



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DEPT. OF TELECOMMUNICATIONS AND MEDIA INFORMATICS

SELECTIVE AUTOMATIC TEST GENERATION FROM
EFSM FORMAL MODELS WITH FAULT AND STRING
EDIT DISTANCE BASED METHODS

Gábor Kovács

Summary of the Ph.D. Dissertation

Supervised by

Dr. Gyula Csopaki and Dr. Katalin Tarnay
High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

Budapest, Hungary
2009

1 Introduction

Protocols in telecommunication systems – just like software systems – get more and more complex and robust nowadays and at the same time they have to meet strict reliability requirements defined by the standards. Several development methodologies have been defined to support the quality of protocols throughout the life-cycle, and some of these have been standardized.

The SDL+ methodology [IT97] was standardized at ITU. Its life cycle model defines the relation among several development activities: Requirement capture is followed by a two staged design phase of draft design and formal modelling. The formal model after validation provides the basis for both implementation of the product and test specification.

The dissertation builds on this life-cycle model and the literature on test generation for different formal models to investigate the possibility of automation of the test specification activity. Automatic test generate for an extended finite state machine (EFSM) model can be done by unfolding the EFSM into a finite state machine (FSM) or by means of a partial, selective search method.

Test generation for FSMs is fully solved: the transition checking method can be used to derive test cases automatically [LY96, vBP94]. Since according to this method each transition must be checked, the resulting test set may contain quasi infinite number of cases for systems with huge state space, which makes this method impractical from the point of view of test execution.

Several EFSM based test generation approaches have been published [SKGH97, BH89, BDA, DÜU04, Kim08] and implemented in software tools. These tools, which are often criticized for generating a large number of test cases, support method based on exhaustive search and random walk.

Several partial state space search methods have been defined beside the brute force method like depth bound, scatter search, guided search, probabilistic search, partial order and random selection in the literature. According to Holzmann [Hol91] random selection *‘is not only the simplest technique to implement, but is also likely to produce the highest quality search.’* [Hol91]

2 Research objectives

The main aim of my research is to define a formal framework for the EFSM and the derivative SDL semantics to support selective automatic test generation. The selectivity gives two criteria for the test generation. On the one hand, the number of automatically generated abstract test cases should be less than the size of the exhaustive test set generated with the brute force method. This aim reduces the cost of test execution with regard to the high cost of bringing back the system under test

to its initial state. On the other hand, the quality metrics of the generated test set should be better than the set generated with random walk.

The results of the thesis are divided into three groups. The first group defines a fault model based formal framework and an automatic test generation method. The selectivity of the algorithm is provided by on-the-fly evaluation the fault detection capability of the test case being generated. This results in smaller test suites with test cases of high fault detection ability.

The second group extends the string edit distance based test selection methodology with a solution that can find the highest compression of a test set for a given density threshold. While existing method can only decide if two test cases are redundant, my method decides which one to keep.

In the third group abstract iterative algorithms are proposed that allow the definition of selective automatic test generation algorithm for any test suite metrics. This approach can reduce the computation requirement of the method in the first group and support the maintenance of test sets when using development life-cycles with iteration.

3 Methodology

I used analytic and simulation based research methods during my research. I used analytic methods to solve problems in algorithm theory, graph theory and complexity theory. I evaluated the models, algorithms proposed in the thesis by means of simulations.

In the first group of results my formal framework builds on different formal semantics: finite state machines, extended finite state machines and process algebra. My definitions adapt *the formal framework for conformance testing* [Tre00], which is defined in process algebra, to SDL semantics. The test generation method I propose uses the mutation analysis [DMLS78] technique. I investigate my approach with simulation based experiments.

The second group of theses is built on the matching problem known in graph theory and the algorithm computing string edit distance [WF74]. The solution reduces the string edit distance based test selection to the so called k-assignment problem. I investigated the efficiency of this method with experiments based on simulations.

The third thesis group builds on the concept of evolutionary algorithms. I evaluated the approach by means of simulations.

4 New results

4.1 Automatic test generation for SDL specifications based on fault models

Thesis group 1 [J2, C1, C6, C7] *I designed a fault model based selective automatic test generation method that derives abstract conformance test cases in MSC formalism for SDL specification using an arbitrary state space search method. I conducted empirical analysis to evaluate the efficiency of the method.*

Research aimed to develop automatic test generation tools and software components states that for systems of larger scale selective methods should be preferred over exhaustive, brute force methods [Hol91, vBP94]. Available software tools mainly support brute force and random walk based methods. AutoLink [SKGH97] that is part of the SDL based tool of Telelogic (now IBM) provides beside the brute force and the random based test generation methods a semi-automatic method with manual navigation in the state space. The tools TorX [TB03] and TGV [JJ02] with process algebra semantics, which are based on the framework of [Tre00], use brute force method for test generation and note the necessity of selectivity.

The cornerstones of the test generation method of the first thesis group are based on the principles of the tools above and are include the requirement of selectivity:

- the test case derivation process must take place without any human intervention, that is, fully automatically
- the number of traces (hence test cases as well) included in the generated set must be limited
- though random subset of all traces, that is, walks have been shown to be “quite satisfactory” [TB03], the strategy should have a better fault detection ability or require less execution time than the same number of random traces
- test cases are generated from the black box perspective, state space search algorithm do not use information related to the current state of the specification model

This thesis group limits the number of test cases to be generated with a fault model. The mutation technique is well-known in the software world: it was originally proposed for test data selection [DMLS78] and used for software validation. Research of the last decade has shown that it is applicable not only for software but for formal models as well [PG91, WL93, FMDM94, SMFLDS00, FMM⁺95, FMSM99, ABM98, BOY00, AB99]. Beside our work SDL has been addressed twice, in both cases the

aim is the validation of the specification. The approach in [1] builds on our work, the latest paper [WRQB08] is independent.

The statements in this thesis are complementary and in line with the statements that appeared in Zoltán Pap’s Ph.D. thesis [Pap06], where a fault model of atomic operators for SDL and an automatic test selection algorithm were proposed. I use that fault model for evaluating the test generation method of this thesis and the test selection method in the third thesis group.

Thesis 1.1 *I designed and formalized a new fault model based formal framework for SDL that supports the automatic application of fault models and the generation of abstract test cases.*

I designed a formal framework for SDL specification that use the fault model of Pap [Pap06] to generate test cases automatically.

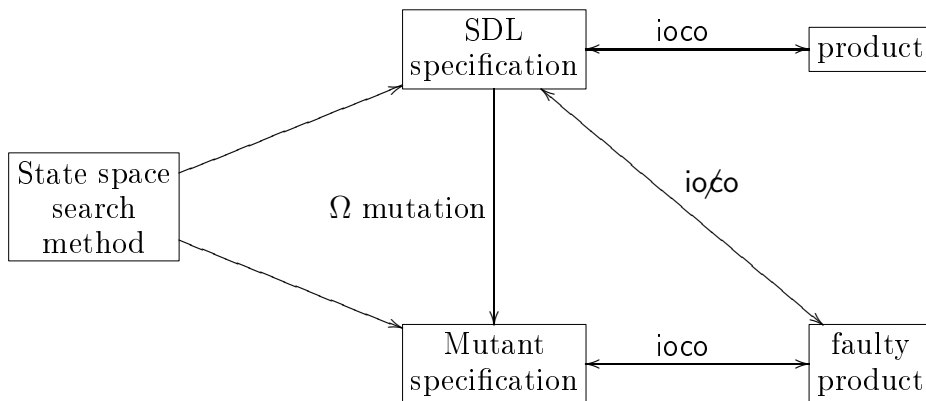


Figure 1: Model of fault model based automatic test generation for SDL

My test generation framework builds on the *ioco* conformance and the refusal preorder implementation relations of *the formal framework for conformance testing* model [Tre00]. The conformance relation check if the implementation can produce the behavior given in the specification. However, this relation does not necessarily hold between the original specification and a faulty implementation, which is conform to a specification with an injected fault.

The model is given in Figure 1. The conformance relation holds both between the SDL specification and the product and between the mutant specification and the faulty product. The mutant specification is automatically generated with an $\omega \in \Omega$ mutation operator from the original SDL specification: the operator either

adds and/or removes – possibly a set of – transitions and/or relabels the transition part of a transition. From the design aspect this means that a requirement is either added and/or removed or modified. From the point of view of implementation, the faulty product that is in ioco relation with that mutant specification implements some optional features, but may not implement mandatory features from the original set of requirements. This enables the formalization of the relation between the original and the mutant specifications, which is a refusal preorder relation:

- mandatory features must be present in the mutant specification as well, and
- the optional features are considered only in the implementation and not in the specification, that is a mutant specification must not specify additional optional features.

To prove that there is a refusal preorder between two systems, one must check all possible sequences. However, to prove that the refusal preorder relation does not hold, a single trace of the original specification must be found that is not present in the mutant specification, and that trace is suitable for checking the ioco relation as well.

Thesis 1.2 *I defined a selective test generation method that derives a set of abstract conformance test cases, while the number of test cases is limited and the fault detection ability is preserved in generation time by a fault model. I evaluated this method with simulations performed using a software tool that implements the formal framework of Thesis 1.1.*

Algorithm 1 generates a test set for SDL specifications automatically. That test set has the highest possible fault detection ability with regard to the given Ω fault model. The size of the resulting test set is not more than the number of mutant system generated.

I investigated the algorithm above with simulations on the INRES system, Conference Protocol and on two real life protocols, on the WAP WTP (Wireless Application Protocol – Wireless Transaction Protocol) and on the SS7 MTP2 (Signalling System No. 7 – Message Transfer Part level 2). I compared with random walk and brute force based test generation methods.

The strength of the proposed method and the implementing software tool is that it automatically generates a test set of limited size and with high fault coverage for an SDL specification based on a fault model. Its weakness is the high computation demand.

The results presented in Chapter 3 in the dissertation are published in [C6, C7, J2, C2]. There are four independent [1, 2, 3, 4] and one dependent [5] citations to these papers.

Algorithm 1: Derivation of test cases from an EFSM specification with mutation analysis

input : $m = (S, V, I, O, T)$, an EFSM/SDL specification; *stop* condition
output: $\Sigma = \{\dots, \sigma, \dots\}$ test set, where σ is an IOTS.

```
1 for  $t \in T$  do
2   for  $\omega \in \Omega$  do
3     Let  $m' = (S, V, I, O, T')$ , where  $t' = \omega(t), t' \in T', t \in T$ ;
4      $\sigma = \text{null}$ ;
5     repeat
6       if stop then return;
7       /* Let an arbitrary test generation method explore the
8         state space of  $M$  and construct  $\sigma$  incrementally */
9        $\sigma := \text{derive}(m)$ ;
10      until  $\text{out}(m, \sigma) \neq \text{out}(m', \sigma)$  ;
11       $\Sigma := \Sigma \cup \{\sigma\}$ ;
12    endfor
13  endfor
14 return  $\Sigma$ 
```

4.2 Optimal string edit distance based test selections

Thesis group 2 [J3, C13] *I extended the string edit distance based test selection method with a procedure that finds the maximum possible reduction with the highest diversity of an input test set with regard to a density threshold. I investigated the proposed method with simulation and compared it with other coverage metrics.*

This thesis uses the method of Vuong and Alilovic-Curgus proposed in [VAC92] and extended in [FGMT02] by Feijs et al. That methodology represents test cases as strings and considers that the similarity of two test cases can be measured by their string edit distance. The former paper by Vuong and Alilovic-Curgus introduces a test coverage metric based on the concept of testing distance between traces. Traces are considered to be similar (redundant) if they can be transformed to each other with a cost no more than a given ε parameter, so that the test set is said to be ε -dense.

In the latter paper Feijs et al. generalize the original idea by introducing a reduction heuristic and a cycling heuristic: The notion of marked traces is proposed to tackle the problem of traces revisiting states of the system via the same loop at most a given number of times determined by the reduction heuristic. Formulae are given to precisely calculate the normed distance between traces containing symbols and marked traces (loops).

That method is able to find out if two test cases are redundant. The extension I propose can tell which of that two test cases should be preserved in the test set.

Thesis 2.1 *I defined a polynomial time algorithm that reduces the string edit distance based test selection to a matching problem in bipartite graphs. The algorithm finds the size of reduced test sets after a maximum reduction of the input test set with regard to the given approximation threshold. I showed that the minimal size can be found in $O(|\Sigma|^3)$ time, where $|\Sigma|$ is the size of the test set to be reduced.*

I defined the distance matrix based on the distance metrics of [VAC92, FGMT02]: $\mathbf{D} = [d_{ij}]$, where $d_{ij} = d(\mu(t_i), \mu(t_j))$ and $1 \leq i, j \leq |\Sigma|, i, j \in \mathbb{N}$, and μ is a test case to marked trace mapping.

The inputs of Algorithm 2 are the distance matrix of the test set Σ and a ε threshold. The output is the minimum cardinality of the reduced set. The test set Σ is divided into two disjoint subsets: a Σ'' subset of test cases that can and a $\Sigma'_0 \subseteq \Sigma'$ subset of test cases that can not be ε -covered by other test cases. The test cases from Σ'_0 must be included in Σ' , and from Σ'' the minimum number of cases must be selected: $\Sigma' = \Sigma'_0 \cup \text{reduce}(\Sigma'')$. Thus the maximum reduction of Σ'' yields the most compact Σ' .

For the computation two matrices \mathbf{A} and \mathbf{A}_ψ of boolean values and a bipartite graph G' are used. The algorithm first determines which test cases of Σ ε -cover each other, and if they do, it is marked in \mathbf{A} with 1 value (lines 4-7). Then, the size of the Σ'_0 set is determined in lines 8-10. In lines 11-15, if exactly the same coverage is found for two test cases, then one of them is eliminated. Lines 16-22 construct a bipartite graph G' based on matrix \mathbf{A} . The Hopcroft-Karp algorithm [CLRS01] is used for finding a maximum cardinality assignment ψ in G' (line 23). The k minimal size returned in line 28 is the size of Σ'_0 plus the rank of the upper or lower triangular matrix of \mathbf{A}_ψ constructed based on the maximum matching in lines 24-27.

Given the distance matrix, the construction of the bipartite graph is $O(|T|^2)$. The worst-case complexity of the Hopcroft-Karp algorithm [CLRS01] applied to the graph $G' = (N', E')$ is $O(\sqrt{|N'|}|E'|)$. Since $|N'| \leq 2|\Sigma|$ and $|E'| \leq |\Sigma|^2$ its complexity is $O(|\Sigma|^{5/2})$. Hence the worst case complexity of this algorithm is determined by the search for same rows in \mathbf{A} , which is $O(\Sigma^3)$.¹

Thesis 2.2 *I reduced the problem of finding the most diverse subset of an input test set to the k -assignment problem defined by Dell'Amico and Martello.*

¹The complexity can further be reduced. Searching for identical rows in matrices can be done more effectively with the Rabin-Karp algorithm [CLRS01], which operates on the hash values of patterns. That algorithm can find all matching rows in $O(\Sigma)$. So the resulting complexity of Algorithm 2 is $O(|\Sigma|^{5/2})$

Algorithm 2: Deriving the string edit distance based test case selection problem to a matching problem in bipartite graphs

```

input :  $\mathbf{D}$  distance matrix of  $\Sigma$  test set;  $\varepsilon$  threshold
output:  $k$ , the maximum number of redundant cases for the given  $\varepsilon$ 
1 data( $\mathbf{A} = [a_{ij}], a_{ij} \in \{0, 1\}; \mathbf{A}_\psi = [a_{\psi ij}], a_{\psi ij} \in \{0, 1\};$ 
2  $G' = (N', E')$  bipartite graph, where  $N' = N'_R \cup N'_C$ )

  /* Initialization */
3  $k := 0, N'_R := \emptyset, N'_C := \emptyset, E' := \emptyset, \mathbf{A} := \mathbf{0}, \mathbf{A}_\psi := \mathbf{0};$ 
  /* Computing the  $\mathbf{A}$  matrix */
4 foreach  $i, j, 1 \leq i, j \leq |\Sigma|$  do
5   if  $d_{ij} < \varepsilon$  then  $a_{ij} = 1;$ 
6   else  $a_{ij} = 0;$ 
7 endfch
  /* Counting the elements of  $\Sigma'_0$  */
8 foreach  $i$  do
9   if  $\sum_j a_{ij} = 0$  then  $k := k + 1;$ 
10 endfch
  /* Finding test cases with the same coverage */
11 foreach  $k, l, 1 \leq k \leq |\Sigma| - 1, k < l \leq |\Sigma|$  do
    if  $\sum_j (a_{kj} \text{ xor } a_{lj}) = 0$  then
12     foreach  $j, 1 \leq j \leq |\Sigma|$  do  $a_{lj} := 0, a_{jl} := 0;$ 
13     endif
14 endfch
15 foreach  $\sigma_i \in \Sigma$  do
    /* Constructing the  $G'$  graph */
16 foreach  $\sigma_i \in \Sigma$  do
17    $N'_R := N'_R \cup \sigma_i;$ 
18    $N'_C := N'_C \cup \sigma_i;$ 
19 endfch
20 foreach  $i, j, 1 \leq i, j \leq |\Sigma|$  do
21   if  $a_{ij} > 0$  then  $E' := E' \cup \{(n'_i, n'_j)\};$ 
22 endfch
23 Let  $E'_\psi$  be the matching selected by the Hopcroft-Karp algorithm;
    /* Marking the pairs of the maximum matching in  $\mathbf{A}_\psi$  */
24 foreach  $i, j, 1 \leq i, j \leq |\Sigma|$  do
25   if  $(n'_i, n'_j) \in E'_\psi$  then  $\mathbf{A}_{\psi ij} = 1;$ 
26   else  $\mathbf{A}_{\psi ij} = \mathbf{0};$ 
27 endfch
28 return  $k := k + \text{rank}(A_\psi^L)$ 

```

The redundancy among test cases in a test set is the smallest, if sum the pairwise distances in the reduced subset is maximal. When more Σ' exist with the minimum cardinality, the one with the highest internal distance sum should be preferred, which can be determined in polynomial time of the size of the input set Σ . The optimization problem is the following:

$$\max \sum_{\sigma_i \in \Sigma', \sigma_j \in \Sigma'} d(\mu(\sigma_i), \mu(\sigma_j)), \quad (1)$$

where $\forall i, j : d(\mu(\sigma_i), \mu(\sigma_j)) > \varepsilon$.

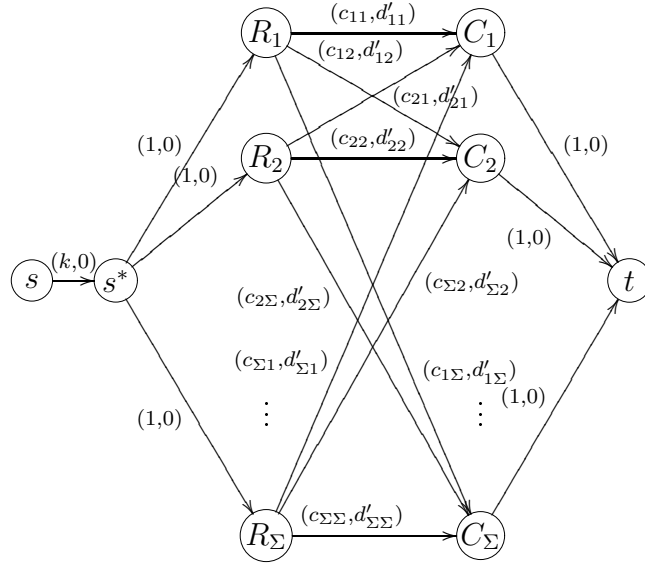


Figure 2: The flow problem equivalent to the maximum distance k -cardinality matching

The minimum cost maximum flow can be found by solving the following optimization with linear programming, that is, the problem is in other words:

$$\max \sum_{i=1}^{|\Sigma|} \sum_{j=1}^{|\Sigma|} c_{ij} d_{ij}, \quad (2)$$

$$\sum_{j=1}^{|\Sigma|} c_{ij} \leq 1, i = 1, \dots, |\Sigma|, \quad (3)$$

$$\sum_{i=1}^{|\Sigma|} c_{ij} \leq 1, j = 1, \dots, |\Sigma|, \quad (4)$$

$$\sum_{i=1}^{|\Sigma|} \sum_{j=1}^{|\Sigma|} c_{ij} = k, \quad (5)$$

where $c_{ij} \in \{0, 1\}$. This problem has been defined as the k -cardinality assignment problem by Dell’Amico and Martello and has been shown to be a P-space problem in [DAM97]

I compared this string edit distance based test selection method with two other test selection method on the INRES system and the Conference Protocol. Test sets derived manually and automatically are used in this investigation. One of these two methods is the fault coverage based test selection method proposed in [Pap06], where the test case requirements are considered to be faults injected systematically into the system according to the given fault model. The other is a transition coverage based method that regards the checking of a transition of an FSM as a test case requirement. Hence the two SDL systems are unfolded into FSM.

Test set	System	Method	Number of selected cases	Number of faults detected	Transitions covered
Automatic (8)	INRES	String	4/8	19/25	11/13
		Transition	3/8	19/25	11/13
		Fault	4/8	23/25	11/13
Manual (13)	INRES	String	6/13	19/25	6/13
		Transition	8/13	19/25	10/13
		Fault	4/13	23/25	8/13
Automatic (40)	Conference	String	14/40	59/78	39/55
		Transition	11/40	59/78	40/55
		Fault	6/40	60/78	25/55

Table 1: Results of the selection experiments

In general according to Table 1, the smallest reduced test and the highest fault coverage ration is provided by the fault based method. The highest transition coverage can be achieved by the transition coverage based method. The string based method provides results close to the transition coverage based method, but this is automatic and requires much less time to execute than the automatic fault based method.

The strength of this method is that is a polynomial time solution. Its weakness is that because of the ε threshold it may not be able to reduce the test set as other coverage metrics.

An open research problem connected to this thesis group is the efficient mapping of test cases to strings.

The results presented in Chapter 4 of the thesis are published in [J3, C13, C4].

4.3 Iterative test specification

Thesis group 3 [J1, C9] *I defined abstract iterative algorithms on the pattern of evolutionary algorithms to reduce the computation cost of selective automatic test generation. The method is also able to support the maintenance of test set over iteration cycles. I introduced test suite metrics and used them as optimization criteria. I conducted empirical analysis to evaluate the efficiency of the method.*

The large computation demands of the test selection method proposed in the first thesis group can be reduced. Recently several research papers [WW06, LI07, KHS08] proposed test data selection solutions with evolutionary methods.

My method is based on test set metrics and the heuristics of partial matching among the set of test cases and a set of test case properties. At a time the optimum is determined for a subset of the test case set, and in each iteration cycle the metrics of the old and new sets are compared.

I defined two abstract algorithms. Both use the same set of operations: derive a new test case (**derive**), compute the cost matrix of the matching problem (**compute**), solve the matching problem (**reduce**), compose two matching problems (**append**). The inputs of the algorithms are an EFSM and a stop condition, the output is a test set with abstract test cases.

In Algorithm 3 selectivity is achieved by not adding test cases considered to be redundant according to a given metric. The inputs are a specification machine, a feature set and a stop condition. The latter is used to break the iteration cycle. The output is a set of test cases.

Test selections problems have been shown to be NP-hard, hence it takes significantly less time to run the optimization for small test suites, rather than to optimize a complete test suite. Algorithm 4 uses this observation to generate a test suite that has already been optimized according to a given metric making further test selection unnecessary.

I concretized the algorithms above by introducing fault coverage based and string edit distance based test suite metrics.

The fault coverage based concretization is based on the `fails_after` function defines as follows: `fails_after` : $\Sigma \times M \rightarrow \mathbb{N}$ a `fails_after`. Let $\forall \sigma \in \Sigma, m \in M$: `fails_after`(σ, m) = $l \iff \text{obs}(\text{substring}(\sigma, l), m) = \text{fail}, \text{obs}(\text{substring}(\sigma, l - 1), m) = \text{pass}$, where $l \leq \text{length}(\sigma)$, $l \in \mathbb{N}$. Hence `obs`(σ, f) = `pass` $\iff \text{fails_after}(\sigma, m) = \text{length}(\sigma)$.

Algorithm 3: Iterative test derivation

```
input :  $m, F, stop$ 
output:  $\Sigma$ 
/* the actual test suite in cycle  $k$ , a matrix of integer values
   in cycle  $k$  */
1 data( $\Sigma, \Phi$ );
   /* Initialization */
2  $k := 0$ ;
3  $Sigma[0] := \emptyset$ ;
4  $\Phi[0] := 0$ 
   /* The  $k^{\text{th}}$  iteration cycle: */
5 repeat
6   switch random choice do
7     case modify existing cases
8       foreach  $\sigma \in \Sigma[k - 1]$  do  $\sigma := \text{derive}(m, \sigma)$ ;
9     case generate a new case
10       $\sigma := \text{derive}(m)$ ;
11       $\Sigma[k] := \Sigma[k - 1] \cup \{\sigma\}$ ;
12   endsw
13   Compute  $\Phi[k]$  based on the actual  $\Sigma[k]$  and  $F \cup \{m\}$ ;
14   if  $\Phi[k] \stackrel{(F)}{>} \Phi[k - 1]$  then
15      $\Sigma := \Sigma[k]$ ;
16   endif
17 until  $stop = \text{false}$  ;
```

Algorithm 4: Iterative test generation with immediate optimization

```
input :  $m, F, stop$ 
output:  $\sigma$ 
/* the actual test suite in cycle  $k$ , a matrix of integer values
   in cycle  $k$ , a matrix of boolean values in cycle  $k$  */
1 data( $\Sigma, \Phi, \mathbf{C}$ );
   /* Initialization */
2  $k := 0$ ;
3  $\Sigma[0] := \emptyset$ ;
4  $\Phi[0] := 0$ ;
5  $\mathbf{C}[0] := 0$ ;
   /* The  $k^{\text{th}}$  iteration cycle: */
6 repeat
7   switch random choice do
8     case modify existing cases
9       foreach  $\sigma \in \Sigma[k - 1]$  do  $\sigma := \text{derive}(m, \sigma)$ ;
10    case generate a new case
11       $\sigma := \text{derive}(m)$ ;
12       $\Sigma[k] := \Sigma[k - 1] \cup \{\sigma\}$ ;
13    endsw
14    Compute  $\Phi[k]$  based on the actual  $\Sigma[k]$  and  $F \cup \{m\}$ ;
15    Compute  $\mathbf{C}[k]$  from  $\Phi[k]$ ;
16    Let
17       $\Sigma[k] := \text{reduce}(\text{append}(\Sigma[k - 1], \Sigma[k]), \text{select}(\text{append}(\mathbf{C}[k - 1], \mathbf{C}[k])))$ ;
18      Let  $\Phi[k] := \text{reduce}(\text{append}(\Phi[k - 1], \Phi[k]), \text{select}(\text{append}(\mathbf{C}[k - 1], \mathbf{C}[k])))$ ;
18 until  $stop = false$  ;
```

Matrix Φ can be computed using the `fails_after` function: $\Phi_{ij} := \text{fails_after}(\sigma_i, \phi_j), \forall i, j \in \mathbb{N}, 1 \leq i \leq |\Sigma|, 0 \leq j \leq |\Phi|$. Therefore the input of the fault based test selection method – proposed in Pap’s thesis [Pap06] – is: $\forall i, j \in \mathbb{N}, 1 \leq i \leq |\Sigma|, 1 \leq j \leq |\Phi|$:

$$\mathbf{C}_{ij} = \begin{cases} 0 & \iff \Phi_{i0} = \Phi_{ij} \iff \text{fails_after}(\sigma_i, m) = \text{fails_after}(\sigma_i, \phi_j). \\ 1 & \iff \Phi_{i0} \neq \Phi_{ij} \iff \text{fails_after}(\sigma_i, m) \neq \text{fails_after}(\sigma_i, \phi_j). \end{cases}$$

The fault coverage of the test set Σ can be computed using these Φ_Σ and \mathbf{C}_Σ matrices. The fitness functions below consider the following properties of Σ : fault detection ability, average length of test cases, number of test cases, uniformity in fault detection ability.

1. The number of faults detected by the Σ test set is a number that can be used to evaluate a test set. Let $cr_1 : \mathbf{C} \rightarrow \mathbb{N}$ be a fitness function, where \mathbf{C} is a matrix of boolean values:

$$cr_1(\mathbf{C}_\Sigma) = \sum_{j=1}^{|\Sigma|} \text{sgn}\left(\sum_{i=1}^{|\Phi|} c_{\Sigma ij}\right),$$

where $\text{sgn}(x)$ is the signum function: $x < 0 \Rightarrow \text{sgn}(x) = -1, \text{sgn}(0) = 0, x > 0 \Rightarrow \text{sgn}(x) = 1$.

2. Let the fitness function $cr_2 : \Sigma \rightarrow \mathbb{R}$ be a fitness function that calculates the the average length of the test cases in Σ :

$$cr_2(\Phi_\Sigma) = \frac{\sum_{i=0}^{|\Phi|} \sum_{j=1}^{|\Sigma|} \phi_{\Sigma ij}}{|\Sigma|}$$

3. Let $cr_3 : \Sigma \rightarrow \mathbb{R}$ be a fitness function for the test set Σ such that: $cr_3(\Sigma) = |\Sigma|$.
4. Let $cr_4 : \mathbf{C}_\Sigma \rightarrow \mathbb{R}$ be a fitness function defined by the uniformity of the fault detection capability of Σ :

$$cr_4(\mathbf{C}_\Sigma) = \frac{\sum_{j=1}^{|\Sigma|} \left(\sum_{i=1}^{|\Phi|} c_{\Sigma ij}\right)^2}{|\Sigma|}$$

I conducted simulation based experiments on the INRES system, Conference Protocol, WAP WTP and SS7 MTP2 to check the efficiency of fault coverage based

iterative test generation methods. I found that the proposed iterative methods reduce the test generation time while they have similar fault coverage values and similar resulting test set sizes as the non-iterative method composed of the generation of a huge test set and performing a test selection.

I showed the string edit distance based iterative automatic test generation method on the example of the INRES system. I used the following string edit distance based metric to evaluate the test set Σ : The pair $(\mu(\Sigma), dd)$ defines a metric space, where Σ is the set of test cases over the $I \cup O$ alphabet, μ is a marked trace transformation of a test case and $dd : \Sigma \times \Sigma \rightarrow \mathbb{R}$ is a distance function such that $\forall \Sigma_1, \Sigma_2 \subseteq \Sigma$:

$$dd(\Sigma_1, \Sigma_2) = \left| \sum_{\sigma_i \in \Sigma_1, \sigma_j \in \Sigma_1} d(\mu(\sigma_i), \mu(\sigma_j)) - \sum_{\sigma_i \in \Sigma_2, \sigma_j \in \Sigma_2} d(\mu(\sigma_i), \mu(\sigma_j)) \right|$$

The strength of the proposed solution is that it can make use of theoretical coverage metrics and reduce the computation time needed to automatically generate already reduced test sets. Its weakness is that the generated set may be larger as the one generated without the evolutionary cycle.

The results presented in Chapter 5 are published in [C9, J1, C13, C4].

5 Application of the results

The results of the dissertation support computer aided design of telecommunication protocols. The proposed selective automatic test generation methods are effective for development life-cycle methodologies that involve a formal modelling activity in the design phase.

All three theses focus on the test specification activity with the purpose of automatically generating test sets with low redundancy and high quality. The solution of the first thesis group is an automatic test generation method for SDL specifications. It can be used for generating a limited set of test cases with low redundancy with regard to the fault detection ability. The approach in the second thesis is a quick and effective automatic method for eliminating redundancies in a test set. With further research on test case to string mapping it can be brought to real life practice. The algorithms of the third thesis transform any coverage metric into a selective automatic test generation algorithm, and the iterative cycle reduces the computation demands for evaluating the coverage metrics for all test cases.

The software tool that implements the proposed methods can be used as extension to commercial tools like Telelogic (now IBM) Tau [Tau] to find redundancies among test cases generated automatically with them.

Acknowledgement

I have been fortunate to have the help and support of many people over the last several years. Without their help this dissertation would not have been completed. First of all, I would like to express my deepest gratitude to my supervisors Gyula Csopaki, who has been supervising my work since my undergraduate studies, and Katalin Tarnay for their encouraging guidance and invaluable support. I wish to thank the High Speed Networks Laboratory for the support of my research activities in the field of protocol engineering over so many years, the possibility of participating in the project MUSE and the long lasting patience. I would also like to acknowledge the help I received from the people at Department of Telecommunications and Media Informatics. My special thanks to my past and present colleagues at the protocol engineering group for the fruitful collaborative work, and to the undergraduate students I supervised for their help in developing the software tools I use in the dissertation.

References

- [AB99] P.E. Ammann and P.E. Black. *A Specification-based Coverage Metric to Evaluate Test Sets*. In *Proceedings of Fourth IEEE International High-Assurance Systems Engineering Symposium (HASE 99)*, pages 239–248, 1999.
- [ABM98] P.E. Ammann, P.E. Black, and W. Majurski. *Using Model Checking to Generate Tests from Specifications*. In *Second IEEE International Conference on Formal Engineering Methods*, pages 46–54, 1998.
- [BDA] C. Bourhfir, R. Dssouli, and E.M. Aboulhamid. *Automatic Test Generation for EFSM-based Systems*. <http://citeseer.nj.nec.com/114451.html>.
- [BH89] L. Bromstrup and D. Hogrefe. Tesdl: Experience with generating test cases from sdl specifications. In *Fourth SDL Forum*, pages 267–279, 1989.
- [BOY00] P.E. Black, V. Okun, and Y. Yesha. *Mutation Operators for Specifications*. In *The Fifteenth IEEE International Conference on Automated Software Engineering, Proceedings ASE 2000*, pages 81–88, 2000.
- [CLRS01] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, second edition, 2001.
- [DAM97] M. Dell’ Amico and S. Martello. The k-cardinality assignment problem. *Discrete Applied Mathematics*, 76:103–121, 1997.
- [DMLS78] R.A. De Millo, R.J. Lipton, and F.G. Sayward. *Hints on Test Data Selection: Help for the Practicing Programmer*. *IEEE Computer*, 11(4):34–41, April 1978.
- [DÜU04] Ali Y. Duale and M. Ümit Uyar. A method enabling feasible conformance test sequence generation for efsm models. *IEEE Transactions on Computers*, 53(5):614–627, 2004.
- [FGMT02] L.M.G. Feijs, N. Goga, S. Mauw, and J. Tretmans. Test selection, trace distance and heuristics. In *Testing Communication Systems XIV*, pages 267–282, Berlin, Germany, 2002.
- [FMDM94] S.C.P.F. Fabbri, J.C. Maldonado, M.E. Delamaro, and P.C. Masiero. *Mutation Analysis Testing for Finite State Machine*. In *Proc. ISSRE’94*

- *Fifth International Symposium on Software Reliability Engineering*, pages 220–229, California, USA, 1994.
- [FMM⁺95] S.C.P.F. Fabbri, J.C. Maldonado, P.C. Masiero, M.E. Delamaro, and E. Wong. *Mutation Testing Applied to Validate Specifications Based on Petri Nets. FORTE'95 - 8th International IFIP Conference on Formal Description Techniques for Distributed Systems and Communications Protocol*, 1995.
- [FMSM99] S.C.P.F. Fabbri, J.C. Maldonado, T. Sugeta, and P.C. Masiero. *Mutation Testing Applied to Validate Specifications Based on Statecharts*. In *Proc. ISSRE'99 - 10th International Symposium on Software Reliability Engineering*, pages 210–219, Florida, USA, 1999.
- [Hol91] G.J. Holzmann. *Design and validation of computer protocols*. Prentice Hall, 1991.
- [IT97] ITU-T. *Recommendation Z.100 – Supplement 1: SDL+ methodology, Use of MSC and SDL (with ASN.1)*, 1997.
- [JJ02] C. Jard and T. Jeron. TGV: Theory, principles and algorithms. In *6th World Conference on Integrated Design and Process Technology (IDPT 2002)*, June 2002.
- [KHS08] A. Kalaji, R.M. Hierons, and S. Swift. Automatic generations of test sequences from EFSM models using evolutionary algorithms. Technical report, Brunel University, <http://hdl.handle.net/2438/2630>, 2008.
- [Kim08] T-H. Kim. Test generation for a protocol specified in sdl with complex loops by event-based efsm modeling. *IJCSNS International Journal of Computer Science and Network Security*, 8(3), March 2008.
- [LI07] R Lefticaru and F. Ipate. Automatic state-based test generation using genetic algorithms. In *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 188–195, Timisoara, 26-29 Sept 2007.
- [LY96] D. Lee and M. Yiannakakis. *Principles and Methods of Testing Finite State Machines – A Survey. Proc. of the IEEE*, 43(3):1090–1123, 1996.
- [Pap06] Z Pap. *Detection, diagnosis and correction of faults in telecommunications software development based on formal methods*. PhD thesis, Budapest University of Technology and Economics, 2006.

- [PG91] R.L. Probert and F. Guo. *Mutation Testing of Protocols: Principles and Preliminary Experimental Results*. In *Proc. Protocol Test Systems, III.*, pages 57–76, 1991.
- [SKGH97] M. Schmitt, B. Koch, J. Grabowski, and D. Hogrefe. Autolink - a tool for the automatic and semi-automatic test generation. In *Seventh GI/ITG Technical Meeting on Formal Description Techniques for Distributed Systems*, 1997.
- [SMFLDS00] S.R.S. Souza, J.C. Maldonado, S.C.P.F. Fabbri, and W. Lopes De Souza. *Mutation Testing Applied to Estelle Specifications*. *Software Quality Journal*, 8(04), 2000.
- [Tau] Telelogic Tau. <http://www.telelogic.com>.
- [TB03] G.J. Tretmans and H. Brinksma. Torx: Automated model-based testing. In *First European Conference on Model-Driven Software Engineering*, pages 31–43, Nuremberg, Germany, December 11-12 2003.
- [Tre00] J. Tretmans. *Specification Based Testing with Formal Methods: A Theory*. In A. Fantechi, editor, *FORTE / PSTV 2000 Tutorial Notes*, Pisa, Italy, October 10 2000.
- [VAC92] Son T. Vuong and Jadranka Alilovic-Curgus. On test coverage metrics for communication protocols. In *Proceedings of the IFIP TC6/WG6.1 Fourth International Workshop on Protocol Test Systems IV*, pages 31–45, 1992.
- [vBP94] G. v. Bochmann and A. Petrenko. *Protocol Testing: Review of Methods and Relevance for Software Testing*. In *ISSTA*, pages 109–124, 1994.
- [WF74] R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, January 1974.
- [WL93] C.-J. Wang and M. T. Liu. *Generating Test Cases for EFSM with Given Fault Models*. In *INFOCOM 93*, volume 2, pages 774–781, 1993.
- [WRQB08] W.E. Wong, A. Restrepo, Y. Qi, and Choi B. *An EFSM-based Test Generation for Validation of SDL specifications*. In *Automation of Software Test 2008*, pages 25–32, Leipzig, Germany, May 2008.
- [WW06] S. Wappler and Joachim Wegener. Evolutionary unit testing of object-oriented software using a hybrid evolutionary algorithm. In G.G. Yen, L. Wang, P. Bonissone, and S.M. Lucas, editors, *2006 IEEE Congress*

on Evolutionary Computation, pages 3193–3200, Vancouver, 6-21 July 2006. IEEE Press.

Publications

Journal articles

- [J1] G. Kovács, Z. Pap, G. Csopaki, and K. Tarnay, “Iterative automatic test generation method for telecommunication protocols,” *Computer Standards and Interfaces*, vol. 28, no. 4, pp. 412–427, 2006.

Journal articles appeared in Hungary

- [J2] G. Kovács, Z. Pap, and G. Csopaki, “Automatic Test Selection based on CEFSM Specifications,” *Acta Cybernetica*, vol. 15, pp. 583–599, 2002.

Journal papers in hungarian

- [J3] G. Kovács, “Teszteset válogatás távolság metrikával,” *Híradástechnika*, vol. LXI, pp. 41–43, January 2006.

Lecture Notes in Computer Science papers

- [C1] G. Csopaki, G. Horváth, and G. Kovács, “Communication Protocol Implementation in Java,” in *Lecture Notes in Computer Science 1905: Interactive Distributed Multimedia Systems and Telecommunication Services*, (Enschede, The Netherlands), pp. 254–265, October 17-20 2000.
- [C2] G. Kovács, Z. Pap, D. Le Viet, A. Wu-Hen-Chang, and G. Csopaki, “Applying Mutation Analysis to SDL Specifications,” in *11th International SDL Forum: SDL 2003: System Design* (R. Reed and J. Reed, eds.), (Stuttgart, Germany), pp. 269–284, 2003.
- [C3] Z. Pap, M. Subramaniam, G. Kovács, and G. Á. Németh, “A bounded incremental test generation algorithm for finite state machines,” in *19th IFIP Testing of Communicating Systems (TestCom/FATES)*, (Tallinn, Estonia), pp. 244–259, June 26-29 2007.
- [C4] G. Kovács, G. Á. Németh, M. Subramaniam, and Z. Pap, “Optimal string edit distance based test suite reduction for SDL specifications,” in *LNCS 5719: 14th International SDL Forum: SDL 2009* (R. Reed, A. Bilgic and R. Gotzhein, eds.), (Bochum, Germany), pp. 82-97, September 22-24 2009.

Conference papers

- [C5] G. Gordos, G. Csopaki, Z. Werner, G. Horvath, G. Kovacs, and T. Kerecsen, “Automatic converion of SDL into modern oop languagues with a Java-based compiler,” in *International Workshop on Intelligent Communication Technologies and Applications*, (Neuchatel, Switzerland), May 5-7 1999.
- [C6] G. Kovács, Z. Pap, and G. Csopaki, “Automatic Test Selection based on CEFSM Specifications,” in *7th Symposium on Programing Languages and Software Tools* (T. Gyimothy, ed.), (Szeged, Hungary), pp. 84–97, June 15-16 2001.
- [C7] G. Kovács, Z. Pap, and G. Csopaki, “Automatic test selection method applied to WAP,” in *EUNICE 2001*, (Paris, France), September 3-5 2001.
- [C8] G. Csopaki, T. Kasza, G. Kovács, and M. Szücs, “Applicability of UML in the protocol development process,” in *Polish-Czech-Hungarian Workshop on Circuit Theory, Signal Processing and Telecommunication Services*, (Budapest), September 14-17 2001.
- [C9] G. Kovács, D. Le Viet, and A. Wu-Hen-Chang, “Iterative automatic test generation,” in *EUNICE 2003*, (Balatonfüred), September 8-10 2003.
- [C10] C. Lukovszki, L. Kovács, G. Kovács, A. Foglar, E. Areizaga, and Z. Ghebretesae, “Revolutionary ipv6 optimized access solution,” in *11th European Conf. on Networks and Optical Communications (NOC)*, (Berlin, Germany), July 11-13 2006.
- [C11] H. Mickelsson, A. van Neerbos, P. Nooren, M. Prins, K. Oberle, D. Jocha, B. Radier, G. Kovács, M. Thakur, I. Pinilla, E. Areizaga, A. Gamelas, and A. Sitek, “Nomadism/fmc use cases and aaa impact,” in *BroadBand Europe*, (Geneva, Switzerland), December 12-14 2006.
- [C12] Z. Kardkovács, E. Lejtovicz, and G. Kovács, “Context identification: A relational database approach,” in *The 3rd Language & Technology Conference* (Z. Vetulani, ed.), (Poznan, Poland), pp. 211–215, October 1-5 2007.
- [C13] G. Kovács, G. Á. Németh, Z. Pap, and M. Subramaniam, “Deriving compact test suites for telecommunication software using distance metrics,” in *Software, Telecommunications and Computer Networks (SoftCOM)*, (Split-Dubrovnik, Croatia), September 25-27 2008.

Other publications

- [O1] D. Jocha and G. Kovács, “Introduction to the MUSE FMC architecture,” in *MUSE Summer School*, (Budapest), July 5 2007.

Citations

- [1] T. Sugeta, J. Maldonado, and W. Wong, “Mutation testing applied to validate sdl specifications,” in *Testing of Communicating Systems. 2004.03.17-2004.03.19 193-208. 2004.*, (Oxford, England), pp. 193–208, March 17-19 2004.
- [2] M. Hong and Z. Lu, “A framework for testing web services and its supporting tool,” in *SOSE 2005: IEEE International Workshop on Service-Oriented System Engineering*, (Beijing, China), pp. 199–206, October 20-21 2005.
- [3] Y. Jiang, S. Hou, J. Shan, Z. L., and B. Xie, “Contract-based mutation for testing components,” in *ICSM 2005: 21st IEEE International Conference on Software Maintenance*, (Budapest, Hungary), pp. 483–492, September 26-29 2005.
- [4] Y. Jiang, S. Hou, J. Shan, L. Zhang, and B. Xie, “An approach to testing black-box components using contract-based mutation,” *INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING 18: (1) 93-117 (2008)*, vol. 18, no. 1, pp. 93–117, 2008.
- [5] Z. Andriska, G. Bátor, D. Le Viet, A. Wu-Hen-Chang, and G. Csopaki, “Tool for automatic test selection based on formal specification,” *Híradástechnika*, vol. LVII, pp. 37–43, December 2002.
- [6] K. Oberle, S. Wahl, and A. Sitek, “Enhanced methods for SIP based session mobility in a converged network,” in *16th Mobile and Wireless Communications Summit*, (Budapest, Hungary), pp. 1–5, July 1-5 2007.