Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

STEFAN HEINZ[*] MARTIN SACHENBACHER[**]

Using Model Counting to Find Optimal Distinguishing Tests

[*] Supported by the DFG Research Center MATHEON *Mathematics for key technologies* in Berlin.
[**] Technische Universität München, Institut für Informatik Boltzmannstraße 3, 85748 Garching, Germany

ZIB-Report 08-32 (August 2008)

Using Model Counting to Find Optimal Distinguishing Tests

Stefan Heinz[1]* and Martin Sachenbacher[2]

[1] Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany
heinz@zib.de
[2] Technische Universität München, Institut für Informatik
Boltzmannstraße 3, 85748 Garching, Germany
sachenba@in.tum.de

Abstract. Testing is the process of stimulating a system with inputs in order to reveal hidden parts of the system state. In the case of non-deterministic systems, the difficulty arises that an input pattern can generate several possible outcomes. Some of these outcomes allow to distinguish between different hypotheses about the system state, while others do not.

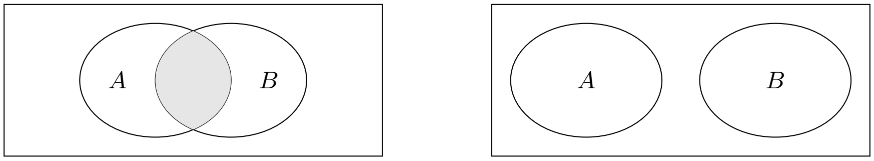
In this paper, we present a novel approach to find, for non-deterministic systems modeled as constraints over variables, tests that allow to distinguish among the hypotheses as good as possible. The idea is to assess the quality of a test by determining the ratio of distinguishing (good) and not distinguishing (bad) outcomes. This measure refines previous notions proposed in the literature on model-based testing and can be computed using model counting techniques. We propose and analyze a greedy-type algorithm to solve this test optimization problem, using existing model counters as a building block. We give preliminary experimental results of our method, and discuss possible improvements.

1 Introduction

In natural sciences, it often occurs that one has several different hypotheses (models) for a system or parts of its state. *Testing* asks whether one can reduce their number by stimulating the system with appropriate inputs, called test patterns, in order to validate or falsify hypotheses from observing the generated outputs. Applications include, for example, model-based fault analysis (checking technical systems for the absence or presence of faults [8, 15]), autonomous systems (acquiring sensory inputs to discriminate among competing state estimates [4]), and bioinformatics (designing experiments that help to distinguish between different possible explanations of biological phenomena [16]).

For deterministic systems where each input generates a unique output, such as digital circuits, it has been shown how generating test inputs can be formulated and solved as a satisfiability problem [6, 10]. The *non-deterministic* case,

* Supported by the DFG Research Center MATHEON *Mathematics for key technologies* in Berlin.



(a) overlapping (b) non-overlapping

Figure 1. Given two non-deterministic models, a test input can either lead to overlapping (a) or non-overlapping (b) output sets A and B.

however, where the output is not uniquely determined by the inputs, is more frequent in practice. One reason is that in order to reduce the size of a model, for example, to fit it into an embedded controller [13, 18], it is common to aggregate the domains of system variables into small sets of values such as 'low', 'med', and 'high'; a side-effect of this abstraction is that the resulting models can no longer be assumed to be deterministic functions, even if the underlying system behavior was deterministic [17]. Another reason is the test situation itself: even in a rigid environment such as an automotive test-bed, there are inevitably variables or parameters that cannot be completely controlled while testing the device.

The difficulty of test generation with non-deterministic models is that each input pattern can generate a set of possible outcomes instead of a single outcome. For two hypotheses and a fixed test input, let A and B be the sets of possible outputs. These sets can either overlap or be disjoint as illustrated in Figure 1. Assuming that at least one hypothesis captures the actual behavior of the system, there are two possible cases: (i) the actual observed output of the system could either fall into the intersection of A and B or (ii) outside the intersection. In the first case no information is gained, as none of the hypotheses can be refuted. In the latter case, however, one of the hypotheses can be refuted. Thus, if the sets overlap as depicted in Figure 1(a), the test input *might* distinguish between the two hypotheses, whereas if the sets are disjunct as shown in Figure 1(b), the test input *will certainly* distinguish among them. Note that, if the assumption that at least one hypothesis captures the actual behavior of the system fails, there is a third possible outcome, where the observed output lies outside of both sets. In this case, both hypotheses can be refuted since they do not describe the actual behavior of the system.

This *qualitative* distinction of tests for non-deterministic models was noted in several research areas. In the field of model-based diagnosis with first-order logical models, Struss [15] introduced so-called *possibly* and *definitely discriminating tests*, for short PDT and DDT, respectively. The first type of test (PDT) might distinguish between fault hypotheses and corresponds to Figure 1(a), whereas the second type (DDT) will necessarily do so, which corresponds to Figure 1(b). Struss [15] further provided a characterization of PDTs and DDTs in terms of relational (logical) models, together with an ad-hoc algorithm to compute them. In the field of automata theory, Alur et al. [3] have studied the analogous problem of generating so-called *weak* and *strong distinguishing sequences*. These are input sequences for a non-deterministic finite state machine, such that based on the generated outputs, one can determine the internal state either for some or all feasible runs of the machine. Finding weak and strong sequences with a length less than or equal to a bound $k \in \mathbb{N}$ can be reduced to the problem of finding PDTs and DDTs, by unrolling automata into a constraint network using k copies of the transition relation and the observation relation [8].

In previous work [12], we have shown how PDTs and DDTs can be formalized and computed using *quantified constraint satisfaction problems*, a game-theoretic extension of constraint satisfaction problems. In the next section, we summarize this constraint-based framework for testing. In section 3, we then propose a novel, *quantitative* distinction of tests that refines and generalizes the previous notions of weak versus strong and possibly versus definitely discriminating tests. The key idea is to measure the quality of a test by determining the actual ratio of distinguishing and not distinguishing outcomes, corresponding to the ratio of non-intersecting and intersecting areas in Figure 1. Because test inputs that maximize this measure distinguish among given hypotheses as good as possible, we call them *optimal distinguishing tests* (ODTs). We show how in a constraint-based framework, ODTs can be defined and computed using model counting techniques. In Section 4, we propose a greedy algorithm that can quickly find distinguishing tests, using existing model counters as a building block (in our experiments, we used a model counting extension of a constraint integer programming solver SCIP [1, 2]). We give preliminary experimental results of our method using a small real-world problem from automotive industry. Finally, in the last section we discuss possible improvements and directions for future work.

2 Distinguishing Tests

We briefly introduce the theory of constraint-based testing similar to [12, 15]. We first define the notion of a constraint satisfaction problem (CSP).

Definition 1 (Constraint Satisfaction Problem). *A* constraint satisfaction problem M *is a triple* $M = (\mathcal{V}, \mathcal{D}, \mathcal{C})$*, where* $\mathcal{D} = D(v_1) \times \ldots \times D(v_n)$ *are the finite domains of finitely many variables* $v_j \in \mathcal{V}$*,* $j = 1, \ldots, n$*, and* $\mathcal{C} = \{C_1, \ldots, C_m\}$ *is a finite set of constraints with* $C_i \subseteq \mathcal{D}$*,* $i = 1, \ldots, m$*. The task is to find an* assignment $\boldsymbol{x} \in \mathcal{D}$ *to the variables such that all constraints are satisfied, that is,* $\boldsymbol{x} \in C_i$ *for* $i = 1, \ldots, m$.

We denote by X the set of all solutions of a given constraint satisfaction problem. That is,

$$X = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{D},\ \mathcal{C}(\boldsymbol{x})\},\ \text{with } \mathcal{C}(\boldsymbol{x}) :\Leftrightarrow \boldsymbol{x} \in C_i\ \forall i = 1, \ldots, m.$$

Testing attempts to discriminate between hypotheses about a system – for example, about different kinds of faults – by stimulating it in such a way that the hypotheses become observationally distinguishable. Thereby, the system under investigation defines a set of *controllable* (input) variables $\mathcal{I}$ and a set of *observable* (output) variables $\mathcal{O}$. Formally, a hypothesis M for a system is a CSP where the variable set $\mathcal{V}$ contains the input and output variables of the system.

Definition 2 (Hypothesis). *A* hypothesis *for a system is a CSP whose variables are partitioned into $\mathcal{V} = \mathcal{I} \cup \mathcal{O} \cup \mathcal{S}$, such that $\mathcal{I}$ and $\mathcal{O}$ are the input and output variables of the system, and for all assignments to $\mathcal{I}$, the CSP is solvable. The remaining variables $\mathcal{S} = \mathcal{V} \setminus (\mathcal{I} \cup \mathcal{O})$ are called internal state variables.*

Note that the internal state variable sets $\mathcal{S}$ can differ for two different hypotheses. We denote by $\mathcal{D}(\mathcal{I})$ and $\mathcal{D}(\mathcal{O})$ the cross product of the domains of the input and output variables, respectively:

$$\mathcal{D}(\mathcal{I}) = \bigtimes_{v \in \mathcal{I}} D(v) \quad \text{and} \quad \mathcal{D}(\mathcal{O}) = \bigtimes_{v \in \mathcal{O}} D(v).$$

The goal of testing is then to find assignments of the input variables $\mathcal{I}$ that will cause different assignments of output variables $\mathcal{O}$ for different hypotheses. For a given hypothesis M and an assignment $\boldsymbol{t} \in \mathcal{D}(\mathcal{I})$ to the input variables we define the *output function* $\mathcal{X}$ as follows:

$$\mathcal{X} : \mathcal{D}(\mathcal{I}) \to 2^{\mathcal{D}(\mathcal{O})} \text{ with } \boldsymbol{t} \mapsto \{\boldsymbol{y} \mid \boldsymbol{y} \in \mathcal{D}(\mathcal{O}), \exists\, \boldsymbol{x} \in X : \boldsymbol{x}[\mathcal{I}] = \boldsymbol{t} \wedge \boldsymbol{x}[\mathcal{O}] = \boldsymbol{y}\},$$

where $2^{\mathcal{D}(\mathcal{O})}$ denotes the power set of $\mathcal{D}(\mathcal{O})$, and $\boldsymbol{x}[\mathcal{I}]$, $\boldsymbol{x}[\mathcal{O}]$ denote the restriction of the assignment vector $\boldsymbol{x}$ to the input variables $\mathcal{I}$ and the output variables $\mathcal{O}$, respectively. Note that since M will always yield an output, $\mathcal{X}(\boldsymbol{t})$ is non-empty.

Definition 3 (Distinguishing Tests). *Consider $k \in \mathbb{N}$ hypotheses $M_1, \ldots, M_k$ with input variables $\mathcal{I}$ and output variables $\mathcal{O}$. Let $\mathcal{X}_i$ be the output function of hypothesis M_i with $i \in \{1, \ldots, k\}$. An assignment $\boldsymbol{t} \in \mathcal{D}(\mathcal{I})$ to the input variables $\mathcal{I}$ is a* possibly distinguishing test *(PDT), if there exists an $i \in \{1, \ldots, k\}$ such that*

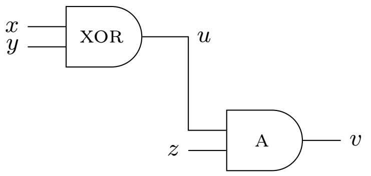
$$\mathcal{X}_i(\boldsymbol{t}) \setminus \bigcup_{j \neq i} \mathcal{X}_j(\boldsymbol{t}) \neq \varnothing.$$

An assignment $\boldsymbol{t} \in \mathcal{D}(\mathcal{I})$ is a definitely distinguishing test *(DDT), if for all $i \in \{1, \ldots, k\}$ it holds that*

$$\mathcal{X}_i(\boldsymbol{t}) \setminus \bigcup_{j \neq i} \mathcal{X}_j(\boldsymbol{t}) = \mathcal{X}_i(\boldsymbol{t}).$$

Verbally, a test input is a PDT if there exists a hypothesis for which this test input can lead to an output which is not reachable for any other hypothesis. On the other hand, an assignment to the input variables is a DDT if for all hypotheses the possible outputs are pairwise disjoint. This means, there exists no overlapping of the possible outcomes at all.

In the following, we restrict ourselves to the case where there are only two possible hypotheses, for example corresponding to normal and faulty behavior of the system.



XOR: x	y	u
L	L	L
L	H	H
H	L	H
H	H	L

A: u	z	v
L	L	L
L	H	L
L	H	H
H	L	L
H	L	H
H	H	H

A$'$: u	z	v
L	L	L
L	H	L
H	L	L
H	H	L

Figure 2. Circuit with a possibly faulty adder.

To illustrate the above definitions, consider the system in Figure 2. It consists of five variables x, y, z, u, and v, where x, y, and z are input variables and v is an output variable. Furthermore, the system has two components, one comparing signals x and y with result u and the other adding signals u and z. The signals have been abstracted into qualitative values 'low' (L) and 'high' (H). This means, each variable of the system has the same domain set $\{L, H\}$; thus, for instance, values L and H can add up to the value L or H, and so on. Assume we have two hypotheses M_1 and M_2 about the system that we want to distinguish from each other: the first hypothesis is that the system is functioning normally, which is modeled by the constraint set $\{\text{XOR}, \text{A}\}$ (see Figure 2). The second hypothesis is that the adder is *stuck-at-L*, which is modeled by the constraints $\{\text{XOR}, \text{A}'\}$. Note that only the second constraint of both hypotheses contains a non-deterministic behavior. The assignment $(x, y, z) = (L, H, L)$, for example, is a PDT, since it leads to the observation $v = L$ or $v = H$ for M_1, and $v = L$ for M_2. One the other hand, the assignment $(x, y, z) = (L, H, H)$ is a DDT for the two hypotheses, since this assignment leads to the observation $v = H$ and $v = L$ for the hypotheses M_1 and M_2, respectively.

Testing can be extended from the above case of logical, state-less models to the more general case of *automata models* that have internal states. This means that we are no longer searching for a single assignment to input variables, but rather for a sequence of inputs over different time steps. The following definitions are adapted from [5] and [7]:

Definition 4 (Plant Hypothesis). *A (partially observable) plant is a tuple $P = \langle x_0, S, I, \delta, O, \lambda \rangle$, where S, I, O are finite sets, called the* state space*,* input space*, and* output space*, respectively, $x_0 \in S$ is the* start state*, $\delta \subseteq S \times I \times S$ is the* transition relation*, and $\lambda \subseteq S \times O$ is the* observation relation*.*

Such plant models are for instance used in NASA's Livingstone [19] or MIT's Titan model-based system [18]. Note that a plant need not be deterministic, that is, the state after a transition may not be uniquely determined by the state before the transition and the input. Likewise, a plant state may be associated with several possible observations.

For technical convenience, it is assumed that the relations δ and λ are *complete*, that is for every $x \in S$ and $i \in I$ there exists at least one $x' \in S$ such that $(x, i, x') \in \delta$ and at least one $o \in O$ such that $(x, o) \in \lambda$. We write $\delta(x, i, x')$ for

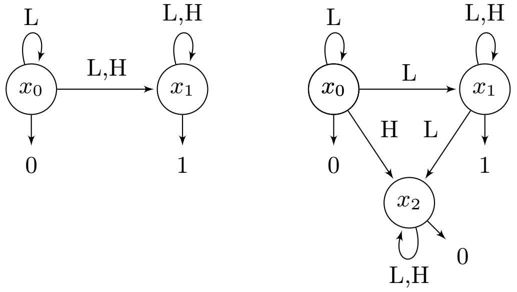


Figure 3. Two plants P_1 (left) and P_2 (right).

$(x, i, x') \in \delta$, and $\lambda(s, o)$ for $(x, o) \in \lambda$. A *feasible trace* of a plant P is a pair (σ, ρ), where $\sigma = i_1, i_2, \ldots, i_k \in I^*$ is a sequence of k inputs and $\rho = o_0, o_1, \ldots, o_k \in O^*$ is a sequence of $k + 1$ outputs, such that there exists a sequence $x_0, x_1, \ldots, x_k$ of states with $\delta(x_{j-1}, i_j, x_j)$ for all $1 \leq j \leq k$ and $\lambda(x_j, o_j)$ for all $0 \leq j \leq k$.

Definition 5 (Distinguishing Test Sequences). *Given two plants $P_1 = \langle x_0, S, I, \delta, O, \lambda\rangle$ and $P_2 = \langle y_0, Y, I, \eta, O, \mu\rangle$, a sequence of inputs $\sigma \in I^*$ is a* weak test*, if there exists a sequence of outputs $\rho \in O^*$ such that (σ, ρ) is a feasible trace of P_1 but not of P_2. The sequence σ is a* strong test *for P_1 and P_2, if and only if for all sequences of outputs ρ, it holds that if (σ, ρ) is a feasible trace P_1 then it is not a feasible trace of P_2.*

Notice that due to the assumptions about completeness, for every input sequence $\sigma \in I^*$ there exist output sequences $\rho, \tau \in O^*$ such that (σ, ρ) is a feasible trace of P_1 and (σ, τ) is a feasible trace of P_2.

Analogous to PDTs and DDTs, a weak test is a sequence that *may* reveal a difference between two hypotheses, whereas a strong test is a sequence that *will necessarily* do so. For example, Figure 3 shows two plants P_1 and P_2 with $I = \{L, H\}$ and $O = \{0, 1\}$. The input sequence $\sigma = L, L$ is a weak test for the two plants, because, for example, $0, 1, 0$ is a possible output sequence of P_2 but not of P_1. The sequence $\sigma' = H, H$ is a strong test for P_2 and P_1, because the only possible output sequence $0, 0, 0$ of P_2 cannot be produced by P_1.

From a practical point of view, it is often sufficient to consider *bounded test sequences* that do not exceed a certain length $k \in \mathbb{N}$. In this case, the problem of finding weak and strong tests for automata models can be reduced to finding PDTs and DDTs:

Remark 1. Finding weak and strong tests with a length less than or equal to a bound $k \in \mathbb{N}$ can be reduced to the problem of finding PDTs and DDTs, by unrolling automata into a constraint network using k copies of the transition relation and the observation relation [8].

In the following, we consider only tests with such a bounded length. Therefore, we assume the hypotheses are given as CSPs over finite-domain variables

(Definition 2). This covers both the case of logical models and (bounded) automata models.

3 Optimal Distinguishing Tests

In [12], we have shown how PDTs and DDTs can be formalized and computed using *quantified constraints satisfaction problems* (QCSP), a game-theoretic extension of CSPs. However, for larger hypotheses, the computational cost of solving such QCSPs can be prohibitive. Moreover, due to limited observability or a high degree of non-determinism in the system under investigation, it is not uncommon that a DDT for the hypotheses does not exist, and one can instead only find PDTs.

In the following, we therefore propose a novel, *quantitative measure* for tests that refines and generalizes the previous, qualitative notions of PDTs and DDTs. The key idea is to determine the ratio of distinguishing and not distinguishing outcomes of a test input, corresponding to the degree of overlap between the output sets shown in Figure 1. This measure provides a way to further distinguish between different PDTs. In addition, even if computing this measure is by itself not easier than finding PDTs and DDTs, approximations of it can be used as a *guiding heuristic* in the search for tests, providing a basis for greedy methods to quickly find good tests.

The main assumption underlying our approach is that for a test input and a non-deterministic hypothesis, the possible outcomes (feasible assignments to the output variables) are all (roughly) equally likely. Then, a PDT will be more likely to distinguish among two given hypotheses compared to another PDT, if the ratio of possible outcomes that are unique to a hypothesis versus the total number of possible outcomes is higher.

This intuition is captured in the following definitions.

Definition 6 (Distinguishing Ratio). *Given a test input $\boldsymbol{t} \in \mathcal{D}(\mathcal{I})$ for two hypotheses M_1, M_2 with input variables $\mathcal{I}$ and output variables $\mathcal{O}$, we define $\Gamma(\boldsymbol{t})$ to be the ratio of feasible outputs that distinguish among the hypotheses versus all feasible outputs:*

$$\Gamma(\boldsymbol{t}) := \frac{|\mathcal{X}_1(\boldsymbol{t}) \cup \mathcal{X}_2(\boldsymbol{t})| - |\mathcal{X}_1(\boldsymbol{t}) \cap \mathcal{X}_2(\boldsymbol{t})|}{|\mathcal{X}_1(\boldsymbol{t}) \cup \mathcal{X}_2(\boldsymbol{t})|} = 1 - \frac{|\mathcal{X}_1(\boldsymbol{t}) \cap \mathcal{X}_2(\boldsymbol{t})|}{|\mathcal{X}_1(\boldsymbol{t}) \cup \mathcal{X}_2(\boldsymbol{t})|}.$$

Γ is a measure for test quality that can take on values in the interval $[0, 1]$. It refines the notion of PDTs and DDTs in the following precise sense: if Γ is 0, then the test does not distinguish at all, as both hypotheses lead to the same observations (output patterns). If the value is 1, then the test is a DDT, since both hypotheses always lead to different observations. If the value is between 0 and 1, then the test is a PDT (there is some non-overlap in the possible observations). Note that Γ is well-defined since for any chosen $\boldsymbol{t} \in \mathcal{D}(\mathcal{I})$, the sets $\mathcal{X}_1(\boldsymbol{t})$ and $\mathcal{X}_2(\boldsymbol{t})$ are non-empty (see Definition 2).

Remark 2. For computing the distinguishing ratio for a fixed test input $\boldsymbol{t}$ it is only necessary to compute (model count) the value $|\mathcal{X}_1(\boldsymbol{t}) \cap \mathcal{X}_2(\boldsymbol{t})|$, $|\mathcal{X}_1(\boldsymbol{t})|$, and $|\mathcal{X}_2(\boldsymbol{t})|$, since

$$\Gamma(\boldsymbol{t}) = 1 - \frac{|\mathcal{X}_1(\boldsymbol{t}) \cap \mathcal{X}_2(\boldsymbol{t})|}{|\mathcal{X}_1(\boldsymbol{t}) \cup \mathcal{X}_2(\boldsymbol{t})|} = 1 - \frac{|\mathcal{X}_1(\boldsymbol{t}) \cap \mathcal{X}_2(\boldsymbol{t})|}{|\mathcal{X}_1(\boldsymbol{t})| + |\mathcal{X}_2(\boldsymbol{t})| - |\mathcal{X}_1(\boldsymbol{t}) \cap \mathcal{X}_2(\boldsymbol{t})|}.$$

Based on this measure, we can formalize our goal of finding tests that discriminate among two hypotheses as good as possible:

Definition 7 (Optimal Distinguishing Test). *An assignment $\boldsymbol{t} \in \mathcal{D}(\mathcal{I})$ is an* optimal distinguishing test (ODT) *for two hypotheses M_1, M_2 with input variables $\mathcal{I}$ and output variables $\mathcal{O}$ if its distinguishing ratio is maximal, that is, $\Gamma(\boldsymbol{t}) = \max_{\boldsymbol{x} \in \mathcal{D}(\mathcal{I})} \Gamma(\boldsymbol{x})$.*

Note that each DDT is also an ODT. To illustrate the previous definition, consider again the example in Figure 3. The input sequence $\boldsymbol{t} = (L, L)$ is a weak test or equivalently, a PDT if the automata are expanded into suitable constraint networks. The possible outcomes (output patterns) for P_1 and P_2 are

$$\mathcal{X}_1(\boldsymbol{t}) = \{(0,0,0), (0,0,1), (0,1,1)\}$$
$$\mathcal{X}_2(\boldsymbol{t}) = \{(0,0,0), (0,0,1), (0,1,1), (0,1,0)\}.$$

Thus, for this test there is only one possible outcome $(0,1,0)$ that is unique to a hypothesis, out of a total of four possible outcomes. Hence, $\Gamma(\boldsymbol{t}) = \frac{1}{4}$. There exists another weak test (PDT), namely the input sequence $\boldsymbol{t'} = (L, H)$, with possible outcomes

$$\mathcal{X}_1(\boldsymbol{t'}) = \{(0,0,1), (0,1,1)\}$$
$$\mathcal{X}_2(\boldsymbol{t'}) = \{(0,0,0), (0,1,1)\}.$$

This test has two possible outcomes $\{(0,0,0), (0,0,1)\}$ that are unique to a hypothesis, out of three possible outcomes $\{(0,0,0), (0,0,1), (0,1,1)\}$. This leads to $\Gamma(\boldsymbol{t'}) = \frac{2}{3}$. Note that for this example, there exists a test $\boldsymbol{t''} = (H, H)$ with $\Gamma(\boldsymbol{t''}) = 1$, which is a DDT and therefore an ODT.

Now we present a general lower bound on the optimal distinguishing ratio.

Theorem 1. *Consider a system with input variable set $\mathcal{I}$ and output variable set $\mathcal{O}$. Furthermore, let M_1 and M_2 be two hypotheses for this system. Let*

$$X_i[\mathcal{I}, \mathcal{O}] = \{(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x} \in \mathcal{D}(\mathcal{I}),\, \boldsymbol{y} \in \mathcal{D}(\mathcal{O}),\, \exists\, \boldsymbol{t} \in X_i : \boldsymbol{t}[\mathcal{I}] = \boldsymbol{x} \wedge \boldsymbol{t}[\mathcal{O}] = \boldsymbol{y}\},$$

where X_i is the set of all feasible solutions of the hypothesis M_i, $i \in \{1, 2\}$. Then,

$$1 - \frac{|X_1[\mathcal{I}, \mathcal{O}] \cap X_2[\mathcal{I}, \mathcal{O}]|}{|X_1[\mathcal{I}, \mathcal{O}] \cup X_2[\mathcal{I}, \mathcal{O}]|}$$

is a lower bound on the optimal distinguishing ratio.

Proof. Let $\mathcal{I}$ and $\mathcal{O}$ be the input and output variable sets of an arbitrary system and M_1 and M_2 two hypotheses. Furthermore, let $X_1[\mathcal{I},\mathcal{O}]$ and $X_2[\mathcal{I},\mathcal{O}]$ be the sets to the hypotheses as defined in the theorem.

Given an input variable $v \in \mathcal{I}$, we denote by T_d the subset of $\mathcal{D}(\mathcal{I})$ which is restricted to the elements where the input variable v is fixed to $d \in D(v)$. That is, $T_d = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{D}(\mathcal{I}) \wedge \boldsymbol{x}[\{v\}] = d\}$. These subsets form a partition of $\mathcal{D}(\mathcal{I})$. This means, $\mathcal{D}(\mathcal{I}) = \bigcup_{d \in D(v)} T_d$ and $T_d \cap T_k = \varnothing$ for all $d, k \in D(v)$ with $d \neq k$. Hence, these subsets can be used to partition $X_i[\mathcal{I},\mathcal{O}]$ as follows:

$$X_i[\mathcal{I},\mathcal{O}] = \bigcup_{d \in D(v)} \{(\boldsymbol{x},\boldsymbol{y}) \mid \boldsymbol{x} \in T_d,\, \boldsymbol{y} \in \mathcal{D}(\mathcal{O}),\, \exists\, \boldsymbol{t} \in X_i : \boldsymbol{t}[\mathcal{I}] = \boldsymbol{x} \wedge \boldsymbol{t}[\mathcal{O}] = \boldsymbol{y}\}.$$

Therefore,

$$|X_i[\mathcal{I},\mathcal{O}]| = \sum_{d \in D(v)} |\{(\boldsymbol{x},\boldsymbol{y}) \mid \boldsymbol{x} \in T_d,\, \boldsymbol{y} \in \mathcal{D}(\mathcal{O}),\, \exists\, \boldsymbol{t} \in X_i : \boldsymbol{t}[\mathcal{I}] = \boldsymbol{x} \wedge \boldsymbol{t}[\mathcal{O}] = \boldsymbol{y}\}|.$$

We claim that

$$\frac{|X_1[\mathcal{I},\mathcal{O}] \cap X_2[\mathcal{I},\mathcal{O}]|}{|X_1[\mathcal{I},\mathcal{O}] \cup X_2[\mathcal{I},\mathcal{O}]|} \geq \min_{d \in D(v)} \frac{|X_1[\mathcal{I},\mathcal{O}] \cap X_2[\mathcal{I},\mathcal{O}] \cap (T_d \times \mathcal{D}(\mathcal{O}))|}{|(X_1[\mathcal{I},\mathcal{O}] \cup X_2[\mathcal{I},\mathcal{O}]) \cap (T_d \times \mathcal{D}(\mathcal{O}))|}.$$

To this end, let $d^* \in D(v)$ be a domain value of v, which attains the minimum on the right hand side. In a first step, we decompose the left hand side using that the subsets T_d are a partition of $\mathcal{D}(\mathcal{I})$:

$$\frac{|X_1[\mathcal{I},\mathcal{O}] \cap X_2[\mathcal{I},\mathcal{O}]|}{|X_1[\mathcal{I},\mathcal{O}] \cup X_2[\mathcal{I},\mathcal{O}]|} = \frac{\sum_{d \in D(v)} |X_1[\mathcal{I},\mathcal{O}] \cap X_2[\mathcal{I},\mathcal{O}] \cap (T_d \times \mathcal{D}(\mathcal{O}))|}{\sum_{d \in D(v)} |(X_1[\mathcal{I},\mathcal{O}] \cup X_2[\mathcal{I},\mathcal{O}]) \cap (T_d \times \mathcal{D}(\mathcal{O}))|}$$

Now we substitute

$$\begin{aligned} a_d &:= |X_1[\mathcal{I},\mathcal{O}] \cap X_2[\mathcal{I},\mathcal{O}] \cap (T_d \times \mathcal{D}(\mathcal{O}))| \\ \tilde{a}_d &:= |(X_1[\mathcal{I},\mathcal{O}] \cup X_2[\mathcal{I},\mathcal{O}]) \cap (T_d \times \mathcal{D}(\mathcal{O}))|. \end{aligned}$$

To prove the claim, it is left to show that

$$\frac{\sum_{d \in D(v)} a_d}{\sum_{d \in D(v)} \tilde{a}_d} \geq \frac{a_{d^*}}{\tilde{a}_{d^*}} \quad \text{with} \quad \frac{a_d}{\tilde{a}_d} \geq \frac{a_{d^*}}{\tilde{a}_{d^*}} \quad \forall\, d \in D(v).$$

This follows since

$$\frac{\sum_{d \in D(v)} a_d}{\sum_{d \in D(v)} \tilde{a}_d} \geq \frac{a_{d^*}}{\tilde{a}_{d^*}} \Leftrightarrow \sum_{d \in D(v)} a_d \geq \sum_{d \in D(v)} \tilde{a}_d \cdot \frac{a_{d^*}}{\tilde{a}_{d^*}}$$

and

$$\sum_{d \in D(v)} a_d = \sum_{d \in D(v)} \frac{a_d \cdot \tilde{a}_d}{\tilde{a}_d} \geq \sum_{d \in D(v)} \frac{a_{d^*} \cdot \tilde{a}_d}{\tilde{a}_{d^*}} = \sum_{d \in D(v)} \tilde{a}_d \cdot \frac{a_{d^*}}{\tilde{a}_{d^*}}.$$

Therefore, we have proven, that it is possible to fix any input variable such that the claimed lower bound holds. Doing this sequentially for all input variables leads to an assignment which has as distinguishing ratio which is at least as good as the claimed lower bound. □

```
Input: Hypotheses $M_1$ and $M_2$ with set of input and output variables $\mathcal{I}$ and $\mathcal{O}$
Output: Test $\boldsymbol{t} \in \mathcal{D}(\mathcal{I})$
$T \leftarrow \mathcal{D}(\mathcal{I})$;
foreach $v \in \mathcal{I}$ do
    bestratio $\leftarrow -1$;
    bestfixing $\leftarrow \infty$;
    foreach $d \in D(v)$ do
        $T' \leftarrow \{\boldsymbol{x} \mid \boldsymbol{x} \in T \wedge \boldsymbol{x}[\{v\}] = d\}$;
        ratio $\leftarrow \Gamma(T')$;
        if ratio > bestratio then
            bestratio $\leftarrow$ ratio;
            bestfixing $\leftarrow d$;
        end
    end
    $T \leftarrow T \cap \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{D}(\mathcal{I}) \wedge \boldsymbol{x}[\{v\}] = \text{bestfixing}\}$;
end
return $\boldsymbol{t} \in T$
```

Algorithm 1: Greedy algorithm for distinguishing test input generation.

4 Greedy Algorithm for Distinguishing Test Generation

In the previous section we stated the optimization problem of computing an optimal distinguishing test (ODT). In this section, we propose and analyze a *greedy-type* algorithm to solve this problem, which can use existing model counting methods (exact or approximate) as a building block.

The idea of the greedy algorithm is to select at each step an input variable which is not fixed yet. For each possible value of this variable, the algorithm computes a local form of the distinguishing ratio (comparison of model counts, as defined in Section 3) for assigning this value. The variable is then fixed to a value that attains the maximal (local) distinguishing ratio.

To formalize this idea, we canonically extend the function Γ from single assignments $\boldsymbol{t} \in \mathcal{D}(\mathcal{I})$ to sets of assignments $T \subseteq \mathcal{D}(\mathcal{I})$ by defining

$$\mathcal{X}(T) = \bigcup_{\boldsymbol{t} \in T} \mathcal{X}(\boldsymbol{t}).$$

Algorithm 1 shows the algorithm Greedy. It takes as input the controllable and observable variable sets $\mathcal{I}$ and $\mathcal{O}$ defined by the system under investigation and two hypotheses M_1 and M_2 to distinguish. As output it returns an assignment for the input variables.

For example, consider the system shown in Figure 4. It has two input variables $\mathcal{I} = \{v_1, v_2\}$ and one output variable $\mathcal{O} = \{v_3\}$. Let $D(v_1) = D(v_2) = \{0, 1\}$ and $D(v_3) = \{0, 1, 2\}$. Consider two hypotheses M_1 and M_2 for this system,

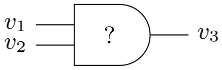


Figure 4. Example system schema.

where both hypotheses have no internal state variables. Each hypothesis has one constraint

$$\begin{aligned}&C_1 = D(v_1) \times D(v_2) \times D(v_3)\\&C_2 = D(v_1) \times D(v_2) \times D(v_3) \setminus \{(0,0,1),(0,0,2),(0,1,2),(1,0,2)\},\end{aligned}$$

where C_1 and C_2 belong to hypothesis M_1 and M_2, respectively. Assume the algorithm selects the variables in the order v_1, v_2. Then for the two values of v_1, it computes the two ratios

$$\begin{aligned}v_1 = 0 \rightarrow \mathit{\Gamma}(T') &= 1 - \frac{|\{0,1,2\} \cap \{0,1\}|}{|\{0,1,2\} \cup \{0,1\}|} = \tfrac{1}{3}\\ v_1 = 1 \rightarrow \mathit{\Gamma}(T') &= 1 - \frac{|\{0,1,2\} \cap \{0,1,2\}|}{|\{0,1,2\} \cup \{0,1,2\}|} = 0.\end{aligned}$$

It chooses value 0 for v_1, since it has the highest ratio. Continuing with v_2, its ratios are determined as

$$\begin{aligned}v_2 = 0 \rightarrow \mathit{\Gamma}(T') &= 1 - \frac{|\{0,1,2\} \cap \{0\}|}{|\{0,1,2\} \cup \{0\}|} = \tfrac{2}{3}\\ v_2 = 1 \rightarrow \mathit{\Gamma}(T') &= 1 - \frac{|\{0,1,2\} \cap \{0,1\}|}{|\{0,1,2\} \cup \{0,1\}|} = \tfrac{1}{3}\end{aligned}$$

and thus value 0 for v_2 is chosen. The computed input $(0,0)$ is an ODT for this example.

4.1 Properties of the Algorithm

Note that if the system consists only of one input variable, Greedy computes an ODT, since the algorithm just enumerates all possible variable assignments for the input variable and selects the assignment that maximizes the distinguishing ratio. In general, however, the Greedy algorithm has no constant approximation factor.

Theorem 2. *The* Greedy *algorithm has no constant approximation factor. That is, there exists no constant c such that for all instances*

$$\max_{\boldsymbol{t} \in D(\mathcal{I})} \mathit{\Gamma}(\boldsymbol{t}) \leq c \cdot \mathit{\Gamma}(\boldsymbol{x}^*),$$

where $\boldsymbol{x}^*$ is the solution computed by Greedy.

Proof. Consider again the system stated in Figure 4, and let the domain of the input variables be $D(v_1) = D(v_2) = \{0, 1\}$ and of the output variable v_3 be $D(v_3) = \{0, \dots, n\}$, $n \in \mathbb{N}$, and $n > 2$. W.l.o.g. let the domains be ordered as:

$$\mathcal{D} = D(v_1) \times D(v_2) \times D(v_3).$$

Let M_1 and M_2 be defined by the following sets of feasible solutions:

$$\begin{aligned} X_1 &= \{(0,0,0), (1,0,0)\} \cup \{(x,1,z) \mid x \in D(v_1) \wedge z \in \{2, \dots, n\}\} \subset \mathcal{D} \\ X_2 &= \{(0,1,0), (1,0,0), (1,1,1)\} \cup \{(x,x,z) \mid x \in \{0,1\} \wedge z \in \{2, \dots, n\}\} \subset \mathcal{D}, \end{aligned}$$

where X_1 and X_2 belong to hypothesis M_1 and M_2, respectively. Both hypotheses have no internal state variables. It is assumed that the GREEDY algorithm selects v_1 first. The best possible assignment for this variable is $v_1 = 1$, since

$$\begin{aligned} v_1 = 0 \rightarrow \mathit{\Gamma}(T') &= 1 - \frac{|\{0,2,\dots,n\} \cap \{0,2,\dots,n\}|}{|\{0,2,\dots,n\} \cup \{0,2,\dots,n\}|} = 0 \\ v_1 = 1 \rightarrow \mathit{\Gamma}(T') &= 1 - \frac{|\{0,2,\dots,n\} \cap \{0,\dots,n\}|}{|\{0,2,\dots,n\} \cup \{0,\dots,n\}|} > 0. \end{aligned}$$

In the final step GREEDY has to fix variable v_2 with respect to the previous fixing of $v_1 = 1$. The best possible decision is, to fix v_2 also to 1, since fixing v_2 to 0 leads to an distinguishing ratio of zero and for $v_2 = 1$ we have:

$$\mathit{\Gamma}((1,1)) = 1 - \frac{|\{2,\dots,n\} \cap \{1,\dots,n\}|}{|\{2,\dots,n\} \cup \{1,\dots,n\}|} = \tfrac{1}{n}.$$

Note that the computed test input $(v_1, v_2) = (1, 1)$ is independent of the chosen n.

An ODT for this problem, however, is $(v_1, v_2) = (0, 0)$, which is also a DDT. This test input has, therefore, a distinguishing ratio of 1. For n tending to infinity, the distinguishing ratio of the test input computed by GREEDY tends to zero. This proves that the GREEDY algorithm has no constant approximation factor. □

Note that if GREEDY would choose variable v_2 first, it would compute an ODT for this example. This rises the question, whether there exist always a permutation of the input variables such that the GREEDY algorithm computes an ODT. The following theorem answers this question.

Theorem 3. *In general, the* GREEDY *algorithm does not compute an ODT even if it is allowed to try all possible input variable permutations.*

Proof. Again, consider the abstract system depicted in Figure 4 with the input variable set $\mathcal{I} = \{v_1, v_2\}$, $D(v_1) = D(v_2) = \{0, 1\}$, the output variable set $\mathcal{O} = \{v_3\}$, and $D(v_3) = \{1, 2, 3, 4\}$. Let M_1 and M_2 be two hypotheses given through the constraints:

$$\begin{aligned} C_1 &= \{(0,0,1), (1,0,2), (0,1,2), (1,1,2), (1,1,3), (1,1,4)\} \subset \mathcal{D} \\ C_2 &= \{(x,y,2) \mid x, y \in \{0,1\}\} \subset \mathcal{D}, \end{aligned}$$

where C_1 belongs to hypothesis M_1, C_2 to hypothesis M_2, and $\mathcal{D} = D(v_1) \times D(v_2) \times D(v_3)$.

The test input $(v_1, v_2) = (0, 0)$ is the unique DDT and, therefore, the unique ODT. If we show that the Greedy algorithm fixes in the first iteration, independently of the chosen input variable, this variable to 1, then we have proven the theorem.

Independently from the chosen input variable, Greedy fixes this variable to 1 since for $i \in \{1, 2\}$ it follows:

$$v_i = 0 \to \mathit{\Gamma}(T') = 1 - \frac{|\{1,2\} \cap \{2\}|}{|\{1,2\} \cup \{2\}|} = \tfrac{1}{2}$$
$$v_i = 1 \to \mathit{\Gamma}(T') = 1 - \frac{|\{2,3,4\} \cap \{2\}|}{|\{2,3,4\} \cup \{2\}|} = \tfrac{2}{3}.$$

□

In the example at the beginning of Section 4, the sequence of distinguishing ratios $\mathit{\Gamma}(T')$ computed by Greedy increases monotonically. This observation can also be made later in the computational results for the automotive example (see Table 2). However, this needs not be the case in general.

Theorem 4. *In general, the sequence of distinguishing ratios computed by* Greedy *is not monotonically increasing.*

Proof. Consider the system stated in Figure 4, and let the domain of the input variables be $D(v_1) = \{0\}$, $D(v_2) = \{0, 1\}$, and of the output variable v_3 be $D(v_3) = \{0, 1, 2\}$. W.l.o.g. let the domains be ordered as:

$$\mathcal{D} = D(v_1) \times D(v_2) \times D(v_3).$$

Let M_1 and M_2 be defined by the following sets of feasible solutions:

$$X_1 = \{(0,0,0), (0,1,0), (0,1,1)\} \subset \mathcal{D}$$
$$X_2 = \{(0,0,0), (0,1,0), (0,0,2)\} \subset \mathcal{D}$$

where X_1 and X_2 belong to hypothesis M_1 and M_2, respectively. Both hypotheses have no internal state variables. It is assumed that the Greedy algorithm selects v_1 first. Since v_1 has only one possible value in its domain, Greedy fixes v_1 to this value 0. The (local) distinguishing ratio yields:

$$\mathit{\Gamma}(T) = 1 - \frac{|\{0,1\} \cap \{0,2\}|}{|\{0,1\} \cup \{0,2\}|} = \tfrac{2}{3} \quad \text{with} \quad T = \mathcal{D}$$

In the final step Greedy has to fix variable v_2 with respect to the previous fixing of $v_1 = 0$:

$$\mathit{\Gamma}(T') = 1 - \frac{|\{0\} \cap \{0,2\}|}{|\{0\} \cup \{0,2\}|} = \tfrac{1}{2} \quad \text{with} \quad T' = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{D} \wedge \boldsymbol{x}[\{v_2\}] = 0\}$$
$$\mathit{\Gamma}(T'') = 1 - \frac{|\{0,1\} \cap \{0\}|}{|\{0,1\} \cup \{0\}|} = \tfrac{1}{2} \quad \text{with} \quad T'' = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{D} \wedge \boldsymbol{x}[\{v_2\}] = 1\}.$$

Independently of the chosen fixing for the variable v_2, the (local) distinguishing ratio decreases. □

Table 1. Model counts for the four hypotheses in the automotive example.

	CORRECT	NO-ENGINE	NO-PIPE	NO-THROTTLE
$\lvert X \rvert$	329	6552	25356	8560
$\lvert X[\mathcal{I}, \mathcal{O}] \rvert$	43	552	168	127
$\lvert \mathcal{X}(\mathcal{D}(\mathcal{I})) \rvert$	13	72	41	22

Table 2. Distinguishing ratios computed by Greedy for the automotive example.

		sequence of distinguishing ratio $\Gamma(T')$		
permutation	lower bound	0 iteration	1 iteration	2 iterations
CORRECT vs. NO-ENGINE				
(v_1, v_2)	$1 - \frac{43}{552} = 0.922$	$1 - \frac{13}{72} = 0.819$	$1 - \frac{3}{24} = 0.875$	$1 - \frac{1}{24} = 0.958$
(v_2, v_1)	$1 - \frac{43}{552} = 0.922$	$1 - \frac{13}{72} = 0.819$	$1 - \frac{4}{72} = 0.944$	$1 - \frac{1}{24} = 0.958$
CORRECT vs. NO-PIPE				
(v_1, v_2)	$1 - \frac{43}{168} = 0.744$	$1 - \frac{13}{41} = 0.683$	$1 - \frac{3}{27} = 0.889$	$1 - \frac{1}{15} = 0.933$
(v_2, v_1)	$1 - \frac{43}{168} = 0.744$	$1 - \frac{13}{41} = 0.683$	$1 - \frac{4}{23} = 0.826$	$1 - \frac{1}{15} = 0.933$
CORRECT vs. NO-THROTTLE				
(v_1, v_2)	$1 - \frac{43}{127} = 0.661$	$1 - \frac{13}{22} = 0.409$	$1 - \frac{3}{22} = 0.864$	$1 - \frac{1}{9} = 0.889$
(v_2, v_1)	$1 - \frac{43}{127} = 0.661$	$1 - \frac{13}{22} = 0.409$	$1 - \frac{5}{10} = 0.5$	$1 - \frac{5}{10} = 0.5$

4.2 Computational Results

We have implemented Algorithm 1 using the constraint integer programming solver SCIP [1, 2] as (exact) model counter.

We ran our prototype implementation on a small real-world automotive example. The example is based on a mixed discrete-continuous model of an engine air intake test-bed [11]. It has been turned into a coarse CSP model by abstracting continuous system variables into suitable finite domains with up to 12 values, corresponding to different operating regions. The system consists of the three major components ENGINE, PIPE, and THROTTLE; for each component, a fault model is defined that simply omits the respective constraint from the model. Thus, there are four diagnostic hypotheses (CORRECT, NO-ENGINE, NO-PIPE, and NO-THROTTLE), corresponding to all components functioning normally and one of them failing. The goal is to find an assignment to two controllable variables (throttle angle v_1, valve timing v_2), such that one can discriminate among hypotheses based on two observable variables (engine speed and air flow) in the system.

Table 1 shows the model counts (total number of solutions $|X|$ and the total number of projected solutions $|X[\mathcal{I}, \mathcal{O}]|$ and $|\mathcal{X}(\mathcal{D}(\mathcal{I}))|$) for the four hypotheses. Table 2 shows the computational results of the Greedy algorithm for finding tests to distinguish the normal system behavior from the faults. The first column states the used permutation, the second column gives the general lower bound on the optimal distinguishing ratio, as stated in Theorem 1, and the last three columns the sequence of the distinguishing ratios as Greedy iterates through the input variables. In all cases except the last (finding a test input to identify a NO-PIPE fault given the variable order v_2, v_1), the test input generated by the algorithm is an ODT. The last test shows that in general, the Greedy algorithm does not compute a test input whose distinguishing ratio is at least as good as the general lower bound. The run-time of the algorithm on this example is in the order of a few seconds.

5 Conclusion and Future Work

We presented a method for generating tests to distinguish system hypotheses modeled as constraints over variables. It is based on maximizing the number of non-overlapping versus overlapping observable outcomes and extends previous notions of testing for non-deterministic systems. We showed how this proposed test quality measure can be computed as a ratio of model counts. Therefore, we argue that test generation is a promising application area where model counting techniques can be fruitfully applied.

Challenges arise from the computational complexity of generating optimal distinguishing tests, since computing the optimal distinguishing ratios can be very expensive. We proposed an algorithm that greedily assigns input variables and thus requires only a limited number of model counts, but sometimes misses the optimal solution. An alternative approach that we would like to investigate in the future is to use a complete (branch-and-bound like) algorithm, but to combine it with *approximate* counting methods that compute confidence intervals for solution counts [9].

In practice, testing problems often have additional structure: for instance, like in the automotive example in Section 4.2, pairs of hypotheses often share significant identical portions. There exist decomposition techniques in test generation that can exploit such structures [5]. Therefore, an interesting question is whether these can be adapted to model counting approaches.

Another extension concerns relaxing the simplifying assumption that the possible outcomes of a non-deterministic hypothesis all have similar likelihood. In this context, methods for *weighted* model counting [14] could be used to capture, for instance, probability distributions in the hypotheses.

References

1. T. Achterberg, *Constraint Integer Programming*, PhD thesis, TU Berlin, 2007.
2. T. Achterberg, S. Heinz, and T. Koch, *Counting solutions of integer programs using unrestricted subtree detection*, in Proc. of CPAIOR-08, vol. 5015 of LNCS, 2008, pp. 278–282.
3. R. Alur, C. Courcoubetis, and M. Yannakakis, *Distinguishing tests for nondeterministic and probabilistic machines*, in Proc. of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 1995, pp. 363–372.

4. L. Blackmore and B. C. Williams, *Finite horizon control design for optimal discrimination between several models*, in Proc. IEEE Conference on Decision and Control, 2006, pp. 1147–1152.
5. S. Boroday, A. Petrenko, and R. Groz, *Can a model checker generate tests for non-deterministic systems?*, Electr. Notes Theor. Comput. Sci., 190 (2007), pp. 3–19.
6. S. Brand, *Sequential automatic test pattern generation by constraint programming*, in Proc. CP-01 Workshop on Modelling and Problem Formulation, 2001.
7. A. Cimatti, C. Pecheur, and R. Cavada, *Formal verification of diagnosability via symbolic model checking*, in Proc. of the Eighteenth International Joint Conference on Artificial Intelligence, 2003, pp. 363–369.
8. M. Esser and P. Struss, *Fault-model-based test generation for embedded software*, in Proc. of the 20th International Joint Conference on Artificial Intelligence, 2007, pp. 342–347.
9. C. Gomes, A. Sabharwal, and B. Selman, *Model counting: A new strategy for obtaining good bounds*, in Proc. of AAAI-06, 2006, pp. 54–61.
10. T. Larrabee, *Test pattern generation using boolean satisfiability*, IEEE Trans. on CAD of Integrated Circuits and Systems, 11 (1992), pp. 4–15.
11. J. Luo, K. R. Pattipati, L. Qiao, and S. Chigusa, *An integrated diagnostic development process for automotive engine control systems*, IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews, 37 (2007), pp. 1163–1173.
12. M. Sachenbacher and S. Schwoon, *Model-based testing using quantified CSPs: A map*, in ECAI 2008 Workshop on Model-based Systems, 2008, pp. 37–41.
13. M. Sachenbacher and P. Struss, *Task-dependent qualitative domain abstraction*, Artif. Intell., 162 (2005), pp. 121–143.
14. T. Sang, P. Beame, and H. Kautz, *Solving bayesian networks by weighted model counting*, in Proc. of AAAI-05, 2005.
15. P. Struss, *Testing physical systems*, in Proc. of AAAI-94, 1994, pp. 251–256.
16. I. Vatcheva, H. de Jong, O. Bernard, and N. J. Mars, *Experiment selection for the discrimination of semi-quantitative models of dynamical systems*, Artif. Intell., 170 (2006), pp. 472–506.
17. D. S. Weld and J. de Kleer, eds., *Readings in qualitative reasoning about physical systems*, Morgan Kaufmann Publishers, 1990.
18. B. C. Williams, M. D. Ingham, S. H. Chung, and P. H. Elliott, *Model-based programming of intelligent embedded systems and robotic space explorers*, Proc. of the IEEE, 91 (2003), pp. 212–237.
19. B. C. Williams and P. P. Nayak, *A model-based approach to reactive self-configuring systems*, in Proc. of AAAI-96, 1996, pp. 971–978.