Automated Test Reduction

P. Struss\textsuperscript{1,2}

\textsuperscript{1}Technische Universität München, Boltzmannstr. 3, D-85748 Garching, Germany
\textsuperscript{2}OCC'M Software, Gleissentalstr. 22, D-82041 Deisenhofen, Germany
struss@occm.de, struss@in.tum.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 variables 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 reducing the cost of set-up actions and measurements. We define different kinds of irrelevant variables, 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 and 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 behavior model of the system?” Now, we address the problem of determining minimal sets of stimuli and observations. 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. We also briefly summarize the basis for automated model-based test generation, although our solution to test reduction is independent of the way the tests were produced. The basis for this is the manipulation of finite relations as they are given by qualitative 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 \( TI = \{t_i\} \) be the set of possible test inputs (stimuli), \( OBS = \{obs\} \) the set of possible observations (system responses), and
The basic idea underlying model-based test generation ([Struss 94]) is then that the construction of test inputs is done 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, $p_{\text{obs}}(R)$, 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 $R_2$](image)

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 $(p_{\text{obs}}(R_1) \cap p_{\text{obs}}(R_2))$ shaded in Figure 1. These test inputs we find by projecting the intersection to the causal variables: $$p_{\text{cause}}(p_{\text{obs}}(R_1) \cap p_{\text{obs}}(R_2)) =$$

The complement of this is the complete set of all test inputs that are guaranteed to produce different system responses under the two hypotheses: $DT_H = DOM(\Sigma_{\text{caus}}} \setminus p_{\text{cause}}(p_{\text{obs}}(R_1) \cap p_{\text{obs}}(R_2))$.

**Lemma 1**

If $h_1 = R_1$, $h_2 = R_1$, $T_I = DOM(\Sigma_{\text{caus}})$, and $OBS = DOM(\Sigma_{\text{obs}})$, then $DT_H$ is the set of all definitely discriminating test inputs for $[h_1, h_2]$. 

Please, note that we assume that the projections of $R_1$ and $R_2$ cover the entire domain of the causal variables which corresponds to condition (i) in the definition of the test input (an assumption which may be relaxed in the otherwise identical discriminability/detectability analysis presented in [Dressler-Struss 03]).

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 $R_{adm} \subseteq DOM(\Sigma_{caus})$, and we obtain $DTI_{adm, ij} = R_{adm} \setminus \{p_{caus}(R_i) \cap p_{obs}(R_j)\}$.

In a similar way as $DTI$, 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:

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

Then the non-discriminating test inputs are

$$NTI_{ij} = DOM(\Sigma_{caus}) \setminus (p_{caus}(R_i) \cup p_{caus}(R_j))$$

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 $\{p_{obs}(R_i) \cup p_{obs}(R_j)\}$. 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 $n^2 - n$. 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 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 $R_i$.

In practice, we did not encounter computational limitations in the test generation step, given the projection $\{p_{obs}(R_i)\}$. The bottleneck lies in the construction of $R_i$, which is given as a compositional model and its projection $\{p_{obs}(R_i)\}$. It is obvious that computing the join and eliminating the unobservable variables can and should be done in an interleaved way. However, even though the overall result of this construction may be very small, intermediate results can grow big and sometimes too big even for a representation as an OMDD. Developing good heuristics to avoid the explosion of the space requirement is complicated by the fact that the number of nodes in the OMDD, which represents the relation as a graph, is not simply a function of the number of tuples in the relation. It can be strongly dependent on structural features, predominantly the one of the order of the variables in the graph, in a way that is not understood well.

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

The abovementioned algorithms aim at reducing the costs of testing by reducing the number of tests to be performed, given sets of observable and causal variables. This means, the 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 the 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 $DTI_{ij} \subseteq DOM(\Sigma_{caus})$ be the set definitely discriminating test inputs. The question is whether there is a subvector $\Sigma_{caus} \subseteq \Sigma_{caus}$ 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 $\text{DTI}_j$.

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 $S_1$ and $S_2$, and $L$ can be observed. If we assume that resistor $R$ is not too small, all one has to do is close $S_1$ and observe whether or not $L$ is lit (assuming there is a voltage supply). For this test, the position of $S_2$ 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](image)

Regarding the circuit in Figure 3, we can observe the following: the position of switch $S_1$ 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 $S_1$ up, $S_2$ must be closed; otherwise, $S_3$ has to be closed. This means, the position of $S_1$ 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).

![Figure 3 The position of S1 is weakly irrelevant; positions of S2 and S3 are conditionally](image)

The same circuit can be used to illustrate a third kind of irrelevance of a causal variable: the position of $S_2$ is irrelevant if $S_1$ is in down position. Hence, it is not totally irrelevant, but only under certain conditions. This is still practically important, because once the condition is satisfied, we can save by avoiding actions related to $S_2$’s position. This variable is conditionally irrelevant, and so is $S_3$’s position, of course.

To generalize the intuition gained from the examples and to formalize them: for some subvector $\mathbf{v}'_{\text{cause}} \subseteq \mathbf{v}_{\text{cause}}$ we distinguish the following cases (for which Figure 4 shows abstract examples): for all value assignments from $\text{DOM}(\mathbf{v}'_{\text{cause}})$, $\text{DTI}_j$ 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 $\text{DTI}$ 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) $\mathbf{v}'_{\text{cause}}$, while the horizontal axis represents the remaining ones. In case (i), $\text{DTI}$ is the cross product of the entire domain of $\mathbf{v}'_{\text{cause}}$ and a certain value assignment to the remaining variables. Case (ii) can be characterized by the fact that the projection of $\text{DTI}$ to $\mathbf{v}'_{\text{cause}}$
causal variables. However, this is not the case due to the following lemma.

**Lemma 5**

(v1) and (v2) are both totally irrelevant
\[ \iff \exists \gamma_{\text{cause}} = (v_1, v_2) \text{ is totally irrelevant.} \]

(v1) and (v2) are both weakly irrelevant
\[ \iff \exists \gamma_{\text{cause}} = (v_1, v_2) \text{ is weakly irrelevant.} \]

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, the lemma does not apply to conditionally irrelevant variables.

**Remark**

If (v1) and (v2) are both conditionally irrelevant, then \( \exists \gamma_{\text{cause}} = (v_1, v_2) \) is not necessarily conditionally independent.

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 \( p_k (DTI) = \text{DOM}(v_k) \)
then \( v_k \) is weakly irrelevant to \( DTI \).

If \( p_k (DTI) \neq \text{DOM}(v_k) \)
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
\[ p_A(DTI') \times \text{DOM}(V_1) \subset DTI'. \]

where \( p_A \) is the projection to the causal variables except \( v_k : \Sigma_{\text{cause}} \{ v_k \} \).

We do so by computing the projection of the maximal \( DTI' \)
\[ DTI' : p_A(DTI) \]
and checking whether it is empty, and we compute it by computing its complement.

\( DTI' \) comprises all value assignments to \( \Sigma_{\text{cause}} \{ v_k \} \) 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:
\[ \text{DOM}(\Sigma_{\text{cause}} \{ v_k \}) \setminus DTI' = \{ v_{k,0} \in \text{DOM}(\Sigma_{\text{cause}} \{ v_k \}) \mid \exists v_{k,0} \in \text{DOM}(v_k) \land v_{k,0} \neq v_{k,0} \in \text{DOM}(\Sigma_{\text{cause}} \{ v_k \}) \setminus DTI \}. \]

But this is the projection of the complement of \( DTI' \):
\[ p_A(DOM(\Sigma_{\text{cause}} \{ v_k \}) \setminus DTI). \]

This yields the following lemma which underlies the second check.

**Lemma 7**

Let \( v_k \) be a causal variable, \( p_A \) the projection to the other causal variables, and \( DTI \) a non-empty set of definitely discriminating test inputs. Furthermore, let
\[ DTI' : p_A\left(\text{DOM}(v_k) \setminus DTI \right). \]

If \( DTI' = \emptyset \)
then \( v_k \) is not conditionally or totally irrelevant to \( DTI \).

If \( DTI' \neq \emptyset \)
then \( v_k \) is conditionally irrelevant to \( DTI \)
If \( DTI' = p_A\left(\text{DOM}(v_k) \setminus DTI \right) \)
the \( v_k \) is totally irrelevant to \( DTI \).

Please note that \( DTI' \) 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 \( p_A(DTI) = \text{DOM}(v_k) \)
THEN
IF \( DTI' = \emptyset \)
THEN "WEAKLY IRRELEVANT"
ELSE IF \( DTI' = p_A(DTI) \)
THEN "TOTA LLY IRRELEVANT"
ELSE "IRRELEVANT UNDER DTI'"
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 the system is 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 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 $DI^i$ 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 $DI^i$ is satisfied in the current situation, and if so, they do not have to be touched and an arbitrary assignment of values out of $DI^i$ 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. This is done to investigate whether and to what extent the solution can be applied to testing of software. 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 to test 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


[OCC’M 05] www.occm.de


