levesque. A knowledge-level account of abduction. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1061–1067, Detroit, MI, 1989.

[40] David McAllester. A widely used truth-maintenance system. Ai lab memo, MIT, Cambridge, MA, 1985.

[41] Kenneth S. Murray. KI: An experiment in automating knowledge integration. Technical Report AI88-90, Artificial Intelligence Laboratory, Department of Computer Sciences, The University of Texas at Austin, 1988.

[42] Hwee Tou Ng. Model-based, multiple fault diagnosis of time-varying, continuous physical devices. In *Proceedings of the Sixth IEEE Conference on Artificial Intelligence Applications*, pages 9–15, Santa Barbara, CA, 1990. To appear in *Readings in Model-based Diagnosis*, edited by Walter Hamscher, Luca Console, and Johan de Kleer.

[43] Hwee Tou Ng. Model-based, multiple-fault diagnosis of dynamic, continuous physical devices. *IEEE Expert*, 6(6):38–43, 1991.

[44] Hwee Tou Ng. *A General Abductive System with Application to Plan Recognition and Diagnosis*. PhD thesis, Department of Computer Sciences, University of Texas at Austin, Austin, TX, May 1992.

[45] Hwee Tou Ng and Raymond J. Mooney. Abductive explanation in text understanding: Some problems and solutions. Technical Report AI89-116, Artificial Intelligence Laboratory, Department of Computer Sciences, The University of Texas at Austin, October 1989.

[46] Hwee Tou Ng and Raymond J. Mooney. On the role of coherence in abductive explanation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 337–342, Boston, MA, 1990.

[47] Hwee Tou Ng and Raymond J. Mooney. An efficient first-order abduction system based on the ATMS. Technical Report AI91-151, Artificial Intelligence Laboratory, Department of Computer Sciences, The University of Texas at Austin, January 1991.

[48] Hwee Tou Ng and Raymond J. Mooney. An efficient first-order Horn-clause abduction system based on the ATMS. In *Proceedings of the National Conference on Artificial Intelligence*, pages 494–499, Anaheim, CA, 1991.

[49] Peter Norvig. *Unified Theory of Inference for Text Understanding*. PhD thesis, Computer Science Division, University of California at Berkeley, Berkeley, CA, 1987. Technical Report No. UCB/CSD 87/339.

[50] Peter Norvig. *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp*. Morgan Kaufmann, San Mateo, CA, 1992.

[51] Peter Norvig and Robert Wilensky. A critical evaluation of commensurable abduction models for semantic interpretation. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, Helsinki, Finland, August 1990.

[52] Paul O'Rorke, Steven Morris, and David Schulenburg. Theory formation by abduction: Initial results of a case study based on the chemical revolution. In *Proceedings of the Sixth International Workshop on Machine Learning*, 1989.

[53] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[54] Charles Sanders Peirce. *Collected Papers of Charles Sanders Peirce (1839–1914)*. Harvard University Press, Cambridge, MA, 1958.

[55] Yun Peng and James A. Reggia. *Abductive Inference Models for Diagnostic Problem-Solving*. Springer-Verlag, New York, 1990.

[56] Martha E. Pollack. Plan recognition beyond STRIPS. In *Working Notes of the Second AAAI Workshop on Plan Recognition*, Detroit, Michigan, August 1989.

[57] David Poole. Representing knowledge for logic-based diagnosis. In *Proceedings of the International Conference on Fifth Generation Computing Systems*, pages 1282–1290, Tokyo, Japan, 1988.

[58] David Poole. A methodology for using a default and abductive reasoning system. Technical Report 89-20, Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada, September 1989.

[59] David Poole. Normality and faults in logic-based diagnosis. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1304–1310, Detroit, MI, 1989.

[60] Harry E. Pople, Jr. On the mechanization of abductive logic. In *Proceedings of the Third International Joint Conference on Artificial Intelligence*, pages 147–152, 1973.

[61] James A. Reggia, Dana S. Nau, and Pearl Y. Wang. Diagnostic expert systems based on a set covering model. *International Journal of Man-Machine Studies*, 19:437–460, 1983.

[62] James A. Reggia, Dana S. Nau, and Pearl Y. Wang. A formal model of diagnostic inference. I. problem formulation and decomposition. *Information Sciences*, 37:227–256, 1985.

[63] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

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

[65] Bart Selman and Hector J. Levesque. Abductive and default reasoning: A computational core. In *Proceedings of the National Conference on Artificial Intelligence*, pages 343–348, Boston, MA, 1990.

[66] Mark E. Stickel. A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. Technical Note 451, SRI International, September 1988.

[67] Mark E. Stickel. A Prolog technology theorem prover: Implementation by an extended Prolog compiler. *Journal of Automated Reasoning*, 4:353–380, 1988.

[68] Mark E. Stickel. Upside-down meta-interpretation of the model elimination theorem-proving procedure for deduction and abduction. Technical Report TR-664, Institute for New Generation Computer Technology, Tokyo, Japan, May 1991.

[69] Peter Struss and Oskar Dressler. Physical negation — integrating fault models into the general diagnostic engine. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1318–1323, Detroit, MI, 1989.

[70] Siddarth Subramanian. Belief revision in the context of abductive explanation. dissertation proposal, 1992.

[71] Siddarth Subramanian and Raymond J. Mooney. Combining abduction and theory revision. In *Proceedings of the First International Workshop on Multistrategy Learning*, pages 207–214, Harpers Ferry, W. Va., 1991.

[72] Paul Thagard. Explanatory coherence. *Behavioral and Brain Sciences*, 12(3):435–467, September 1989.

[73] Stanley Tuhrim, James Reggia, and Sharon Goodall. An experimental study of criteria for hypothesis plausibility. *Journal of Experimental and Theoretical Artificial Intelligence*, 3:129–144, 1991.

[74] Robert W. Wilensky. *Understanding Goal-Based Stories*. PhD thesis, Department of Computer Science, Yale University, New Haven, CT, September 1978. Technical Report 140.

[75] Robert W. Wilensky. *Planning and Understanding: A Computational Approach to Human Reasoning*. Addison-Wesley, Reading, MA, 1983.