Testing for Discrimination of Diagnoses

Peter Struss
Technical University of Munich
Computer Science Dept.
Orleansstr. 34, D-81667 Munich, Germany
struss@informatik.tu-muenchen.de

Abstract

In order to discriminate among the diagnostic candidates which are hypothesized by a diagnosis system at a certain stage may require the application of tests, i.e. causal inputs to the device to be diagnosed that promise to reveal distinctions between the candidates through observable features. We develop a theory and an algorithm for generating and applying tests in diagnosis based on a behavior representation by relations. The approach is general and handles also continuous systems (in theory and, through model abstraction, also in practice), thus going beyond the area of digital circuits. We also present a test selection strategy based on a minimum entropy criterion which, as a spin-off, comprises a generalization of the widely used probe selection strategy in consistency-based diagnosis.

1 INTRODUCTION

An important task in diagnosis, particularly in model-based diagnosis, is to discriminate among diagnostic candidates, i.e. fault hypotheses generated at some stage of the diagnostic process. Model-based diagnostic systems can tackle this task in two different ways:

- Use more *knowledge* that is present in the *model*, but has not yet been exploited (e.g. expand the structural focus, activate more powerful models, or use fault models).
- Obtain more information about the actual state and physical condition of the device, i.e. perform and exploit more measurements and observations.

The general diagnostic engine (GDE) comprises an approach to the latter, namely a minimum-entropy-based strategy for probe selection (de Kleer & Williams 1986): by a one-step-lookahead the variable is determined whose observation promises the highest gain in eliminating a subset of the diagnostic candidates *in the current state* of the system to be diagnosed (i.e. under a certain input).

In some cases this is too limited, because measurements may be impossible or expensive, or the actual state of the device, as determined by the current inputs, does not allow further discrimination of the candidates. In such cases, testing may help, i.e. shifting the device by an appropriate input into a state that allows to obtain useful information. In our case, observations are useful, if they support discrimination of the existing diagnostic candidates which again means in consistency-based diagnosis, which we follow: observations that are likely or guaranteed to rule out at least some of the candidates.

Each diagnostic candidate represents a hypothesis about the current physical condition of the device, specifying, in particular, which component(s) may be broken and, possibly, in which way they malfunction. In model-based diagnosis, such a specification also defines a description of the respective device behavior. This may be a partial decription, especially if there are no fault models available or not used at this stage. Discrimination among the candidates can only succeed if there are (observable) distinctions between the respective behaviors, and testing for discrimination is the task of generating and applying inputs to the device that reveal these distinctions. We present a principled solution to this problem that builds upon our recent work on relational and multiple modeling (Struss 1992) and on testing (Struss 1994, 1994a, 1994b). It is general in that its theoretical foundation covers arbitrary domains of variables, as opposed to specific solutions in the area of digital circuits ((Genesereth 1984), (Meerwijk & Preist 1992)). In (Struss 1994, 1994a, 1994b) we dealt with the task of *confirming* (or disconfirming) a particular behavior (usually the correct one, e.g. after the device has been manufactured or assembled). Here we have the goal to *identify* the current behavior, or, more precisely, to determine which behavior out of a given set (specified by the candidates) is actually present (if any).

In the following section, we will specify formally the goal of our theory as mainly captured by the concept of a discriminating input set. An illustrative example (section 3) is then used to provide the intuition for the solution to generating such input sets based on analyzing the distinctions between model relations, as presented in section 4. We discuss how to obtain model relations for diagnostic candidates (section 5) and how to exploit model

abstraction to cope with complexity (section 6). Section 7 shows the application of the generated sets of test inputs is discussed. Finally, we develop a probability-based test selection strategy and demonstrate that GDE's probe selection forms a specialized instance of this strategy.

2 THE TASK

The situation considered is the following: at a certain stage, the diagnosis sytem has generated a set of possible diagnoses $\Delta = \{\delta_i\}$. In theory, we only require Δ to be finite, but in practice it should be a small set, for instance the most likely 5 or 10 candidates. Each diagnosis δ_i associates a behavior mode $m_{k_j}(C_j)$ to each constituent C_j of the device:

$$\delta_i = \{m_{k_j}(C_j)\}.$$

For most C_j, the mode will be $OK(C_j)$, for a few, it will be a particular failure mode, $Fault_{k_j}(C_j)$, or some unspecified $\neg OK(C_j)$. Assigning a behavior characterization to each constituent means also to characterize the behavior of the entire device, which is constituted by the local behaviors interacting according to the device stucture, although possibly in an incomplete way (in case of using $\neg$OK or other incomplete specifications of (classes of) faults). This way, a set of behaviors of the entire device is given:

$$BEHVS_\Delta = \{B_i\}.$$

where B_i is obtained from δ_i. The behaviors B_i are different for different δ_i (although the distinctions are not necessarily observable), and the behaviors are exclusive:

$$i \neq j \Rightarrow (B_i \Rightarrow \neg B_j).$$

It is believed that the actually correct diagnosis is contained in Δ which translates into

$$B1 \vee B2 \vee \ldots \vee B_{nbehvs},$$

where n_{behvs} is the cardinality of $BEHVS_\Delta$. This can be regarded as a working hypothesis which may have to be retracted during or after testing.

The ultimate goal of the discrimination task is to determine the actual behavior $B_i \in BEHVS_\Delta$ (or refute the working hypothesis), and along the way all other behaviors have to be ruled out. So, informally, we are looking for a set of inputs to the system with the property that the resulting set of observations entails the actual behavior, if it is in $BEHVS_\Delta$, or reveals that this is not the case.

Summarizing briefly some concepts from (Struss 1992, 1994, 1994a, 1994b), we formalize the concept of a discriminating set of test inputs as follows: We assume that behavior (of single constituents as well as the entire device) can be sufficiently characterized by a vector of variables

$$\underline{v} = (v_1, \ldots, v_k)$$

with a domain

$$DOM(\underline{v}) = DOM(v_1) \times \ldots \times DOM(v_k)$$

(which may vary in multiple modeling as in (Struss 1992)).

Only a subset of the variables may be observable, and a subset of the observables can be manipulated to influence the device in testing ("inputs" or causes), which gives us the following inclusions

$$CAUSE(\underline{v}) \subset OBS(\underline{v}) \subseteq VARS(\underline{v}) = \{v_i\}$$

the respective vectors $\underline{v}_{cause}$ and $\underline{v}_{obs}$, and projections

$$\begin{aligned} p_{obs}: &\quad DOM(\underline{v}) \rightarrow DOM(\underline{v}_{obs}) \\ p_{cause}: &\quad DOM(\underline{v}) \rightarrow DOM(\underline{v}_{cause}) \\ p'_{cause}: &\quad DOM(\underline{v}_{obs}) \rightarrow DOM(\underline{v}_{cause}) \end{aligned}$$

If $V = \{vi\} \subset DOM(\underline{v})$ is a set of value tuples, then φ_v denotes the proposition that all vi hold in one of the test situations (and, hence, can, in principle, be observed):

$$\varphi_v \equiv \forall \underline{v}_i \in V \; \exists s_i \in SIT \; \underline{v}(s_i) = \underline{v}_i$$

where SIT is the set of real (physically possible) situations, and $\underline{v}(s_i) = \underline{v}_i$ means that $\underline{v}_i$ is a value of $\underline{v}$ in situation s_i. Finally, we use the notion of a *hitting set*. A set $S = \{s_i\}$ is called a hitting set of a set of sets $\{T_i\}$ iff it contains an element of each T_i:

$$\forall T_i \; \exists s_i \in S \; s_i \in T_i.$$

Test inputs are then subsets of $DOM(\underline{v}_{cause})$, and what we expect a discriminating input set to do is the following: we may pick one input from each test input of the set (thus selecting a hitting set) and apply it to the device. The resulting set of observations will tell us the actual behavior out of $BEHVS_\Delta$. This is stated by the following definition:

DEFINITION 2.1 (DISCRIMINATING INPUT SET)

A test input is a non-empty relation on $DOM(\underline{v}_{cause})$:

$$TI_i \subseteq DOM(\underline{v}_{cause}).$$

A set of test inputs $\{TI_i\}$ is discriminating for a set of behaviors BEHVS iff for all sets

$$V = \{\underline{v}_i\} \subseteq DOM(\underline{v}_{obs})$$

whose set of causes $\{p'_{cause}(\underline{v}_i)\}$ forms a hitting set of $\{TI_i\}$, observation of V entails a behavior $B_i \in BEHVS$: There exists $B_i \in BEHVS$ such that

$$\varphi_v \vdash B_i$$

In other words, φ_v refutes (at least) all but one behavior (and possibly all of them, thus creating an inconsistency with the assumption that the actual behavior is contained in BEHVS).

To motivate and illustrate our solution (rather than to present a serious application), we discuss the example already used in (Struss 1994, 1994a, 1994b), a thyristor.

3 AN ILLUSTRATIVE EXAMPLE

A thyristor is a semi-conductor with anode, A, cathode, C, and gate, G, that operates as a (directed) switch: it works in two states, either conducting current in a specified direction with almost zero resistance (exaggerated by the upper line of the simplified characteristic curve in Fig. 3.1a), or blocking current like a resistor with almost

infinite resistance (the horizontal line). The transition from the OFF state to ON is controlled by the gate; if it receives a pulse the thyristor "fires", provided the voltage drop exceeds a threshold, V_{Th}. There is a second way to fire a thyristor (which is normally avoided, but may occur in certain circuits and situations), namely if the voltage drop exceeds the breakover voltage, V_{Bo} as is indicated by the characteristic in Fig. 3.1a. The annotation with 1 and 0 indicates the presence and absence of a gate pulse. So, for instance, for $\Delta V > V_{Bo}$ the thyristor is ON (i>0), no matter whether or not it receives a gate pulse and, hence, the annotation with 0 and 1. In contrast, the section $V_{Th} < \Delta V < V_{Bo}$, i>0 is annotated with 1, because a gate pulse is required for firing. This representation is based on the assumption that switching happens instantaneously (turn-on time and spreading time 0).

Assume we would like to determine whether the thyristor works properly and, if not, which fault is present.

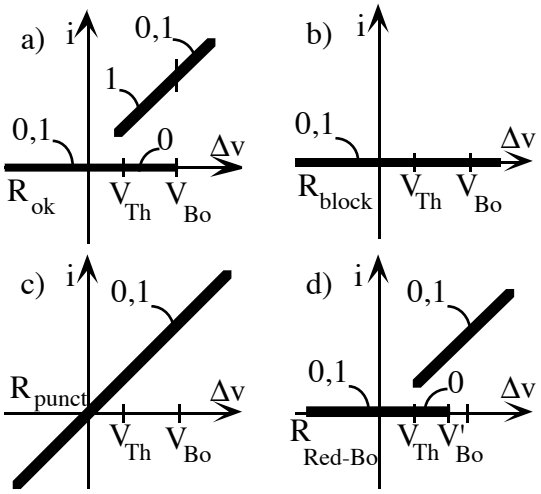

Figure 3.1 The Characteristics of the Behaviors of a Thyristor: a) Correct b) Blocking c) Punctured d) with a Reduced Breakover Voltage

What are the possible faults of a thyristor? A thyristor may be *punctured*, i.e. acting like a wire, or *blocking* like an open switch. A third kind of fault may be due to the fact that the actual *breakover voltage is less than the nominal one*, with the result that the thyristor fires at a voltage drop well below V_{Bo} without a gate pulse. With V'_{Bo} we denote the lowest tolerable actual breakover voltage (or the highest one which is considered to characterize a faulty behavior). Fig. 3.1 shows the (idealized) characteristics of these behaviors in comparison to the correct behavior.

How to discriminate among the behaviors? Applying a voltage between V_{Th} and V_{Bo} together with a gate pulse would lead to an ON-state for all behaviors except the blocking mode. A negative voltage drop would distinguish the punctured thyristor from the rest. What remains is to discriminate between the reduced break-over voltage and correct behavior. This can be done by applying a voltage between V'_{Bo} and V_{Bo} without a gate pulse (which would also be suited to distinguish the blocking fault mode from the other two faults). After performing these tests, we know in any case which of the four behaviors is present.

4 GENERATING DISCRIMINATING INPUT SETS

It is then quite obvious what we have to do in order to find discriminating input sets: they have to uncover the distinctions between any two behaviors in $BEHVS_{\Delta}$. Hence, we can determine the distinction for each pair of behaviors and then look for the inputs that would reveal the distinction. Finally, we can take into account that some inputs may be used for several pairs.

The normal representation of behaviors of physical systems (even for digital circuits) is by relations on the set of variables

$$R \subset DOM(\underline{v})$$

describing the restrictions the behaviors impose on the possible value tuples. The behavior model M(R) defined by R denotes the statement that all value tuples that might occur in a real physical situation are contained in R. We do not require, though, that all tuples in R can actually be realized, because we do not need this restriction for refuting behavior models and behaviors: If M(R) is a proper model of behavior B, this means

$$B \Rightarrow M(R),$$

and observation (or inference) of any value tuple outside R refutes M(R) and, thus, B:

$$\underline{v}_0 \notin R \wedge \underline{v}(s)=\underline{v}_0 \Rightarrow \neg M(R) \Rightarrow \neg B.$$

The behavior representation by relations, i.e. sets, makes it conceptually simple to represent behavior distinctions, namely by *set differences*: if behaviors B_i, B_j have models $M(R_i)$, $M(R_j)$ then their distinction is covered by

$$R_i \setminus R_j \cup R_j \setminus R_i.$$

Since we are interested in the *observable* distinctions only, we have to take the projection to observables:

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

and the respective inputs are obtained by the projection to causes:

$$p'_{cause}(p_{obs}(R_i) \setminus p_{obs}(R_j) \cup p_{obs}(R_j) \setminus p_{obs}(R_i)).$$

However, we must note that the inputs taken from this set *may* lead to different observations under the two behaviors, but they are not guaranteed to. If we use non-deterministic models[1] (and each model reflecting imprecision of measurements and/or disturbances can be considered as such), then it cannot be excluded, that an input chosen as above may result in the same observation

1. A non-deterministic model allows more than one (observable) value tuple for one input:

$$\exists \underline{v}_0, \underline{v}'_0 \in R \quad \underline{v}_0 \neq \underline{v}'_0 \ \wedge \ p_{cause}(\underline{v}_0) = p_{cause}(\underline{v}'_0)$$

for both behaviors, i.e. hit the intersection of the observable parts of the relations. To avoid this, we have to exclude inputs from the causal projection of this intersection:

$$DOM(\underline{v}_{cause}) \setminus p'_{cause}(p_{obs}(R_i) \cap p_{obs}(R_j)).$$

This set contains only inputs that create responses of the device that always differ for the two behaviors, i.e. they will definitely discriminate. If these sets, for different pairs of behaviors, have a non-empty intersection, we obtain inputs supporting several disctinctions. Therefore, we call a set of sets $\{T_k\}$ a hitting set of sets of another set of sets $\{S_i\}$ if it contains non-empty subsets of all S_i:

$$\forall S_i \; \exists T_k \; \emptyset \neq T_k \subseteq S_i \square$$

Now, we can formulate the basic theorem for the generation of discriminating input sets.

THEOREM 4.1

Let $\{R_i \mid R_i \subset DOM(v)\}$

be a set of modeling relations for $BEHVS_\Delta$*:*

$$\forall B_j \in BEHVS_\Delta \; \exists R_j \; B_j \Rightarrow M(R_j).$$

and

$$DI_{ij} := DOM(\underline{v}_{cause}) \setminus (p'_{cause}(p_{obs}(R_i) \cap p_{obs}(R_j)))$$

If a set of test inputs $\{TI_k\}$ *is a hitting set of sets of* $\{DI_{ij} \mid i<j \leq n_{behvs}\}$*, then it is a discriminating input set for* $BEHVS_\Delta$*.*

According to Theorem 4.1 we have to construct a hitting set of sets of $\{DI_{ij} \mid i<j \leq n_{behvs}\}$. Of course, we could take the set $\{DI_{ij}\}$ itself, but this would ignore the chance of obtaining a smaller set of test inputs which cover more than one behavior distinction. The algorithm shown in Fig. 4.1 attempts to reduce the set by intersecting the DI_{ij} if possible, without guaranteeing a minimal set of tests.

We present a version that can work incrementally: it is assumed, that a set of discriminating input sets, TI-SET, exists for a set OLD-BEHVS containing m behaviors. A set NEW-BEHVS (indexed m+1 through n_{behvs}) is added, and its elements have to be discriminated from the old behaviors and from each other. In the initial step or in the non-incremental use, we invoke the procedure with OLD-BEHVS = Ø and TI-SET = Ø.

The algorithm considers all necessary pairs of model relations R(I), R(J) and, in step (1), computes the set DI of inputs that deterministically create different observations for them according to Theorem 4.1. The only difference is that, instead of using the whole space of causes, $DOM(\underline{v}_{cause})$, inputs are restricted to some set of *admissible* causes, R_{adm}, which reflect the fact, that some inputs may be dangerous or not advisable probably under consideration of the current fault hypotheses. For instance, triggering the thyristor with $\Delta V > V_{Bo}$ is usually not to be recommended. If this set DI is empty, there is no input that deterministically distinguishes between the two relations, which is reported in step (2). In this case, requirements are weakened and the set of those inputs are constructed, that

ALGORITHM GENERATE-NEW-TEST-INPUT

```
OLD-BEHVS (1…M)
NEW-BEHVS (M+1…N-BEHVS)
R (1…N-BEHVS)
FOR I FROM M+1 TO N-BEHVS DO
  FOR J FROM 1 TO I – 1 DO
(1) DI = R_adm \ p'_cause(p_obs(R(I)) ∩ p_obs(R(J)))
(2) IF DI = Ø
      THEN "No (adm.) discriminating input for" I,J
(3)   DI = R_adm ∩ p'_cause(p_obs(R(I)) \ p_obs(R(J)) ∪
                            p_obs(R(J)) \ p_obs(R(I)))
      IF DI = Ø
      THEN "No (adm.) observable discrimina-
                            ting test for" I,J
           GOTO .NEXT
    Select TI from TI-SET WITH DI ∩ TI ≠Ø
    IF TI exists
(4)   THEN TI = TI ∩ DI
           Append (I,J) TO D-LIST(TI)
(5)   ELSE Append DI TO TI-SET
           D-LIST(DI) = ((I,J))
    .NEXT
  END FOR
END FOR
RETURN TI-SET
```

Figure 4.1 An Algorithm for Incrementally Generating (Preferably Discriminating) Input Sets from Model Relations R(I)

may lead to distinct observations (3). If this also fails, this implies that

$$p_{obs}(R_i) = p_{obs}(R_j),$$

i.e. the behaviors cannot be distinguished under the set of observables in the given representation, and the pair has to be skipped. Otherwise, the algorithm seeks for already constructed input sets that have non-empty intersection with the new DI. If successful, it replaces one of them by this intersection (4). (As a special case, if $TI \subseteq DI$, we can simply add (i,j) to the D-LIST of TI, and we may do so for each DI with this property, if the algorithm finds several.) Otherwise, the new DI constitutes a new input set and is appended to the list (5). In both cases, the pair (i,j) is recorded in the D-LIST associated with the input set, so as to indicate which distinctions are to be revealed by the input (Note that the list can be incomplete, since there may be several test inputs intersecting DI, but only one was chosen).

This information can be used when applying the tests, as discussed below, but also in the generation phase. Different specializations and heuristics may be applied in the algorithm, in particular in step (4), the selection of the TI to be replaced by the intersection reflecting the goal to construct test inputs with high discriminating power. For instance, an input with a D-LIST containing two disjoint pairs, (i,j), (k,l), guarantees to rule out at least two

behaviors when applied. In a combination ((i,j), (j,k)), two out of three may survive. However, a "cycle" of length three, like ((i,j), (j,k), (i,k)), indicates that definitely at least two out of three behaviors will be refuted. There is some field for activity for people who like juggling with prime implicands/implicates: a D-LIST ((i,j), (k,l), …) can be translated into $(\neg B_i \vee \neg B_j) \wedge (\neg B_k \vee \neg B_l) \wedge \ldots$. Also probabilities of behaviors and/or of value tuples can be exploited.

Note, that we have two choices to make: select the input, $\underline{v}_{cause}$, and determine what to observe, $\underline{v}_{obs}$ (with the only restriction that $\underline{v}_{obs}$ contains $\underline{v}_{cause}$). This is important, because, so far, we did not take observation cost into account: two observable vectors, $\underline{v}_{obs}$ and $\underline{v}'_{obs}$, may be equally suited to reveal the distinction between candidates, but it could be the case that the distinction can be read off from a single variable in $\underline{v}_{obs}$ (i.e. other non-causal observable variables are redundant), whereas we need to measure many variables in $\underline{v}'_{obs}$ to make distinctions evident. This may not bother us, if $\underline{v}_{obs}$ is the vector of observables of the aggregate system which are considered to be easily obtainable. However, if additional, more costly probing points are included, the strategy has to be complemented by a cost criterion (e.g the cardinality of $OBS(\underline{v}) \setminus CAUSE(\underline{v})$) and we run the algorithm for varying vectors $\underline{v}_{obs}$ (starting with the cheap ones) in order to find an acceptable trade-off between number of tests and the observation cost in each test (e.g. minimizing the product). Furthermore, in some applications, also the number, type, and cost of the input may matter, and, finally, for dynamic systems, the necessary duration of some input (sequence) can be important.

Before we discuss issues of applying test inputs, we answer two questions that remained open, so far:

- How do we obtain the model relations for diagnostic candidates (as opposed to single constituents like a thyristor)?
- How can we handle test generation for continuous systems?

5 RELATIONAL MODELS OF CANDIDATES

As we emphasized earlier, a candidate is given as a mode assignment for all components:

$$\delta_i = \{m_{k_j}(C_j)\}.$$

For the constituents, C_j, models are assumed to exist.

In consistency-based diagnosis, we need at least models of the correct component behavior. Additionally, there may be fault models or descriptions covering a whole set of faults, such as $M(R_{Red\text{-}Bo})$ which covers an infinite set of behaviors with an actual breakover voltage between V_{Th} and V'_{Bo}. In particular, an unknown fault (including the unspecified $\neg OK$) can be regarded as imposing no restriction on the behavior and, hence, being modelled by $M(DOM(\underline{v}))$. This way, each behavior mode $m_{k_j}(C_j)$ assigned by δ_i has an associated model $M(R_{k_j})$, and the overall behavior B_i can be modelled by intersecting these constituent models:

$$B_i \Rightarrow M(R_i) \text{ where } R_i := \bigcap_{m_{k_j} \in \delta_i} R_{k_j}$$

Here, we assume, for the sake of simplicity, that the behavior relations for the constituents are specified using the global variable set of the entire device (leaving all non-local variables unrestricted). In this sense, there is no fundamental distinction between the models of a single constituent and of an aggregate system (for more details, see (Struss 1994b)). What can make a difference in practice is that observables and inputs usually are a smaller fraction of all variables for aggregate systems. This may affect the problem *whether* a discriminating input set exists, but not *the way* it can be generated.

6 MODEL ABSTRACTION FOR COPING WITH COMPLEXITY

A serious objection is that it may be extremely complicated to represent, intersect and project relations on continuous domains. This is the place multiple models and model abstraction come into play. If we can, for instance, apply a qualitative abstraction of the model relations such that only those distinctions are maintained that are essential for characterizing and comparing behaviors (such as V'_{Bo} for the thyristor), then test generation can be performed in this qualitative, finite representation. The test inputs obtained can be re-translated to the original continuous domain. Intuitively, the justification for the validity of this approach stems from the fact that any distinction revealed by the qualitative model remains true in a strictly stronger representation. In order to formalize this procedure and prove its correctness, we briefly summarize the formal foundations for multiple modeling as developed in (Struss 1992).

The key idea is simple: If $M(R_i)$ is a model of B_i, i.e. $B_i \Rightarrow M(R_i)$, and if R'_i is another relation (preferably in a finite domain) that specifies a weaker model, i.e. $M(R_i) \Rightarrow M(R'_i)$, then refuting $M(R'_i)$ suffices to rule out B_i. Hence, we can build test sets from such finite relations R'_i. The task is then to find conditions and a systematic way to generate models that are guaranteed to be weaker (in the logical sense specified above) by switching to a different representation $(\underline{v}', DOM'(\underline{v}'))$ with finite domains.

In (Struss 1992), a large class of transformations between representations is characterized by conditions that are both rather weak and natural:

DEFINITION 6.1 (REPRESENTATIONAL TRANSFORMATION)

A surjective mapping $\tau: DOM(\underline{v}) \rightarrow DOM'(\underline{v}')$ is a representational transformation iff it has the following properties

$$\underline{v}(s) = \underline{v}_0 \;\Rightarrow\; \underline{v}'(s) = \tau(\underline{v}_0)$$
$$\underline{v}'(s) = \underline{v}'_0 \;\Rightarrow\; \exists \underline{v}_0 \in \tau^{-1}(\underline{v}'_0)\ \ \underline{v}(s) = \underline{v}_0.$$

This simply means that, in the same situation, variables in the different representations have values related by τ.

Under such representational transformations, models are preserved (Struss 1992):

LEMMA 6.1

If τ: DOM'($\underline{v}$') $\rightarrow$ DOM($\underline{v}$) is a representational transformation, then

$$M(R') \Rightarrow M(\tau(R')) \quad \text{and} \quad M(R) \Rightarrow M(\tau^{-1}(R)).$$

This means, if we map a model relation from some original representation into a different one under a representational transformation the image will specify a weaker model, as required. In particular, we can choose a representation with a finite domain, construct disrciminating input sets in this representation from the transformed model relations and map them back to the original detailed representation.

The following theorem states that this actually yields disrciminating input sets in the original representation, thus justifying the intuitive approach:

THEOREM 6.2

Let τ_{obs}: DOM($\underline{v}_{obs}$) $\rightarrow$ DOM'($\underline{v}'_{obs}$)
and τ_{cause}: DOM($\underline{v}_{cause}$) $\rightarrow$ DOM'($\underline{v}'_{cause}$)
be representational transformations.

If $\{TI_i\}$ is a disrciminating input set for $BEVHS_\Delta$, then so is $\{TI_i\}:=\{\tau^{-1}_{cause}(TI'_i)\}$.

In particular, qualitative abstraction (mapping real numbers to a set of landmarks and the intervals between them) is a representational representation. In the thyristor example, the landmarks can be chosen as 0, V_{Th}, V'_{Bo}, V_{Bo} for ΔV and 0 for i. Ignoring the purely theoretical problem of separately treating the landmark points, this introduces quantity spaces Q_5 consisting of

neg := $(-\infty, 0]$
small := $(0, V_{Th}]$
medium := $(V_{Th}, V'_{Bo}]$
high := $(V'_{Bo}, V_{Bo}]$
too high := (V_{Bo}, ∞)

for ΔV and Q_3 with

\- := $(-\infty, -\delta)$
0 := $[-\delta, \delta]$
\+ := (δ, ∞)

for i, respectively. Mapping real values for ΔV and i to the intervals of Q_5 and Q_3, respectively, they are contained in, defines the qualitative abstraction

$$\tau_q : \mathbb{R} \times \{0,1\} \times \mathbb{R} \rightarrow Q_5 \times \{0,1\} \times Q_3$$

by

$$\tau_q((\Delta V_0, \text{ gate}_0, i_0)) = (q_{v0}, \text{gate}_0, q_{i0}).$$

where

$$\Delta V_0 \in q_{v0} \in Q_5 \text{ and } i_0 \in q_{i0} \in Q_3$$

Under the reasonable assumption that the holding current is greater than the leakage current:

$$(\text{res} - \Delta) * V_{Th} > \delta,$$

the representational transformation then induces model relations

$$R'_i := \tau_q(R_i) \subset Q_5 \times \{0,1\} \times Q_3$$

from the relations R_i in the real-valued representation. They are displayed in Table 6.1.

ΔV	gate	i				DI					
		R'_{ok}	$R'_{Red\text{-}Bo}$	R'_{block}	R'_{punct}	ok ↔ Red-Bo	ok ↔ block	ok ↔ punct	punct ↔ Red-Bo	punct ↔ block	block ↔ Red-Bo
neg	0	0	0	0	-			x	x	x	
neg	1	0	0	0	-			x	x	x	
small	0	0	0	0	+			x	x	x	
small	1	0	0	0	+			x	x	x	
medium	0	0	0,+	0	+			x		x	
medium	1	+	+	0	+		x			x	x
high	0	0	+	0	+	x		x		x	x
high	1	+	+	0	+		x			x	x
toohigh	0	+	+	0	+		x			x	x
toohigh	1	+	+	0	+		x			x	x

Table 6.1 The tuples constituting the relations R'_i and the sets DI_{ij} in the qualitative representation.
The values for the current i in the respective column complements the pair for (ΔV,gate) in each line to give R'_i.
The DI_{ij} are represented by collections of "x" indicating the respective inputs (ΔV, gate).

The table also shows the respective sets DI_{ij}. From these, the test generation algorithm produces the discriminating input set

$$\{\{(high,0)\},\ \{medium,high\} \times \{1\},\ \{neg, small\} \times \{0, 1\}\}.$$

Of course, the abstract representation may be too coarse to allow for the separation of particular behaviors. We can use this as a criterion for selecting representations and behavior models, for instance, as the highest level that still allows to distinguish one behavior from the others.

7 APPLYING TESTS

There are different ways to organize the interaction between the generation of diagnostic candidates and the generation and application of tests:

One possible strategy is to generate discriminating tests for the current candidates and recompute the set of preferred candidates after applying one test. If this set changes, new tests have to be generated for the changed $BEHVS_\Delta$. This reflects the fact that elimination of some candidates may push new ones into Δ. For instance, if a certain fault of a constituent is refuted, another, almost equally likely, fault may have to be considered. This strategy generalizes the loop of candidate generation and probing as performed in GDE's sequential diagnosis (de Kleer & Williams 1987). It does not require the generation of a whole discriminating input set as done by the above algorithm, but only the generation of one promising next test. In this case, the algorithm presented in section 4 could stop as soon as one acceptable test input has been produced, which promises to rule out a significant portion of $BEHVS_\Delta$ (w.r.t. its cardinality, probability weight, etc.).

In the second case, no new candidates are added until the discriminating input set has done its job, namely obtained a single diagnosis or a contradiction, in case all candidates have been refuted.

The test application algorithm in Fig. 7.1 covers both strategies: the former reruns the test generation algorithm after proposing additional candidates, perhaps using a variant that retuns only one test input. The latter might use a version of the diagnosis engine that does not shift new candidates into the focus until all previous ones have been refuted. There is a little more to be considered than simply applying blindly an input from TI and letting the diagnosis engine prune the candidate set after each set of observations. The selection of the next input to be applied should assess the anticipated gain which can be judged from an analysis of the D-LIST as mentioned above. Although the overall result is independent of the order in which the tests are applied, clever selection of the next test can pay off in several ways: Ruling out as many behaviors as possible

- can *save computation time* of model-based prediction, because refuted mode assignments can be removed from the focus of attention of the predictive engine
- can render some further tests unnecessary, thus *saving real time* which is even more significant.

The reason for the latter is that any behavior that has been refuted by one test does not require discrimination from other behaviors through further testing. This is why the algorithm removes all pairs containing the index of a refuted behavior from the D-LISTs. If the D-LIST of an input set becomes empty, the test is obsolete and eliminated from the TI-LIST. Note that a test may refute behaviors not mentioned in his D-LIST for two reasons: one has been mentioned before, namely that a different input might have been chosen to cover this behavior explicitly. Second, the resulting observations may contradict a behavior, although this could not be anticipated deterministically.

```
BEHVS (1…N-BEHVS)
WHILE TI-SET ≠ Ø
  Select and remove (best) TI from TI-SET
  Apply v_cause_o ∈ TI to obtain v_obs_o
  Run Consistency-Based Diagnosis
  For all newly refuted BEHVS(I)
    Remove BEHVS(I) from BEHVS
    For TI in TI-SET
      Remove all (I,J) and (J,I) from D-LIST(TI)
      Remove all TI with D-LIST = Ø from TI-SET
  GENERATE-NEW-TEST-INPUT(BEHVS, newly
                          proposed candidates, TI-SET)
```

Figure 7.1 The Test Application Algorithm

8 PROBABILISTIC TEST SELECTION

If probabilities of candidates are available they could be used for selecting the best test input, i.e. the one that promises the greatest gain in disciminating information. This will lead us to a test selection strategy, which comprises a general probe selection strategy which in turn covers the one used in GDE (de Kleer & Williams 1987) as a special instance. As in this work, we use the entropy

$$H = -\sum_{\delta_i \in \Delta} p(\delta_i) \log p(\delta_i)$$

to measure the amount of discriminative information. As before, we are looking for test inputs that guarantee or are likely to supply us with observations that support discrimination among the modeling relations, R_i, of the behaviors B_i defined by the candidates of δ_i. To account for non-deterministic models, we pretend that for each behavior B_i and, hence, candidate δ_i and for each causal input $\underline{v}_{cause_o}$ we know the probability distribution

$$p(\underline{v}_{obs_o} \mid \delta_i, \underline{v}_{cause_o})$$

Since the causal variables are a subset of the observables, this probability is non-zero only for observable tuples that match the causal input:

$$p'_{cause}(\underline{v}_{obs_o}) \neq \underline{v}_{cause_o} \Rightarrow p(\underline{v}_{obs_o} \mid \delta_i, \underline{v}_{cause_o}) = 0$$

If we denote the "slice" of observable tuples in DOM($\underline{v}_{obs}$) that correspond to $\underline{v}_{cause_o}$ by $O(\underline{v}_{cause_o})$:

$$O(\underline{v}_{cause_o}) := p'^{-1}_{cause}(\underline{v}_{cause_o}),$$

then

$$(1) \qquad \sum_{\underline{v}_{obs_o} \in O(\underline{v}_{cause_o})} p(\underline{v}_{obs_o} \mid \delta_i) = 1$$

for each δ_i and $\underline{v}_{cause_o}$[1]. (Of course, the probability is zero outside the modeling relation R_i, i.e.

$$\underline{v}_{obs_o} \in O(\underline{v}_{cause_o}) \setminus R_i \;\Rightarrow p(\underline{v}_{obs_o} | \delta_i, \underline{v}_{cause_o}) = 0.)$$

To make the following equations more readable, we replace

$$\sum_{\underline{v}_{obs_o} \in O(\underline{v}_{cause_o})} \text{ by } \sum_{\underline{v}_{obs_o}}$$

and

$$\sum_{\delta_i \in \Delta} \text{ by } \sum_{\delta_i}$$

whenever it is not ambiguous in a context.

What we are interested in is the expected entropy after applying some input $\underline{v}_{cause_o}$ and observing $\underline{v}_{obs}$. This is

$$(2) \qquad H_e(\underline{v}_{cause_o}) = \sum_{\underline{v}_{obs_o}} p(\underline{v}_{obs_o}) H(\underline{v}_{obs_o}),$$

where

$$(3) \qquad H(\underline{v}_{obs_o}) = -\sum_{\delta_i} p(\delta_i \mid \underline{v}_{obs_o}) \log p(\delta_i \mid \underline{v}_{obs_o})$$

is the entropy for a particular observed tuple $\underline{v}_{obs_o}$. We apply Bayes' rule to obtain

$$p(\delta_i \mid \underline{v}_{obs_o}) = \frac{p(\underline{v}_{obs_o} \mid \delta_i)\, p(\delta_i)}{p(\underline{v}_{obs_o})}.$$

Substitution in (3) yields

$$H(\underline{v}_{obs_o}) = -\sum_{\delta_i} \frac{p(\underline{v}_{obs_o} \mid \delta_i)\, p(\delta_i)}{p(\underline{v}_{obs_o})} \cdot$$
$$(\log p(\underline{v}_{obs_o} \mid \delta_i) + \log p(\delta_i) - \log p(\underline{v}_{obs_o})),$$

and (2) turns into

$$H_e(\underline{v}_{cause_o}) = -\sum_{\underline{v}_{obs_o}} \sum_{\delta_i} p(\underline{v}_{obs_o} \mid \delta_i)\, p(\delta_i)$$
$$(\log p(\underline{v}_{obs_o} \mid \delta_i) + \log p(\delta_i) - \log p(\underline{v}_{obs_o})).$$

Rearranging summation and their ordering, we obtain

$$(4) \quad H_e(\underline{v}_{cause_o}) = \sum_{\underline{v}_{obs_o}} \log p(\underline{v}_{obs_o}) \sum_{\delta_i} p(\underline{v}_{obs_o} \mid \delta_i)\, p(\delta_i)$$
$$- \sum_{\delta_i} p(\delta_i) \log p(\delta_i) \sum_{\underline{v}_{obs_o}} p(\underline{v}_{obs_o} \mid \delta_i)$$
$$- \sum_{\delta_i} p(\delta_i) \sum_{\underline{v}_{obs_o}} p(\underline{v}_{obs_o} \mid \delta_i) \log p(\underline{v}_{obs_o} \mid \delta_i).$$

1. We only treat the case of $\underline{v}_{obs}$ having a finite domain because, first, it is as theoretically unproblematic as practically infeasible to replace the sum by an integral and, second, our approach aims at exploiting finite domains created by appropriate abstractions, anyway.

Because of

$$\sum_{\delta_i} p(\underline{v}_{obs_o} \mid \delta_i)\, p(\delta_i) = p(\underline{v}_{obs_o}),$$

the first term becomes

$$\sum_{\underline{v}_{obs_o}} p(\underline{v}_{obs_o}) \log p(\underline{v}_{obs_o})$$

which is the (negative) entropy of the observable tuples. With equation (1), the second term in (4) is discovered to be

$$-\sum_{\delta_i} p(\delta_i) \log p(\delta_i) = H\,,$$

i.e. the entropy before obtaining information based on the input. Hence, the information gain from the test input $\underline{v}_{cause_o}$ is

$$(5) \qquad H - H_e(\underline{v}_{cause_o}) = -\sum_{\underline{v}_{obs_o}} p(\underline{v}_{obs_o}) \log p(\underline{v}_{obs_o})$$
$$+ \sum_{\delta_i} p(\delta_i) \sum_{\underline{v}_{obs_o}} p(\underline{v}_{obs_o} \mid \delta_i) \log p(\underline{v}_{obs_o} \mid \delta_i).$$

The best test should be the one for which this gain is maximal. The two terms in (5) can be interpreted as representing two different (counteracting) influences on it: on the one hand, it grows with the entropy of the probabilities of $\underline{v}_{obs_o}$ in $O(\underline{v}_{cause_o})$, which is maximal for an equal distribution of the tuples. On the other hand, it is diminished by the second term. This represents the expected information about $\underline{v}_{obs_o}$ if the candidate was determined and is minimal if the candidates deterministically predict one value. Intuitively speaking, the best situation is given if a) the tuples are equally distributed and b) this is caused by the candidates making distinctive predictions about them. In contrast, it does not help to apply $\underline{v}_{cause_o}$ if the high entropy of tuple probabilities is due to the fact that all candidates are indifferent about $\underline{v}_{obs}$. Also, observing $\underline{v}_{obs}$ is useless if all candidates are very specific about $\underline{v}_{obs}$ – but predict the same value. Hence, we can intuitively justify the following

Probabilistic Test Selection Strategy

In order to discriminate among candidates $\delta_i \in \Delta$, choose a test input $\underline{v}_{cause_o}$ and an observable vector $\underline{v}_{obs}$ such that

$$(6) \qquad -\sum_{\underline{v}_{obs_o} \in O(\underline{v}_{cause_o})} p(\underline{v}_{obs_o}) \log p(\underline{v}_{obs_o})$$
$$+ \sum_{\delta_i \in \Delta} p(\delta_i) \sum_{\underline{v}_{obs_o} \in O(\underline{v}_{cause_o})} p(\underline{v}_{obs_o} \mid \delta_i) \log p(\underline{v}_{obs_o} \mid \delta_i)$$

is maximal, where the probabilities of observable tuples are determined from the candidate-specific distributions and probabilities:

$$(7) \qquad p(\underline{v}_{obs_o}) = \sum_{\delta_i \in \Delta} p(\underline{v}_{obs_o} \mid \delta_i)\, p(\delta_i).$$

The same remarks we made in section 4 about variations in $\underline{v}_{obs}$ apply to the probabilistic test generation, as well. In particular, this kind of test generation includes a probe

selection strategy, if we fix the input $\underline{v}_{cause_o}$:

Probabilistic Probe Selection Strategy

In order to select the best probing point $\underline{v}_{obs}$ for discrimination among candidates $\delta_i \in \Delta$ under a given input, fix $\underline{v}_{cause_o}$ to be this input, vary $\underline{v}_{obs}$ by appending the different non-causal observables $v_{obs} \in OBS(\underline{v}) \setminus CAUSE(\underline{v})$ to $\underline{v}_{cause}$, apply the Test Selection Strategy, and select $\underline{v}_{obs}$ that maximizes (6).

This gives us a generalization of GDE's probe selection strategy (de Kleer & Williams 1987) for non-deterministic models with probability distributions for values. GDE is a special instance in considering only extremes of such distributions: candidates that predict exactly on value, $\underline{v}_{obs_o}$, with probability one, gathered in candidate sets $S_{\underline{v}_{obs_o}} \subseteq \Delta$:

$$S_{\underline{v}_{obs_o}} := \{\delta_i \in \Delta \mid p(\underline{v}_{obs_o} \mid \delta_i) = 1\},$$

and candidates in $U_{\underline{v}_{obs}}$ that are completely uncommitted w.r.t. $\underline{v}_{obs}$, i.e. predict all values with equal probability:

$$U_{\underline{v}_{obs}} := \{\delta_i \in \Delta \mid \forall \underline{v}_{obs_0}, \underline{v}_{obs_1} \in O(\underline{v}_{cause_o}) \; p(\underline{v}_{obs_o} \mid \delta_i) = p(\underline{v}_{obs_1} \mid \delta_i)\}$$

GDE's strategy can be characterized by considering a single non-causal observable, asuming

$$\Delta := \bigcup_{\underline{v}_{obs_o} \in O(\underline{v}_{cause_o})} S_{\underline{v}_{obs_o}} \cup U_{\underline{v}_{obs}},$$

and by finite domains. If m is the cardinality of the domain of the variable v_{obs} to be probed,

$$m = \mid DOM(v_{obs}) \mid = \mid \underline{v}_{cause_o} \times DOM(v_{obs}) \mid = \mid O(\underline{v}_{cause_o}) \mid,$$

then values have a uniform probability of $\frac{1}{m}$ for all $\delta_i \in U_{\underline{v}_{obs}}$, and the computation of $p(\underline{v}_{obs_o})$ according to (7) turns into

$$p(\underline{v}_{obs_o}) = \sum_{\delta_i \in S_{\underline{v}_{obs_o}}} p(\delta_i) + \sum_{\delta_i \in U_{\underline{v}_{obs}}} \frac{1}{m} p(\delta_i)$$

(8) $$p(\underline{v}_{obs_o}) = p(S_{\underline{v}_{obs_o}}) + \frac{1}{m} p(U_{\underline{v}_{obs}}).$$

In the selection criterion (6) the second term becomes

(9) $$\sum_{\delta_i \in U_{\underline{v}_{obs}}} p(\delta_i) \sum_{\underline{v}_{obs_o}} \frac{1}{m} \log \frac{1}{m} = p(U_{\underline{v}_{obs}}) \log m.$$

(8) and (9) reproduce GDE's probe selection criterion in the form it is stated in (de Kleer 1990).

9 DISCUSSION

We presented a theoretical foundation and an algorithm for generating tests for discrimination of diagnoses based on relational behavior models. The worst-case complexity of the algorithm (n^4_{behvs}) prohibits the application if the set of candidates is large. But what makes the approach work for continuous domains, performing test generation on (qualitative) abstractions of models, may also reduce the set of candidates through collapsing them into qualitatively described classes of behaviors which can then be discriminated.

More can be done, in particular in assessing the expected gain of tests and applying heuristics that exploit domain-specific characteristics. However, generality is an advantage of the presented theory which can serve as a framework for the systematic design of test generation and application in diagnosis and for analyzing the preconditions and the impact of different strategies and heuristics.

Acknowledgements

Many thanks to Ulli Heller and Andreas Malik for their support.

References

de Kleer, J., Williams, B. C., 1987, *Diagnosing Multiple Faults,* Artificial Intelligence 32(1987).

de Kleer, J., 1990, *Using Crude Probability Estimates to Guide Diagnosis,* Artificial Intelligence 45(3)(1990)

Genesereth, M.R., 1984, *The Use of Design Descriptions in Automated Diagnosis,* Artificial Intelligence 24(1984), 411-436

Meerwijk, A., and Preist, C., 1992, *Using Multiple Tests for Model-based Diagnosis,* Working Papers of the Third International Workshop on Principles of Diagnosis, Rosario

Struss, P., 1992, *What's in SD? Towards a Theory of Modeling for Diagnosis,* in: Hamscher, W. Console, L., and de Kleer, J. eds., Readings in Model-based Diagnosis. San Mateo: Morgan Kaufmann: 419-449.

Struss, P., 1994, *A Theory of Testing Physical Systems Based on First Principles,* Technical Report 94/63, Christian-Doppler-Labor, Technical University of Vienna.

Struss, P., 1994a, *Testing Physical Systems,* AAAI-94, Seattle, Washington.

Struss, P., 1994b, *Model Abstraction for Testing of Physical Systems,* International Workshop on Qualitative Reasoning QR-94, Nara, Japan.