Generating Tests using Abduction

Sheila McIlraith
Department of Computer Science
University of Toronto
Toronto, M5S 1A4 Canada

Abstract

Suppose we are given a theory of system behavior and a set of candidate hypotheses. Our concern is with generating tests which will discriminate these hypotheses in some fashion. We logically characterize test generation as abductive reasoning. Aside from defining the theoretical principles underlying test generation, we are able to bring to bear the abundant research on abduction to show how test generation can be embodied in working systems. Furthermore, we address the issue of computational complexity. It has long been known that test generation is NP-complete. This is consistent with complexity results on the generation of abductive explanations. By syntactically restricting the description of our theory of system behavior or by limiting the completeness of our abductive reasoning, we are able to gain insight into tractable test generation problems.

1 INTRODUCTION

Diagnostic reasoning is often viewed as an iterative generate-and-test process. Given a description of a system together with observations of system behavior, a set of candidate diagnoses is produced which account for the observed behavior. From the set of candidate diagnoses, one or more tests is generated, executed and the observed behavior fed back into the diagnostic problem solver to determine a new set of candidate diagnoses. In this paper we specifically examine the task of test generation as it applies to hypothetical reasoning, and in particular to diagnosis.

Consider a set of hypotheses HYP which we entertain about some state of affairs represented by a first-order sentence Σ. We are concerned with generating tests to discriminate these hypotheses relative to some hypothetical reasoning goal. In a diagnosis setting, the hypotheses could represent potential diseases, the diagnostic goal, to eliminate a particular disease candidate from consideration and the tests, observations of symptoms or medical test results. In an active vision setting, the hypotheses could represent candidate interpretations for an object in a scene, the goal, to uniquely identify the object by candidate elimination and the tests, observation of new visual features resulting from a camera movement. Hypothetical reasoning covers a range of AI applications, all characterized by the objective of generating hypotheses and then distinguishing these hypotheses relative to some theory through the use of testing. Diagnosis, plan recognition, image understanding and aspects of natural language understanding are all instances of hypothetical reasoning problems.

Hardware designers have examined the problem of test generation for years. It is acknowledged to be computationally costly; even the problem of generating tests for simple combinational Boolean circuits is NP-complete (Ibarra and Sahni, 1975). Much of what is found in the traditional design literature is test generation algorithms for specific classes of digital circuits. These algorithms are not directly applicable to the diversity of test generation problems in hypothetical reasoning domains. In the AI test generation literature, the emphasis has also been on diagnosis of digital circuits. DART (Genesereth, 1984) and GDE (de Kleer and Williams, 1987) for example, both provide mechanisms for rudimentary test generation within their diagnostic frameworks. Much of the AI literature focuses on strategies to deal with complexity, such as the use of hierarchical designs (Shirley and Davis, 1983), (Genesereth, 1984), probabilities (de Kleer, 1991) and look-up tables (Meerwijk and Preist, 1992). Computational architectures have been proposed for generating tests for circuits (Shirley, 1986). Interestingly, there has been little to no formal analysis of the problem of test generation in the AI literature. Our objective is to move beyond the specific problem of testing digital circuits and to examine the general problem of test generation for hypothetical reasoning, including diagnosis.

In an earlier paper, McIlraith and Reiter (McIlraith and Reiter, 1992) provided a logical characterization of testing for hypothetical reasoning. They characterized tests in terms of the prime implicates $PI(\Sigma)$ of Σ. Since the ATMS computes (many) $PI(\Sigma)$ in generating diagnoses, it was shown that some propositional tests could simply be "read off" from $PI(\Sigma)$ with no further computation necessary. Many tests are thus generated *for free*. While a nice result for ATMS-based problem solvers, it is of limited use for hypothetical reasoning problem solvers that do not compute the prime implicates of Σ.

In this paper we take the logical characterization of tests introduced in (McIlraith and Reiter, 1992) and use it as a basis for examining the task of test generation. We augment and extend the framework to a first-order characterization. Then we recast test generation as abduction. In so doing, we are able to apply the abundant research on abduction to gain insight into the generation of tests. Specifically, we show how the theoretical characterization can be embodied in a variety of a different computational mechanisms. Finally, we examine the issue of complexity, gaining insight into tractable and intractable test generation problems.

2 PRELIMINARIES

We review and expand upon the testing framework provided in (McIlraith and Reiter, 1992). A fixed first-order language is assumed throughout. Σ will be a fixed sentence of the language, and will serve as the relevant background knowledge describing the system under analysis. For example, in the case of circuits, Σ might describe the individual circuit components, their normal input/output behavior, their fault models, the topology of their interconnections, and the legal combinations of circuit inputs (e.g. (de Kleer and Williams, 1987), (Reiter, 1987)). We also assume a fixed set HYP of hypotheses. In the case where Σ describes a circuit, HYP might be the set of diagnoses which we currently hold for this device. How we arrived at the set HYP will be largely irrelevant for our purposes. HYP could be a set of abductive hypotheses (Poole, 1989), the result of a consistency-based diagnostic procedure (de Kleer et al., 1992), or any other conceivable form of hypothesis generation. We make two assumptions about $H \in HYP$. The first assumption is that H be a conjunction of distinguished ground literals of the language. The second assumption is that the truth status of the hypotheses is unknown, i.e., $\forall H \in HYP$, $\Sigma \not\models H$ and $\Sigma \not\models \neg H$.

A test specifies certain initial conditions which may be established by the tester, together with an observation whose outcome in the physical world determines the test conclusions. The initial conditions must be consistent with the theory and with the current hypotheses being entertained. For example, in circuit diagnosis the initial conditions of a test might be the provision of certain fixed circuit inputs, and the observation might be the resulting value of a circuit output, or the value of an internal probe. In the medical setting, the initial conditions might involve performing a laboratory procedure like a blood test, and the observation might be the white cell count. In an active vision setting, the initial conditions might involve changing the camera angle or moving objects in the scene, and the observation might be some aspect of the corresponding image. Some tests do not dictate initial conditions. Such is the case when the test involves simply reading a sensor value or querying a user.

To provide a formal definition of a test, we distinguish a subset of ground literals of our language, called the *achievable literals*. These will specify the initial conditions for a test. In addition, we define a distinguished subset of the literals of our language called the *observables*. Thus, a test specifies some initial condition A which the tester establishes, and an observable o whose truth or instantiated value the tester is to determine from the physical world.

Definition 1 (Test) *A test is a pair (A, o) where A is a conjunction of achievable literals and o is an observable.*

We distinguish between two types of tests, **truth tests**, which tell us *whether* the observable is true in the physical world, and **value tests**, which tell us *what* instance of the observable is true in the physical world.

Definition 2 (Truth Test) *Let the observable o be a ground literal. A truth test is a test (A, o), whose outcome α is one of o, $\neg o$.*

$(blood\_test, hepatitis\_A\_virus)$ is an example of a truth test. As a result of performing $(blood\_test, hepatitis\_A\_virus)$ in the physical world, the truth value of $hepatitis\_A\_virus$ is established; the outcome of the test is either $hepatitis\_A\_virus$, otherwise $\neg hepatitis\_A\_virus$.

In contrast, a value test establishes the existence and truth status of an instance of the observable in the physical world.

Definition 3 (Value Test) *Let the observable o contain at least one uninstantiated variable. A value test is a test (A, o) whose outcome α is a ground literal o', the instantiation of the observable o, or its negation.*

An example of a value test would be $(\{\}, colour(object, X))$, where X is an uninstantiated variable. As a result of performing $(\{\}, colour(object, X))$ in the physical world, the outcome might be $colour(object, red)$, establishing the existence and truth value of a particular instance of the observable.

Definition 4 (Confirmation, Refutation) *The outcome α of the test (A, o) confirms $H \in HYP$ iff $\Sigma \wedge A \wedge H$ is satisfiable, and $\Sigma \wedge A \models H \supset \alpha$. α refutes H iff $\Sigma \wedge A \wedge H$ is satisfiable, and $\Sigma \wedge A \models H \supset \neg\alpha$.*

Not all conjunctions A of achievable literals will be legal initial conditions, for example simultaneously making a digital circuit input 0 and 1. Since Σ will encode constraints determining the legal initial conditions, we require that $\Sigma \wedge A$ be satisfiable. Moreover, hypothesis H could conceivably further constrain the possible initial conditions A permitted in a test. For example, the hypothesis that a patient is pregnant would prevent a test in which an x-ray is performed. In such a case, Σ would include a formula of the form $pregnant \supset \neg x\_ray$ so that $\Sigma \wedge pregnant \wedge x\_ray$ would be unsatisfiable, in which case the very idea of a confirming or refuting outcome of such a test would be meaningless.

(McIlraith and Reiter, 1992) show that a refuting test outcome allows us to reject H as a possible hypothesis, regardless of how we arrived at our space of hypotheses, HYP. A confirming test outcome is generally of no deterministic value except in the case where our space of hypotheses is defined abductively and HYP is comprised of all and only the hypotheses being considered. In such a case, it was shown that there is a duality between confirming and refuting tests and that a confirming test outcome has discriminatory power, eliminating hypotheses which do not explain it, by virtue of the definition of abductive hypothesis.

Discriminating tests are characterized as those tests (A, o) which are guaranteed to discriminate an hypothesis space HYP, i.e., which will refute at least one hypothesis in HYP, regardless of the test outcome.

Definition 5 (Discriminating Tests) *A test (A, o) is a discriminating test for the hypothesis space HYP iff $\Sigma \wedge A \wedge H$ is satisfiable for all $H \in HYP$ and there exists H_i, $H_j \in HYP$ such that the outcome α of test (A, o) refutes either H_i or H_j, no matter what that outcome might be.*

By definition, a discriminating test must refute at least one hypothesis in the hypothesis space.

Definition 6 (Minimal Discriminating Tests) *A discriminating test (A, o) for the hypothesis space HYP is minimal iff for no proper subconjunct A' of A is (A', o) a discriminating test for HYP.*

Minimal discriminating tests preclude unnecessary initial conditions, for example unnecessary medical tests, camera movement, etc. Only those conditions necessary for producing the test outcome are invoked.

In many instances our theory will not provide us with discriminating tests. Relevant tests are those tests (A, o) which have the *potential* to discriminate an hypothesis space HYP, but which cannot be guaranteed to do so. Given a particular outcome α, a relevant test may refute a subset of the hypotheses in the hypothesis space HYP, but may not refute any hypotheses if $\neg\alpha$ is observed. Since there is no guarantee a priori of the outcome of a test, these tests are not guaranteed to discriminate an hypothesis space.

Definition 7 (Relevant Tests) *A test (A, o) is a relevant test for the hypothesis space HYP iff $\Sigma \wedge A \wedge H$ is satisfiable for all $H \in HYP$ and the outcome α of test (A, o) either confirms a subset of the hypotheses in HYP or refutes a subset.*

By definition, a relevant test confirms or refutes at least one hypothesis in HYP.

Definition 8 (Minimal Relevant Tests) *A relevant test (A, o) for the hypothesis space HYP is minimal iff for no proper subconjunct A' of A is (A', o) a relevant test for HYP.*

Example 1. To illustrate, consider a simple medical diagnosis problem where we suspect that a patient is suffering from either mumps, measles, chicken pox or flu.

$HYP = \{mumps, measles, chicken\_pox, flu\}$
$\Sigma =$
$(measles \supset red\_spots) \wedge (chicken\_pox \supset red\_spots)$
$\wedge\ (mumps \supset swollen\_glands) \wedge (flu \supset fever)$

Both the hypothesis that the patient has measles and the hypothesis that the patient has chicken pox, infer the observation of red spots. However, neither the hypothesis that the patient has mumps or the hypothesis that the patient has the flu infer anything about the existence or lack of existence of red spots. As a result, the outcome of a test to observe red spots will only provide discriminatory information if we observe red_spots to be false. In such a case we can refute both $chicken\_pox$ and $measles$. However, if we observe red_spots to be true, we are unable to reject any of the four hypotheses. Thus, the test $(\{\}, red\_spots)$ is an example of a minimal relevant test. No discriminating test exists for our theory Σ and hypothesis space HYP.

(McIlraith and Reiter, 1992) further showed that if HYP contains all and only the hypotheses to be considered, and if the space of hypotheses is defined abductively, then every relevant test acts as a discriminating test. In our example above, if these conditions are met, then the outcome red_spots of the relevant test $(\{\}, red\_spots)$ would eliminate flu and $mumps$ since neither hypothesis abductively explains red_spots.

Example 2. Consider a bin-picking problem where a smart computer vision system is trying to identify fruit coming down a conveyor belt. The fruit is limited to apples, lemons, limes and bananas. Hypotheses are defined abductively.

$HYP =$
$\{is(object, apple),\ is(object, lemon),\ is(object, lime),$
$is(object, banana)\}$
$\Sigma =$
$(is(object, apple) \supset colour(object, red)) \wedge$
$(is(object, lemon) \supset colour(object, yellow)) \wedge$
$(is(object, lime) \supset colour(object, green)) \wedge$
$(is(object, banana) \supset colour(object, yellow)) \wedge$
$(is(object, apple) \oplus^1 is(object, lemon) \oplus$
$is(object, lime) \oplus is(object, banana))$

By performing the value test $colour(object, X)$ with outcome $colour(object, red)$, that one test would allow us to uniquely identify the object on the conveyor belt as an apple. In contrast, we might have had to perform a number of truth tests before arriving at the same hypothesis space.

Finally, there is an even weaker notion of a test which has the potential to provide further information about the hypothesis space, but which generally does not uniquely discriminate hypotheses.

Definition 9 (Constraining Test) *A test (A, o) is a constraining test for the hypothesis space HYP iff $\Sigma \wedge A \wedge H$ is satisfiable for all $H \in HYP$ and the outcome α of test (A, o) either confirms or refutes a conjunction of hypotheses drawn from HYP.*

A constraining test has the potential to further constrain or limit the hypothesis space, but not in itself eliminate any hypotheses, except in the limiting case. The limiting case occurs when the conjunction of hypotheses contains only one hypothesis. In such a case, a constraining test becomes a relevant test.

To illustrate the notion of a constraining test, consider $\Sigma \models H_1 \wedge H_2 \supset b$, and consider the test (A, b). If the outcome α of test (A, b) is $\neg b$, then $\Sigma \wedge A \wedge \neg b \models \neg H_1 \vee \neg H_2$. The outcome of the test constrains the hypothesis space and refutes the *conjunction* of hypotheses $H_1 \wedge H_2$. Although the test did not refute an individual hypothesis, it has provided further discriminatory information. Given this test outcome, if another test results in the refutation of $\neg H_1$ (i.e., the entailment of H_1), then the additional information from the constraining test (A, b) enables refutation of H_2 (i.e., the entailment of $\neg H_2$).

Proposition 1 (Test Relationships) *Every discriminating test is a relevant test. Every relevant test is a constraining test.*

[1]The connective $\oplus$ is used for notational brevity to indicate exclusive-or

3 TEST GENERATION AS ABDUCTION

Suppose we are given a theory of system behavior, a set of hypotheses, a set of achievables and a set of observables. The task is to generate a test, drawn from the set of achievables and observables which will meet some hypothetical reasoning objective. The objective could be to refute a particular hypothesis, to confirm a particular hypothesis, or simply to discriminate the space.

Intuitively, the generation of tests, particularly the generation of observable outcomes seems to be deductive in nature. Given a theory Σ and achievable A, conjoin the hypothesis H and predict observations. Test whether those observations are indeed true, and if they are false, refute H.

There are several problems with using deduction to generate tests. Theorem provers generally use resolution refutation to deduce whether or not a particular proposition is true, not to deduce what is true (i.e., all logical consequences of a theory). Furthermore, deduction alone does not resolve the problem of identifying both the achievables and the observables of a test.

A better formulation of test generation is as *theory formation*. Given Σ and the objective of generating a test to attempt to eliminate $H \in HYP$, what test could be conjoined to Σ to potentially refute H? (i.e., Find a test (A, o) with outcome α such that $\Sigma \wedge A \wedge \alpha \models \neg H$.)

The pattern of inference is easily recognized as **abduction**.
It is logically equivalent to finding a test (A, o) with outcome α such that $\Sigma \wedge A \wedge H \models \neg\alpha$.

In this section, we characterize test generation as abduction (Cox and Pietrzykowski, 1986), (Poole et al., 1987). We limit ourselves to the examination of truth tests, but the generation of value tests are a simple extension of these results. The sections to follow examine some practical benefits of this theoretical characterization.

Definition 10 (Abductive Explanation) *Given a first-order theory Σ and a ground literal obs, E, a conjunction of literals is an abductive explanation for obs iff $\Sigma \wedge E \models obs$ and $\Sigma \wedge E$ is satisfiable.*

Definition 11 (Min. Abductive Explanation) *E is a minimal abductive explanation for obs iff no proper subconjunct of E is an abductive explanation for obs.*

Testing is performed to meet some hypothetical reasoning objective. Often the objective is simply to perform tests which will eliminate the maximum number of hypotheses. In other instances, it may be desirable to confirm a particular hypothesis, to refute a particular hypothesis or to discriminate (and thus eliminate) some subset of hypotheses in the hypothesis space HYP. The strategy for selecting the type of test to generate may depend on the user's or system's goals and objectives. It may be influenced by decision theoretic measures of utility such as the cost (in dollars, time or human terms) of computing or executing a test, the criticality of a particular hypothesis being true or false, information gain etc. Strategies for generating tests may also depend on probabilities relating to the expected outcome of a particular test or the probability that a particular hypothesis is true or false. We do not address these issues in this paper, but rather focus on the underlying task of test generation. We characterize the notion of confirmation and refutation in terms of abductive explanations. Further, we demonstrate how a variety of tests may be characterized and hence generated abductively. The following proposition is a direct result of Definition 4.

Proposition 2 (Confirmation, Refutation) *The outcome α of the test (A, o) confirms $H \in HYP$ iff $\Sigma \wedge A \wedge H$ is satisfiable, and $A \wedge \neg\alpha$ is an abductive explanation for $\neg H$. α refutes H iff $\Sigma \wedge A \wedge H$ is satisfiable, and $A \wedge \alpha$ is an abductive explanation for $\neg H$.*

Example 3. Returning to the axioms provided in Example 1, the outcome *red_spots* of the test $(\{\}, red\_spots)$ confirms *measles* and *chicken_pox* since $\neg red\_spots$ is an abductive explanation for both $\neg measles$ and $\neg chicken\_pox$. Similarly, the outcome $\neg swollen\_glands$ of the test $(\{\}, swollen\_glands)$ would refute *mumps* since $\neg swollen\_glands$ is an abductive explanation for $\neg mumps$.

If our objective is to establish the truth or falsity of a particular hypothesis $H_i \in HYP$, we ideally want to generate and perform a minimal individual discriminating tests. As a result of this test, we will know either H_i or $\neg H_i$. For example, we may want to establish whether or not a patient is suffering from the particularly virulent hepatitis A. In a vision application, we may want to pick out all the apples from a bowl of fruit. Both examples may be addressed by performing individual discriminating tests.

Theorem 1 (Individual Discriminating Tests) *(A, o) is an individual discriminating test for $H_i \in HYP$ iff*

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$A \wedge o$ is an abductive explanation for $\neg H_i$; $\Sigma \wedge A \not\models \neg H_i$;*
3. *$A \wedge \neg o$ is an abductive explanation for H_i; $\Sigma \wedge A \not\models H_i$.*

The condition that $\Sigma \wedge A \not\models \neg H_i$ and $\Sigma \wedge A \not\models H_i$ ensure that it is the observable and not the achievable which is refuting the hypotheses. Note that this condition is addressed through the use of minimal abductive explanations in every minimal test defined in this section.

Corollary 1 (Min. Ind. Discriminating Tests) *(A, o) is a minimal individual discriminating test for $H_i \in HYP$ iff*

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$A' \wedge o$ is a minimal abductive explanation for $\neg H_i$;*
3. *$A'' \wedge \neg o$ is a minimal abductive explanation for H_i;*
4. *$A = A' \wedge A''$.*

The following corollary also pertains to minimal individual discriminating tests.

Corollary 2 (Min. Ind. Discriminating Tests) *(A, o) is a minimal individual discriminating test for $H_i \in HYP$ iff*

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$A \wedge o \wedge \neg o$ is a minimal abductive explanation for $H_i \vee \neg H_i$.*

Condition 2 of Corollary 2 is trivial in the sense that $o \wedge \neg o$ is vacuously *false*, $H_i \vee \neg H_i$ is vacuously *true* and $false \supset true$. However, this corollary will still be of assistance in computing minimal individual discriminating tests in the sections to follow.

Example 4. To illustrate the concepts in this section, we take liberties with our domain and extend the Σ described in Example 1 by conjoining the following three axioms:
$(hepatitis\_A \wedge blood\_test \supset hepatitis\_A\_virus) \wedge (\neg hepatitis\_A \wedge blood\_test \supset \neg hepatitis\_A\_virus) \wedge (mumps \supset \neg red\_spots) \wedge (hepatitis\_A \supset jaundice)$

$HYP = \{mumps,\ measles,\ chicken\_pox,\ flu,\ hepatitis\_A\}$

$(blood\_test, hepatitis\_A\_virus)$ is a minimal individual discriminating test for *hepatitis_A* since $blood\_test \wedge hepatitis\_A\_virus$ is a minimal abductive explanation for *hepatitis_A*, and $blood\_test \wedge \neg hepatitis\_A\_virus$ is a minimal abductive explanation for $\neg hepatitis\_A$.

As noted previously, many domains do not provide discriminating tests. In such cases, we must settle for a relevant test in order to attempt to eliminate hypotheses. Relevant tests are those tests which have the potential to discriminate an hypothesis space, but which cannot be guaranteed to do so since they only discriminate if α is observed, but not if $\neg\alpha$ is observed. In this instance we want to generate and perform an individual relevant test.

Theorem 2 (Individual Relevant Tests)
(A, o) is an individual relevant test for the hypothesis space HYP iff

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$A \wedge o$ is an abductive explanation for $\neg H_i$; $\Sigma \wedge A \not\models \neg H_i$.*

Corollary 3 (Min. Ind. Relevant Tests)
(A, o) is a minimal individual relevant test for the hypothesis space HYP iff

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$A \wedge o$ is a minimal abductive explanation for $\neg H_i$.*

Further to Example 4., $(\{\}, jaundice)$ is a minimal individual relevant test for the hypothesis *hepatitis_A*, since $\neg jaundice$ is a minimal abductive explanation for $\neg$*hepatitis_A*. A test outcome of *jaundice* provides no discriminatory information. Recall again that when the space of hypotheses is defined abductively and when HYP represents all the hypotheses to be considered, then the observation of *jaundice* would result in the elimination of all hypotheses in HYP except *hepatitis_A*, since it is the only hypothesis which explains the observation of *jaundice*.

Unless we are interested in focusing on a particular hypothesis, our testing objective will likely be to perform tests which refute a maximum number of hypotheses in the hypothesis space. Ideally we want to generate minimal discriminating tests because they guarantee that the outcome, when conjoined to Σ will refute at least one hypothesis in HYP.

Theorem 3 (Discriminating Tests)
(A, o) is a discriminating test for the hypothesis space HYP iff

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$\exists H_i \in HYP$ such that $A \wedge o$ is an abductive explanation for $\neg H_i$; $\Sigma \wedge A \not\models \neg H_i$;*
3. *$\exists H_j \in HYP$ such that $A \wedge \neg o$ is an abductive explanation for $\neg H_j$; $\Sigma \wedge A \not\models \neg H_j$.*

Corollary 4 (Minimal Discriminating Tests)
(A, o) is a minimal discriminating test for the hypothesis space HYP iff

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$\exists H_i \in HYP$ such that $A^{'} \wedge o$ is a minimal abductive explanation for $\neg H_i$;*
3. *$\exists H_j \in HYP$ such that $A^{''} \wedge \neg o$ is a minimal abductive explanation for $\neg H_j$;*
4. *$A = A^{'} \wedge A^{''}$.*

Again, when discriminating tests do not exist or are not achievable, relevant tests are the next best alternative.

Theorem 4 (Relevant Tests)
(A, o) is a relevant test for the hypothesis space HYP iff

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$\exists H_i \in HYP$ such that $A \wedge o$ is an abductive explanation for $\neg H_i$; $\Sigma \wedge A \not\models \neg H_i$.*

The definition of minimal relevant test follows from the theorem above, as per Corollary 2.

Finally, if no relevant test exists or is unachievable, a constraining test may be desirable.

Theorem 5 (Constraining Tests)
(A, o) is a constraining test for the hypothesis space HYP iff

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$A \wedge o$ is an abductive explanation for $\bigvee_{H_i \in HYP} \neg H_i$;*
3. *$\Sigma \wedge A \not\models \bigvee_{H_i \in HYP} \neg H_i$.*

We add the third condition to eliminate both the case where the test achievable alone causes the conjunction of hypotheses to be refuted and the case where the conjunction of hypotheses is already refuted by Σ. For example, if our theory states that H_i and H_j are mutually exclusive (i.e., $\neg(H_i \wedge H_j)$) then no test is needed to discriminate them. If condition 3 is violated, then a constraining test must be designed using a subset of HYP for which condition 3 holds.

Corollary 5 (Minimal Constraining Tests)
(A, o) is a minimal constraining test for the hypothesis space HYP iff

1. *$\Sigma \wedge A \wedge H$ is satisfiable $\forall H \in HYP$;*
2. *$A \wedge o$ is a minimal abductive explanation for $\bigvee_{H_i \in HYP} \neg H_i$.*

4 PRACTICAL BENEFITS

There are many benefits to formal specification of a reasoning task. Primarily, it provides a non-procedural specification of the task from which meta-theoretic properties may be proven. From it, we are able to assess the impact of assumptions, of syntactic restrictions etc. Furthermore, it enables us to realize the task relative to the specification and to establish correctness proofs for our algorithms. In this particular instance, we are fortunate that we have characterized test generation in terms of abduction, an inference procedure that boasts a large body of research. As a result, we are able to immediately exploit research in abduction to gain valuable insight into test generation.

Here, we examine two issues: the mechanization of test generation and tractable abductive test generation.

4.1 MECHANIZING TEST GENERATION

By characterizing test generation as abduction we may employ existing abductive reasoning mechanisms to generate tests. In this section we propose several different approaches for generating tests abductively using theorem proving techniques. Some of the mechanisms are propositional, while others are first order. Recall that a first-order theory of finite domain can be transformed into a propositional theory; thus enabling the use of propositional machinery.

The general problem of abductive test generation is to find a test (A, o) satisfying a logical formula of the form $\Sigma \wedge A \wedge O \vdash X$, where O represents o or $\neg o$ and X represents an individual (negated) hypothesis or a disjunction of negated hypotheses. O and X are determined by the type of test and are specified in Theorems 1-5 and Corollaries 1-5 of the previous section. For example, when generating an individual relevant test, as specified in Theorem 2, O would be o and X would be $\neg H_i$.

By Proposition 1, we know that every discriminating test is a relevant test and that every relevant test is a constraining test. Thus, the various tests can be generated from a basic core. If we are interested in individual tests to refute H_i, then we can try to find a minimal individual relevant test which provides an abductive explanation for $\neg H_i$, as per Corollary 3. The resulting test (A, o) may then be examined to see whether it can satisfy the further requirements of a minimal individual discriminating test. Alternatively, we can use Corollary 2 to attempt to generate an individual discriminating test which provides an abductive explanation for $H_i \vee \neg H_i$, but unlike the previous strategy, if this attempt fails we have no test to fall back on.

To eliminate random hypotheses drawn from HYP, a strategy which minimizes the possibility of producing no test is to employ the criteria of Corollary 5 to generate a minimal constraining test and then examine whether it fulfills the more stringent requirements of a minimal relevant test or a minimal discriminating test (Corollary 4).

In order to compute tests, we must perform both consistency testing and actual generation of the abductive explanations. For formula F, $\Sigma \cup F$ is consistent iff $\Sigma \not\vdash \neg F$. First-order logic is semi-decidable. (i.e., Given first-order proof theory and a closed formula, a proof will be found if the formula is valid, but the proof procedure may not terminate if the formula is not valid.) Consequently, there is no decision procedure for determining the consistency of first-order formulae in general. Fortunately, there are decidable first-order theories. In particular, first-order Horn theories without function symbols are decidable. Similarly, some applications with finite domains may be rewritten as propositional theories, which are decidable. If all else fails, consistency checking can be approximated. For example, if after a certain outlay of resources the formulae have not been proven to be inconsistent, then assume that they are consistent. It is up to the developer of an individual application to ensure that consistency checking is decidable either by syntactic restrictions on Σ or by using some reasonable approximation of consistency checking.

The problem of finding an abductive explanation $A \wedge O$ for X may be computed in several different ways. By recasting the problem $\Sigma \wedge A \wedge O \vdash X$ using the deduction theorem, we can categorize the different approaches to generating abductive explanations.

- Proof-tree completion
  - $\Sigma \wedge A \wedge O \wedge \neg X \vdash \bot$,
- Direct-proof method
  - $\Sigma \vdash A \wedge O \supset X$,
  - $\Sigma \wedge \neg X \vdash \neg A \vee \neg O$,
- Model Generation[2]
  - $\Sigma^* \wedge X \vdash \bigvee_i O_i$.

Recall that X and O are defined as per Theorems 1–5 and Corollaries 1–5, and are limited by the restrictions of the specific computational machinery.

4.1.1 Proof-tree Completion

$\Sigma \wedge A \wedge O \vdash X$ is equivalent to $\Sigma \wedge A \wedge O \wedge \neg X \vdash \bot$. As such, the problem of generating an abductive explanation for X may be recast as finding a refutation proof for X which employs literals drawn from a distinguished set of achievables and observables. Currently the most popular mechanism for computing abductive explanations, this technique is often referred to as *proof-tree completion*.

To generate tests, Σ and $\neg X$ may be conjoined and converted to clausal form. Linear resolution may be used to attempt to derive $\bot$. The proof will fail, but will result in so-called *dead ends*. If these dead ends can resolve with achievables and observables to derive $\bot$ then the minimal achievables and observables required for the proof constitute an abductive explanation for X and may constitute a test if they adhere to the specific test criteria defined in Theorems 1-5 or Corollaries 1-5.

Example 5. Returning to Example 4, in order to find at least a minimal constraining test

[2]Severe restricitons apply. See Section 4.1.3.

for HYP given Σ, we must convert Σ to clausal form and conjoin $\neg \bigvee_{H_i \in HYP} \neg H_i$. Thus, we conjoin $\neg(\neg mumps \vee \neg measles \vee \neg chicken\_pox \vee \neg flu \vee \neg hepatitis\_A)$ (which is equivalent to $mumps \wedge measles \wedge chicken\_pox \wedge flu \wedge hepatitis\_A$) to Σ. The proof will terminate at several dead ends including red_spots, $\neg red\_spots$, $swollen\_glands$ etc. The addition of any of these observables would complete the proof, but only the observable red_spots will fulfill the criteria for a minimal discriminating test defined in Corollary 4. Thus, $(\{\}, red\_spots)$ is a minimal discriminating test for Σ and HYP.

There are several proof-tree-completion-style abductive inference engines (e.g., (Pople, 1973), (Cox and Pietrzykowski, 1986), (Cox and Pietrzykowski, 1987), (Poole, 1988), (Poole et al., 1987)). The Theorist framework (Poole, 1989) is one such engine, but the implementation differs slightly in that the distinguished explanation literals (achievables and observables, in our case) are added to Σ a priori and rather than deriving dead ends, Theorist merely notes the distinguished explanation literals which were employed in the refutation proof.

The available implementation of Theorist provides a more sophisticated development environment for users to perform both abductive explanation and prediction. The prediction facilities, like our deductive theorem provers tell us whether or not a particular formula is true, not *what* formulae are true. Theorist classifies user-provided formulae as *Facts, Defaults, Conjectures* and *Observations*. Both *Defaults* and *Conjectures* are used to generate abductive *Explanations* for *Observations*. *Defaults* are also used for prediction.

There are several ways in which the Theorist development environment may be employed to generate tests. The simplest way is to define achievables and observables as *Conjectures* and to use them to generate abductive explanations for a user-supplied X as per Theorems 1–5 and Corollaries 1–5. Alternatively, the Theorist environment could be modified to enable test generation to occur in conjunction with hypothesis generation. It would require the creation of a new set of user-provided formulae called *Conjecturable-tests*, which would contain either achievables and observables, or predefined tests. Taking advantage of the abductive explanation generation machinery already in place, Theorist could take the set of *Explanations* (hypotheses equivalent to HYP) and generate abductive explanations drawn from *Conjecturable-tests* as per Theorems 1–5 and Corollaries 1–5. This would provide the tests to discriminate the original hypothesis space *Explanations* and enable hypothesis generation and test generation to be performed simultaneously.

Finally, off related interest, Sattar and Goebel (Sattar and Goebel, 1991) provided a mechanism within the Theorist system for recognizing so-called *crucial literals* which provides a basis for identifying discriminating tests of the form $(\{\}, o)$. They compute the crucial literals using consistency trees.

4.1.2 Direct-proof Method

Aside from proof-tree completion, there are several ways of generating tests using a direct proof method. The term *direct proof method* is often used to refer to the task of *consequence finding* – finding the consequences of a theory. $\Sigma \wedge A \wedge O \vdash X$ may be recast as both $\Sigma \vdash A \wedge O \supset X$ and $\Sigma \wedge \neg X \vdash \neg A \vee \neg O$ (assuming $\Sigma \wedge \neg X$ is consistent). In both cases, tests may be found from the logical consequences of Σ and $\Sigma \wedge \neg X$, respectively. Unfortunately, while resolution is refutation complete (complete for proof-finding), it is not deductively complete and so does not find all the logical consequences of a theory.

Fortunately, in the case of test generation, we are only interested in a subset of the logical consequences of our theories. Specifically, we want the minimal[3] clauses of the form $\neg A \vee \neg O \vee X$ and $\neg A \vee \neg O$ respectively, and we don't need them all, unless we want to select the *best* tests. Recent advances have been made in developing complete consequence-finding theorem provers for first-order and propositional theories. In particular, Inoue (Inoue, 1991) has developed a complete resolution procedure for consequence-finding, generalized to finding only interesting clauses having certain properties. A set of so-called *characteristic clauses* can be defined to specify both a set of distinguished literals from which the characteristic clauses must be drawn and any other conditions to be satisfied. In our case, the characteristic clauses would be of the form $\neg A \vee \neg O \vee X$ and $\neg A \vee \neg O$ respectively. The augmentation of the theorem prover with a skip rule allows it to focus on generating only the characteristic clauses, rather than generating all minimal logical consequences and further pruning to retrieve the desired subset of clauses. Following Theorems 1–5 and Corollaries 1–5, we can then use such a consequence-finding system to generate tests.

The RESIDUE system (Finger and Genesereth, 1985) used in the implementation of Genesereth's well-known Design Automated Reasoning Tool (DART) is also a first-order consequence-finding procedure; however, RESIDUE does not focus search as extensively as Inoue's system (Inoue, 1991). RESIDUE was employed in DART to generate potential diagnosis candidates by direct proof, and was also used for rudimentary test generation.

When dealing with propositional theories, the task of finding the minimal logical consequences of a theory is by definition equivalent to computing the prime implicates of that theory.

[3] We use the term minimal as per Definition 11.

Definition 12 (Prime implicates) *C is a prime implicate for Σ iff $\Sigma \models C$, and for no proper subset C' of C does $\Sigma \models C'$.*

At the core of the well-known assumption-based truth maintenance system (ATMS) (de Kleer, 1986) is the computation of certain prime implicates of a propositional Horn theory, Σ (Reiter and de Kleer, 1987). Thus, the ATMS contains a propositional consequence-finding procedure for Σ. In this discussion, we refer to the ATMS in the broadest context, to include its extensions beyond Horn theories, to include probabilistic focusing and to include those systems which compute prime implicates incrementally (Kean and Tsiknis, 1990).

The ATMS identifies a distinguished set of literals called *assumptions* which act as the primitive abducible literals for production of abductive explanations. (McIlraith and Reiter, 1992) identified one way of acquiring certain tests from the side effects of the ATMS's computations for generating diagnoses, H_i. Since the ATMS calculates prime implicates of Σ of the form $H_i \supset o$, some tests of the form $(\{\}, o)$ would be generated *for free* through the normal operation of the ATMS. In order to actually *generate* tests using the ATMS, we take advantage of the fact that the ATMS is an abduction engine and make the achievables and observables *assumptions*. This is almost like operating the ATMS backwards. Rather than diagnostic candidates being the abductive explanations for observations, the tests are the abductive explanations for refutable hypotheses. Tests are those (A, o) for which $A \wedge O \supset X$ is a prime implicate of Σ and (A, o) satisfies all other criteria specified for the test.

Depending upon the application, there may be many achievables and observables and this may not be the most efficient mode of test generation. On a positive note, tests are generally composed of one observable and a minimal number of achievables, so the potential for an exponential number of environments is limited. This, along with probabilistic focusing of the ATMS may make the ATMS or one its generalizations a viable mechanism for test generation.

4.1.3 Model Generation

We mention this last approach to abductive test generation only to be thorough, because it is of very limited use in practical test generation applications. Model generation (e.g., Satchmo (Manthey and Bry, 1988),(Denecker and de Schreye, 1992)) and model finding techniques (e.g., GSAT (Selman et al., 1992) may be used to generate abductive hypotheses in limited cases. Console et al. (Console et al., 1991) showed that abductive explanations could be generated deductively from the Clark's completion Σ^* of a causal Horn theory Σ. By augmenting a causal Horn theory with completion axioms, we explicitly provide that a particular effect, e entails a disjunction of possible causes, c_i, i.e., $\Sigma^* \wedge e \vdash \bigvee_i c_i$. The causes, c_i can be determined by generating the minimal models of $\Sigma^* \wedge e$.

These results may be applied to a very restricted class of test generation problems. In particular, those problems for which tests are restricted to $(\{\}, o)$ and for which completion of the theory can be computed with respect to the refutation of causes, not with respect to the potential effects. Specifically, the completion axioms would state that a particular refuted hypothesis (not_H_i) implied a disjunction of possible test outcomes $o_1 \vee not\_o_2 \vee \ldots o_n$. In order to compute a test to refute the hypothesis H_i, the set of minimal models for $\Sigma^* \wedge not\_H_i$ would be computed using a model generator. The tests $(\{\}, o)$ would be the distinguished literals o_i retrieved from the minimal models.

This approach seems both awkward and impractical. A better proposal for producing tests via model generation is to generate the model for Horn theory $\Sigma \wedge \neg X$. Tests of the form $(\{\}, o)$ could then be retrieved as the distinguished observables of the minimal Herbrand model.

4.2 TRACTABLE TEST GENERATION

From the computer hardware literature, we know that the general problem of test generation, even for simple combinational Boolean circuits is NP-complete (Ibarra and Sahni, 1975). Similarly, we know that finding an abductive explanation in the general case is NP-hard (Selman, 1990). The challenge with computationally hard problems is either to attempt to deal with the worst-case complexity by employing problem-specific strategies such as probabilistic focusing of algorithms or alternatively to define tractable classes of the problem. Tractable classes may often be achieved by limiting the expressive power of a theory, or by limiting the completeness of reasoning. In the abduction research, there are a few simple classes of tractable abduction problems. In this section, we examine the complexity results on abduction to attempt to provide insight into classes of tractable tests generation problems.

In defining tractable abductive test generation problems, we may avail ourselves of certain properties of test generation that occur generally or in certain hypothetical reasoning domains. They are as follows:

1. **There is no need to generate all tests**
   In generating tests, there is always a trade-off between the cost of computing tests and the cost of performing tests. In many instances, the cost of performing a test is cheap while the generation of tests is expensive. Consequently, we need not calculate *all* tests or even the best test. Computing any relevant test is generally of value.
2. **Some application tests are limited to $(\{\}, o)$**
   There are many application domains for which

tests require no achievable literals. This issue was discussed in (McIlraith and Reiter, 1992). For example, some applications have a great deal of sensor data available. It is the job of test generation to select which sensor data to "observe"; no achievable preconditions are required. In other domains, tests of the form $(\{\}, o)$ may be performed by simply querying the user as to the truth value of the test proposition o. This may be the case for certain medical diagnosis problems or when performing certain natural language understanding tasks.

3. **An exponential number of tests is unlikely** Many tests are composed of one observable literal and few if any achievable literals. As such, the number of minimal tests generated as abductive explanations is unlikely to be exponential in the number of observables and achievables.

Selman (Selman, 1990), Levesque (Levesque, 1989) and Bylander et al. (Bylander et al., 1991) have all defined classes of tractable abductive reasoning problems. There are some gaps in the complexity results that need to be filled in to deal fully with test generation, however from the existing results we can gain some insight into what makes test generation problems tractable, or for that matter, intractable. We focus here on test generation from *propositional* theories.

Complexity results for abduction are often based on the ATMS. Consequently, the term *assumption* refers to the distinguished set of literals from which explanations are composed. It is equivalent to our set of observable and achievable literals when abduction is applied to test generation.

It has long been known that there may be exponentially many abductive explanations for a given literal ((McAllester, 1985), (de Kleer, 1986)) and so listing them all would take exponential time. For test generation, we are often uninterested in listing all tests as explained by Property 1 above. Even if we were, by Property 3, we would be unlikely to have an exponential number of tests. Assuming, there are not an exponential number of tests, we proceed to define certain complexity results for test generation, viewed as an abductive task.

Selman (Selman, 1990) states that the problem of generating abductive explanations for theories composed of arbitrary clauses is NP-hard, because of the consistency check on Σ. Consequently it follows directly from (Selman, 1990) that:

Proposition 3 *If Σ is a conjunction of arbitrary clauses, the problem of generating a test is NP-hard.*

We would hope that the story would be better for Horn clause theories. Selman further shows that even when Σ is composed of Horn clauses, that finding an abductive explanation for a letter q, where the explanation must be derived from a set of assumptions, is NP-hard. This seems discouraging, but upon analysis of the complexity proof, we see some hope. The proof is based upon a reduction from the NP-complete decision problem "path with forbidden pairs." In this instance, the forbidden pairs are mutually incompatible assumptions drawn from our assumption set. It would appear that if we got rid of the problem of forbidden pairs, that the complexity problem would be resolved. This indeed appears to be the case.

Bylander et al. (Bylander et al., 1991) define the class of *independent abduction problems*. This class of problems has a polynomial time algorithm for finding an explanation, if one exists. The trick is to get rid of Selman's forbidden pairs – to ensure that no assumptions are mutually incompatible in the one instance and to then additionally ensure that there are no cancellation interactions among the assumptions.

If our tests are composed of single literals, then we don't have to concern ourselves with the compatibility of achievables/observables. Property 2 shows that this is a reasonable assumption for tests in certain application domains. Following (Bylander et al., 1991), we show that:

Proposition 4 *If Σ is a conjunction of Horn clauses and tests are of the form $(\{\}, o)$, then a test may be generated in polynomial time, if such a test exists*

This follows directly from the results in (Bylander et al., 1991).

For the general case, the question remains as to whether it seems reasonable to assume that no achievables/observables are mutually incompatible. Note that achievables/observables S_1 and S_2 are defined to be mutually incompatible iff $\Sigma \models \neg(S_1 \wedge S_2)$.

To be able to assume no mutually incompatible achievables/observables, we would have to assume that for every achievable A_i and observable o_i that $\Sigma \not\models \neg (A_1 \wedge A_2)$, $\Sigma \not\models \neg (o_1 \wedge o_2)$ and $\Sigma \not\models \neg (A_1 \wedge o_1)$. While it may be possible to make this assumption in specific instances, it is unlikely to be true in the general case. In circuit diagnosis for example, let A_1 be $input = 1$, A_2 be $input = 0$, obviously $\Sigma \models \neg(A_1 \wedge A_2)$. Similarly, since observations can generally be positive and negative literals, if we let $o_1 = \neg o_2$ then $\Sigma \models \neg(o_1 \wedge o_2)$. We state the following proposition for those situations where there are no mutually incompatible achievables/observables.

Proposition 5 *If Σ is a conjunction of Horn clauses and no two literals drawn from the set of achievable and observable literals are mutually incompatible with respect to Σ, then a test may be generated in polynomial time, if such a test exists.*

Finally, Levesque (Levesque, 1989) and Selman (Selman, 1990) define a linear time algorithm for finding certain explanations of a literal from Horn clause theories. Although motivated by different concerns, their algorithm and results are virtually the same. The explanations produced are those that are *explicitly* represented (Levesque, 1989) in Σ. Further, it is not required that they be drawn from a set of distinguished literals.

The algorithm searches through the clauses of Σ to find clauses containing the literal q, the literal to be explained. The negation of the other literals in the clause form the explanations. For example, if $\neg H$ is to be explained and $x \vee y \vee \neg H$ is a clause in Σ, then the abductive explanation $\neg x \wedge \neg y$ would be found in linear time. Levesque proposes using this algorithm to define a form of limited abductive reasoning in which explicit explanations are determined first, followed by a chaining process to find implicit explanations.

These results tell us that if we have tests (A, o) explicitly represented in Σ as $\neg A \vee \neg o \vee \neg H_i$, then they can be found in linear time, (along with other extraneous explanations that do not contain the desired distinguished literals and thus are not tests per se). Simple *causal theories* where clauses in Σ are of the form *hypothesis* $\supset$ *observable* (e.g., *disease* $\supset$ *symptom*) would contain such explicit tests. This is an argument in favor of encoding or even caching tests explicitly in a theory to make them computationally easy to generate (Meerwijk and Preist, 1992).

Definition 13 (Explicit test) (A, o) *is an explicit test to potentially refute* $H \in HYP$ *if* $\neg A \vee \neg o \vee \neg H$ *is a clause in* Σ.

Proposition 6 *If* Σ *is a conjunction of Horn clauses, an explicit test may be generated in linear time, if such a test exists.*

This follows from results in (Levesque, 1989) and (Selman, 1990).

5 SUMMARY

We provide three main contributions towards research in test generation. First, we characterize test generation as abductive reasoning. As a consequence, we are able to define the notions of discriminating tests, relevant tests, individual discriminating and relevant tests, and constraining tests all in terms of abductive explanation. We then outline a variety of approaches to abductive reasoning which can be modified and employed to perform test generation. Finally, we examine the research on tractable abductive reasoning to gain insight into tractable and intractable test generation problems.

This paper provides both a theoretical and computational framework for test generation, which is lacking in the test generation literature. From this framework and some of the proposed procedures for test generation, there is opportunity for experimental work to analyze the efficacy in practice of some of these alternative approaches to test generation.

Acknowledgements

We would like to thank the following people who contributed towards the development or improvement of this work: Ray Reiter, Chris Preist, David Poole, Mike Gruninger and Javier Pinto. We also acknowledge the generous support of Alberta Research Council and the Natural Science and Engineering Research Council.

References

T. Bylander, D. Allemang, M. Tanner and J. Josephson (1991). The computational complexity of abduction, *Artificial Intelligence Journal* **49**:25–60.

L. Console, D. Theseider Dupre and P. Torasso (1991). On the relationship between abduction and deduction, *Journal of Logic and Computation* **1**(5):661–690.

P. Cox and T. Pietrzykowski (1986). Causes for events: their computation and applications, *Eighth International Conference on Automated Deduction*, 608–621.

P. Cox and T. Pietrzykowski (1987). General diagnosis by abductive inference, in *Proceedings of the IEEE Symposium on Logic Programming*, 183–189.

J. de Kleer (1986). An assumption-based TMS, in *Artificial Intelligence Journal* **28**:127–162.

J. de Kleer (1991). Focusing on probable diagnoses, in *Proceedings of the National Conference on Artificial Intelligence (AAAI-91)*, 842–848.

J. de Kleer and A. Mackworth and R. Reiter (1992). Characterizing diagnoses and systems, in *Artificial Intelligence Journal* **56**:197–222.

J. de Kleer and B. Williams (1987). Diagnosing multiple faults, in *Artificial Intelligence Journal* **32**:97–130.

M. Denecker and D. de Schreye (1992). On the duality of abduction and model generation, in *Proceedings of the Fifth Generation Computer Systems Conference (FGCS-92)*, 650–657.

J. Finger and M. Genesereth (1985). RESIDUE: a deductive approach to design synthesis, Technical Report No. HPP-85-5, Department of Computer Science, Stanford University, CA.

M. Genesereth (1984). The use of design descriptions in automated diagnosis, in *Artificial Intelligence Journal* **24**:411–436.

O. Ibarra and S. Sahni (1975). Polynomially complete fault detection problems, in *IEEE Transactions on Computers* **24**(3):242–249.

K. Inoue (1991). Consequence-finding based on ordered linear resolution, in *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, 158–164.

A. Kean and G. Tsiknis (1990). An incremental method of generating prime implicants/implicates, in *Journal of Symbolic Computing* **9**:185–206.

H. Levesque (1989). A knowledge-level account of abduction, in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* 1061–1067.

R. Manthey and F. Bry (1988). SATCHMO: a theorem prover implemented in Prolog, in *Ninth International Conference on Automated Deduction.*

D. McAllester (1985). A widely used truth-maintenance system, AI-lab Memo Department of Computer Science, MIT Cambridge, MA.

S. McIlraith and R. Reiter (1992). On tests for hypothetical reasoning, in W. Hamscher, L. Console and J. de Kleer, editors, *Readings in model-based diagnosis*, 89–96. Morgan Kaufmann Publishers, San Mateo, CA.

A. Meerwijk and C. Preist (1992). Using multiple tests for model-based diagnosis, in *Proceedings of the Third International Workshop on Principles of Diagnosis*, 30–39, Orcas Island, Washington, USA.

D. Poole (1988). A logical framework for default reasoning, in *Artificial Intelligence Journal* **36**(1):27–47.

D. Poole (1989). Explanation and prediction: an architecture for default and abductive Reasoning, in *Computational Intelligence* **5**:97–110.

D. Poole and R. Goebel and R. Aleliunas (1987). Theorist: a logical reasoning system for defaults and diagnosis, in N. Cercone and G. McCalla, editors, *The Knowledge Frontier: Essays in the Representation of Knowledge*, 331–352, Springer Verlag.

H. Pople (1973). On the mechanization of abductive logic, in *Proceedings of the Third International Joint Conference on Artificial Intelligence (IJCAI-73)*, 147–152.

R. Reiter (1987). A theory of diagnosis from first principles. *Artificial Intelligence Journal* **32**:57–95.

R. Reiter and J. de Kleer (1987). Foundations for assumption-based truth maintenance systems: preliminary report, *Proceedings of the National Conference on Artificial Intelligence (AAAI-87)*,183–188.

A. Sattar and R. Goebel (1991). Using crucial literals to select better theories, in *Computational Intelligence* **7**(1):11–22.

B. Selman (1990). Tractable default reasoning. Ph.D. Thesis, Department of Computer Science, University of Toronto, Canada.

M. Shirley (1986). Generating tests by exploiting designed behavior, in *Proceedings of the National Conference on Artificial Intelligence (AAAI-86)* 884–890.

M. Shirley and R. Davis (1983). Generating distinguishing tests based on hierarchical models and symptom information. in *Proceedings International Conference on Computer Design.*