Test Strategy Generation using Quantified CSPs

Martin Sachenbacher and Paul Maier

Technische Universität München, Institut für Informatik
Boltzmannstraße 3, 85748 Garching, Germany
{sachenba,maierpa}@in.tum.de
http://www9.in.tum.de

Abstract. Testing is the process of stimulating a system with inputs in order to reveal hidden parts of the system state. We consider a variant of the testing problem that was put forward in the model-based diagnosis literature, and consists of finding input patterns that definitely discriminate between different constraint-based system models. We show that this problem can be framed as a game played between two opponents, and naturally lends itself towards a formulation in terms of quantified CSPs. This QCSP-based formulation is a starting point to extend testing to a new classes of practically relevant applications – namely, systems with limited controllability – where tests consist of stimulation *strategies* instead of simple input patterns.

Key words: Test generation, adversarial planning, quantified CSPs

1 Introduction

As the complexity of technical devices grows, methods and tools that can automatically check such systems for the absence or presence of faults become more and more important. *Diagnosability* asks whether a certain fault can ever go undetected in a system due to limited observability. It has been shown how this question can be framed and solved as a SAT-based verification problem [2, 3]. *Testing* instead asks whether a certain fault will ever lead to observable differences if the system under scrutiny is actively stimulated with inputs (test patterns). In this paper, we study a variant of the testing problem introduced in [1]: finding so-called *definitely discriminating tests* (DDTs) asks whether it is possible to generate inputs that can unambiguously reveal a certain fault in a system. Because DDTs allow one to definitely verify or exclude the presence of a certain fault in a system, generating DDTs is a problem of great practical importance. For instance, in [5, 4], the framework was applied to real-world scenarios from the domain of railway control and automotive systems. [1] also provided a characterization of this problem in terms of relational (constraint-based) models.

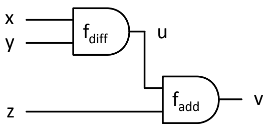
In this paper, we build a bridge from this earlier, application-oriented work to newer developments in the area of constraint programming. In particular, we show how the DDT problem can be conveniently formulated using quantified CSPs (QCSPs). QCSPs can be viewed as an extension of CSPs to multi-agent scenarios, and consequently, we describe how DDTs correspond to winning strategies in an adversarial game. This leads to several contributions: first, it turns out that the problem of generating definitely discriminating tests [1] is in a different complexity class than the diagnosability problem described in [2, 3] (PSPACE rather than NP). Second, formulating DDT generation as an instance of QCSP solving allows to leverage recent progress in QCSP/QBF solvers in order to effectively compute DDTs. Third, we observe that our QCSP (adversarial planning) formulation of the testing problem is more powerful than previous formalizations, as it allows to generate complex test strategies instead of simple input patterns, thus extending testing to a new range of applications (systems with limited controllability) that could not be handled by previous frameworks.

1.1 Discriminating Tests

We briefly review the theory of constraint-based testing of physical systems as introduced in [1]. Testing attempts to discriminate between different hypotheses about a system (for example, about different kinds of faults) by stimulating the system in such a way that the hypotheses become observationally distinguishable. Formally, let $M = \bigcup_i M_i$ be a set of different models (hypotheses) for a system, where each M_i is a set of constraints over variables V. Let $I = \{i_1, \ldots, i_n\} \subseteq V$ be the subset of input (controllable) variables, $O = \{o_1, \ldots, o_m\} \subseteq V$ the subset of observable variables, and $U = \{u_1, \ldots, u_k\} = V - (I \cup O)$ the remaining (uncontrollable and unobservable) variables. The goal is then to find assignments to I (input patterns) that will cause different assignments to O (output patterns) for the different models M_i:

Definition 1 (Discriminating Tests [1]). *An assignment t_I to I is a* possibly discriminating test *(PDT), if for all M_i there exists an assignment t_O to O such that $t_I \wedge M_i \wedge t_O$ is consistent and for all M_j, $j \neq i$, $t_I \wedge M_j \wedge t_O$ is inconsistent. The assignment t_I is a* definitely discriminating test *(DDT), if for all M_i and all assignments t_O to O, if $t_I \wedge M_i \wedge t_O$ is consistent then for all M_j, $j \neq i$, it follows that $t \wedge M_j \wedge o$ is inconsistent.*

In the following, we restrict ourselves to the case where there are only two possible hypotheses, corresponding to normal and faulty behavior of the system. For example, consider the circuit in Fig. 1. It consists of five variables x, y, z, u, v with domain {L,H}, where x, y, z are input variables, and v is observable. The behavior of the two components is captured by two constraints f_{diff} and f_{add}, respectively; for instance, values L and H can add up to the value L or H, etc. Now assume we have two hypotheses 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 $M_1 = \{f_{\text{diff}}, f_{\text{add}}\}$. The second hypothesis is that the adder is stuck-at-L, which is modeled by $M_2 = \{f_{\text{diff}}, f_{\text{addstuck}}\}$. Then for example, the assignment $x \leftarrow L, y \leftarrow H, z \leftarrow L$ is a PDT for $M = \{M_1, M_2\}$ (it leads to the observation $v \leftarrow L$ or $v \leftarrow H$ for M_1, and $v \leftarrow L$ for M_2), while the assignment $x \leftarrow L, y \leftarrow H, z \leftarrow H$ is a DDT for M (it leads to the observation $v \leftarrow H$ for M_1, and $v \leftarrow L$ for M_2).



f_{diff}:

x	y	u
L	L	L
L	H	H
H	L	H
H	H	L

f_{add}:

u	z	v
L	L	L
L	H	L
L	H	H
H	L	L
H	L	H
H	H	H

$f_{add\text{-}stuck}$:

u	z	v
L	L	L
L	H	L
H	L	L
H	H	L

Fig. 1. Circuit with a possibly faulty adder.

1.2 Test Generation as QCSP Solving

It turns out that the two forms of tests in Def.1 correspond to different classes of constraint problems. The first form of testing, finding PDTs, corresponds to solving a CSP and is captured by the formula

$$\exists i_1 \dots i_n \exists o_1 \dots o_m \exists u_1 \dots u_k . \overline{M_1} \vee \overline{M_2} \tag{1}$$

where $\overline{M_i}$ denotes the complement of M_i. QCSPs [?] are a generalization of CSPs where variables can be existentially or universally quantified. As this extension corresponds to a step from single-agent to multi-agent scenarios, QCSPs are significantly more expressive than CSPs, and are suited to model problems like two-player games. In the following, we show how the second (stronger) form of testing, finding DDTs, can be characterized as a game played between two opponents: the first player (∃-player) tries to reveal the fault by choosing inputs for which the two hypotheses yield disjunct observations. The second player (∀-player) instead tries to hide the fault by choosing observations (outputs) that overlap for the two hypotheses. There are rules that the ∀-player must adhere to: he can only choose among observations that are consistent with the model of the system, as not all observations are possible in all situations. The goal of the game is then that exactly one hypothesis becomes true. It is easy to see that using this game-theoretic characterization, a DDT exists if and only if the first player has a winning strategy.

There exist several approaches how problems described as games can be turned into an equivalent QCSP or QBF (boolean version of QCSP) formulation [6, 7]. In analogy to [?], we get the following QCSP formula to capture DDTs:

$$\exists i_1 \dots i_n \forall o_1 \dots o_m \forall u_1 \dots u_k . \overline{M_1} \vee \overline{M_2} \tag{2}$$

Note that while CSPs are in the class of NP-complete problems, QCSPs are in the complexity class PSPACE that is believed to comprise of even harder problems. The formulation 2 of the existing theory of testing [1] as an instance of QCSP solving allows us to leverage recent progress in QCSP solving in order to actually compute such tests. But it is also a starting point to extend the theory to a new range of applications. In Def. 1, tests are assumed to consist of assignments to the controllable variables; the underlying assumption is that the these variables characterize all relevant causal inputs to the system. However, one can conceive situations where this assumption is too restrictive; in practice, there might be variables/parameters who influence the system's behavior, but whose values cannot be completely controlled. This scenario of *testing under limited controllability* can be captured using a slight modification of formula 2. Let I be partitioned into input variables $I_c = \{i_1 \dots i_s\}$ that can be controlled (set during testing), and input variables $I_{nc} = \{i_{s+1} \dots i_n\}$ that can be observed but not controlled. Then a definitely discriminating test exists iff the following formula is satisfiable:

$$\forall i_{s+1} \dots i_n \exists i_1 \dots i_s \forall o_1 \dots o_m \forall u_1 \dots u_k . \overline{M_1} \vee \overline{M_2} \tag{3}$$

Note that while solutions to Eqn. 1 and Eqn. 2 are simply assignments to the values of the input variables, solutions to Eqn. 3 are in general more complex and correspond to a *strategy* or *policy* that states how the values of the controllable variables I_c must be set depending on the values of the non-controllable variables I_{nc}. To illustrate this, consider again the example in Fig.1, but assume that variable x can't be controlled. According to Def. 1 no DDT exists in this case, as the possible observations for v will always overlap for the two models (hypotheses) M_1 and M_2. However, there exists a test strategy to distinguish M_1 from M_2, which consists of setting y depending on the value of x: choose $y \leftarrow H, z \leftarrow H$ if $x = L$, and choose $y \leftarrow L, z \leftarrow H$ if $x = H$). Generating such strategies goes beyond the theory in [1], which assumed that tests consists of assignments (patterns) for the input variables, but it is possible in our QCSP framework.

We conducted preliminary experiments of QCSP-based DDT generation with the solvers Qecode [7] and sKizzo [8] (since the present version of Qecode does not allow one to extract solutions from satisfiable instances, we transform the instance into QBF and use sKizzo to extract solutions). Figure 2 shows solutions generated from Eqn. 3 for the example in Fig. [**?**]. The solutions are represented in the form of BDDs with complemented arcs (see [8]), where $\neg x$ stands for $x \leftarrow L$, x stands for $x \leftarrow H$, etc. The left-hand side of the figure shows the strategy (in this case, a simple set of assignments) that is generated if variables x, y, z are specified as controllable (input) variables, whereas the right-hand side of the figure shows the strategy when only y, z are controllable (in this case, y must be set depending on the value of x). No solution (definitely discriminating test strategy for the fault) exists if only z is assumed to be controllable.

1.3 Conclusion and Directions for Future Work

We reviewed an existing theory of testing for physical systems, which defines a weaker (PDTs) and a stronger form (DDTs) of test inputs, and showed how it can be framed as CSP and QCSP solving, respectively.

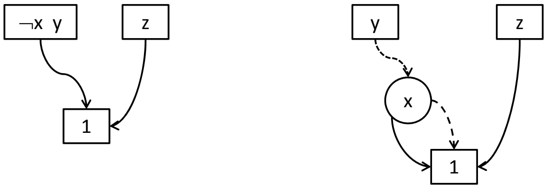


Fig. 2. Test strategies generated for the circuit in Fig. 1.

Assumptions in this theory about the complete controllability of system inputs can be relaxed and lead to a more powerful class of tests, where inputs are intelligently set in reaction to observed values. Such *test strategies* go beyond the test pattern approach of the existing theory, but they can be captured in the QCSP framework.

We are currently working on larger, more realistic examples to evaluate our QCSP-based testing approach. We are also extending our framework to systems with dynamic behavior (transition systems), in order to complement (passive) verification tools for embedded autonomous controllers [2] with a capability to generate test strategies to (actively) reveal faults.

References

1. Peter Struss: Testing Physical Systems, Proceedings AAAI-94, pp. 251–256, 1994.
2. Alessandro Cimatti, Charles Pecheur, and Roberto Cavada: Formal Verification of Diagnosability via Symbolic Model Checking, Proceedings IJCAI-05, pp. 363–369, 2003.
3. Jussi Rintanen and Alban Grastien: Diagnosability Testing with Satisfiability Algorithms, Proceedings IJCAI-07, pp. 532–537, 2007.
4. Michael Esser and Peter Struss: Fault-Model-Based Test Generation for Embedded Software, Proceedings IJCAI-07, pp. 342–347, 2007.
5. Reiner Inderst: Automatische Testgenerierung auf der Basis einer qualitativen Modellierung physikalischer Systeme, Master's thesis, Technische Universität München, Germany, 1995.
6. Carlos Ansótegui, Carla Gomes, and Bart Selman: The Achilles' Heel of QBF, Proceedings AAAI-05, pp. 275–281, 2005.
7. Marco Benedetti, Arnaud Lallouet, and Jérémie Vautard: QCSP Made Practical by Virtue of Restricted Quantification, Proceedings IJCAI-07, pp. 38–43, 2007.
8. Marco Benedetti: sKizzo: A Suite to Evaluate and Certify QBFs, Proceedings CADE-05, 2005.