Qualitative Fault Modeling Supported Test Generation

András Pataricza, Balázs Polgár, Imre Kocsis, András Kövi
Budapest University of Technology and Economics

Automated test program generation (ATPG) faces huge computational complexity challenges, as it has to cope with all the potential faulty instances, or at least with a representative subset promising proper fault coverage. The search space for test sequences is typically huge due to the large number of faulty system instances to be evaluated and the large state space of the individual instances. Formal methods are widely used for proof of correctness in IT system verification, but modeling and analysis complexity originating in the large number of faulty cases limits their use in dependability analysis.

The basic approaches for complexity reduction are: aggregation of equivalent or dominant faults into a single fault class represented by a single instance (reduction of the number of faulty instances); abstraction of the data/control model of the program under test (reduction of detailedness, thus test search spaces). However, abstraction has to preserve enough faithfulness of the model to assure a good correlation with the anticipated set of faults and avoid the generation of redundant or inefficient tests as far as possible. Another main practical approach is priorization of ATPG coverage requirements according to the severity of the potential impacts of the different faults. A high coverage of faults causing critical safety failures is a priority objective, while low severity faults can be neglected if ATPG efforts are limited.

This paper (based on [Pataricza, 2006] and [Pataricza, 2008]) presents a *qualitative modeling approach* at the level of faults, errors and failure modes. The approach facilitates the reuse of existing formal methods to *identify the faults resulting in critical effects w.r.t. safety and dependability requirements*. The qualitative model based pre-analysis focuses the subsequent detailed, thus costly ATPG on the *critical faults* necessitating a test set of a complete or at least of a high coverage. Qualitative fault modeling uses *abstraction* in the form of *spatial and temporal compaction* to master the complexity related issues.

Figure 1 shows the relations between the concrete and abstract models. Spatial compaction shrinks the state space of the concrete model potentially by orders of magnitude. It aggregates those fully detailed reference-actual value pairs in the concrete system into a single class, which result in similar impacts and represents them by a single qualitative value in the abstract model. The dynamics of error propagation is represented by a dynamic model preserving the control flow in the original one. The subsequent temporal compaction turns the dynamic model into a static one by grouping error sequences having a similar impact and representing them by a single qualitative syndrome. The topology and function of the concrete system are component-wise mapped into an error sensitivity constraint network. The chain of special and temporal compaction delivers a proper abstraction in the sense that each concrete test sequence has a counterpart in the abstract domain. The reverse is generally not true; that is, spurious abstract solutions may occur having no counterpart in the concrete domain.

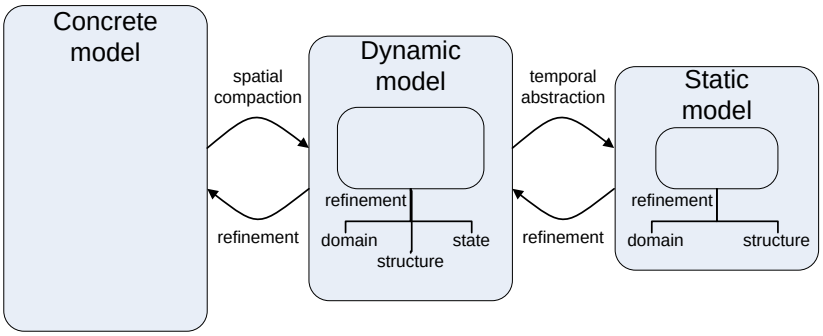

Fig 1: Model abstraction and refinement

Abstract models can be utilized according to the following general rules and conditions:

- First, the abstract counterparts of the mutations (abstract mutant models) are generated. The abstraction process is facilitated by using easy to estimate over-abstractions in *the abstract model space. As they deliver an upper cover of the feasible solution space*, thus the guarantee of systematicness of the search process in the concrete model is retained (no test is overlooked).
- Solving the test generation problem in the usually small abstract space delivers abstract solutions. A part of these solutions corresponds to true solutions in the detailed concrete domain while another part is formed by spurious solutions due to abstraction not detecting faults in the concrete model. The level of compaction essentially influences the occurrence frequency of spurious abstract. The abstract ATPG is similar to a semi-decision i.e. in the abstract domain it is undecidable whether an abstract test sequence detects any faults in the concrete domain.
- Abstract tests will be used as guiding heuristics in the concrete detailed (SAT based) test generation by confining the search space of ATPG in the concrete model as constraints prescribing the correspondence to a particular abstract test sequence. If the previously generated abstract test sequence has a counterpart in the concrete domain than the resulting drastic reduction of the search space has a huge impact on computation time. However, the elimination of a spurious abstract solution needs an exhaustive but unsuccessful search in the concrete domain.

Qualitative abstraction *allows focusing ATPG to critical faults* by processing those test abstract sequences first which result in critical fault manifestations. Moreover, testability analysis in the abstract domain detects non-testable faults (i.e., checking whether a given fault can be tested by considering the points of control and observation in the system).

Another potential application of qualitative fault modeling includes control of fault injection experiments. Abstract models can pre-filter candidate patterns used in fault injection campaigns by excluding faults from further injection resulting only in already checked failures. Additionally, uncovered failures can be estimated together with their respective test sets (aiming at a better focusing of fault injection campaigns); moreover, an approximate injection coverage metrics can be calculated.

The main advantage of the proposed modeling technique is the reusability of existing formal methods for analyzing fault effects. *Static analysis* can be implemented by means of constraint satisfaction programming. The accuracy of the abstract analysis phase can be further refined by applying the more costly *dynamic analysis*. The guarantee that *no critical behavior will be overlooked* fulfills the expectations against a worst case analysis. The price of simplified analysis is the generation of spurious results that can be eliminated by a subsequent exhaustive analysis in the detailed domain.

[Pataricza, 2006] A. Pataricza. Model-based Dependability Analysis. DSc Thesis. Hungarian Academy of Sciences, 2006.

[Pataricza, 2008] A. Pataricza. Systematic Generation of Dependability Cases from Functional Models. In G. Tarnai and E. Schnieder (eds.): Formal Methods for Automation and Safety in Railway and Automotive Systems. Proc. Symposium FORMS/FORMAT, October 9-10, Budapest, Hungary, pp 17-24, L'Harmattan, Budapest.