

# Avoiding Combinatorial Explosion in Automatic Test Generation: Reasoning about Measurements is the Key

Harald Lange, Ralf Möller, Bernd Neumann<sup>1</sup>

University of Hamburg, Computer Science Department, Laboratory for Artificial Intelligence  
Vogt-Kölln-Str. 30, 22527 Hamburg  
{hlange,moeller,neumann}@informatik.uni-hamburg.de

**Abstract:** The main thesis of this paper is that reasoning about measurements can be used as a basic mechanism for generating test plans for analogical circuits. Motivated by an application scenario, reasoning about measurements incorporates domain knowledge about testing conditions, local behavior of circuit components (fault modes covered by measurements) and the topological structure of the circuit to be tested. With the test generation architecture introduced in this paper, a combinatorial explosion which is problematic in model-based test generation approaches can be avoided.

## 1 Introduction

The automatic generation of test plans for technical systems becomes more and more important, especially since the producer of a technical product can be made liable for any damage that is caused by the product. Test generation and quality control is not only important for new products. Systems that have been repaired or maintained must also be tested again. A test plan is a sequence of tests (or measurements) being used to indicate that the behavior of a system is correct w.r.t. a formal specification. The composition of tests must consider several criteria. First of all, the test plan must be complete in some sense, i.e. if there is a fault in a component, it should be detected by at least one test. In most cases, non-trivial assumptions must be made to guarantee completeness: the fault model completeness assumption [13], the single-fault assumption and the non-intermittency assumption [11]. Thus, at the current state of the art, completeness can be guaranteed only in a weaker sense (but see [3] for extensions to handle multiple faults). Second, tests or measurements are associated with costs and the goal of test plan generation is also to reduce these costs. Third, user interface concerns are also important, i.e. in some domains, tests must be composed in such a way that a human engineer (operator of the test machinery) is adequately supported.

Test plan generation is related to technical diagnosis. The theory about technical diagnosis can provide the basis for test generation because a sequence of measurements is used in a similar way to refute all possible fault models for circuit components. However, instead of determining a faulty component, with test generation systems the correct behavior of all components must be shown. In this paper, we focus on computing confirming test sets and do not consider test sets that identify the reason for a certain fault (tests for identification).

---

1. Work in this project has been done in collaboration with "DTK: Gesellschaft für Technische Kommunikation mbH", Hamburg. We would like to thank U. Haferstroh, A. Josub, M. Orlick and M. Schmidt. Many thanks also to our students A. Kaplunova and H. Paulsen who not only implemented large parts of the interface and the test generator but also contributed many good ideas for the PETS architecture. The project was supported by the Wirtschaftsbehörde of Hamburg.

Because of the enormous commercial importance, the generation of tests for *digital* circuits has been considered in numerous publications. See e.g. the D-Algorithm of Roth [12] and the subsequent work of Fujiwara [5]. Distributed model-based test generation has been proposed for VHDL descriptions of digital circuits [7]. Even in simple circuits, there is always the danger of combinatorial explosion. The problem is that components can usually not be measured in isolation but must be tested in an aggregate which can be very complex.

Other approaches known in VLSI testing like graph-based algorithms for evaluating boolean functions introduced by Bryant (OBDDs: ordered binary decision diagrams, see e.g. [2]) exchange computational complexity with data structure size (exponential storage complexity in the worst case). The idea behind OBDDs is to combine the variables of models for basic components in a recursive bottom-up process until the complete circuit is modeled. The graph structure provides a fast way to determine the value of a single variable given the value of other variables. However, the data structures grow exponentially.

Much less work has been published for the generation of test plans for *analogical* circuits. Modeling the behavior of physical systems in general and qualitative modeling in particular (cf. Struss [15] [16] [17]) is necessary when tests for single constituents are to be generated from first principles. Though, in principle, it has been shown that model abstraction [16] and model-based test generation for single constituents can also be applied to aggregate testing [15], the computational complexity is NP-complete. Furthermore, as Struss emphasizes, the solution introduced in [15] “shifts the burden to the hard task of modeling”.

Although the selection of models has been automated in some domains (e.g. [10], [13]), models are usually defined manually. Modeling also comes at a cost that must not be neglected in practical applications (see [8] for a discussion of different aspects and domain characteristics). In addition, in some domains, the limited observability of physical quantities must be taken into account. Thus, the application domain defines several constraints that should not be neglected for efficient test generation.

The real world application scenario discussed in this paper has the following characteristics and constraints:

- Analogical and digital components must be dealt with.
- Circuits are composed of a large amount of components (1000-5000 per assembly) with high interconnectivity.
- Components cannot be measured in isolation (approx. 10-20 terminals only).
- If components and their models must be adapted or extended, this must be done by domain experts, not experts of the test generation system, i.e. formal models for component behavior should be as simple as possible to reduce training time.
- The actual testing machinery is given in the industrial context, i.e. the set of possible measurements is restricted.
- The structure of the circuits and the derivation of measurements is influenced by the user interface for the operator of the testing machine (for details see below).

In collaboration with our project partner DTK whose employees are experienced in manual test plan generation we developed a new approach to automatic test plan generation for combined analogical and digital circuits. Manual test plan generation takes up to four person months per assembly, and because several hundreds of different assembly types have to be tested, automatic test plan generation saves an enormous amount of money. The system we developed is called PETS (in German: Prüfplan-Erstellung für Technische Systeme). It is now in operation at DTK.

The approach is to some extent coupled to the problems in the domain and is not as general as the work of Struss. In the group of Struss, some experiments on combining model-based test generation techniques with the OBDD approach of Bryant indicate that combinatorial explosion is likely to occur even in small circuits (see Inderst [6]). A first approach of Inderst uses some test data from PETS but, in reasonable time, his system can handle only half of the small circuit presented in Figure 1 (see below).

Our approach is different. The constraints imposed by the application domain provide a lot of insights into how complexity can be dealt with. Inspired by the actions taken by the domain experts, we *model test generation as reasoning about necessary and possible measurements* of circuit elements (and circuit element *aggregates*) rather than reasoning primarily about component behavior. The test generation algorithm directly works on the kind of possible measurements defined by the test environment. Reasoning about measurements can be seen as supplementary to reasoning about component behavior because it relates local component behavior and global system structure.

The paper is structured as follows. In the next section, we discuss the application domain and explain components and structure of the circuits to be tested. The discussion derives a set of optimization criteria for test plans. The third section introduces our approach to solve the test generation problem. After discussing the relations to model-based test generation in the fourth section, the paper concludes with a summary and an outlook.

## 2 PETS: The Application Scenario

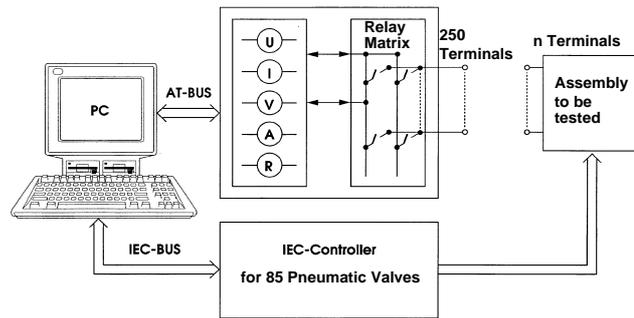
Even today, most railroad systems are controlled by relay circuits because relays are quite reliable. After repair or maintenance, relay assemblies must be tested. An assembly consist of 10 to 40 relays, each of which controls up to fourteen contacts (switches). According to the relay type, contacts are either opened or closed when the relay is switched. Relays can influence each other (coupled relays). In addition, in assemblies there are many types of analogical components: resistors, lamps, relay coils, diodes, capacitors, transformers, etc. Besides source and ground terminals, a relay assembly consists of several other external terminals which can be used for tests and measurements.

### 2.1 The PETS interface

Circuits are described by drawings which are organized into several sheets (connected with “continuation elements” known from flow-charts). For test generation an internal (object-oriented) representation of circuit elements and circuit structure is required. Depending on the element class, every circuit element is characterized by a set of attributes which are shown in the drawings. To support the creation of the internal circuit representation, a small circuit editor (in the spirit of a domain-specific CAD system) has been developed. The interface is also used to visualize test generation results. Figure 1 presents a small sample circuit (called SC for “switch control”).

In the PETS interface, the usual icons of railroad signal technology are used. Contacts are represented by small bars. Contacts can be open (bar on one side of the wire) or closed (bars on both sides of the wire, think of a water pipe which is interrupted by a sheet metal plate). The relay type is indicated by different arrows which also indicate whether the contact is either open or closed when the relay is excited. The coil (or coils) of a relay are separated from the contacts and can possibly be found on different sheets.





**Fig. 2.** Overview of the measurement automaton in the maintenance workshop.

in a car repair workshop, due to the available measurement equipment and training of personnel, only a limited set of measurements (limited w.r.t. to the physically possible measurements) can be carried out. The set of possible measurements covers several fault models for different components. Furthermore, the set of fault models for contacts (to be ruled out by refutation measurements [14]) is rather small (stuck-at-open, stuck-at-closed). These fault models directly correspond to a conduction and a non-conduction measurement, respectively. For analogical element types, there are specialized measurements, too (diode and capacity measurements). Thus, for each circuit element, there exists a set of necessary measurements. The final test plan should include all required measurements for each component that can be tested by external measurements. Every measurement is characterized by a specific relay position. For instance, in order to carry out a resistance measurement for a lamp, a path from two terminals at the outside of the assembly must be found. The path must not contain a shortcut around the elements to be tested. Note that it is not always possible to directly measure a single circuit element because, due to the connection topology, in some circuits every measurement path (in any kind of relay position) also contains other elements. Even worse, for some components, the connection topology of a circuit might not even allow an external measurement at all. For instance, in the lower left part of Figure 1 two contacts are arranged in parallel to two lamp elements, i.e. there is no way to do an external non-conduction measurement for the contacts because it is not possible to set the relays of the circuit in such a way that the shortcut of the contacts by the lamps can be avoided. All that PETS can do in situations like these is to report the circuit elements that cannot be tested.

### 2.3 Optimization Criteria for Test Plan Generation

In the PETS context, the initial relay position is defined by the test automaton (Figure 2). However, the subsequent test states of the circuit are subject to optimization because the number of relay switches should be minimized. Note that in other domains (e.g. vehicle diagnosis [8]), test generation is simplified drastically because the static test states of the system are prescribed.

In general, it is desirable to minimize the number of measurements. But since there might be several elements on a measurement path, a measurement does not necessarily directly reveal a broken element (nor the exact cause of the fault). In order to minimize search time for the electrical engineer, measurement paths should be rather short. Furthermore, for the engineer's user interface (see Figure 2), the sequencing of measurements must correspond to the organization of circuits into sheets. The complete circuit will be tested sheet by sheet and measurements should be local to a single sheet. Thus, in this scenario several conflicting optimization criteria are important:

- minimization of the number of relay switches,
- minimization of the length of measurement paths,

- minimization of the number of measurements,
- minimization of sheet crossings.

However, these criteria are no hard constraints.

## 2.4 Worst-Case Complexity Considerations

Every relay position results in a different circuit topology. Thus, since there are up to 40 relays in a circuit, in principle,  $2^{40}$  different circuits have to be considered. Assuming that there are 10 to 20 terminals which are candidates for two-point measurements in each circuit, it becomes clear that a general model-based test generation system can be used only for small circuits (see the results of Inderst [6]).

Considering the test automaton, we focus on two-point measurements between terminals rather than on local definitions of component behavior. For each type of circuit element, a set of necessary measurements is defined. However, due to circuit topology, several circuit elements must be combined in order to find two-point measurements between terminals. The main idea behind the PETS approach is to reduce complexity by focussing on measurements rather than on component behavior. For example, for a serialization of ten contacts only eleven measurements are necessary (one conduction measurement with all contacts closed and ten non-conduction measurements with one contact open in each measurement). However, in the worst case, a behavioral model has to consider  $2^{10}$  different states (either by simulation or encoded with OBDDs).

In the following chapter we will discuss how test plan generation can be modeled as reasoning about measurement combinations. We will see that the circuit topology itself imposes some sort of global constraints on the complete set of measurements combined in a test plan.

## 3 Reasoning about Measurements

In the PETS architecture, measurements are first-class objects which can be combined to new measurements according to a set of rules. The parameters of different measurements depend on the attributes of the elements being tested. Before the operations on measurements are described and necessary measurements are derived, we introduce several kinds of circuit transformations that are required to implement reasoning about measurements.

### 3.1 Circuit Transformations and Measurement Composition

#### Two-Pin Circuit Elements

The circuit might contain elements that have more than two pins (e.g. transformers). In a first step, these elements are replaced by two-pin elements. For instance, a transformer is replaced by two resistors representing the coils.<sup>2</sup>

#### Topological Connectors: Considering Possible Faults in Structure

The connection topology is another example for a circuit transformation. It should be noted that in the PETS scenario the model for the topology of a circuit is incomplete. The “wires” shown in the drawings are only logical connections. They do not directly

---

2. There may also be a shortcut between the two substitute resistors as an additional fault model for a transformer. These situations are covered by a set of additional shortcut tests which are not discussed in this paper.

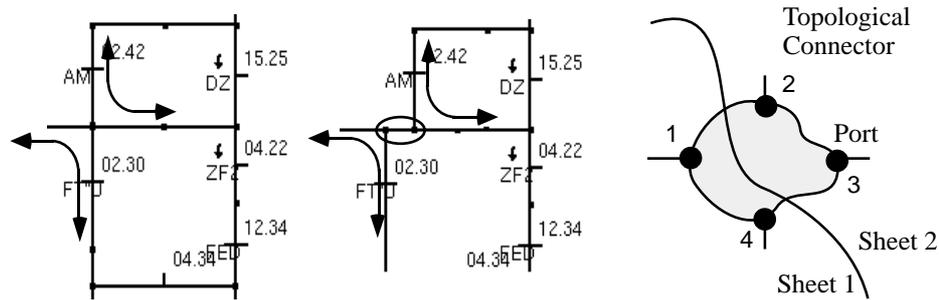


Fig. 3. Motivation of measurements for a topological connector.

correspond to physical wires but describe the connection topology between circuit elements only. In the left part of Figure 3, a small clipping of a circuit is shown. Let us assume, the four-point connector below the contact of the relay  $AM$  is “covered” by two measurements (indicated by arrows). It is easy to see that an additional measurement would be necessary to exclude broken wires if the actual physical wiring were defined as presented in the middle of Figure 3. Since models of the concrete physical wiring of circuit elements are definitely not available, the situation shown in Figure 3 cannot be ruled out and some kind of worst case reasoning must be applied to cope with *faults in structure* (see also [1]).

So-called *topological connectors* are abstractions from the “wires” between circuit elements. The circuit is transformed into another representation where topological connectors are abstract nodes between two-pin circuit elements. In order to correctly handle the worst case, (i) all ports must be covered by the set of measurements “crossing” the topological connector, and (ii) the cover must be connected. Thus, besides the two measurements in the left part of Figure 3, an additional measurement is required (for example from Port 1 to Port 3). Other combinations are also possible. A topological connector might cross the boundaries of sheets (see Figure 3 for an example). Sheet information is important because, in general, measurements across sheet boundaries should be avoided to support a more ergonomic interaction style for the operator interface. Thus, although the set of measurements  $\{1-2, 2-4, 3-4\}$  satisfies the “ports-must-be-connected” constraint, it is less optimal considering the above-mentioned optimization criteria. The final test plan must include enough measurements such that the ports of every topological connector are connected. Ports of a topological connector might be terminals and the “ports-must-be-connected” constraint ensures that all terminals are tested in the final test plan.

### Serial-Parallel-Analysis

A reduction of circuit size can be achieved by the well-known serial-parallel analysis (sp-analysis) which has also been successfully applied to problems in qualitative diagnosis [8]. The compound elements (aggregates) derived step by step are also associated with measurements. However, not all combinations of measurements are allowed and some high-level combinations might rule out measurement candidates which were possible at lower level aggregates.

Table 1 and Table 2 define the serial and parallel combinations of measurements, respectively. Each entry in the table defines the type of the compound measurement and characterizes the basic circuit elements that can be marked as measured (row = only the elements of the row aggregate are measured, col = only the elements of the column

aggregate are measured, bth = the elements of both indices, the row and and the column aggregate, can be measured, non = no element can be marked as measured in the compound circuit). For instance, when a conduction measurement (column) is combined with a non-conduction measurement (row) in a serial combination, the resulting measurement will be a non-conduction measurement but only the circuit elements of the row aggregate can be marked as measured. The measurement status “exp” means that currently, the parameters of measurements cannot be determined by PETS and have to be determined “experimentally” by model-based simulation (either quantitative or qualitative, see below).

Measurements are also associated with a direction. Combining, for example, diode measurements with different directions is not valid. The complete composition tables for the direction of aggregate measurements are given in Table 3 (serial combinations) and Table 4 (parallel combinations).

**Table 1.** Serial combination of measurement directions (b-e = begin-to-end, e-b = end-to-begin, b-n = both necessary, b-p = both-possible, exp = experimental, n-p = none-possible). Measurements with direction n-p will no longer be considered.

<b>serial</b>	<b>b-e</b>	<b>e-b</b>	<b>b-n</b>	<b>b-p</b>	<b>n-p</b>	<b>exp</b>
<b>b-e</b>	b-e	n-p	exp	b-e	n-p	exp
<b>e-b</b>	n-p	e-b	exp	e-b	n-p	exp
<b>b-n</b>	exp	exp	b-n	b-n	n-p	exp
<b>b-p</b>	b-e	e-b	b-n	b-p	n-p	exp
<b>n-p</b>	n-p	n-p	n-p	n-p	n-p	n-p
<b>exp</b>	exp	exp	exp	exp	n-p	exp

**Table 2.** Parallel combination of measurement directions.

<b>par</b>	<b>b-e</b>	<b>e-b</b>	<b>b-n</b>	<b>b-p</b>	<b>n-p</b>	<b>exp</b>
<b>b-e</b>	b-e	b-n	b-n	exp	b-e	exp
<b>e-b</b>	b-n	e-b	b-n	exp	e-b	exp
<b>b-n</b>	b-n	b-n	b-n	exp	b-n	exp
<b>b-p</b>	exp	exp	exp	b-p	b-p	exp
<b>n-p</b>	b-e	e-b	b-n	b-p	n-p	exp
<b>exp</b>	exp	exp	exp	exp	exp	exp

**Table 3.** Serial combination of measurements (con = conduction, nco = non-conduction, res = resistance, cap = capacity, dio = diode, exp = experimental measurement).

<b>serial</b>	<b>con</b>	<b>nco</b>	<b>res</b>	<b>cap</b>	<b>dio</b>	<b>exp</b>
<b>con</b>	con/bth	nco/col	res/bth	cap/bth	dio/bth	exp/bth
<b>nco</b>	nco/row	nco/non	nco/row	nco/row	nco/row	nco/row
<b>res</b>	res/bth	nco/col	res/bth	exp/bth	exp/bth	exp/bth
<b>cap</b>	cap/bth	nco/col	exp/bth	cap/bth	exp/bth	exp/bth
<b>dio</b>	dio/bth	nco/col	exp/bth	exp/bth	exp/bth	exp/bth
<b>exp</b>	exp/bth	nco/col	exp/bth	exp/bth	exp/bth	exp/bth

**Table 4.** Parallel combination of measurements.

<b>par</b>	<b>con</b>	<b>nco</b>	<b>res</b>	<b>cap</b>	<b>dio</b>	<b>exp</b>
<b>con</b>	con/non	con/row	con/row	con/row	con/row	con/row
<b>nco</b>	con/col	nco/bth	res/bth	cap/bth	dio/bth	exp/bth
<b>res</b>	con/col	res/bth	res/bth	exp/bth	exp/bth	exp/bth
<b>cap</b>	con/col	cap/bth	exp/bth	cap/bth	exp/bth	exp/bth
<b>dio</b>	con/col	dio/bth	exp/bth	exp/bth	exp/bth	exp/bth
<b>exp</b>	con/col	exp/bth	exp/bth	exp/bth	exp/bth	exp/bth

Sp-Analysis is also done for contact constellations. For instance, a parallel combination of contacts can be used for a non-conduction measurement (of both contacts, see Table 2). A conduction measurement cannot be performed on two parallel contacts. However, because sp-aggregation can proceed at a higher level, the resulting measurement will still be a conduction measurement but none of the contacts is marked as measured (see the corresponding entry in Table 2). Dual results can be found for serial combinations of non-conduction measurements (see Table 1).

After sp-analysis, more abstract compound circuit elements are defined. Each compound element defines a set of necessary and possible measurements (with attributes for directions and required relay positions). As has been mentioned before, a specific aggregate of ten serial contacts requires only eleven measurements to refute all fault models. Besides the derivation of possible and necessary measurements, the sp-analysis ensures that larger nets can be handled.

For each of the compound elements, *paths* from the external “pins” of a compound to terminals (in different topological connectors) must be found. In order to find these paths, the circuit is treated as a labyrinth.

### 3.2 Finding Measurement Paths: Traversing a Labyrinth

Starting from the initial state of the circuit (with all relays being in a predefined starting state), PETS defines a search space of possible relay switches to generate serial measurement paths. The current version of PETS does not search for relay states where parallel configurations of compound elements can be found (e.g. for non-conduction measurements).

#### Finding Half-Paths

Considering each compound element of the transformed circuit, paths to topological connectors with terminals must be found. For each outgoing pin of a compound element, PETS traverses the corresponding circuit graph of topological connectors and other compound elements and in a breadth-first manner because shorter paths should be generated first. The extension of a path is possible when the measurements associated with the new head (either a topological connector or a compound element) are compatible to the measurement of the path (serial composition, see Table 1 and Table 3). PETS treats all possible combinations of measurements as different path candidates (so-called half-paths). All path candidates are expanded until a topological connector with a terminal is found. Sometimes, the extension of a path will only be possible when a contact is closed, i.e. a relay must be switched. However, the labyrinth is tricky and switching a relay is not possible in all situations because of the interdependencies of the contacts of a relay.

PETS must ensure that switching a relay for path extension does not open a contact on the path itself! Therefore, with every half-path candidate a relay position vector is associated. For each relay, the relay position vector defines the required state (either excited or non-excited) or contains a wildcard when both states are still possible. Note that in some circuits, switching a relay might also switch another relay because they can be coupled.

### Combining Half-Paths and the Associated Measurements

After the half paths have been generated, they must be combined. Half-paths can be combined when the intersection of the elements is empty and the associated relay state vectors as well as the associated measurements are compatible. Relay state vectors are compatible when the states of the corresponding relays are either equal or at least one of the states is undefined (wildcard). Measurements are combined according to Table 1 and Table 3. Measurements are compatible when at least one basic circuit element can still be measured.

### Avoiding Shortcuts: Blocking Cycles

The serial combination of half-paths is only a necessary condition for combining two half-paths. In addition, there must exist a relay position such that the circuit elements on the path are not shorted by side-paths connected to the main path. Thus, for generating a full path, the octopus arms starting from the “main” path (see Figure 2 for an example) must be followed and shortcuts must be detected. If there is a shortcut, a contact on the shortcut path must be opened, i.e. another relay must be switched and PETS must find a contact whose relay position is not yet bound by the relay state vector.<sup>3</sup>

A (full) measurement path is a sequence of topological connectors and compound elements. It starts and ends with a different topological connector. The space of measure-

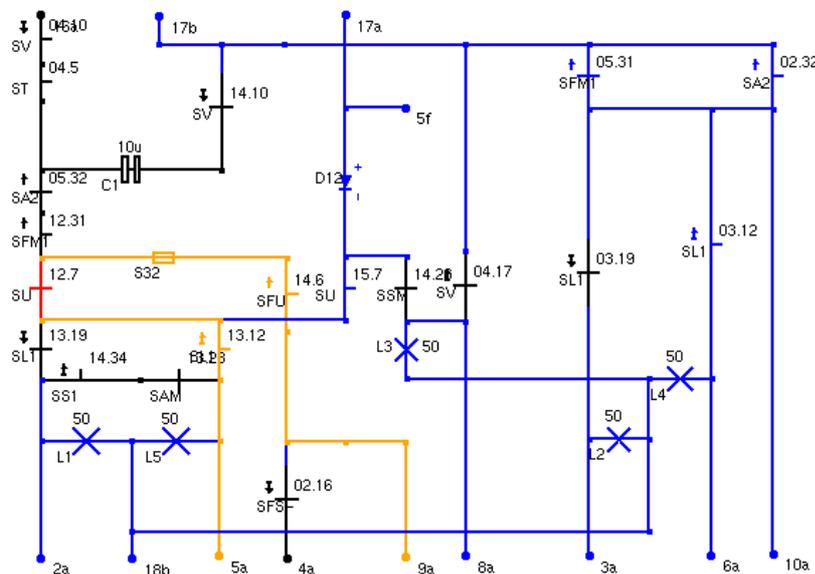


Fig. 4. Another measurement path for a non-conduction measurement (see also Figure 2).

3. The current implementation takes the first contact that is found while following a side-path and whose relay can be switched. This is not always the best solution because relay switches are not minimized. PETS sacrifices optimality for computation time in this respect.

ment path candidates for compound elements (aggregates after sp-analysis) is generated by delayed evaluation programming techniques, i.e. only those candidates which are really needed are actually generated. In the next section, we will discuss how PETS combines the candidates to a complete test plan.

### 3.3 Generating the Test Plan: A Two-Stage Approach

The task of PETS is to find a sequence of measurements (a test plan) such that all elements are tested (necessary measurements), all constraints are fulfilled (e.g. topological connector constraints) and the above-mentioned optimality criteria are satisfied. Test plan generation is realized by a two-stage process indicated by the following procedure:

```

procedure test-plan-generation(circuit)
  circuit-state := initial-circuit-state(circuit);
  while unmeasured-elements-existing(circuit)
    and measurements-possible(circuit) do
    measurements := applicable-measurements(circuit-state);
    while not(emptylist(measurements)) do
      m := first(sort(measurements, g1));
      measurements := rest(measurements);
      mark-elements-as-measured(elements(m), circuit);
      add-to-test-plan-of-circuit(circuit, m);
    end
    circuit-state := next-circuit-state(circuit, g2);
  end
end

```

The initial relay state (i.e. the corresponding circuit state) is defined by the measurement machinery. When there are still circuit elements for which measurements are required, the list of measurements that are applicable in the chosen circuit state are determined. Measurement candidates have been generated by the process described in the previous section. A measurement is applicable in a certain circuit state when the associated relay state vectors are compatible. i.e. when the associated relay state vector either requires the same relay state as found in the current circuit state or contains a wildcard. PETS orders the set of possible measurements with the cost function  $g_1(m) = w_1 \text{elements}(m) - w_2 \Delta b(m)$  where  $\text{elements}(m)$  is the number of basic circuit elements that are tested with measurement  $m$ , and  $\Delta b(m)$  is the number of new elements that can be tested with  $m$ . The idea is (i) to generate short measurement paths and (ii) to reduce the number of measurements.

The measurement with the lowest cost is added to the test plan. PETS continues with the previous step and computes a new ordering relation. When in a certain relay state no more measurements are available or no more elements can be tested by subsequent measurements, the next relay position is determined. PETS considers every possible successor state and takes the one that minimizes  $g_2(s) = w_3 \Delta R(s) - w_4 \Delta B(s)$  where  $\Delta R(s)$  returns the number of relay switches in comparison to the previous circuit state and  $\Delta B(s)$  defines the number of new circuit elements that can be tested in the new relay state. The weights of the local optimization criteria can be changed by the PETS user (the interactive interface provides sliders to determine reasonable values). They allow a rough consideration of the optimality criteria. A global optimization (over all circuit states and two-point measurements) would be computationally much more expensive with only limited value in practice.

### 3.4 Evaluation and Performance Considerations

As discussed before, by switching relays PETS only generates serial measurements path. There is no attempt to look for circuit states where parallel configurations of compound elements are created. Parallel compounds are useful for non-conduction measurements. However, in all circuits that have been used as test circuits for PETS, the serial combination of compounds on a measurement path was sufficient to compute measurements for all components that can be tested with the prescribed test automaton.

Experiences with PETS have shown that in some rare cases, for aggregation not only the direction of measurements are important but also the attribute values of circuit elements. For instance, due to measurement inaccuracies, a serial combination of a 100 Ohm resistor R1 and a 10kOhm resistor R2 will hardly reveal resistance deviations of R1. In this case, the serial combination should be allowed for R2 (there might be no other way to measure it), but for R1 an additional measurement path should be found. The generation of a test plan for the small circuit in Figure 1 with 14 relays and 20 contacts generates 33 measurements and takes approximately 30 seconds on a SPARC 10. All elements are tested except two contacts parallel to two lamps (see the discussion above). Other nets with up to 40 relays and 500 contacts and other elements take up to one hour of computation time.

## 4 Conclusion

In this paper, we have introduced a methodology for generating test plans that avoids combinatorial explosion. The key idea is to generate a test plan by reasoning about measurements. For reasoning about measurements PETS represents knowledge about measurement combinations (see the measurement combination tables above) and knowledge about testing the topological circuit structure (see the treatment of topological connectors). The combinatorial complexity can be reduced because there are far less measurements possible than system states (defined by local models of component behavior). In order to derive the complete set of measurements for a test plan, only a fragment of the system states has to be considered. However, a shortcoming of the current version of PETS is that parameters for some measurements must still be manually derived (so-called experimental measurements). This is the place where traditional model-based testing techniques could be integrated. For local element constellations on a measurement path, a more elaborate reasoning strategy based on models about component behavior can indeed be tractable. Thus, both approaches can complement each other, and it would be very interesting to combine PETS with a model-based simulation component. New approaches to model-based diagnosis reduce computation cost by interleaving model-based inference phases with probing (or measurement) phases (de Kleer and Raiman [4]). The key idea is to gain new information about components at some state of model-based inferencing to reduce the set of diagnosis candidates. Thus, computation time is taken into account as an optimality criterion. However, in the PETS scenario, concrete results of measurements are not available during test plan computation.

At the end, it should be noted that multiple faults are no problem in the PETS scenario as long as they do not cancel each other out. When a measurement reveals that one element on a measurement path must be faulty, the engineer will manually look for the culprit and, after the repair, the whole assembly will be tested again. But, unfortunately, one can easily construct circuits where a fault will remain *undetected* by a test plan because of interfering multiple faults. Further work is required to extend test plans with additional measurements to rule out these multiple fault problems.

## References

1. Böttcher, C., *No Faults in Structure? – How to Diagnose Hidden Interactions*, in: Proc. IJCAI'95, Morgan Kaufmann Publ., 1995, pp. 1728-1734.
2. Bryant, R.E., *On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplications*, in: IEEE Transactions on Computers, Vol. 40, No. 2, February 1991, pp. 205-213.
3. De Kleer, J., Williams, B.C., *Diagnosing Multiple Faults*, in: Readings in Model-Based Diagnosis, Hamscher, W., Console, L., De Kleer, J. (Eds.), Morgan Kaufmann Publ., 1992, pp. 100-117.
4. De Kleer, J., Raiman, O., *Trading off the costs of inference vs. probing in diagnosis*, in: Proc. IJCAI'95, Morgan Kaufmann Publ. 1995, pp. 1736-1741.
5. Fujiwara, H., Shimono, T., *On the Acceleration of Test Generation Algorithms*, in: Proc. IEEE Transactions on Computers, pp., 1137-1144, 1983.
6. Inderst, R., *Automatische Testgenerierung auf der Basis einer qualitativen Modellierung physikalischer Systeme*, in German, Diploma Thesis, University of Munich, Feb. 1995.
7. Khaani, F., Navabi, Z., *VHDL Structural Models for the Implementation of Path Sensitation Test Generation*, [http://www.ece.neu.edu/info/vhdl/Postscript\\_papers/](http://www.ece.neu.edu/info/vhdl/Postscript_papers/).
8. Mauss, J., Neumann, B., *Diagnosis by Algebraic Modeling and Fault-Tree Induction*, in: Proc. DX-95, 6th Int. Workshop on Principles of Diagnosis, Goslar, Germany, pp. 73-80, 1995.
9. Mauss, J., Neumann, B., *How to Guide Qualitative Reasoning about Electrical Circuits by Series-Parallel Trees*, in: Proc. QR'96, 10th Int. Workshop on Qualitative Reasoning, Stanford, CA, 1996.
10. Nayak, P.P., *Automated Modeling of Physical Systems*, Lecture Notes in Artificial Intelligence 1003, Springer 1995.
11. Raiman, O., de Kleer, J., Sarasvat, V., Shirley, M., *Characterizing Non-Intermittent Faults*, in: Proc. AAAI'91, Morgan Kaufmann Publ., 1991, pp. 849-854.
12. Roth, G.P. *Computer Logic, Testing, and Verification*, Computer Science Press, 1980.
13. Struss, P., *What's in SD? Towards a Theory of Modeling for Diagnosis*, in: Readings in Model-Based Diagnosis, Hamscher, W., Console, L., De Kleer, J. (Eds.), Morgan Kaufmann Publ., 1992, pp. 419-449.
14. Struss, P., Dressler, O., *Physical Negation - Integrating Fault Models into the General Diagnostic Engine*, in: Proc. IJCAI 89, Morgan Kaufmann Publ., 1989, pp. 1318-1323.
15. Struss, P., *Testing Physical Systems*, in: Proc. AAAI'94, Morgan Kaufmann Publ., pp. 251-256.
16. Struss, P., *Model Abstraction for Testing of Physical Systems*, in: Proc. QR'94, 8th Int. Workshop on Qualitative Reasoning, Nara, Japan, 1994.
17. Struss, P., Malik, A., Sachenbacher, M., *Qualitative Modeling is the Key*, in: Proc. DX-95, 6th Int. Workshop on Principles of Diagnosis, Goslar, Germany, pp. 73-80, 1995.