Generating Tests using Abduction

S. McIlraith
Department of Computer Science
University of Toronto
Toronto, M5S 1A4 Canada
email: mcilrait@ai.toronto.edu
telephone: 416-978-1929, fax: 416-978-1455

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. In the AI literature on hypothetical reasoning and in particular diagnosis, there has been a great deal of research on generating candidate hypotheses, but much less work on the generation of tests to discriminate these hypotheses. This paper represents *preliminary work* on the generation of tests for hypothetical reasoning.

We logically characterize test generation as abductive reasoning. Aside from providing a formal specification of test generation, there is immediate practical benefit from our theoretical characterization. We are able to bring to bear the abundant research on abduction to advance research on the generation of tests. We show how several abductive reasoning frameworks may be used to generate tests. Furthermore, we address the issue of computational complexity. It has long been known that test generation is NP-complete [10]. 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 define notions of tractable test generation.

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 propositional 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 [10]. 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 [9] and GDE [7] 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 [24], [9], probabilities [5] and look-up tables [15]. 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 [14] 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 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 [14] and use it as a basis for examining the task of test generation. We characterize 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 demonstrate how tests may be generated using several different abductive reasoning frameworks. Additionally, we examine the issue of complexity, gaining insight into tractable and intractable test generation problems.

2 Preliminaries

We follow the framework provided in [14], which is reviewed in this section. A fixed propositional 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. [7], [21]). 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 [17], the result of a consistency-based diagnostic procedure [6], 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 literals of the propositional language. The second assumption is that for all $H \in HYP$, $\Sigma \not\models H$ and $\Sigma \not\models \neg H$.

Informally, the notion of a test provides for certain initial conditions which may be established by the tester, together with the specification of an observation whose outcome determines what the test conclusions are to be. 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. We provide for a formal definition of a test by distinguishing a subset of literals of our propositional language, called the *achievable literals*. These will specify the initial conditions for a test. In addition, we require a distinguished subset of the propositional symbols of our language called the *observables*. These will specify the observations to be made as part of a test.

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

A test specifies some initial condition A which the tester establishes, and an observation o whose truth value the tester is to determine.

Definition 2 (Outcome of a test) *The outcome of a test (A, o) is one of o, $\neg o$.*

In other words, as a result of performing the test (A, o) in the physical world, the truth value of o is observed. If o is observed to be true, the outcome of the test is o, otherwise $\neg o$.

Definition 3 (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 $\Sigma \wedge A \models H \supset \neg\alpha$.*

At first, the requirement in this definition that $\Sigma \wedge A \wedge H$ be satisfiable might seem odd. However, 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 radioactivity has escaped within a reactor would prevent a test in which humans enter the reactor chamber. In such a case, Σ would include a formula of the form *radioactivity* $\supset \neg$*enter-chamber* so that $\Sigma \wedge$ *enter-chamber* $\wedge$ *radioactivity* 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 [14] 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 [14] 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.

Definition 4 (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 5 (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 circuit inputs, laboratory tests, 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, relevant tests will 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 6 (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.*

Definition 7 (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.*

Again, minimal relevant tests preclude unnecessary initial conditions.

Example 1.

Consider a simple medical diagnosis problem where we suspect that a patient is suffering from either mumps, measles, chicken pox or the flu.

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

$\Sigma =$

$measles \supset red\_spots$

$chicken\_pox \supset red\_spots$

$mumps \supset swollen\_glands$

$flu \supset fever$

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.

Note, in [14] it was shown that if we assume 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 red_spots would eliminate flu and $mumps$ since neither hypothesis abductively explains red_spots.

3 Characterizing 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 hypothesis 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 to see 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) such that $\Sigma \cup (A, o) \models \neg H$.)

The pattern of inference is easily recognized as **abduction**. [2], [18].

In this section, we characterize test generation as abduction. The sections to follow examine some practical benefits of these theoretical results.

Definition 8 (Abductive Explanation) *Given a propositional theory* Σ *and a propositional formula* O*,* E*, a conjunction of literals is an abductive explanation for* O *iff* $\Sigma \wedge E \models O$ *and* $\Sigma \wedge E$ *is satisfiable.*

Definition 9 (Minimal Abductive Explanation) E *is a minimal abductive explanation for* O *iff no proper subconjunct of* E *is an abductive explanation for* O*.*

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. We characterize the notion of confirmation and refutation in terms of abductive explanations. Further, we demonstrate how discriminating tests and relevant tests may be characterized and hence generated abductively.

Proposition 1 (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* $\neg H$*.*

This follows directly from Definition 3.

Example 2.

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 eliminate hypotheses in HYP, we ideally want to generate minimal discriminating tests. Whether the outcome of our test (A, o) is o or $\neg o$, a discriminating test guarantees that the outcome, when conjoined to Σ will refute at least one hypothesis in HYP.

Theorem 1 (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. $A \wedge o$ *is an abductive explanation for* $\bigvee_{H_i \in HYP} \neg H_i$
3. $A \wedge \neg o$ *is an abductive explanation for* $\bigvee_{H_i \in HYP} \neg H_i$
4. $\Sigma \not\models \bigvee_{H_i \in HYP} \neg H_i$

We add the fourth condition to eliminate the null test. 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. A similar problem arises if the disjunction of hypotheses is a tautology (i.e., $(H_i \vee \neg H_i)$).

If condition 4 is violated, then a discriminating test must be designed using a subset of HYP for which condition 4 holds. The point is to find a test (A, o) such that for some H_m and $H_n \in HYP$, $A \wedge o$ is an abductive explanation for $\neg H_m$ and $A \wedge \neg o$ is an abductive explanation for $\neg H_n$.

Corollary 1 (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. $A' \wedge o$ *is a minimal abductive explanation for* $\bigvee_{H_i \in HYP} \neg H_i$
3. $A'' \wedge \neg o$ *is a minimal abductive explanation for* $\bigvee_{H_i \in HYP} \neg H_i$
4. $A = A' \wedge A''$
5. $\Sigma \not\models \bigvee_{H_i \in HYP} \neg H_i$

All the minimal tests defined here preclude unnecessary initial conditions.

Example 3.
To illustrate the concepts in this section, we take liberties with our domain and extend the Σ described in Example 1 with the the following three axioms.
$hepatitis \supset pos\_blood\_test$
$\neg hepatitis \supset \neg pos\_blood\_test$
$mumps \supset \neg red\_spots$
$HYP = \{mumps, measles, chicken\_pox, flu, hepatitis\}$
To find a minimal discriminating test, we must find a test (A, o) such that:
$A \wedge o$ is a minimal abductive explanation for
$\neg mumps \vee \neg measles \vee \neg chicken\_pox \vee \neg flu \vee \neg hepatitis$, and
$A \wedge \neg o$ is a minimal abductive explanation for
$\neg mumps \vee \neg measles \vee \neg chicken\_pox \vee \neg flu \vee \neg hepatitis$
$(\{\}, red\_spots)$ is such a test since red_spots explains $\neg mumps$ and $\neg red\_spots$ explains $\neg chicken\_pox$ and $\neg measles$.

If our objective is to establish the truth or falsity of a particular hypothesis in HYP, we ideally want to generate minimal individual discriminating tests. As a result of this test, we will know either H_i or $\neg H_i$.

Theorem 2 (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 $\neg H_i$*
3. *$A \wedge \neg o$ is an abductive explanation for H_i*

Corollary 2 (Minimal Individual 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. *$A = A' \wedge A''$*

To illustrate, if our hypothetical reasoning objective is to establish the truth or falsity of *hepatitis*, then $(\{\}, pos\_blood\_test)$ would be an individual discriminating test for *hepatitis*.

As noted previously, many domains do not provide discriminating tests. In such a case, we must generate relevant tests in order to attempt to eliminate hypotheses in HYP. 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.

Theorem 3 (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. *$A \wedge o$ is an abductive explanation for $\bigvee_{H_i \in HYP} \neg H_i$*
3. *$\Sigma \not\models \bigvee_{H_i \in HYP} \neg H_i$*
4. *$A \wedge o$ is not an abductive explanation for $\neg H_i, \forall H_i \in HYP$*

The fourth condition ensures that some discrimination occurs – that an outcome of a test would not result in the refutation of *all* hypotheses. It may be the case in some applications that it is desirable to eliminate all the hypotheses in HYP. In such a case, the fourth condition may be ignored.

Corollary 3 (Minimal Relevant Tests)
(A, o) is a minimal 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 $\bigvee_{H_i \in HYP} \neg H_i$*
3. *$\Sigma \not\models \bigvee_{H_i \in HYP} \neg H_i$*
4. *$A \wedge o$ is not a minimal abductive explanation for $\neg H_i, \forall H_i \in HYP$*

$(\{\}, \neg swollen\_glands)$ is an example of a minimal relevant test. If $\neg swollen\_glands$ is observed, then $mumps$ is refuted. Conversely, if $swollen\_glands$ is observed, then the test is of no discriminatory value in the general case. Recall again that when the space of hypotheses is defined abductively and when HYP represents all the hypotheses to be considered, then all relevant tests act as discriminating tests [14] and the observation of $swollen\_glands$ from the test $(\{\}, \neg swollen\_glands)$ would result in the elimination of all hypotheses in HYP except $mumps$.

The concept of a minimal individual relevant test follows trivially from the descriptions of minimal individual discriminating tests and minimal relevant tests. It is not listed here.

4 Practical Benefits of our Characterization

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.

In particular, 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. Most of the logical abductive reasoning frameworks are based on resolution theorem provers. In this section we demonstrate several different mechanisms for generating tests abductively using resolution theorem provers.

The general problem 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 hypothesis or disjunction of negated hypotheses. O and X are determined by the type of test and are specified in Theorems 1-3 and Corollaries 1-3 of the previous section. For example, when generating a relevant test, as specified in Theorem 3, O would be o and X would be $\bigvee_{H_i \in HYP} \neg H_i$.

$\Sigma \wedge A \wedge O \vdash X$ 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 (e.g., [20], [2], [3], [16] [19]), 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 1-3 or Corollaries 1-3.

Example 4.

Returning to Example 3, in order to find a minimal discriminating test 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 $\neg(\neg mumps \vee$

$\neg measles \vee \neg chicken\_pox \vee \neg flu \vee \neg hepatitis$) (which is equivalent to $mumps \wedge measles \wedge chicken\_pox \wedge flu \wedge hepatitis$) 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 1. Thus, $(\{\}, red\_spots)$ is a minimal discriminating test for Σ and HYP.

There are several proof-tree-style abductive inference engines (e.g., [20], [2], [3], [16] [19]). The Theorist framework [17] is one such engine, but the implementation differs slightly in that the distinguished literals (achievables and observables, in our case) are added to Σ a priori and rather than deriving dead ends, Theorist merely notes the distinguished 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* formulas are true. Theorist classifies user-provided formulas 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 X as per Theorems 1–3 and Corollaries 1–3. Theorist could also be modified by the addition of another set of user-provided formulas called *Conjecturable-tests* which would contain either achievables and observables, or predefined tests. Theorist could be further modified to take the set of *Explanations* (equivalent to HYP) and to generate abductive explanations drawn from *Conjecturable-tests* as per Theorems 1–3 and Corollaries 1–3.

Of related interest, Sattar and Goebel [22] 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.

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. There are several well-known resolution-based algorithms for consequence finding. In particular, the ATMS [4] calculates certain prime implicates (minimal logical consequences) of Σ. ATMS-like algorithms have been used extensively for generating consistency-based diagnoses, but they are also documented abduction engines. The ATMS identifies a distinguished set of literals called *assumptions* which act as the primitive literals for production of abductive explanations. [14] identified one way of acquiring certain tests from the side effects of the ATMS's computations for generating diagnoses. In order to actually *generate* tests using the ATMS, we make the achievables and observables *assumptions*. This is almost like operating the ATMS backwards. Rather than diagnostic candidates being the abductive explanations, the tests are. 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 a viable option.

Finger's RESIDUE system [8] is another example of a first-order consequence finding procedure that may be used to generate tests. It was employed in the DART system [9] to generate potential diagnosis candidates by direct proof. By conjoining $\neg X$ to Σ, RESIDUE will entail so-called *residues* of the proof procedure – clauses which cannot be further resolved. The residues are comparable to the dead ends of the proof-tree completion technique. More recent work on the computation of logical consequences [11] has enabled better focussing of the consequence finding algorithm on clauses containing distinguished literals.

Finally, for propositional causal horn clause theories, tests of the form $(\{\}, o)$ may be generated by deduction on the Clark completion of the causal theory. The relationship between abduction, closure and deduction has been outlined by Console et al [1] among others.

4.2 Tractable Abductive 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 [10]. Similarly, we know that finding an abductive explanation in the general case is NP-hard [23]. 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 define 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. **For some applications, tests are of the form $(\{\}, o)$**
   There are many application domains for which tests require no achievable literals. This issue was discussed in [14]. 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 are unlikely to be exponential in the number of observables and achievables.

Selman [23], Levesque [12] and Bylander et al. [25] 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.

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 [13], [4] 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.

Selman [23] 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 [23] that:

Proposition 2 *If Σ is a conjunction of arbitrary clauses, the problem of generating a test which will potentially refute an hypothesis 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. [25] defines 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 in the other case.

If our tests are composed of single literals, then we don't have to concern ourselves with the compatibility of assumptions. Property 2 shows that this is a reasonable assumption for tests in certain application domains.

Proposition 3 *If Σ is a conjunction of Horn clauses and tests are of the form $(\{\}, o)$, then a test to potentially refute an hypothesis may be generated in polynomial time, if such a test exists*

This follows directly from the results in [25].

For the general case, the question remains as to whether it seems reasonable to assume that no assumptions are mutually incompatible.

Definition 10 *Assumptions S_1 and S_2 are mutually incompatible iff $\Sigma \models \neg(S_1 \wedge S_2)$.*

In the case of test generation, the assumptions are the set of achievables and observables. To be able to assume no mutually incompatible assumptions, 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 $A_1 \equiv \mathit{input} = 1$, $A_2 \equiv \mathit{input} = 0$, 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 perhaps unlikely situations where there are no mutually incompatible achievables and observables.

Proposition 4 *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 to potentially refute an hypothesis may be generated in polynomial time, if such a test exists.*

Finally, Levesque [12] and Selman [23] 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 [12] 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. This would be akin to the look-up tables employed in [15].

Definition 11 (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 5 *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 [12] and [23].

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, individual discriminating tests and relevant tests in terms of abductive explanations and hence compute them abductively. We briefly discuss several abductive mechanisms which may be used to generate tests. Finally, we examine the research on tractable abductive reasoning to gain insight into tractable and intractable test generation. The latter will be clarified and formalized in the final version of this paper.

References

[1] L. Console, D. Theseider Dupre, and P. Torasso. Abductive reasoning through direct deduction from completed domain models. In Z. Ras, editor, *Methodologies for Intelligent Systems 4*, pages 175–182. North Holland, 1989.

[2] P. Cox and T Pietrzykowski. Causes for events: their computation and applications. In *Eighth International Conference on Automated Deduction*, pages 608–621, 1986.

[3] P. Cox and T Pietrzykowski. General diagnosis by abductive inference. In *Proc. IEEE Symposium on Logic Programming*, pages 183–189, 1987.

[4] J. de Kleer. An assumption-based tms. *Artificial Intelligence*, 28:127–162, 1986.

[5] J. de Kleer. Focusing on probable diagnoses. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 842–848, Anaheim, CA., 1991.

[6] J. de Kleer, A.K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, to appear.

[7] J. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[8] J. Finger and M. Genesereth. Residue: a deductive approach to design synthesis. Technical Report HPP-85-5, Department of Computer Science, Stanford University, 1985.

[9] M.R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24:411–436, 1984.

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

[11] K. Inoue. Consequence-finding based on ordered linear resolution. In *Proceedings of the Twelth International Joint Conference on Artificial Intelligence*, pages 158–164, 1991.

[12] H. L. Levesque. A knowledge-level account of abduction. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1061–1067, 1989.

[13] D. McAllester. A widely used truth-maintenance system. Technical report, Department of Computer Science, AI-lab Memo, MIT, Cambridge, MA, 1985.

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

[15] A. Meerwijk and C. Preist. Using multiple tests for model-based diagnosis. In *Proc. of The Third International Workshop on Principles of Diagnosis*, pages 30–39, 1992.

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

[17] D. Poole. Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence*, 5:97–110, 1989.

[18] D. Poole, R.G. Goebel, and R. Aleliunas. 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*, pages 331–352. Springer Verlag, 1987.

[19] D. Poole, R.G. Goebel, and R. Aleliunas. Theorist: a logical reasoning system for defaults and diagno sis. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier: Essays in the Representation of Knowledge*, pages 331–352. Springer Verlag, 1987.

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

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

[22] A. Sattar and R. Goebel. Using crucial literals to select better theories. Technical Report TR 89-27, Department of Computing Science, University of Alberta, 1989.

[23] B. Selman. *Tractable Default Reasoning.* PhD thesis, Department of Computer Science, University of Toronto, 1990.

[24] M. Shirley and R. Davis. Generating distinguishing tests based on hierarchical models and symptom information. In *Proceedings International Conference on Computer Design*, 1983.

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