

# Deriving Compact Test Suites for Telecommunication Software Using Distance Metrics

Gábor Kovács, Gábor Árpád Németh, Zoltán Pap and Mahadevan Subramaniam

Original scientific paper

**Abstract**—This paper proposes a string edit distance based test selection method to generate compact test sets for telecommunication software. Following the results of previous research, a trace in a test set is considered to be redundant if its edit distance from others is less than a given parameter. The algorithm first determines the minimum cardinality of the target test set in accordance with the provided parameter, then it selects the test set with the highest sum of internal edit distances. The selection problem is reduced to an assignment problem in bipartite graphs.

**Index terms:** test set maintenance, test selection, string edit distance, assignment problem

## I. INTRODUCTION

Despite the well established theoretical background, the introduction of automatic abstract test generation methods into the telecommunication software development process has been difficult under industrial requirements. One of the main problems is the extensive size of automatically generated test sets, which are often impossible to run in practice.

A major challenge of automatic test generation is the detection of redundancies among the large number of test cases derived from a – formal – model. The purpose of automatic test selection methods is to address this problem; their objective is to minimize the cardinality of the target test set without sacrificing its quality.

There are different measures of quality used to guide the reduction process like symbol (path) coverage [1], fault coverage based on a fault model [2], test sub-purpose coverage [3] or trace distance heuristics [4]. The selection problem can be solved by exact algorithms such as linear programming methods [5] [6], but considering the large number of generated test cases and the hardness of the selection problem usually approximate solutions, for example genetic algorithms [3], are applied.

This paper follows the idea proposed in [1] and extended in [4]. The former paper introduces a test coverage metric based on the concept of testing distance between traces. This metric

The material in this paper was presented in part at the 16<sup>th</sup> International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2008), Split - Dubrovnik, Croatia, Sept. 2008.

G. Kovács, G. Á. Németh and Z. Pap are with the Dept. of Telecommunications and Media Informatics, Budapest University of Technology and Economics (e-mail: {kovacs, gabor.nemeth, pap}@tmit.bme.hu).

M. Subramaniam is with the Computer Science Department University of Nebraska at Omaha (e-mail: subramaniam@mail.unomaha.edu)

is used to approximate differences among patterns of system behavior. Traces are considered to be similar (redundant) if they can be transformed to each other with a cost no more than a given  $\varepsilon$  parameter. The latter paper generalizes the original idea by introducing a cycling heuristic: The notion of marked traces is proposed to tackle the problem of traces revisiting states of the system, and formulae are given to precisely calculate distance between traces containing loops.

The current paper extends Vuong's and Feijs' approach as described in [1] and [4]. Two selection criteria are introduced to find the most compact test set with the highest possible diversity with regard to the distance based coverage metric. First the minimum cardinality of the test set for a given  $\varepsilon$  parameter is determined by reducing the distance based selection problem to an assignment problem in bipartite graphs. Then a test set with the highest overall internal edit distance is selected with the help of the  $k$ -assignment problem defined by Dell'Amico and Martello in [7].

The rest of the paper is organized as follows. A brief overview of our assumptions and notations is given in Section 2. Section 3 describes the procedure of distance maximization among the test cases. In Section 4 a summary is given.

## II. PRELIMINARIES

Finite state machines (FSMs) have been widely used for decades to model telecommunication software. In Mealy machines an output event, which is like sending a packet, is produced on a state transition triggered with an input event, which is like a reception of a packet. Several specification languages, such as SDL [8] and ESTELLE [9], are extensions of the FSM formalism.

A finite state machine  $M$  is a quadruple [10]  $M = (I, O, S, h)$  where  $I$  is the finite set of input symbols,  $O$  is the finite set of output symbols,  $S$  is the finite set of states,  $h: D \rightarrow 2^{O \times S}$  is a behavior function where  $D \subseteq S \times I$  is the specification domain and  $2^{O \times S}$  is the set of all subsets of the set  $S \times O$ .

In case the specification domain  $D = S \times I$ , the behavior function is defined for all state-input combinations and the FSM  $M$  is said to be completely specified (or completely defined). If for each pair  $(s, i) \in D$  it holds that  $|h(s, i)| \leq 1$  then FSM  $M$  is said to be deterministic. In case of a deterministic FSM we may separate the behavior function  $h$  into two functions, the next state function  $\delta: S \times I \rightarrow S$  and the output function  $\lambda: S \times I \rightarrow O$ .

Let the input string  $x = i_1, \dots, i_k \in I^*$  take FSM  $M$  successively through the states  $s_1, \dots, s_{k+1}$  such that  $s_{j+1} = \delta(s_j, i_j)$ ,  $j = 1, \dots, k$ , and produce the output string  $\lambda(s_1, x) = o_1, \dots, o_k \in O^*$ , where  $k \in \mathbb{N}$ ,  $1 \leq k < \infty$ . The input/output sequence  $i_1 o_1 i_2 o_2 \dots i_k o_k$  is then called a trace of  $M$ . Note that if an FSM is deterministic then all its traces are deterministic, because there are no transitions with different next states and/or outputs for the same state-input combination.

We say that machine  $M$  has a reset capability if there is an initial state  $s_0 \in S$  and an input symbol  $r \in I$  that takes the machine from any state back to  $s_0$ . That is,  $\delta(s_j, r) = s_0$  for all states  $s_j \in S$ . The reset is reliable if it is guaranteed to work properly in any implementation machine  $M'$ , i.e.,  $\delta'(s'_j, r) = s'_0$  for all states  $s'_j \in S'$ , otherwise it is unreliable. Note that reset  $r$  is an input symbol as well. Thus, if  $M$  has reset then  $M$  is considered to be strongly connected if all the other states can be reached from the initial state  $s_0$ .

#### A. Test Cases and their String Representation

If a specification of the system is given as a finite automaton then its traces may be used as test cases. Even if the specification has finite states, the set of executable traces may be in practice infinite and therefore exhaustive testing based on all traces is generally impossible. The purpose of test selection (and test generation in its essence) is to identify a subset of traces sufficient to establish a required level of confidence in the correctness of the system.

Throughout this paper we consider systems with reliable reset capability. A test set consists of a set of test cases, where each test case is a finite trace starting with a reset input. Traces are generated by means of random-walk, but any other test derivation algorithm may be considered as well. Note that in this paper we use the terms test sets and trace sets interchangeably.

Traces can be represented as strings on an arbitrary alphabet  $C$ . A mapping  $M : \{I \cup O\}^+ \rightarrow C$  defines a set of pairs of an event sequence and a string of alphabet  $C$ :  $\langle (x_{i1} x_{i2} \dots x_{im}), c_i \rangle$ , where  $x_{ik} \in I \cup O$ ,  $k = 1 \dots m$ ,  $k \in \mathbb{N}$ ,  $c_i \in C$ . Note that according to this definition more than one successive events may be mapped to a character of the given alphabet. Such mapping  $M$  is the marked trace notation defined in [4].

**Example** Consider the system in Figure 1. The machine has three states, the initial state is  $s_0$ . Let its event alphabet be  $I \cup O = \{a, b, c\}$ . Let the alphabet be  $C = \{V, W, X, Y, Z\}$  and let the mapping  $M$  between the event and character alphabets be defined by the table below. For instance  $aa$  and  $bc$  are loops in the state machine around  $s_0$ .

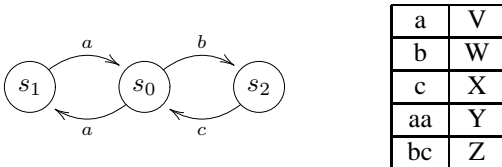


Figure 1. A sample FSM and its event to character mappings

Let  $T = \{abca, aabb, bcaa, abab\}$  be a test set, its elements are denoted with  $t_i$ ,  $i = 1, 2, 3, 4$ . The initial state of the test case derivation is  $s_0$ . The string set  $S$  corresponding to  $T$  and defined by the mapping  $M$  contains the following strings:  $S = \{VZV, YWW, ZY, VWVW\}$ .

#### B. Trace distance

The edit distance between strings  $\sigma_1$  and  $\sigma_2$  is the minimum number of edit operations (character insert, character delete and character overwrite) needed to transform  $\sigma_1$  to  $\sigma_2$ . Different edit operations may have distinct costs assigned, but for the sake of simplicity we consider unit edit operator costs in the current paper.

Let  $\Sigma$  be a finite set of strings over the alphabet  $C$ .

$d : \Sigma \times \Sigma \rightarrow \mathbb{R}$  is a distance metric if  $d(\sigma_1, \sigma_1) = 0$ ,  $d(\sigma_1, \sigma_2) = d(\sigma_2, \sigma_1)$  and  $d(\sigma_1, \sigma_3) \leq d(\sigma_1, \sigma_2) + d(\sigma_2, \sigma_3)$ , for all  $\sigma_1, \sigma_2, \sigma_3 \in S$ . If the lengths of  $\sigma_1$  and  $\sigma_2$  are  $l_1$  and  $l_2$  respectively, the time and space complexity of computing the distance is  $O(l_1 l_2)$ . The distance metric computation above is according to one defined in [11], but other approaches can be considered as well.

**Definition 2.1: Distance matrix.** Let  $\mathbf{D} = [d_{ij}]$  be a distance matrix of the set of traces  $T$ . Let  $d_{ij} = d(M(t_i), M(t_j))$ , where  $1 \leq i, j \leq |T|$ ,  $i, j \in \mathbb{N}$ .

Note that  $\mathbf{D}$  is symmetric because of the symmetric nature of distance and the size of the matrix is  $|T| \times |T|$ . Let  $\mathbf{D}_U$  and  $\mathbf{D}_L$  denote the strictly upper and lower triangular matrices of  $\mathbf{D}$  respectively.

The metric space involves the normalization of edit distance values as it is done in [1] and [4]. For the sake of simplicity – and without any loss of generality – in this paper we dispense with the normalization and consider that the distances are non-negative integer numbers.

**Example** The distance matrix of trace set  $T$  in the previous example is:

$$\mathbf{D} = \begin{bmatrix} 0 & 3 & 2 & 2 \\ 3 & 0 & 3 & 2 \\ 2 & 3 & 0 & 4 \\ 2 & 2 & 4 & 0 \end{bmatrix}. \quad (1)$$

### III. MAXIMIZING THE DIVERSITY IN THE SETS

This section proposes a method for selecting a test set with the highest possible diversity with regard to the distance based coverage metric. The inputs of the method are the  $\mathbf{D}$  distance matrix of the traces, and an  $\varepsilon$  parameter. The method consists of two stages: First the minimum cardinality of the target test set is calculated assuming the  $\varepsilon$  parameter, then the test set with the maximum internal distance is selected.

For our discussions we assume the notion of  $\varepsilon$ -approximation defined by Feijs et. al. in [4]:

**Definition 3.1:  $\varepsilon$ -cover.**  $T'$  is an  $\varepsilon$ -cover of  $T$ , where  $T' \subseteq T$ ,  $\varepsilon \geq 0 \iff \forall t \in T : \exists t' \in T' : d(M(t), M(t')) \leq \varepsilon$ .

Definition 3.1 implies that for each  $T$  and  $\varepsilon$  there exists at least one minimal cardinality  $T'$   $\varepsilon$ -cover of  $T$ , for which  $\forall t'_i, t'_j \in T' : d(M(t'_i), M(t'_j)) > \varepsilon$ .

The next definition (Definition 3.2) defines a graph representation of the distance matrix  $\mathbf{D}$  showing which traces  $\varepsilon$ -cover

each other. According to Lemma 3.3 the minimum cardinality  $\varepsilon$ -coverage problem – finding the greatest possible reduction of  $T$  – is reduced to a matching problem in bipartite graphs. Theorem 3.4 shows that the maximum matching can be found in polynomial time of the size of the trace set.

**Definition 3.2: Constructing a bipartite graph from a matrix.** Let  $G' = (V', E')$  be an undirected graph constructed from a matrix  $\mathbf{A} = [a_{ij}]$  of size  $R \times C$ , where  $R$  and  $C$  denote the cardinality of its row and column index sets, respectively. Let  $G'$  comprise vertices corresponding to rows and columns of  $\mathbf{A}$ :  $V' = V'_R \cup V'_C$ . Let  $(v'_i, v'_j) \in E' \iff a_{ij} > 0$ , where  $v'_i \in V'_R, v'_j \in V'_C$ . Thus  $G'$  is a bipartite graph with  $|V'| \leq R + C$  vertices and  $|E'| \leq R * C$  edges.

Note that if  $\mathbf{A}$  is a distance matrix  $\mathbf{D}$  according to Definition 2.1 then  $R = C = |T|$ .

Finding a maximum cardinality assignment  $\mu$  is identical to finding the maximum number of independent edges in the bipartite graph  $G'$ . The  $\mu$  in the bipartite graph  $G' = (V', E')$  is a function:  $E' \rightarrow \{0, 1\}$ .  $\mu(e_i) = 1 \iff e_i$  is selected from  $E'$  by a maximum cardinality assignment algorithm (e.g. Ford-Fulkerson, Hopcroft-Karp or Edmonds-Karp algorithms [12]).

**Lemma 3.3:** The maximum cardinality assignment  $\mu$  in the bipartite graph  $G'$  selects the maximum number of linearly independent elements of the matrix  $\mathbf{A} = [a_{ij}]$  of size  $n \times n$ .

*Proof:* Definition 3.2 implies that:

- for each  $i$  row of  $\mathbf{A}$  :  $\sum_{j=1}^n \mu((v'_i, v'_j)) = 1$ , hence exactly one element is selected from a row, and
- for each  $j$  column of  $\mathbf{A}$  :  $\sum_{i=1}^n \mu((v'_i, v'_j)) = 1$ , hence exactly one element is selected from a column.

The maximum cardinality of the selected edges implies that maximum number of non-zero elements are selected from  $\mathbf{A}$ , because if  $\mu((v'_i, v'_j)) = 1, v'_i, v'_j \in V'$  then  $a_{ij} > 0$ . ■ ■

In the following we show that among the matches in the bipartite graph there is at least one with the minimum cardinality, which means that there exists a maximum reduction of  $T$  with regard to  $\varepsilon$  as shown in the next theorem.

**Theorem 3.4:** A minimum cardinality  $\varepsilon$ -cover  $T'$  of the test set  $T$  with the given distance matrix  $\mathbf{D}$  can be found in  $O(|T|^{5/2})$ .

*Proof:* The test cases in test set  $T$  can be divided into two disjoint subsets: subset of test cases that can and ones that can not be  $\varepsilon$ -covered by other test cases. The ones can not be covered must be included in  $T'$ , and from the rest the minimum number of cases must be selected. Thus a  $T'$  set with the minimum cardinality can be selected by finding a  $T'' = T \setminus T'$  with maximum cardinality. If  $T''$  is a set of maximum cardinality then its elements do not  $\varepsilon$ -cover each other.

Let  $\mathbf{A} = [a_{ij}]$  be a matrix such that  $a_{ij} \in \{0, 1\}$  and  $\forall i, j$  :

$$a_{ij} = \begin{cases} 0 & \iff i = j, \\ 0 & \iff d_{ij} > \varepsilon, i \neq j, \\ 1 & \iff d_{ij} \leq \varepsilon, i \neq j \end{cases},$$

where  $\mathbf{D} = [d_{ij}]$  is the distance matrix of  $T$ . Note that  $\mathbf{A}$  is symmetric, and a 1 value shows that the corresponding test

case is  $\varepsilon$ -covered by an other one. If a row or column contains only 0s then the corresponding test case must be included in  $T'$ . Therefore the new goal is to find a maximum set of linearly independent 1 values in  $\mathbf{A}$ .

Construct  $G'$  from  $\mathbf{A}$  as given in Definition 3.2 and use the Hopcroft-Karp algorithm [12] to find the maximum cardinality assignment  $\mu$  of  $G'$ . Because of Lemma 3.3 the selected elements of  $\mathbf{A}$  are candidates for a  $T''$  with maximum cardinality. For each  $\mu$  in  $G'$  there are two  $T''$ s because of the symmetry of  $\mathbf{A}$ , thus it is enough to consider the selected elements of either its strictly upper triangular matrix  $\mathbf{A}_U$  or its strictly lower triangular matrix  $\mathbf{A}_L$ . Therefore  $t_i \in T''_L \subset T \iff \exists i > j : e = (v'_i, v'_j) \in \mu$  and  $t_i \in T''_U \subset T \iff \exists j > i : e = (v'_i, v'_j) \in \mu$ .

The space complexity is  $|T|^2$ , because the distances are stored in a symmetric matrix of type  $|T| \times |T|$ . The time complexity of the construction of  $\mathbf{A}$  is  $O(|T|^2)$ . The search for identical rows in  $\mathbf{A}$  has  $O(|T|^3)$  time complexity. However, this can be done more effectively with the Rabin-Karp algorithm [12], which operates on the hash values of patterns and finds all matching rows in  $O(|T|)$  while it does not increase the space complexity. The worst-case time complexity of the Hopcroft-Karp algorithm [12] applied to the graph  $G' = (V', E')$  is  $O(\sqrt{|V'|}|E'|)$ . Since  $|V'| \leq 2|T|$  and  $|E'| \leq |T|^2$  because of Definition 3.2, the resulting time complexity is  $O(|T|^{5/2})$ . ■ ■

Let  $\mathbf{A}_{\mu L}$  and  $\mathbf{A}_{\mu U}$  denote the upper and lower triangular matrices containing only the selected linearly independent elements.

**Example** The  $\varepsilon$ -coverage of matrix  $\mathbf{D}$  of the previous example with  $\varepsilon = 2$  is (note that we dispense with normalization):

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \quad (2)$$

The bipartite graph  $G'$  constructed from  $\mathbf{A}$  is in Figure 2. Edges show that a trace  $\varepsilon$ -covers an other test case.

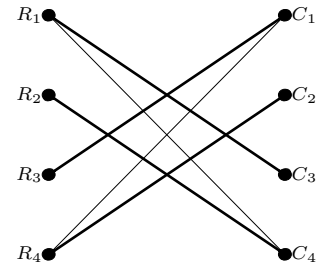


Figure 2. The bipartite graph  $G'$  constructed from matrix  $\mathbf{A}$

Bold lines show a maximum cardinality matching: the selected elements of  $\mathbf{A}$  are  $(1, 3), (2, 4), (3, 1), (4, 2)$ . In general more maximum cardinality assignments may exist, but all with the same cardinality.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \boxed{1} & 1 \\ 0 & 0 & 0 & \boxed{1} \\ \boxed{1} & 0 & 0 & 0 \\ 1 & \boxed{1} & 0 & 0 \end{bmatrix} \quad (3)$$

Hence,  $T_U'' = \{t_1, t_2\}$  and  $T_L'' = \{t_3, t_4\}$ . The resulting  $T'$  candidates are  $T_1' = \{t_3, t_4\}$  and  $T_2' = \{t_1, t_2\}$ .

According to [1] and [4] two patterns of behavior are approximated to be less similar if the distance between their string representation is greater. The redundancy among the test cases in a test set is the least, if the sum of all pairwise distances is maximal. Hence, if more than one minimal cardinality  $T'$  solutions exist, the one with the maximum internal distance should be preferred.

**Theorem 3.5:** Selecting the  $T'$  test set with minimal cardinality from the test set  $T$ , such that  $T'$  is an  $\varepsilon$ -cover of  $T$  and the test cases differ from each other as much as possible can be calculated in a polynomial time of  $|T|$ . This means that the sum of distances between all pairs of the test cases of  $T'$  is maximal, therefore the optimization problem is:

$$\max \sum_{t_i \in T', t_j \in T'} d(M(t_i), M(t_j)), \quad (4)$$

where  $\forall i, j : d(M(t_i), M(t_j)) > \varepsilon$ .

*Proof:* A  $T'$  with the minimum  $k$  cardinality can be determined according to Theorem 3.4. There may be many  $T'$ s with this  $k$  cardinality, and there is at least one with the maximum distance sum.

Let  $\mathbf{C} = [c_{ij}]$ , where  $c_{ij} \in \{0, 1\}$  a capacity matrix, and let the distance matrix  $\mathbf{D} = [d_{ij}]$  of  $T$  be a cost matrix. This problem equivalent with the following minimum cost maximum flow problem:

Let  $G' = (V', E')$  be a bipartite graph of distance matrix  $\mathbf{D}$  according to Definition 3.2. Let  $G = (V, E)$  be directed weighted graph extending  $G'$  such that  $V = \{s, s^*\} \cup V' \cup \{t\}$  and  $E = \{(s, s^*), (s^*, v'_i)\} \cup E' \cup \{(v'_j, t)\}$ , for all  $v'_i \in V'_R, v'_j \in V'_C$  and let all edges between nodes  $V'_R$  and  $V'_C$  be directed from  $V'_R$  to  $V'_C$ . Each edge is assigned with a capacity value and a cost defined as follows. Let the capacity of all edges  $e \in E$  be  $c = 1$ , except for  $(s, s^*)$  that has a capacity of  $k$ . Hence, the maximum flow capacity is determined by the  $\{(s, s^*)\}$  cut and it equals to  $k$ . Let the cost of edges  $(s, s^*), (s^*, v'_i)$  and  $(v'_j, t)$  be 0, and let the cost of edges  $(v'_i, v'_j)$ , where  $v'_i \in V'_R, v'_j \in V'_C$ , be  $d'_{ij} = \max_{i,j} (d_{ij}) - d_{ij}$  for all  $i$  and  $j$ .

The minimum cost maximum flow can be found by solving the following optimization with linear programming, that is,

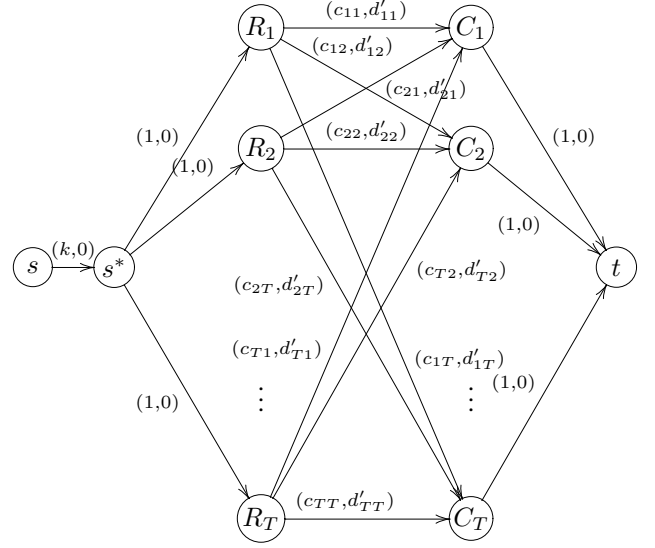


Figure 3. The flow problem equivalent to the maximum distance  $k$ -cardinality matching

the problem is in other words:

$$\max \sum_{i=1}^{|T|} \sum_{j=1}^{|T|} c_{ij} d_{ij}, \quad (5)$$

$$\sum_{j=1}^{|T|} c_{ij} \leq 1, i = 1, \dots, |T|, \quad (6)$$

$$\sum_{i=1}^{|T|} c_{ij} \leq 1, j = 1, \dots, |T|, \quad (7)$$

$$\sum_{i=1}^{|T|} \sum_{j=1}^{|T|} c_{ij} = k, \quad (8)$$

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 [7]. ■

**Example** It has been shown that  $k = 2$  for the  $\varepsilon = 2$  case and therefore the “best” solution contains  $|T| - k = 4 - 2 = 2$  test cases. In the example two linearly independent solutions have been found. The two solutions have the same internal distance sum, and are therefore equivalent in the  $\varepsilon = 2$  case.

$$\sum_{i=1}^4 \sum_{j=1}^4 d_{ij} a_{\mu U_{ij}} = \sum_{i=1}^4 \sum_{j=1}^4 d_{ij} a_{\mu L_{ij}} = 4 \quad (9)$$

#### IV. CONCLUSIONS

This paper proposed a method that selects a subset of a sequence set in polynomial time by searching for similar patterns of events. The method is based on string distance metrics and uses an  $\varepsilon$ -approximation. The presented approach is able to detect redundancies among sequences generated by random walk even without using the practical heuristics of [4].

Nevertheless it is best to be guided by other selection methods (such as the one proposed in [2]), which do not allow the fault detection capability of the remaining trace set (or test set) to be reduced, and used as an additional selection criterion.

A generalization of the method is possible for test cases represented as a labeled, rooted, unordered trees. By calculating the distance matrix of such trees the method is applicable for test case selection without fundamental changes. Another extension of the presented approach is to build an iterative cycle to reduce the computation requirement for long test cases and to suite the requirements of test set maintenance. In the future we plan to investigate these extensions and conduct experiments to compare how this automatically applicable analytical technique works against other selection methods.

#### REFERENCES

- [1] S. T. Vuong and J. 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*, 1992, pp. 31–45.
- [2] G. Kovács, Z. Pap, D. L. Viet, A. Wu-Hen-Chang, and G. Csopaki, "Applying Mutation Analysis to SDL Specifications," in *Proceedings of the Eleventh SDL Forum "System Design"*, 2003, pp. 269–284.
- [3] T. Csöndes, B. Kotnyek, and J. Szabó, "Application of heuristic methods for conformance test selection," *European Journal of Operational Research*, August 2001.
- [4] L. Feijs, N. Goga, S. Mauw, and J. Tretmans, "Test selection, trace distance and heuristics," in *Proceedings of Testing Communication Systems XIV*, Berlin, Germany, 2002, pp. 267–282.
- [5] T. Csöndes and B. Kotnyek, "Automated test case selection based on subpurposes," in *Proceedings of the IFIP TC6 12th International Workshop on Testing Communicating Systems*, 1999, pp. 251–266.
- [6] A. Williams and R. Probert, "Formulation of the interaction test coverage problem as an integer program," in *Proceedings of Testing Communication Systems XIV*, Berlin, Germany, 2002, pp. 283–298.
- [7] M. Dell'Amico and S. Martello, "The k-cardinality assignment problem," *Discrete Applied Mathematics*, vol. 76, pp. 103–121, 1997.
- [8] ITU-T, "Recommendation Z.100: Specification and description language," 2000.
- [9] I. TC97/SC21, "Estelle – a formal description technique based on an extended state transition model. international standard 9074," 1988.
- [10] A. Gill, *Introduction to the Theory of Finite State Machines*. New-York: McGraw-Hill, 1962.
- [11] R. Wagner and M. Fischer, "The string-to-string correction problem," *Journal of the ACM*, vol. 21, no. 1, pp. 168–173, January 1974.
- [12] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw-Hill, 2001.



**Gábor Kovács** received the M.Sc. degree from Budapest University of Technology and Economics (BME) in 2000. Currently he is a Ph.D. candidate. From 2000 he works at BME initially as a Ph.D. student and later as research fellow. His research interests include formal modelling of telecommunication protocol, aspect-oriented software design, Ethernet access networks and indoor positioning.



**Gábor Árpád Németh** received the M.Sc. degree from Budapest University of Technology and Economics in 2007. He is a Ph.D. student at the Dept. of Telecommunications and Media Informatics, Budapest University of Technology and Economics. His current research topic is efficient testing of telecommunication protocols in case of changing specifications.



**Zoltán Pap** received M.Sc. degree in Electrical Engineering in 2000 at Budapest University of Technology and Economics (BME). In 2002 he received M.Sc. degree in Business Administration at the Corvinus University of Budapest. From 2000 to 2006 he worked at BME on various research fields, initially as a Ph.D. student and later as research fellow. His main research topics are telecommunication networks and protocols, grid computing, peer to peer networks, advanced software technologies and testing. He received Ph.D degree at BME in 2006.

Since 2006 he is working at Ericsson Hungary as systems engineering expert and in part time at BME as research fellow.



**Mahadevan Subramaniam** is an associate professor in the computer science department at the University of Nebraska at Omaha. His research interests are in the area of formal methods, software testing, and software maintenance and evolution. He graduated with a Ph.D in Computer Science from SUNY Albany, and has extensive industrial experience in the design test, and verification of complex software and hardware products.