Abductive and Default Reasoning: A Computational Core

Bart Selman and Hector J. Levesque*
Dept. of Computer Science
University of Toronto
Toronto, Canada M5S 1A4

Abstract

Of all the possible ways of computing abductive explanations, the ATMS procedure is one of the most popular. While this procedure is known to run in exponential time in the worst case, the proof actually depends on the existence of queries with an exponential number of answers. But how much of the difficulty stems from having to return these large sets of explanations? Here we explore abduction tasks similar to that of the ATMS, but which return relatively small answers. The main result is that although it is possible to generate some non-trivial explanations quickly, deciding if there is an explanation containing a given hypothesis is NP-hard, as is the task of generating even one explanation expressed in terms of a given set of assumption letters. Thus, the method of simply listing all explanations, as employed by the ATMS, probably cannot be improved upon. An interesting result of our analysis is the discovery of a subtask that is at the core of generating explanations, and is also at the core of generating extensions in Reiter's default logic. Moreover, it is this subtask that accounts for the computational difficulty of both forms of reasoning. This establishes for the first time a strong connection between computing abductive explanations and computing extensions in default logic.

Introduction

Of all the possible ways of computing abductive explanations, the procedure employed by an assumption-based truth-maintenance system (ATMS) is one of the most popular (de Kleer 1986a; Reiter and de Kleer 1987). It is therefore somewhat surprising that so little effort has gone into understanding in precise terms the nature of the computational task performed by an ATMS, that is, the *what* for which an ATMS is a *how*.

What do we know in general about this task? It has been known since at least 1985 that in the worst case, any procedure that computes what the ATMS computes will need time that is exponential in the length of its input (McAllester 1985). This is because there are problems for which the desired set of answers (where intuitively, each answer is a set of assumptions that would *explain* a given condition) is exponentially large. Perhaps this simple fact has discouraged further theoretical analysis into the worst-case difficulty of computing explanations.

But it doesn't tell the whole story. Is the fact that an ATMS can take exponential time *only* due to cases where an exponential number of answers need to be returned? What if instead of generating all the answers, we only required a procedure to reply to simple yes/no questions about them, such as whether or not there is an explanation containing a given assumption? Furthermore, in many (if not most) applications, we expect to be dealing with a very small number of explanations. For example, in circuit diagnosis, explanations involve sets of possibly faulty components (Reiter 1987; Poole 1988), and one would not expect k components to break down independently, for large k. Is it still the case that generating a small number of explanations is hard? In other words, if an ATMS still runs in exponential time for problems like this (as it apparently does (Provan 1987)), should we be looking for a different procedure, or is this the best that can be expected?

In this paper, we attempt to answer these questions. In particular, we show that certain natural variations of the ATMS task that do not require listing all the answers are nonetheless NP-hard. In our view, this provides concrete evidence that the ATMS is doing as well as can be expected.

But something more fundamental came out of the analysis. We were surprised to discover a strong connection between computing explanations on the one hand, and computing extensions in Reiter's default logic (Reiter 1980), on the other. It turns out that both tasks share a common computational core. Moreover, it is this common subtask that leads to the computational difficulty of both abductive and default reasoning. Apart from the fact that both forms of reasoning use the word "assumption," this is the first result that we know of to show a clear relationship between the

*Fellow of the Canadian Institute for Advanced Research.

computational properties of these two apparently very different forms of reasoning.

The rest of the paper is organized as follows. In the next section, we review Reiter and de Kleer's analysis of the ATMS. Next, we show that while it is always easy to find at least one non-trivial explanation, determining if there is one containing a given assumption, or finding one that is expressed only in terms of a given assumption set is NP-hard. In section 4, we consider a weaker version of the ATMS task (which we call the Support Selection Task), where explanations are not required to be minimal, and show that it too is NP-hard. In section 5, we briefly review the definitions from Reiter's default logic, and show that the problem of computing an extension is a variant of the Support Selection Task where we care about maximality instead of minimality. It turns out that making a support set minimal or maximal is easy; it's finding one in the first place that is hard. Finally, some conclusions are drawn.

Abduction

In this section, we formally define what constitutes an explanation as computed by the ATMS (Reiter and de Kleer 1987). We will assume a standard propositional language $\mathcal{L}$ with propositional letters from the set P. We will use p, q, r, s, and t (possibly with subscripts) to denote propositional letters. A *clause* is a disjunction of literals (a *literal* is either a propositional letter, called a positive literal, or its negation, called a negative literal). We will represent a clause by the set of literals contained in the clause. A clause is called a *unit* clause if it only contains a single literal. A clause is *Horn* if and only if it contains at most one positive literal. A set of Horn clauses will be called a *Horn theory*.

Definition 1 [Explanation] *Given a set of clauses Σ, called the background theory, and a letter q, an explanation for q is a minimal set of unit clauses α such that*

1. $\Sigma \cup \alpha \models q$, and

2. $\Sigma \cup \alpha$ is consistent.

For a discussion on the desirability of the above properties, see Levesque (1989).[2] Instead of expressing explanations as sets of unit clauses, we will often give the logical equivalent form consisting of the conjunction of the literals occurring in the clauses, *e.g.*, we write $p \wedge q \wedge r$ instead of $\{\{p\}, \{q\}, \{r\}\}$.

Example: Let Σ be the set $\{\{p\}, \{q\}, \{\overline{p}, \overline{r}, \overline{s}, t\}\}$. The conjunctions $r \wedge s$ and t are explanations for t. We call t the trivial explanation for t; our interest lies of course in the other, non-trivial explanations.

The notion of explanation defined above, is somewhat more general than the one employed in the ATMS. The ATMS only computes a certain subset of these explanations, namely those drawn from a distinguished subset of the propositional letters, called *assumptions.* Assumptions stand for the hypotheses that we are willing to consider in the explanations, such as the possible failure of a component in circuit diagnosis.

Definition 2 [Formal Specification of the ATMS] *Given a set of Horn clauses Σ, a set of assumptions $A \subseteq P$, and a letter q, called the query, the ATMS procedure computes the following set:*

$$\mathcal{A}[\Sigma, A, q] = \{\alpha \mid \alpha \textit{ is an explanation for } q \textit{ containing only letters from } A\}.$$

The explanations in $\mathcal{A}[\Sigma, A, q]$ will be called *assumption-based* explanations. Note that when the assumption set includes all symbols in the language, every explanation is also an assumption-based one.

Computing explanations

We will now consider the computational cost of generating explanations. As is well-known, there may be exponentially many explanations for a given letter (McAllester 1985; de Kleer 1986b); and therefore, listing all of them may require exponential time.[3] However, this leaves open the question of what the complexity of finding some explanation is. In particular, what is the complexity of finding a non-trivial one?

In case Σ contains arbitrary clauses, finding any explanation is easily shown to be NP-hard.[4] However, the following theorem shows that when Σ is a Horn theory, a non-trivial explanation (if one exists) can be computed efficiently.

Theorem 1 *Given a set of Horn clauses Σ and a letter q, a non-trivial explanation for q can be computed in time $O(kn)$, where k is the number of propositional letters and n is the number of occurrences of literals in Σ.*

Here we only give an outline of the algorithm. Consider a clause in Σ of the following form: $\{\overline{q}_1, \ldots, \overline{q}_k, q\}$ with $k \geq 0$ (if no such clause exists, return "no non-trivial explanation"). Now, clearly $q_1 \wedge \ldots \wedge q_k$ with Σ implies q. Subsequently, try removing a letter from this conjunction while ensuring that the remaining conjunction together with Σ still implies q (testing can be done in linear time, using the Dowling and Gallier (1984) procedure). Repeat this process until no more letters can be removed. If the remaining conjunction is non-empty and combined with Σ is consistent, return that one; otherwise consider another clause containing q and repeat the above procedure. When all clauses

[2]For a quite different definition of explanation, see Reggia (1983) and Allemang et al. (1987).

[3]In fact, there may be exponentially many assumption-based explanations, and therefore the worst case complexity of the ATMS, which lists all of them, is clearly exponential.

[4]Since explanations only exist when Σ is consistent, an explanation procedure can be used to test the satisfiability of a set of clauses.

containing q have been explored and no explanation is found, return "no non-trivial explanation."

It is clear that the above algorithm only generates certain, very particular explanations — which ones depend on the way the background knowledge Σ is expressed. But if there are *some* non-trivial explanations that are easy to find, could it be that in some sense they are *all* easy to find, even if there are too many to list? One way to look at this question is to consider a procedure that generates only a single explanation, but must return different ones for different arguments. For example, if we can ask for an explanation containing the letters in S_1 but not containing those in S_2, clearly we can generate arbitrary explanations.[5] Unfortunately, the following theorem show that there can be no efficient procedure for this form of "goal-directed" abduction, even if the set S_1 contains only a single literal, and S_2 is empty.[6]

Theorem 2 *Given a set of Horn clauses Σ and letters p and q, the problem of generating an explanation for q that contains p is NP-hard.*

The proof of this theorem is based on a reduction from the NP-complete decision problem "path with forbidden pairs" (or PWFP) defined by Gabow, Maheshwari, and Osterweil (1976). An instance of PWFP consists of a directed graph $G = (V, E)$, specified vertices $s, t \in V$, and a collection $\mathcal{C} = \{(a_1, b_1), ..., (a_n, b_n)\}$ of pairs of vertices from V. The question is: does there exist a path from s to t in G that contains at most one vertex from each pair in $\mathcal{C}$? This problem remains NP-complete even if we only consider acyclic graphs.

Given an instance of this restricted version of PWFP, we now construct a background theory Σ. Identifying the vertices of the graph with propositional letters, Σ contains the following clauses: (1) for each directed edge (x, y), the clause $\{\overline{x_y}, \overline{x}, y\}$ where x_y is a new propositional letter, and (2) for each forbidden pair (a_i, b_i), the clause $\{\overline{a}_i, \overline{b}_i\}$. Now, consider an explanation for t that contain s. It can be shown that if such an explanation exists, it will consist of a set of propositional letters of the form x_y that uniquely identify a path from s to t in the original graph (Selman 1990). Moreover, because of the clauses in group (2), such a path goes through at most one vertex of each forbidden pair. Thus, we can reduce the PWFP problem to goal-directed abduction.

Intuitively speaking, theorem 2 shows that certain explanations will be hard to find, even if our background theory Σ is Horn. And, as can be seen from the reduction, this result also holds when Σ consists of an acyclic Horn theory.[7]

Finally, we consider the influence of an assumption set as used in the ATMS. Recall that the assumption set A is a distinguished subset of the propositional letters and that given a query q, the ATMS will generate only explanations that contain letters from among those in A. Note that the assumption set again allows one to select a certain subset of all possible explanations. This way of of selecting certain explanations is related, but *not* identical, to the notion of goal-directed abduction. The following theorem shows that the use of such an assumption set dramatically increases the complexity of finding a non-trivial explanation (compare with theorem 1):

Theorem 3 *Given a set of Horn clauses Σ, a set of assumptions A, and a query letter q, finding an assumption-based explanation for q is NP-hard.*

The proof of this theorem is based on a modification of the reduction used in the proof of theorem 2: add the clause $\{s\}$ to the background theory, and let the assumption-set contain all letters of the form x_y. Now, an assumption-based explanation will consist of a subset of the letters in the assumption set, and as above, this set will uniquely identify a path from s to t not containing any forbidden pair. Again, the problem remains NP-hard even for acyclic theories.

This theorem shows that apart from the fact that the ATMS may have to list an exponential number of explanations, merely finding one of them may require exponential time.

Finally, we consider the experimental observation, reported by Provan (1987), that the ATMS can exhibit exponential behaviour even if the background theory is such that there are only a few assumption-based explanations for the query letter. Provan argues that such restricted theories have practical significance, for example, in scene interpretation. We can now show that in fact the intractability is *inherent* in the task, and not simply caused by the particular procedure employed in the ATMS:

Theorem 4 *Given a set of assumptions A, a query q, and a set of Horn clauses Σ such that q has at most one assumption-based explanation, finding this explanation is NP-hard under randomized reductions.*

To prove this theorem we use a result by Vazirani and Valiant (1988), who show that determining propositional satisfiability remains hard (unless NP = RP, considered to be very unlikely) even if one guarantees that the given instances of SAT has at most one satisfying truth assignment. Since the reduction from SAT to

[5]The set S_1 could be used, for example, to identify components that have a high failure rate when doing circuit diagnosis. For a related approach, see de Kleer and Williams (1989).

[6]For the purpose of this paper, and to keep the provisos to a minimum, we assume that P≠NP. An excellent introduction to the basic concepts of computational complexity theory can be found in Garey and Johnson (1979).

[7]Given a Horn theory Σ, let G be a directed graph containing a vertex for each literal in Σ and an edge from any vertex corresponding to a letter on the left-hand side of a Horn rule to the vertex corresponding to the letter on the right hand-side of that rule. A Horn theory is *acyclic* if and only if the associated graph G is acyclic.

PWFP and the reduction from PWFP to assumption-based explanations are parsimonious (*i.e.*, the number of solutions is preserved), it follows that even if we guarantee that there is at most one assumption-based explanation for the query letter, finding it still remains hard. So, the problem of generating assumption-based explanations even for such special restricted background theories remains intractable, and therefore the ATMS procedure is doing as well as can be expected.

One remaining question is whether the problem is still hard if we are guaranteed that there are only a few explanations overall (including the non-assumption-based ones) for the query. Note that we can always generate at least one non-trivial explanation (see theorem 1); we conjecture however that generating $O(n)$ of them is again NP-hard (possibly under randomized reductions), where n is the number of propositional letters in the language.

The Computational Core

We have shown that finding an assumption-based explanation is intractable, even when the background theory Σ is an acyclic set of Horn clauses. In this section, we will isolate a subtask, called the Support Selection Task, which lies at the core of the computational difficulties.

Definition 3 [Support Selection Task] *Given a set of Horn clauses Σ, a set of letters $A \subseteq P$, and a letter q, find a set of unit clauses α, called a support set, such that the following conditions hold:*

1. $\Sigma \cup \alpha \models q$,

2. $\Sigma \cup \alpha$ is consistent, and

3. α contains only letters form A.

Note that an assumption-based explanation is simply a *minimal* support set. We first consider the complexity of the Support Selection Task:

Theorem 5 *Given a Horn theory Σ, a set $A \subseteq P$, and a letter q, finding a support set for q is NP-hard.*

This result follows directly from a generalization of the reduction used in the proof of theorem 3. Since the reduction does not rely on finding a minimal set of assumptions to support the query, any support set for the query will identify a path that goes from s to t containing at most one vertex from each forbidden pair.

Because an assumption-based explanation is a minimal support set, finding such an explanation is at least as hard as finding support sets. Hence, the intractability of finding an assumption-based explanation is in fact a direct consequence of theorem 5. Stated differently, in order to establish the intractability of finding an assumption-based explanation, one need not use the fact that explanations are minimal.

Furthermore, the minimality requirement does not further increase the computational difficulty of the task, as can be seen from the following argument. Consider a support set α for q. We can minimize this set by removing clauses from it while each time verifying that the reduced set combined with Σ still implies q. Since Σ is a Horn theory this can be done in polynomial time. Finally, note that the Support Selection Task can be shown to be no harder than any problem in NP, and thus neither is generating an assumption-based explanation.

To summarize, the Support Selection Task is at the core of the ATMS-style abduction task. In the next section, we will see how this task also is at the core of goal-directed default reasoning, thereby establishing a computational connection between abductive and default reasoning.

Default Reasoning

Default Logic, introduced by Reiter (1980), is one of the more prominent formal proposals for representing and reasoning with default information. We will first briefly define Default Logic (see Reiter (1980) and Etherington (1986) for further details), and subsequently consider the complexity of default logic theories.

Reiter formalized default reasoning by extending first-order logic with default rules. A ***default theory*** is a pair (D, W) where D is a set of default rules and W a set of ordinary first-order formulas. A rule is of the form:

$$\frac{\alpha : \beta}{\gamma}$$

where α is the *prerequisite*, γ the *conclusion*, and β the *justification* of the rule, each of them formulas. A rule is intuitively understood as meaning that if α is known, and β is consistent with what is known, then γ may be inferred.

An *extension* is a *maximal* set of conclusions that can be drawn from a theory. But care must be taken that none of the justifications of the rules used in the construction of an extension conflict with its final contents, and that every formula in the extension can in fact be derived from W and the rules. The formal definition of an extension (from Reiter (1980), Theorem 2.1) is therefore rather complex:

Definition 4 [Extension] *A set of formulas E is an extension for the theory (D, W) if and only if it satisfies the following equations:*[8]

$$\begin{aligned} E_0 &= W, \text{ and for } i \geq 0 \\ E_{i+1} &= Th(E_i) \cup \\ &\quad \left\{ \gamma \,\middle|\, \frac{\alpha : \beta}{\gamma} \in D, \;\; \alpha \in E_i, \;\; \text{and } \neg\beta \notin E \right\} \\ E &= \bigcup_{i=0}^{\infty} E_i \end{aligned}$$

Note the explicit reference to E in the definition of E_{i+1}. Th denotes logical closure.

[8]We assume that all formulas are closed, *i.e.*, they do not contain free variables.

Computing Extensions

Kautz and Selman (1989) give a detailed analysis of the computational complexity of default reasoning based on Reiter's proposal. They consider a partially ordered space of more and less general propositional default logic theories. For each theory the complexity is determined of the following tasks: finding an extension (*credulous* reasoning), generating an extension that contains a given set of propositions (*goal-directed* reasoning), and the problem of determining what holds in all extensions of a default logic theory (*skeptical* reasoning). To avoid the difficulty of the consistency check needed to determine whether a rule can be applied, Kautz and Selman restrict the default theories to ones in which the set of facts W, the prerequisites, the justifications, and the consequences each consist of a set of literals. Here we will consider a relaxation of these restrictions. In particular, we will allow W to contain Horn clauses.

We will show that even for extremely simple default rules and a Horn theory W goal-directed default reasoning is intractable, and that the computational difficulty is again appropriately characterized by the Support Selection Task.[9] To facilitate our discussion, a rule of the form $:p/p$ will be called an *elementary* default — these rules are the simplest possible defaults.[10]

We have the following result concerning goal-directed default reasoning:

Theorem 6 *Given an acyclic Horn theory W, a set of elementary defaults D, and a letter q, finding an extension of (D,W) that contains q is NP-hard.*[11]

This result strengthens a recent result by Stillman (1989), who showed the task is NP-hard for arbitrary Horn theories with general normal unary defaults. But aside from strengthening Stillman's result, our interest in this result arises from the fact that the Support Selection Task lies again at the root of the computational difficulty of the problem, as we will see below.

To prove theorem 6, we first consider the relation between goal-directed default reasoning and the Support Selection Task. Note that if Σ is a Horn theory and D a set of elementary defaults involving letters from a set A, then each extension of (D, Σ) is of the form $Th(\Sigma \cup \alpha)$, where α is a set of unit clauses drawn from A. Intuitively, α is the set of letters that are added to Σ via the rules in D. We have the following theorem:

[9] Determining the complexity of goal-directed reasoning for such theories was given as one of the central open problems in Kautz and Selman (1989).

[10] Using the terminology of Kautz and Selman (1989), they are normal unary defaults with empty prerequisites and positive consequences.

[11] Since an extension consists of an infinite set of formulas because of the logical closure, we are only looking for a set of unit clauses that represents an extension, as described below.

Theorem 7 *Let Σ be a Horn theory, q be a letter, $A \subseteq P$ be a set of letters, and let $D = \{:p/p \mid p \in A\}$. Then, $Th(\Sigma \cup \alpha)$ is an extension of (D, Σ) that contains q if and only if α is a* maximal *support set of q.*

This theorem follows from a more general result by Reiter (1987). It follows that finding extensions of the default logic theory that contains a given letter q, is at least as hard as finding a support set for q. Thus, theorem 6 follows directly from the fact that the Support Selection Task is NP-hard (theorem 5).

Furthermore, the fact that extensions correspond to *maximal* support sets does not further add to the difficulty of computing extensions: given a support set, one can simply try adding additional letters from the assumption set while maintaining consistency, until a maximal set is obtained. Thus, as for assumption-based explanations, the Support Selection Task is the difficult part of goal-directed default reasoning.

It is suggested in Kautz and Selman (1989) that goal-directed reasoning could be of use in resolution theorem provers that incorporate default information. Our results here suggest that such an integration will most likely run into computational difficulties. Much more promising, are credulous reasoners that search for an arbitrary extension. This task remains tractable for relatively expressive default rules combined with Horn theories (Selman 1990). By contrast, skeptical default reasoning, *i.e.*, determining what holds in all extensions of a theory, can be shown to be strictly harder than goal-directed reasoning, and thus our intractability result carries over to skeptical reasoning.

Conclusions

In this paper, we have examined the problem of computing abductive explanations. We have shown that given a Horn theory and a letter q, some non-trivial explanation for q can be calculated in polynomial time. However, goal-directed abduction or the use of an assumption set renders the problem intractable, even for acyclic Horn theories. Thus, the exponential worst-case complexity of the ATMS is not just a consequence of having to return an exponential number of answers; generating even one explanation containing letters from the assumption set is inherently difficult. It appears unlikely, therefore, that the efficiency of the ATMS algorithm can be significantly improved.

This work also shows that there is a strong connection between computing explanations and computing extensions in default logic. Our Support Selection Task is at the core of both assumption-based abductive reasoning and goal-directed default reasoning. We need to minimize support sets for the former, and maximize them for the latter, but neither is hard for Horn theories. In both cases, the difficult task is deciding on an appropriate set of assumptions to make.

Finally, given the difficulty of dealing with acyclic Horn theories, this work suggests that we may not be able to trade expressiveness for tractability in abductive reasoning. It may turn out that there are *no* interesting restrictions on the background theory that could guarantee efficient abduction (for some class of queries). This is quite unlike the situation with deductive reasoning, where there is a linear time algorithm for propositional Horn theories (Dowling and Gallier 1984), and with default reasoning, where there are polynomial algorithms for certain acyclic default theories (Selman and Kautz 1988; Kautz and Selman 1989). If we want to produce explanations automatically, in a timely fashion, and over a wide class of inputs, there may be no alternative to some notion of "approximate" explanation, or perhaps some principled form of unsound or incomplete abduction (a proposal for which is suggested in Levesque (1989)).[12]

Acknowledgments

This research was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada. We thank Ray Reiter for bringing to our attention the close relation between default reasoning and the ATMS, and for useful suggestions and discussions.

References

Allemang, D., Tanner, M.C., Bylander T., and Josephson, J.R. (1987) On the computational complexity of hypothesis assembly. *Proceedings IJCAI-87*, Milan, Italy, 1987, 1112-1117.

de Kleer, J. (1986a) An assumption-based TMS, *Artificial Intelligence*, Vol. 28, #2, 1986, 127–162.

de Kleer, J. (1986b) Problem solving with the ATMS, *Artificial Intelligence*, Vol. 28, #2, 1986, 197–224.

de Kleer, J. and Williams, B.C. (1989) Diagnosis with behavioral modes. *Proceedings IJCAI-89*, Detroit, MI, 1989, 1325-1330.

Dowling, W.F. and Gallier, J.H. (1984). Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, **3**, 1984, 267–284.

Gabow, H.N., Maheshwari S.N, and Osterweil L. (1976). On two problems in the generation of program test paths. *IEEE Trans. Software Engineering*, 1976, 227–231.

Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability, A Guide to the Theory of NP-Completeness.* New York: W.H. Freeman, 1979.

Kautz, H.A. and Selman, B. (1989) Hard problems for simple default logics. *Proceedings of the First International Conference on Knowledge Representation and Reasoning (KR-89)*, Toronto, Ont., 1989, 189–197.

Levesque, H.J. (1989). A knowledge-level account of abduction. *Proceedings IJCAI-89*, Detroit, MI, 1989, 1061–1067.

McAllester, D. (1985). A widely used used truth-maintenance system. MIT, AI-lab memo, 1985.

Poole, D. (1988). A methodology for using a default and abductive reasoning system. Technical report, Dept. of Computer Science, University of Waterloo, Waterloo, Ont., 1988.

Provan, G.M. (1987). Efficiency analysis of multiple-context TMSs in scene representation. *Proceedings AAAI-87*, Seattle, WA, 1987, 173–178.

Provan, G.M. (1988). The Computational Complexity of Truth Maintenance System. Technical Report 88-11, University of British Columbia, Department of Computer Science, 1988.

Reggia, J. (1983) Diagnostic expert systems based on a set-covering model. *International Journal of Man Machine Studies*, 19(5), 1983, 437–460.

Reiter, R. and de Kleer, J. (1980). Foundations of assumption-based truth maintenance systems. *Proceedings AAAI-87*, Seattle, WA, 1987, 183–188.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1), 1987, 57–96.

Selman, B. and Kautz, H.A. (1988) The Complexity of Model-Preference Default Theories. *Proceedings of the Seventh Biennial Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI-88)*, Edmonton, Alberta, 1988, 102–109. Extended version to appear in *Artificial Intelligence.*

Selman, B. (1990) Tractable default reasoning. Ph.D. Thesis, Dept. of Computer Science, University of Toronto, Toronto, Ont. (forthcoming.)

Stillman (1989) The complexity of Horn theories with normal unary defaults. Technical report, General Electric Research and Development Center, Schenectady, N.Y., 1989.

Valiant, R.G. and Vazirani V.V. (1986) NP is as easy as detecting unique solutions. *Theoretical Computer Science*, #47, 1986, 85–93.

[12] Recently, it has been brought to our attention that Provan (1988) has independently obtained complexity results for the ATMS which are similar to our results, but somewhat weaker.