In: International Joint Conference on Artificial Intelligence (IJCAI). 6th January through 12th January 2007, Hyderabad, India

Model-based Optimization of Testing through Reduction of Stimuli

P.Struss
Technische Universität München, Boltzmannstr. 3 D-85748 Garching, Germany
and
OCC'M Software, Gleissentalstr. 22, D-82041 Deisenhofen, Germany
struss@in.tum.de, struss@occm.de

Abstract

The paper presents the theoretical foundations and an algorithm to reduce the efforts of testing physical systems. A test is formally described as a set of stimuli (inputs to the system) to shift the system into a particular situation or state, and a set of varia-bles whose observation or measurement refutes hypotheses about the behavior mode the system is operating in. Tests (either generated automatically or by humans) may contain redundancy in the sense that some of its stimuli and/or observables maybe irrelevant for achieving the result of the test. Identifying and dropping them contributes to redu-cing the cost of set-up actions and measurements. We define different kinds of irrelevant stimuli, discuss their practical importance, and present criteria and algorithms for computing reduced tests.

1 Introduction

Testing of physical systems is a frequent task in industry: During or after manufacturing of a product it has to be checked whether the process worked properly and the product behaves as designed. Under operation, wearing und breaking of parts may lead to system failures, and it has to be investigated whether and where a fault occurred and of what kind it is. Even though some testing, particularly in manufacturing, is performed automatically and requires no or limited human intervention, saving time and efforts spent on testing is an economical requirement. This becomes more important with the amount of necessary human actions, such as disassembly of parts of a vehicle in a workshop, and the cost of downtime of large equipment.

Designing **effective** test sets or sequences is a demanding and time consuming task, particularly when the systems to be tested come in many variants such as cars and their subsystems. In [Struss 94], we presented the theoretical foundations and implemented algorithms to generate tests for a device based on behavior models of its components.

Designing **efficient** tests is a challenge for the reasons stated above. Our solution presented in [Struss 94] addressed this in one way: it searches for tests that could serve several purposes at once, i.e ruling out more than one hypothesis. This increases efficiency of testing by **reducing** the **number** of tests. However, it was ignorant of another source of efficiency: the **reduction** of the **efforts** spent on an **individual** test. More precisely: so far, the question answered was "Given a set of possible stimuli to a system and a set of observables, how can we stimulate the system such that the observables reveal information about the actual behavior mode of the system?" Now, we address the problem of determining **minimal** sets of stimuli. When combined with an estimation of costs of the respective actions, the solution will contribute to cost-optimal testing. However, this paper is neither addressing costs nor the task of organizing the tests in a sequence or tree, which are different issues.

In the following section, we will introduce a formal definition and representation of tests based on a relational representation of the behavior model of a system or, more generally, the hypotheses to be tested. The basis for this is the manipulation of finite relations as they are given by quali-tative behavior models.

Section 3 provides the formal foundations for test reduction by defining and characterizing redundancy in tests in terms of variables that are irrelevant for a particular test. The algorithms are presented in section 4. Finally, we discuss the practical impact of the solution and the open problems.

2 The Background: Model-based Test Generation

In the most general way, testing aims at finding out which hypothesis out of a set H is correct (if any) by stimulating a system such that the available observations of the system responses to the stimuli refute all but one hypotheses (or even all of them).

This is captured by the following definition.

Definition (Discriminating Test Input)

Let
$TI = \{ti\}$ be the set of possible test inputs (stimuli),
$OBS = \{obs\}$ the set of possible observations (system responses), and
$H = \{h_i\}$ a set of hypotheses.

$ti \in TI$ is called a **definitely** discriminating test input for H if

(i) $\forall\ h_i \in H\ \exists\ obs \in OBS \quad ti \wedge h_i \wedge obs \nvdash \bot$

and

(ii) $\forall\ h_i \in H\ \forall\ obs \in OBS$

if $ti \wedge h_i \wedge obs \nvdash \bot$

then $\forall\ h_j \neq h_i\ ti \wedge h_j \wedge obs \vdash \bot$.

ti is a **possibly** discriminating test input if

(ii´) $\forall\ h_i \in H\ \exists\ obs \in OBS$ such that

$ti \wedge h_i \wedge obs \nvdash \bot$

and $\forall\ h_j \neq h_i\ \ ti \wedge h_j \wedge obs \vdash \bot$.

In this definition, condition (i) expresses that there exists an observable system response for each hypothesis under the test input. It also implies that test inputs are consistent with all hypotheses. i.e. we are able to apply the stimulus, because it is causally independent of the hypotheses. Condition (ii) formulates the requirement that the resulting observation guarantees that at most one hypothesis will not be refuted, while (ii') states that each hypothesis **may** generate an observation that refutes all others.

Usually, one stimulus is not enough to perform the discrimination task which motivates the following definition.

Definition (Discriminating Test Input Set)

$\{ti_k\} = TI´ \subset TI$ is called a discriminating test input set for $H = \{h_i\}$ if

$\forall\ h_i, h_j$ with $h_i \neq h_j\ \ \exists\ ti_k \in TI´$ such that

ti_k is a (definitely or possibly) discriminating test input for $\{h_i, h_j\}$.

It is called **definitely discriminating** if all ti_k have this property, and **possibly discriminating** otherwise. It is called **minimal** if it has no proper subset $TI´´ \subset TI´$ which is discriminating.

Such logical characterizations (see also [McIlraith-Reiter 92]) are too general to serve as a basis for the development of an appropriate representation and algorithms for test generation. Here, the hypotheses correspond to assumptions about the correct or possible faulty behavior of the system to be tested. They are usually given by equations and implemented by constraints, and test inputs and observations can be described as value assignments to system variables.

The system behavior is assumed to be characterized by a vector

$\underline{v}_S = (v_1, v_2, v_3, \ldots, v_n)$

of system variables with domains

$DOM(\underline{v}_S) =$

$DOM(v_1) \times DOM(v_2) \times DOM(v_3) \times \ldots \times DOM(v_n)$.

Then a hypothesis $h_i \in H$ is given as a relation

$R_i \subseteq DOM(\underline{v}_S)$.

Observations are value assignments to a subvector of the variables, $\underline{v}_{obs}$, and also the stimuli are described by assigning values to a vector $\underline{v}_{cause}$ of susceptible ("causal" or input) variables. We make the reasonable assumption that we always know the applied stimulus which means the causal variables are a subvector of the observable ones:

$\underline{v}_{cause} \subseteq \underline{v}_{obs} \subseteq \{v_i\}$

Since the focus of this paper is not on test generation and our solution to test reduction is independent of the way the tests were produced, we only briefly summarize the foundation for automated model-based test generation and refer to for [Struss 94] details. The basic idea is to construct test inputs by computing them from the observable differences of the relations that represent the various hypotheses. Figure 1 illustrates this. Firstly, for testing, only the observables matter. Accordingly, Figure 1 presents only the projections, $p_{obs}(R_i)$, $p_{obs}(R_j)$, of two relations, R_1 and R_2, (possibly defined over a large set of variables) to the observable variables. The vertical axis represents the causal variables, whereas the horizontal axis shows the other observable variables (which represent the observable response of the system).

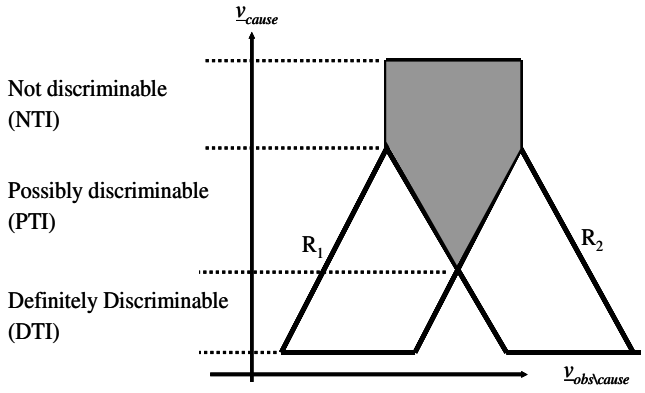

Figure 1 Determining the inputs that do not, possibly, definitely discriminate between R_1 and R_2

To construct a (definitely) discriminating test input, we have to avoid stimuli that can lead to the same observable system response for both relations, i.e. stimuli that may lead to an observation in the intersection

$p_{obs}(R_i) \cap p_{obs}(R_j)$

shaded in Figure 1. These test inputs we find by projecting the intersection to the causal variables:

$p_{cause}(p_{obs}(R_i) \cap p_{obs}(R_j))$.

The complement of this is the complete set of all test inputs that are guaranteed to produce different system responses under the two hypotheses:

$DTI_{ij} = DOM(\underline{v}_{cause}) \setminus p_{cause}(p_{obs}(R_i) \cap p_{obs}(R_j))$.

Lemma 1

If $h_i=R_i$, $h_j=R_j$, $TI=DOM(\underline{v}_{cause})$, and $OBS=DOM(\underline{v}_{obs})$, then DTI_{ij} is the set of all definitely discriminating test inputs for $\{h_i, h_j\}$.

Please, note that we assume that the projections of R_i and R_j cover the entire domain of the causal variables which corresponds to condition (i) in the definition of the test input.

We only mention the fact, that, when applying tests in practice, one may have to avoid certain stimuli because they carry the risk of damaging or destroying the system or to create catastrophic effects as long as certain faults have not

been ruled out. In this case, the admissible test inputs are given by some set $R_{adm} \subseteq DOM(\underline{v}_{cause})$, and we obtain

$DTI_{adm,\,ij} = R_{adm} \setminus p_{cause}(p_{obs}(R_i) \cap p_{obs}(R_j))$.

In a similar way as DTI_{ij}, we can compute the set of test inputs that are guaranteed to create indistinguishable observable responses under both hypotheses, i.e. they cannot produce observations in the difference of the relations:

$(p_{obs}(R_i) \setminus p_{obs}(R_j)) \cup (p_{obs}(R_j) \setminus p_{obs}(R_i))$.

Then the non-discriminating test inputs are

$NTI_{ij} =$

$DOM(\underline{v}_{cause}) \setminus p_{cause}((p_{obs}(R_j) \setminus p_{obs}(R_i)) \cup (p_{obs}(R_i) \setminus p_{obs}(R_j)))$

All other test inputs may or may not lead to discrimination.

Lemma 2

The set of all possibly discriminating test inputs for a pair of hypotheses $\{h_i, h_j\}$ is given by

$PTI_{ij} = DOM(\underline{v}_{cause}) \setminus (NTI_{ij} \cup DTI_{ij})$.

The sets DTI_{ij} for all pairs $\{h_i, h_j\}$ provide the space for constructing (minimal) discriminating test input sets.

Lemma 3

The (minimal) hitting sets of the set $\{DTI_{ij}\}$ are the (minimal) definitely discriminating test input sets.

A hitting set of a set of sets $\{A_i\}$ is defined by having a non-empty intersection with each A_i. (Please, note that Lemma 3 has only the purpose to characterize all discriminating test input sets. Since we need **only one** test input to perform the test, we are not bothered by the complexity of computing all hitting sets.)

This way, the number of tests constructed can be less than the number of necessary pairwise discrimination between n hypotheses, n^2 - n. If the tests have a fixed cost associated, then the cheapest test set can be found among the minimal sets. However, it is worth noting that the test input sets are the minimal ones that **guarantee** the discrimination among the hypotheses in H. In practice, only a subset of the tests may have to be executed, because some of them refute more hypotheses than guaranteed (because they are a possibly discriminating test for some other pair of hypotheses) and render other tests unnecessary.

The computation is based on operations on relations, such as intersection and projection, and will usually practically work only on finite relations. Qualitative abstraction can generate such representations for continuous models and, hence, enable a broad applicability of the algorithm. The many existing test generation algorithms for digital circuits are specializations of it (provided they are sound and complete). Of course, they can exploit the special Boolean domain and, hence, may be more efficient than our general algorithm.

The algorithm has been implemented based on commercial software components of OCC'M's RAZ'R ([OCC'M 05]) which provide a representation and operations of relations as ordered multiple decision diagrams (OMDD). The input is given by constraint models of correct and faulty behavior of components taken from a library which are aggregated according to a structural description. These models are the same ones that can and have been used for model-based diagnosis and detectability and discriminability analysis.

It is important to note that the required operations on the relations are applied to the **observable variables** only (including the causal variables). The projection of the entire relation R_i to this space can be understood as producing a black box model that directly relates the stimuli and the observable response. In many relevant applications, this space will be predefined and small. For instance, when testing of car subsystems exploits the on-board actuators and sensors only, this may involve some 10 - 20 variables or so. The entire workshop diagnosis task has more potential probing points, but still involves only a small subset of the variables in the entire behavior relation R_i.

Finally, we mention that probabilities (of hypotheses and observations) can be used to optimize test sets ([Struss 94a], [Vatcheva-de Jong-Mars 02]).

3 Different Kinds of Irrelevant Causal Variables

In the tests characterized by Lemma 1, test inputs are tuples of values for **all** available causal variables, and the guarantee for discrimination is related to **all** specified observables. However, it may be the case, that a generated test is redundant in the sense that already of subset of inputs and/or observations would provide the same information for discrimination. This is important, because costs are often related to the number of stimuli and observation actions. If we can reduce individual tests to the necessary stimuli and/or observations only, this will contribute to reducing costs for testing.

In the following, we will provide the foundations for reducing the set of input variables. More details can be found in [Strobel 04].

Let $DTI_{ij} \subseteq DOM(\underline{v}_{cause})$ be the set of definitely discriminating test inputs. The question is whether there is a subvector $\underline{v}'_{cause} \subseteq \underline{v}_{cause}$ that can be ignored in some way without losing discrimination information provided by the test. Rather than computing the set of test inputs for various subsets of the causal variables to answer this question, we will identify irrelevant causal variables by analyzing DTI_{ij}.

A closer look reveals that a causal variable can be irrelevant in different ways that have a different impact on the generation and application of tests. Let us first illustrate these cases by simple examples and then define them formally.

Suppose you want to test whether the light bulb L in the tiny circuit of Figure 2 works or is defect (open). The possible stimuli are opening and closing of switches S1 and S2, and L can be observed. If we assume that resistor R is not too small, all one has to do is close S1 and observe whether or not L is lit (assuming there is a voltage supply). For this test, the position of S2 is totally irrelevant: whatever its state may be, it does not influence the actions we have to perform.

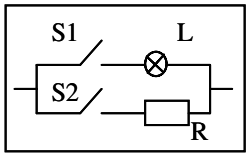

Figure 2 The position of S1 is totally irrelevant for testing L

Regarding the circuit in Figure 3, we can observe the following: the position of switch S1 is irrelevant in the sense that we can test lamp L regardless of whether it is up or down. However, it is not totally irrelevant: in contrast to the first case, the appropriate test inputs depend on the position. For S1 up, S2 must be closed; otherwise, S3 has to be closed. This means, the position of S1 has to be known in order to perform a test, but it does not have to be influenced which allows for omission of an action. We call such a variable weakly irrelevant (in the lack of a better term).

The same circuit can be used to illustrate a third kind of irrelevance of a causal variable: the position of S2 is irrelevant if S1 is in down position. Hence, it is not totally irrelevant, but only under certain conditions. This is still

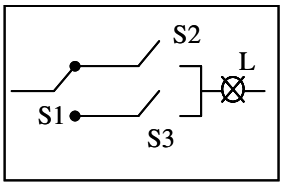

Figure 3 The position of S1 is weakly irrelevant; positions of S2 and S3 are conditionally

practically important, because once the condition is satisfied, we can save by avoiding actions related to S2's position. This variable is conditionally irrelevant, and so is S3's position, of course.

To generalize the intuition gained from the examples and to formalize them: for some subvector $\underline{v'}_{cause} \subseteq \underline{v}_{cause}$ we distinguish the following cases (for which Figure 4 shows abstract examples): for all value assignments from $DOM(\underline{v'}_{cause})$, DTI_{ij} can contain

i. **the same** set of stimuli for the remaining causal variables (total irrelevance)
ii. **some** set of stimuli for the remaining causal variables (weak irrelevance)
iii. **the same** set of stimuli for the remaining causal variables under some **restriction** of the values of the remaining causal variables (conditional irrelevance).

Figure 4 displays the sets *DTI* in the plain of causal variables (**not** the plain of all observables as Figure 1), where the vertical axis corresponds to the irrelevant variable (or subvector of variables) $\underline{v'}_{cause}$, while the horizontal axis represents the remaining ones. In case (i), *DTI* is the cross product of the entire domain of $\underline{v'}_{cause}$ and a certain value assignment to the remaining variables. Case (ii) can be characterized by the fact that the projection of *DTI* to $\underline{v'}_{cause}$

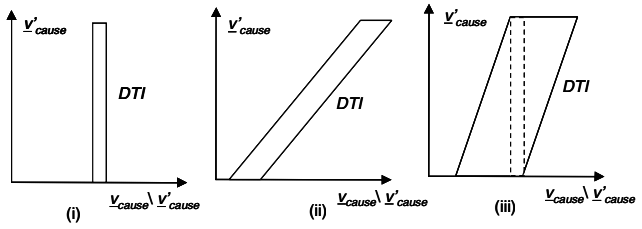

Figure 4 Abstract examples of total irrelevance (i), weak irrelevance (ii), and conditional irrelevance (iii)

covers the entire domain. And case (iii) implies that *DTI* has a subset that is obtained as the cross product $DOM(\underline{v'}_{cause})$ and a set of tuples of the remaining variables (indicated by dotted lines in Figure 4).
This motivates the following definitions:

Definition (Irrelevance of causal variables)
Let $DTI \subset DOM(\underline{v}_{cause})$ be the complete set of definitely discriminating test inputs for two hypotheses. A subset of causal variables $\underline{v'}_{cause} \subseteq v_{cause}$ is called:
(i) **totally irrelevant** if
$DTI = p_{\underline{v}cause \setminus \underline{v'}cause}(DTI) \times DOM(v'_{cause})$
(ii) **weakly irrelevant** if
$p_{\underline{v'}cause}(DTI) = DOM(v'_{cause})$
(iii) **conditionally irrelevant** if there is a non-empty subset $DTI' \subset DTI$ such that
$DTI' = p_{\underline{v}cause \setminus \underline{v'}cause}(DTI') \times DOM(v'_{cause})$

As suggested by Figure 4, there are implication relations among the three types of irrelevance, which can be easily proved based on the above definition.

Lemma 4
If $\underline{v'}_{cause}$ is totally irrelevant
$\Rightarrow \underline{v'}_{cause}$ is also conditionally irrelevant.
If $\underline{v'}_{cause}$ is conditionally irrelevant
$\Rightarrow \underline{v'}_{cause}$ is also weakly irrelevant.

The goal of identifying **sets** of irrelevant causal variables seems to imply that one has to consider the power set of the causal variables. However, this is not the case due to the following lemma.

Lemma 5
(v_1) and (v_2) are both totally irrelevant
$\Leftrightarrow \underline{v'}_{cause} = (v_1, v_2)$ is totally irrelevant.
(v_1) and (v_2) are both weakly irrelevant
$\Leftrightarrow \underline{v'}_{cause} = (v_1, v_2)$ is weakly irrelevant.

This allows us to investigate this kind of irrelevance independently for each variable and then comprise them in one set, which makes the check linear in the number of causal variables. However, a similar lemma does not apply to conditionally irrelevant variables.

Remark
If (v_1) and (v_2) are both conditionally irrelevant, then $\underline{v'}_{cause} = (v_1, v_2)$ is not necessarily conditionally irrelevant.

Obviously, to establish conditional irrelevance of the pair of variables, the conditions for the irrelevance of the two

variables would have to have a non-empty intersection. But Figure 3 provides an example in which they are even exclusive: the position of S2 is irrelevant under conditions that require a particular state of S3 and vice versa.

4 Test Reduction

Based on the definitions and lemmata in the previous section, we developed algorithms for the automated reduction of tests. Whether they have been generated by an algorithm like the one sketched in section 2 or by human experts is irrelevant, as long as they can be represented in the relational style.

Firstly, we exploit lemma 5: we start with the test input set for a maximal set of causal variables and then analyze irrelevance of **each single** causal variable.

Secondly, we check for weak irrelevance first, because lemma 4 allows ruling out also the other kinds of irrelevance in the negative case. This check can be based directly on the definition of weak irrelevance and is formally described as follows.

Lemma 6

Let v_k be a causal variable, p_k the projection to this variable, and *DTI* a set of definitely discriminating test inputs.

If $p_k(DTI) = DOM(v_k)$
then v_k is weakly irrelevant to *DTI*.

If $p_k(DTI) \neq DOM(v_k)$
then v_k is not weakly, conditionally, or totally irrelevant to *DTI*.

In case of a weakly irrelevant variable we can check for conditional and total irrelevance. To get the idea underlying this test, a glance at the abstract example of Figure 4 may be helpful. We have to check whether there exists a non-empty $DTI' \subset DTI$ such that

$p_{-k}(DTI') \times Dom(V_k) \subset DTI'$,

where p_{-k} is the projection to the causal variables except v_k:

$\underline{v}_{cause} \setminus \{v_k\}$.

We do so by computing the projection of the maximal DTI'

$DTI'_{-k} := p_{-k}(DTI')$

and checking whether it is empty, and we compute it by computing its complement.

DTI'_{-k} comprises all value assignments to $\underline{v}_{cause} \setminus \{v_k\}$ that when combined with arbitrary values of v_k always yield a test input of *DTI*. Hence, its complement contains all value assignments that can be combined with at least one value of v_k to yield a test input that does not lie in *DTI*, but in its complement:

$$DOM(\underline{v}_{cause} \setminus \{v_k\}) \setminus DTI'_{-k} = \{v_{-k,0} \in DOM(\underline{v}_{cause} \setminus \{v_k\}) \mid \exists\ v_{k,0} \in DOM(v_k) \wedge v_{-k,0} \circ v_{k,0} \in DOM(\underline{v}_{cause}) \setminus DTI\}.$$

But this is the projection of the complement of *DTI*:

$p_{-k}(DOM(\underline{v}_{cause}) \setminus DTI)$.

This yields the following lemma which underlies the second check.

Lemma 7

Let v_k be a causal variable, p_{-k} the projection to the other causal variables, and *DTI* a non-empty set of definitely discriminating test inputs. Furthermore, let

$DTI'_{-k} := DOM(\underline{v}_{cause} \setminus \{v_k\}) \setminus p_{-k}(DOM(v_{cause}) \setminus DTI)$.

If $DTI'_{-k} = \emptyset$
then v_k is not conditionally or totally irrelevant to *DTI*.

If $DTI'_{-k} \neq \emptyset$
then v_k is conditionally irrelevant to *DTI*

If $DTI'_{-k} = p_{-k}(DTI)$
the v_k is totally irrelevant to *DTI*.

Please note that DTI'_{-k} represents the condition under which v_k is irrelevant. This can be used for investigating the relationship of these conditions for different causal variables. The third implication of the lemma simply reflects the fact that total irrelevance is obtained if the condition comprises **all** value assignments to the other causal variables that occur in *DTI*.

This establishes an algorithm for determining whether a causal variable is irrelevant and if so, of what type:

IF $p_k(DTI) = DOM(v_k)$
  THEN
    IF $DTI'_{-k} = \emptyset$
      THEN "WEAKLY IRRELEVANT"
      ELSE IF $DTI'_{-k} = p_{-k}(DTI)$
        THEN "TOTALLY IRRELEVANT"
        ELSE "IRRELEVANT UNDER DTI'_{-k}"
  ELSE "NOT IRRELEVANT"

Based on the results of this algorithm, the irrelevant variables can be removed from DTI by projection yielding a simplified and cheaper test input set.

What we have presented for the case of definitely discriminating test input sets can obviously be applied in the same way to possibly discriminating test inputs.

5 Discussion and Future Challenges

The generation of a set of test input sets (with or without the reduction described here) provides the starting point for different further processing and use of this information. One can select one test input from each set and generate a fixed sequence or decision tree of tests to be applied. The information could also be used in a dynamic way by making the choice of the next test dependent on the current situation. A characterization of the situation can involve two aspects: firstly, the hypotheses actually refuted so far. We emphasize again, that this is not completely fixed by the tests executed so far, because some of them may have refuted all hypotheses that they can discriminate, and also they may have refuted more hypotheses than were guaranteed to be refuted. Secondly, one can choose the next test based on the current state of the system in order to minimize the number of stimuli that have to be changed.

The different types of irrelevance have a different impact on these strategies. Obviously, totally irrelevant

variables can be eliminated from the respective test inputs, i.e. they do not have to be considered for the respective test actions. However, unless they are irrelevant to all test input sets in the set, they have to be observed during the testing, because they may be weakly irrelevant to some other test input sets and, hence, their value has to be known in order to determine the appropriate values for the other causal variables.

Weakly irrelevant variables do not have to be influenced either in the respective test, but the appropriate values for the other variables have to be determined by restricting *DTI* for the next step to the current values of the weakly irrelevant variables.

For conditionally irrelevant variables, it has to be checked whether the irrelevance condition $DTI'_{\cdot k}$ is satisfied in the current situation, and if so, they do not have to be touched, and an arbitrary assignment of values out of $DTI'_{\cdot k}$ can be chosen for the relevant variables.

In this paper, we focused on the reduction of the number and costs of stimuli actions. This is justified because their costs are often higher than those of observing the system response. Reducing also the cost associated with observations is nevertheless a task that needs to be addressed. However, the solution for the causal variables does not simply carry over, and the tasks are not independent: in principle, a reduction of the set of observables may require the presence of certain stimuli and vice versa.

Another challenge is to investigate how serious a fundamental limitation of our approach is (and to overcome it if necessary and possible): the behavior representation in terms of relations and, hence, a rather static view on the system to be tested. If **dynamic features** are relevant, they can be accommodated by including derivatives in the set of model variables. Another solution is to base the behavior representation on **transitions**. Since they can be represented again by relations (linking the states "before" and "after") the described representations and algorithms remain applicable.

We have explored the latter solution by transforming models given as finite state machines into such a representation with the goal of extending the solution to **testing of software** [Esser-Struss 07]. This provides a challenge in itself, mainly because of the difficulty in establishing appropriate fault hypotheses: While for many physical devices, such hypotheses are determined by the ways the components wear and fail, the ways in which software can fail spans an infinite space and may include structural faults. An extension of the test generation and reduction methods to include software would be highly attractive because it would allow testing embedded software and its physical context in an integrated way.

Acknowledgements

Thanks to Torsten Strobel who implemented the algorithm, Oskar Dressler for discussions and support of this work, and the Model-based Systems and Qualitative Modeling Group at the Technical University of Munich. This work was supported in part by Audi AG, Ingolstadt.

References

[Esser-Struss 07] Esser, M., Struss, P.: *Fault-model-based Test Generation for Embedded Software*. In: Proceedings of the 20th International Joint conference on Artificial Intelligence IJCAI-07, Hyderabad, India, 2007

[McIlraith-Reiter 92] McIlraith, S., Reiter, R.: *On Tests for Hypothetical Reasoning*. In: W. Hamscher, J. de Kleer und L. Console (Hg.). Readings in Model-based Diagnosis: Diagnosis of Designed Artifacts Based on Descriptions of their Structure and Function. Morgan Kaufmann, San Mateo, 1992

[OCC'M 05] www.occm.de

[Strobel 04] Strobel, T., *Ein Algorithmus zur Optimierung automatischer, modellbasierter Testfallgenerierung*, Diploma Thesis, Techn. Univ. Munich, 2004 (in German)

[Struss 94] Struss, P.: *Testing Physical Systems.* In: Proceedings of AAAI-94, Seattle, USA, 1994.

[Struss 94a] Struss, P.: *Testing for Discrimination of Diagnoses*. In: Working Papers of the 5th International Workshop on Principles of Diagnosis (DX-94), New Paltz, USA, 1994.

[Vatcheva-de Jong-Mars 02] Vatcheva, I., de Jong, H., Mars, N.: *Selection of Perturbation Experiments for Model Discrimination*. Proceeddings of ECAI-02, 2002