Reducing the Lengths of Checking Sequences by Overlapping

Hasan Ural and Fan Zhang

School of Information Technology and Engineering
University of Ottawa
ural@site.uottawa.ca, fzhang@site.uottawa.ca

Abstract. There are two main shortcomings in the existing models for generating checking sequences based on distinguishing sequences. First, these models require a priori selection of state recognition sequences (called α-sequences) which may not be the best selection for yielding substantial reduction in the length of checking sequences. Second, they do not take advantage of overlapping to further reduce the length of checking sequences. This paper proposes an optimization model that tackles these shortcomings to reduce the lengths of checking sequences beyond what is achieved by the existing models by replacing the state recognition sequences with a set of basic sequences called α-elements and by making use of overlapping.

1. Introduction

To ensure the correct functioning of implementations of a *Finite State Machine* (FSM) *M*, a *fault detection experiment* can be formed [14]: Such an experiment consists of applying an input sequence (derived from *M*) to an implementation *N* of *M*, observing the actual output sequence produced by *N* in response to the application of the input sequence, and comparing the actual output sequence to the expected output sequence. The applied input sequence is called a *checking sequence* which determines whether *N* is a correct or faulty implementation of *M* [8, 10].

A checking sequence of *M* is constructed in such a way that the output sequence produced by *N* in response to the application of the checking sequence provides sufficient information to verify that every state transition of *M* is implemented correctly by *N*. That is, in order to verify the implementation of a transition from state *a* to state *b* under input *x*, 1) *N* is transferred to the state recognized as state *a* of *M*; 2) the output produced by *N* in response to *x* is checked to be as specified in *M* (to detect an *output fault*); and 3) the state reached by *N* after the application of *x* is recognized as state *b* of *M* (to detect a *transfer fault*). Hence, a crucial part of testing the correct implementation of each transition is recognizing the starting and terminating states of the transition which can be achieved by a distinguishing sequence [8], a characterization set [8] or a unique input-output (UIO) sequence [6]. A *distinguishing sequence* for *M* is an input sequence for which each state of *M* produces a distinct output sequence. It is known that a distinguishing sequence may not exist for every minimal FSM [14], and that determining the existence of a distinguishing sequence for an FSM is PSPACE-complete [15]. However, based on distinguishing sequences, various methods have been proposed for FSM based testing (e.g., [4, 9-11, 16, 17]).

Recent methods for constructing reduced length checking sequences based on distinguishing sequences utilize optimization models. In these models, a distinguishing sequence for M is used to form both α-sequences and test segments [11, 16]. The α-sequences, which consist of consecutive applications of the distinguishing sequence for M, are formed to ensure that each state of M is also a distinct state of N; the test segments, which consist of the application of the input triggering the corresponding transition and the distinguishing sequence for M, are formed to verify that every state transition of M is implemented correctly by N. The α-sequences collectively confirm that if N produces the corresponding distinct output sequence for each state of M, then the distinguishing sequence for M is also a distinguishing sequence for N, that is, the distinguishing sequence used in the formation of the α-sequences defines a bijection between states of M and N. Thus, when a path P of the directed graph G representing M is formed such that the input sequence that induces P on G covers each α-sequence and each test segment, that input sequence is a checking sequence of M.

In these models, however, there are two main shortcomings. These models require *a priori* selection of a set of α-sequences which may not guarantee a substantial reduction in the length of a resulting checking sequence. Also, these models connect the α-sequences and test segments to form a checking sequence and thus do not take advantage of potential overlapping among the α-sequences and test segments that could be used to further reduce the lengths of checking sequences.

This paper proposes a novel optimization model that tackles these shortcomings in generating the minimal-length checking sequences: The proposed model does not require selection of α-sequences in advance. It employs a set of α-elements where there is an α-element for each state of M which consists of the application of the distinguishing sequence for M twice. The set of α-elements are then used for the same purpose as the α-sequences in the earlier models. The proposed model does not simply connect the α-sequences and test segments to form a checking sequence. It facilitates the use of overlapping among the α-elements and test segments to further reduce the lengths of resulting checking sequences.

In the remainder of the paper, the proposed model is presented after some preliminary definitions. An example is used to illustrate the model and the steps of its construction. It is then proven that the proposed model constructs a checking sequence. The extensions and the potential uses of the model are discussed in the concluding remarks.

2. Preliminaries

A deterministic and completely specified *FSM* (*finite state machine*) is a quintuple $M = (S, X, Y, \delta, \lambda)$, where $S = \{s_1, s_2, ..., s_n\}$ is a finite set of states with $n = |S|$ and $s_1 \in S$ as the *initial state*, X is a finite set of inputs, Y is a finite set of outputs, δ is a state transition function that maps $S \times X$ to S, and λ is an output function that maps $S \times X$ to Y. M is *minimal* if, for any different states s_i, $s_j \in S$, there is an input sequence $I \in X^*$ such that $\lambda(s_i, I) \neq \lambda(s_j, I)$. M can be represented by a directed graph (digraph) $G = (V, E)$ (Figure 1) where a set of vertices $V = \{v_1, v_2, ..., v_n\}$ represents the set of states of M and a set of directed edges $E=\{(v_j, v_k; x/y)$: $v_j, v_k \in V\}$ represents all specified transitions of M. More specifically, each edge $e = (v_j, v_k; x/y) \in E$ represents a state transition $t = (s_j, s_k; x/y)$ of M from state s_j to s_k with input $x \in X$ and output $y \in Y$, and the (input/output) pair x/y is the *label* of e.

A *path* $P = (n_1, n_2; x_1/y_1)(n_2, n_3; x_2/y_2) ... (n_{r-1}, n_r; x_{r-1}/y_{r-1})$, $r > 1$, of $G = (V, E)$ is a finite sequence of adjacent (not necessarily distinct) edges in E, where each node n_i, $1 \leq i \leq r$, represents a vertex of V; n_1 and n_r are called *start* and *end* of P, and the input/output sequence $(x_1/y_1)(x_2/y_2) ... (x_{r-1}/y_{r-1})$ is called *label* of P, denoted *label*(P). P is represented by $(n_1, n_r; I/O)$, where I/O is called the *transfer sequence* T from n_1 to n_r, I = "$x_1x_2 ... x_{r-1}$" is called the *input portion* of I/O, O = "$y_1y_2 ... y_{r-1}$" is called the *output portion* of I/O. In this case, I (or I/O) is said to *induce P at* n_1. The *length* (or *cost*) *of an input sequence I* (or *input/output sequence I/O*) is its number of inputs, denoted $|I|$ (or $|I/O|$). The *length* (*or cost*) *of a path* $P = (n_1, n_r; I/O)$ is the length (or cost) of the input sequence I, denoted $|P|$. A sequence $i_1 i_2 \ldots i_k$ is a *subsequence* of $x_1 x_2 \ldots x_m$ if there exists a Δ, $0 \leq \Delta \leq m\text{-}k$, such that for all j, $1 \leq j \leq k$, $i_j = x_{j+\Delta}$. Subsequence $i_1 i_2 \ldots i_k$ is a *prefix* of $x_1 x_2 \ldots x_m$ if $i_1 = x_1$.

A digraph $G = (V, E)$ is *strongly connected* if, for every pair of vertices v_j and v_k, there exists a path from v_j to v_k. A *Rural Postman* (*RP*) *path* from vertex v_i to vertex v_j over a subset of edges E' in $G = (V, E)$ is a path which starts at v_i, ends at v_j, and includes all edges of E'; the *Rural Chinese Postman* (*RCP*) *Problem* is to find an RP path of minimum cost i.e., an RCP path, which is the optimization model we will formulate. Algorithms for solving the RCP problem and its special cases important to testing can be found in [1, 16], which are left outside of the scope of this paper.

Let $M = (S, X, Y, \delta, \lambda)$ denote a completely specified, minimal, and deterministic FSM, which is represented by a strongly connected digraph $G = (V, E)$. Given an FSM M, let $\Phi(M)$ be the set of FSMs each of which has at most n states and the same input and output sets as M. Let N be an FSM of $\Phi(M)$. N is *isomorphic* to M if there is a one-to-one and onto function f on the state sets of M and N such that for any state transition $(s_i, s_j; x/y)$ of M, $(f(s_i), f(s_j); x/y)$ is a transition of N. A *checking sequence* of M is an input sequence starting at the initial state s_1 of M that distinguishes M from any N of $\Phi(M)$ that is not isomorphic to M i.e., the output sequence produced by any such N of $\Phi(M)$ is different from the output sequence produced by M. In the context of testing, this means that in response to this input sequence, any faulty implementation N from $\Phi(M)$ will produce an output sequence different from the expected output, thereby indicating the presence of a fault(s). As stated earlier, a crucial part of testing the correct implementation of each transition of M in N from $\Phi(M)$ is recognizing the starting and terminating states of the transition which lead to the notions of state recognition and transition verification used in algorithms for constructing checking sequences (for example, [11], [16]). These notions are defined below in terms of a given distinguishing sequence D for FSM M.

A *distinguishing sequence* (DS) of M is an input sequence D such that the output sequence produced by M in response to D is different for each state of M (i.e., $\forall$ s_i, s_j $\in S$, $s_i \neq s_j$, $\lambda(s_i, D) \neq \lambda(s_j, D)$). A distinguishing sequence for FSM M_0 is shown in Table 1. Based on this definition, the concepts of *state recognition* and *transition verification* can be defined as follows. Let an IO-sequence Q be the label of a path P $= e_1e_2 \ldots e_r$ of G starting at v_1, where $e_j = (n_j, n_{j+1}; x_j/y_j)$ for all j, $1 \leq j \leq r$, i.e., Q = $(x_1/y_1)\ldots$ (x_r/y_r). Then the following defines state recognition and transition verification as in [16].

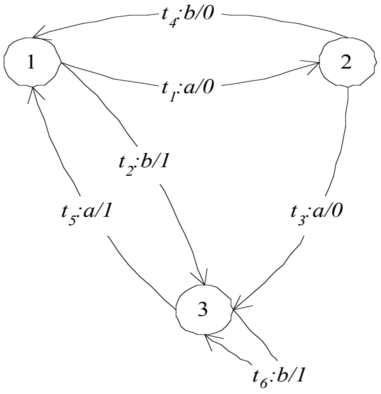

Fig. 1. FSM M_0

Table 1. A DS D = "*aa*" and responses of each state of FSM M_0

Start state	End state	D = "*aa*"
1	3	00
2	1	01
3	2	10

Recognition of a node n_i of P (*in* Q) as some state of M is defined recurrently and is associated with a nonnegative number *depth*(n_i): Let depth(n_i) = ∞ initially;

- 1) A node n_i of P is *d-recognized* (*in* Q) as some state s of M if n_i is the start of a subpath of P whose label is $D/\lambda(s, D)$; depth(n_i) $\leftarrow$ 0.
- 2) Suppose that $(n_q, n_i; T)$ and $(n_j, n_k; T)$ are subpaths of P such that nodes n_q and n_j are d-recognized as state s of M, and n_k is d-recognized as state s' of M but n_i is not d-recognized. Then, node n_i is *t-recognized* as state s' of M; depth(n_i) $\leftarrow$ 1.
- 3) Suppose that $(n_q, n_i; T)$ and $(n_j, n_k; T)$ are subpaths of P such that n_q and n_j are either d-recognized or t-recognized as state s of M, and n_k is either d-recognized or t-recognized as state s' of M. Then, node n_i is *t-recognized* as state s' of M; depth(n_i) $\leftarrow$ min{depth(n_i), 1 + max{depth(n_q), depth(n_j), depth(n_k)}}.
- A node of P is said to be *recognized* if it is either d-recognized or t-recognized as some state of M.

A transition $t = (s, s'; x/y)$ of M is *verified* (in Q) if there is an edge $(n_i, n_{i+1}; x_i/y_i)$ of P such that nodes n_i and n_{i+1} are recognized as states s and s' of M, and $x_i/y_i = x/y$.

Verification of the transitions of M leads to forming checking sequences as shown in Theorem 1 below, which forms the foundation for our proposed model.

Theorem 1 [16] Let Q be the label of a path P of G (for FSM M) such that every transition is verified in Q. Then the input portion of Q is a checking sequence of M.

In the proposed model, recognition of nodes n_i and n_{i+1} of edge $(n_i, n_{i+1}; x_i/y_i)$ of P (of G for FSM M), which corresponds to the transition $t = (s, s'; x/y)$ of M, will be achieved as follows. The node n_{i+1} will be d-recognized in Q as state s' of M, hence there must be a subpath $(n_{i+1}, n_k; D/\lambda(s', D))$ of P. This subpath of P will be used to construct a *test segment* for $t = (s, s'; x/y)$, which is denoted $t' = (n_i, n_{i+1}; x_i/y_i)(n_{i+1}, n_k; D/\lambda(s', D))$ (or $t' = (n_i, n_k; x_iD/\lambda(s, x_iD))$ in short). The collection of test segments for all transitions of M will be denoted P_C, i.e., $P_C = \{(s_i, s_j; x/y)(s_j, \delta(s_j, D); D/\lambda(s_j, D))$: for all $t = (s_i, s_j; x/y)$ of $M\}$ (or $P_C = \{(s_i, \delta(s_i, xD); xD/\lambda(s_i, xD))$: for all $t = (s_i, s_j; x/y)$ of $M\}$ in short). Table 2 shows P_C for FSM M_0.

Table 2. Test Segments of FSM M_0 with DS D = "*aa*"

k	1	2	3	4	5	6
t_k	(1, 2; a/0)	(1, 3; b/1)	(2, 3; a/0)	(2, 1; b/0)	(3, 1; a/1)	(3, 3; b/1)
t_k'	(1,1;aD /001)	(1,2;bD /110)	(2,2;aD /010)	(2,3;bD /000)	(3,3;aD/ 100)	(3,2;bD /110)

The node n_i of P will be t-recognized in Q as state s of M, hence there must be subpaths $(n_j, n_k; T)$ and $(n_q, n_i; T)$ of P such that n_j and n_q are either d-recognized or t-recognized as state s of M, and n_k is either d-recognized or t-recognized as state s' of M. These subpaths will be formed by using what is called ***α-elements***. A set of ***α-elements for M*** is a set of paths $\{\alpha_i = (s_i, \delta(s_i, D); D/\lambda(s_i, D))(\delta(s_i, D), \delta(\delta(s_i, D), D); D/\lambda(\delta(s_i, D), D)): i = 1, \ldots, n\}$ (or $\{\alpha_i = (s_i, \delta(s_i, DD); DD/\lambda(s_i, DD)): i = 1, \ldots, n\}$ in short), denoted by P_α . For example, Table 3 shows P_α for FSM M_0 with D = "*aa*".

Proposition 1: Let Q be the label of a path P of G (for FSM M with a distinguishing sequence D) such that Q contains n subsequences of the form $DD/\lambda(s_i, DD)$, $i = 1, \ldots, n$. If Q induces a path in N of $\Phi(M)$ then D is also a distinguishing sequence for N and defines a bijection from the states of M to the states of N.

Proof: Since D is a distinguishing sequence for M, each of these subsequences of the form $D/\lambda(s_i, D)$, which is a prefix of $DD/\lambda(s_i, DD)$, $i = 1, \ldots, n$, is unique. If Q induces a path of N from $\Phi(M)$ then, since N has at most n states, D must also be a distinguishing sequence for N. This says that if n different responses to D are observed in N, then D defines a one-to-one correspondence between the states of M and N. In this case, we say that the uniqueness of the response of each of the n states of N to D is verified and hence N has n distinct states [13]. □

Proposition 2: Let Q be the label of a path P of G (for FSM M with a distinguishing sequence D) such that each α-element $\alpha_i = (s_i, \delta(s_i, DD); DD/\lambda(s_i, DD))$, $i = 1, \ldots, n$, is a subpath P of G. Then, for each $(s_i, \delta(s_i, D); D/\lambda(s_i, D))$, $1 \le i \le n$, appearing in P as a subpath $(n_j, n_k; D/\lambda(s_i, D))$,

1. the start node n_j of $(n_j, n_k; D/\lambda(s_i, D))$ is d-recognized
2. the end node n_k of $(n_j, n_k; D/\lambda(s_i, D))$ is t-(or d-)recognized

Proof: Part 1) is a direct consequence of the definition state recognition. Part 2) can easily be shown as follows. The α-element $\alpha_i = (s_i, \delta(s_i, DD); DD/\lambda(s_i, DD))$ appears in P as a subpath $(n_q, n_r; D/\lambda(s_i, D))(n_r, n_v; D/\lambda(\delta(s_i, D), D))$. As n_q, n_r and n_j are d-

recognized, the end node n_k of $(n_j, n_k; D/\lambda(s_i, D))$ must be t-recognized as $\delta(s_i, D)$ if it is not d-recognized, by the definition of state recognition. □

Table 3. α-elements for FSM M_0 (with D = "*aa*")

	Start state s_i	End state	label(α_i) = $DD/\lambda(s_i, DD)$
α_1	1	2	*aaaa*/0010
α_2	2	3	*aaaa*/0100
α_3	3	1	*aaaa*/1001

3. The Optimization Model

We wish to pose the following optimization problem: Given an FSM M (represented by a digraph $G= (V, E)$) and DS D for M, generate a minimum-length checking sequence of M starting at the initial state s_1 through composing an RCP path P of G which starts at v_1 and contains every element of $P_\alpha \cup P_C$. As in the earlier models for constructing reduced length checking sequences based on distinguishing sequences, it will be shown that since this RCP path P of G contains every element of $P_\alpha \cup P_C$, it establishes that all states of M are recognized and all transitions of M are verified.

In order to reduce the overall length of the resulting checking sequence, we will take advantage of the overlapping among elements of $P_\alpha \cup P_C$ in generating a minimum-length checking sequence in our model as follows: Let P_1 and P_2 denote two paths of G. If P_1 has a suffix R that is a prefix of P_2, namely, $P_1 = R_1R$ and $P_2 = RR_2$ for some paths R_1 and R_2 of G, we say that P_1 *overlaps* P_2 *by* R. In this case, a new path $P_{1,2}$ of G can be formed by overlapping P_1 and P_2 by R, namely, $P_{12} = R_1RR_2$, with $|P_{12}| = |P_1| + |P_2| - |R|$. Furthermore, if label($P_2$) has D as the prefix of its input portion, we call overlap of this type *D-overlap* by R. This definition offers a way to check if P_1 D-overlap P_2 or not by first checking if D is the prefix of the input portion of label(P_2) and then identifying the maximal overlapping portion R.

D-overlap of a sequence of paths (of G) $P_1, P_2, \ldots, P_k$, where $k > 2$, can be defined inductively as follows: If D-overlapping of the sequence $P_1, \ldots, P_{k-1}$ forms a new path $P_{1,k-1}$ and if this $P_{1,k-1}$ D-overlaps P_k forming a path $P_{1,k}$, then D-overlapping of the sequence $P_1, P_2, \ldots, P_k$ forms $P_{1,k}$.

The proposed algorithm for the solution of the optimization problem augments $G = (V, E)$ to form a digraph $G^* = (V^*, E^*)$ and then formulates the construction of a minimum-length checking sequence for M starting at the initial state s_1 as finding an RCP path P of G^* which starts at v_1 and contains every element of $P_\alpha \cup P_C$.

The proposed algorithm is given as follows:

Initially, $G^* = (V^*, E^*) \leftarrow G = (V, E)$

1. For every $\tau = (s_i, s_j; I_\tau/O_\tau)$ that is either an α-element or a test segment,
   a) add to V^* two new vertices $s'_{i\tau}$, $s''_{j\tau}$ for the start and end states of τ, resp.
   b) add to E^*
      - an edge $(s'_{i\tau}, s''_{j\tau}; I_\tau/O_\tau)$ with cost $|I_\tau|$
      - an edge $(s''_{j\tau}, s_j)$ with cost 0
      - an edge $(s_i, s'_{i\tau})$ with cost 0

2. a) Add to V^* an artificial node $\boldsymbol{s_1}$***** (representing the initial state s_1)
   b) Add to E^* an edge $(s_1^*, s'_{1\tau})$ with cost 0, for each of those $\tau = (s_1, s_j; I_\tau/O_\tau) \in P_\alpha \cup P_C$ such that D is a prefix of I_τ
3. For any pair of two different $\tau = (s_i, s_j; I_\tau/O_\tau)$ and $\mu = (s_k, s_r; I_\mu/O_\mu)$, each being either an α-element or a test segment, such that τ D-overlaps μ by R (which can be determined by first checking if D is the prefix of I_μ and if yes then identifying the maximal overlapping portion R of τ and μ), add to E^* an edge $(s''_{j\tau}, s'_{k\mu})$ with (***negative***) cost $-|R|$ (which reflects the effect of D-overlapping).
4. Find an RCP path P of G^* starting from s_1^* traversing at least once the edges representing the α-elements and the test segments. Use the input portion of label(P) as a checking sequence of M.

   More specifically, the algorithm is as follows:

   Construct $G^* = (V^*, E^*)$ whose vertex-set and edge-set are
   $V^* = V \cup V' \cup V'' \cup \{ s_1^* \}$ and $E^* = E \cup E_0 \cup E_\alpha \cup E_C \cup E' \cup E'' \cup E_D$
   from $G = (V, E)$ representing a given FSM M, where

   $V' = \{s'_{i\tau}$: for all $\tau = (s_i, s_j; I_\tau/O_\tau) \in P_\alpha \cup P_C\}$,
   $V'' = \{s''_{j\tau}$: for all $\tau = (s_i, s_j; I_\tau/O_\tau) \in P_\alpha \cup P_C\}$,
   $E_0 = \{(s_1^*, s'_{1\tau}; \varepsilon)$ with cost 0: for all $\tau = (s_1, s_j; I_\tau/O_\tau) \in P_\alpha \cup P_C$ such that D is a prefix of $I_\tau\}$,
   $E_\alpha = \{(s'_{i\tau}, s''_{j\tau}; I_\tau/O_\tau)$ with cost $|I_\tau|$: for all $\tau = (s_i, s_j; I_\tau/O_\tau) \in P_\alpha\}$,
   $E_C = \{(s'_{i\tau}, s''_{j\tau}; I_\tau/O_\tau)$ with cost $|I_\tau|$: for all $\tau = (s_i, s_j; I_\tau/O_\tau) \in P_C\}$,
   $E' = \{(s_i, s'_{i\tau}; \varepsilon)$ with cost 0: for all $\tau = (s_i, s_j; I_\tau/O_\tau) \in P_\alpha \cup P_C\}$,
   $E'' = \{(s''_{j\tau}, s_j; \varepsilon)$ with cost 0: for all $\tau = (s_i, s_j; I_\tau/O_\tau) \in P_\alpha \cup P_C\}$, and
   $E_D = \{(s''_{j\tau}, s'_{k\mu}; \varepsilon)$ with cost $-|R|$: for all $\tau = (s_i, s_j; I_\tau/O_\tau)$, $\mu = (s_k, s_r; I_\mu/O_\mu) \in P_\alpha \cup P_C$ such that τ D-overlaps μ by some $R\}$.

   Find an RCP path P of G^* that starts at $s'_{1\tau}$ and contains all edges of $E_\alpha \cup E_C$.

   The input portion of *label*(P) is a checking sequence of M.

Example: For FSM M_0 with D = "*aa*", Table 4 shows D-overlapping between pairs of elements of $P_\alpha \cup P_C$ and the resulting negative cost from each overlapping. More specifically, it shows all pairs $\tau, \mu \in P_\alpha \cup P_C$ such that τ D-overlaps μ by R with negative cost $-|R|$. Figure 2 shows an example of the result of application of the proposed algorithm to G (for M_0), using only part of D-overlapping in Table 4 (so that Figure 2 does not become too complicated to follow). In Figure 2, thick lines represent edges of $E_\alpha \cup E_C$ and for simplicity, s'_i, s''_j are used for $s'_{i\tau}$ and $s''_{j\tau}$. Note that in Tables 4-6, we dropped output portion of the paths for ease of presentation. An RCP path P (starting at vertex 1* and ending at vertex 2") is found as

$P = (1^*,1'; \varepsilon)t_1'\text{-}\alpha_1\text{-}t_3'\text{-}\alpha_2\text{-}t_5'\text{-}\alpha_3\,(1'',1; \varepsilon)(1,1'; \varepsilon)t_2'(2'',2; \varepsilon)(2,2'; \varepsilon)t_4'(3'',3; \varepsilon)(3,3'; \varepsilon)t_6'$

where each hyphen sign indicates an occurrence of D-overlapping. Its corresponding input sequence (without overlapping) is:

"ε*aaa-aaaa-aaa-aaaa-aaa-aaaa*εε*baa*εε*baa*εε*baa*".

The checking sequence obtained from P with D-overlapping starting at state 1 is

"ε*aaa-~~aaa~~a-~~aaa~~-~~aaa~~a~~-aaa-aaa~~a baa baa baa*"

whose length is 15.

For the same example, a reduced length checking sequence was found to have length 32 in [11].

Table 4. Overlapping among $P_\alpha \cup P_C$ for FSM M_0

$\tau = P_1$	$\mu = P_2$	R	$-\|R\|$	$E \in E_D$	Inuse*
$\alpha_1 = (1, 2, aaaa)$	$\alpha_2 = (2, 3; aaaa)$	$(2, 2; aaa)$	-3	$(2_\tau'', 2_\mu')$	y
α_1	$\alpha_3 = (3, 1; aaaa)$	$(3, 2; aa)$	-2	$(2_\tau'', 3_\mu')$	y
α_1	$t'_1 = (1, 1; aaa)$	$(1, 2; a)$	-1	$(2_\tau'', 1_\mu')$	
α_1	$t'_3 = (2, 2; aaa)$	$(2, 2; aaa)$	-3	$(2_\tau'', 2_\mu')$	y
α_1	$t'_5 = (3, 3; aaa)$	$(3, 2; aa)$	-2	$(2_\tau'', 3_\mu')$	
$\alpha_2 = (2, 3; aaaa)$	$\alpha_1 = (1, 2, aaaa)$	$(1, 3; aa)$	-2	$(3_\tau'', 1_\mu')$	y
α_2	$\alpha_3 = (3, 1; aaaa)$	$(3, 3; aaa)$	-3	$(3_\tau'', 3_\mu')$	y
α_2	$t'_1 = (1, 1; aaa)$	$(1, 3; aa)$	-2	$(3_\tau'', 1_\mu')$	
α_2	$t'_3 = (2, 2; aaa)$	$(2, 3; a)$	-1	$(3_\tau'', 2_\mu')$	
α_2	$t'_5 = (3, 3; aaa)$	$(3, 3; aaa)$	-3	$(3_\tau'', 3_\mu')$	y
$\alpha_3 = (3, 1; aaaa)$	$\alpha_1 = (1, 2, aaaa)$	$(1, 1; aaa)$	-3	$(1_\tau'', 1_\mu')$	y
α_3	$\alpha_2 = (2, 3; aaaa)$	$(2, 1; aa)$	-2	$(1_\tau'', 2_\mu')$	y
α_3	$t'_1 = (1, 1; aaa)$	$(1, 1; aaa)$	-3	$(1_\tau'', 1_\mu')$	y
α_3	$t'_3 = (2, 2; aaa)$	$(2, 1; aa)$	-2	$(1_\tau'', 2_\mu')$	
α_3	$t'_5 = (3, 3; aaa)$	$(3, 1; a)$	-1	$(1_\tau'', 3_\mu')$	
$t'_1 = (1, 1; aaa)$	$\alpha_1 = (1, 2, aaaa)$	$(1, 1; aaa)$	-3	$(1_\tau'', 1_\mu')$	y
t'_1	$\alpha_2 = (2, 3; aaaa)$	$(2, 1; aa)$	-2	$(1_\tau'', 2_\mu')$	y
t'_1	$\alpha_3 = (3, 1; aaaa)$	$(3, 1; a)$	-1	$(1_\tau'', 3_\mu')$	
t'_1	$t'_3 = (2, 2; aaa)$	$(2, 1; aa)$	-2	$(1_\tau'', 2_\mu')$	y
t'_1	$t'_5 = (3, 3; aaa)$	$(3, 1; a)$	-1	$(1_\tau'', 3_\mu')$	
$t'_2 = (1, 2; baa)$	$\alpha_1 = (1, 2, aaaa)$	$(1, 2; a)$	-1	$(2_\tau'', 1_\mu')$	
t'_2	$\alpha_3 = (3, 1; aaaa)$	$(3, 2; aa)$	-2	$(2_\tau'', 3_\mu')$	y
t'_2	$t'_1 = (1, 1; aaa)$	$(1, 2; a)$	-1	$(2_\tau'', 1_\mu')$	
t'_2	$t'_5 = (3, 3; aaa)$	$(3, 2; aa)$	-2	$(2_\tau'', 3_\mu')$	y
$t'_3 = (2, 2; aaa)$	$\alpha_1 = (1, 2, aaaa)$	$(1, 2; a)$	-1	$(2_\tau'', 1_\mu')$	
t'_3	$\alpha_2 = (2, 3; aaaa)$	$(2, 2; aaa)$	-3	$(2_\tau'', 2_\mu')$	y
t'_3	$\alpha_3 = (3, 1; aaaa)$	$(3, 2; aa)$	-2	$(2_\tau'', 3_\mu')$	y
t'_3	$t'_1 = (1, 1; aaa)$	$(1, 2; a)$	-1	$(2_\tau'', 1_\mu')$	
t'_3	$t'_5 = (3, 3; aaa)$	$(3, 2; aa)$	-2	$(2_\tau'', 3_\mu')$	y
$t'_4 = (2, 3; baa)$	$\alpha_1 = (1, 2, aaaa)$	$(1, 3; aa)$	-2	$(3_\tau'', 1_\mu')$	y
t'_4	$\alpha_2 = (2, 3; aaaa)$	$(2, 3; a)$	-1	$(3_\tau'', 2_\mu')$	
t'_4	$t'_1 = (1, 1; aaa)$	$(1, 3; aa)$	-2	$(3_\tau'', 1_\mu')$	y
t'_4	$t'_3 = (2, 2; aaa)$	$(2, 3; a)$	-1	$(3_\tau'', 2_\mu')$	
$t'_5 = (3, 3; aaa)$	$\alpha_1 = (1, 2, aaaa)$	$(1, 3; aa)$	-2	$(3_\tau'', 1_\mu')$	
t'_5	$\alpha_2 = (2, 3; aaaa)$	$(2, 3; a)$	-1	$(3_\tau'', 2_\mu')$	
t'_5	$\alpha_3 = (3, 1; aaaa)$	$(3, 3; aaa)$	-3	$(3_\tau'', 3_\mu')$	y
t'_5	$t'_1 = (1, 1; aaa)$	$(1, 3; aa)$	-2	$(3_\tau'', 1_\mu')$	y
t'_5	$t'_3 = (2, 2; aaa)$	$(2, 3; a)$	-1	$(3_\tau'', 2_\mu')$	
$t'_6 = (3, 2; baa)$	$\alpha_1 = (1, 2, aaaa)$	$(1, 2; a)$	-1	$(2_\tau'', 1_\mu')$	
t'_6	$\alpha_3 = (3, 1; aaaa)$	$(3, 2; aa)$	-2	$(2_\tau'', 3_\mu')$	y
t'_6	$t'_1 = (1, 1; aaa)$	$(1, 2; a)$	-1	$(2_\tau'', 1_\mu')$	
t'_6	$t'_5 = (3, 3; aaa)$	$(3, 2; aa)$	-2	$(2_\tau'', 3_\mu')$	y

The last column with “y” indicates the overlapping is used in Figure 2 for generating an RCP path. The ones with blank space indicated the overlapping is not considered in Figure 2 as we do not want the Figure too complicated.

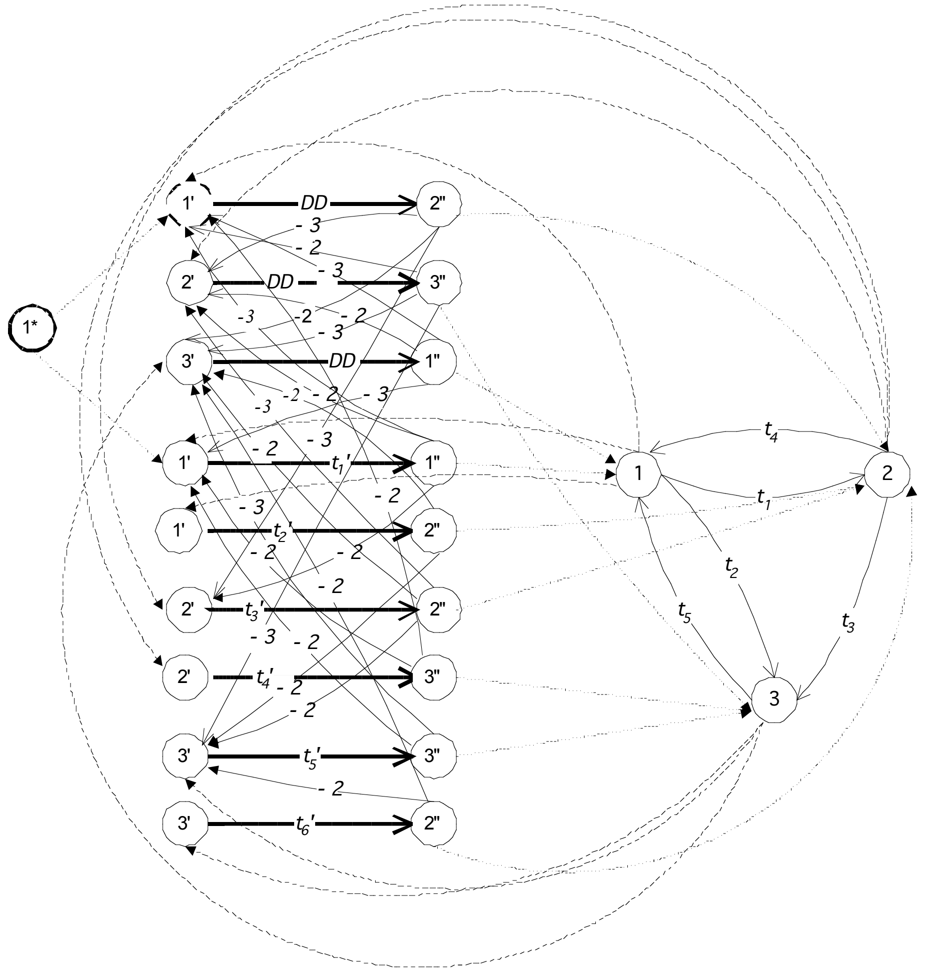

Fig. 2. Optimization Model for FSM M_0

Note that D-overlapping between two elements $P_\alpha \cup P_C$ is shown explicitly in the optimization model whereas D-overlapping among a sequence of elements are formed automatically in the process of finding an RCP path P and can be identified in P. To prove the correctness of the proposed algorithm, let Π denote a set paths (of G) obtained from D-overlapping among elements (paths) of $P_\alpha \cup P_C$, such that every element of $P_\alpha \cup P_C$ is a subpath of a path of Π. In G^*, such a Π is naturally formed to consist of the maximal paths generated from D-overlapping among elements of $P_\alpha \cup P_C$ and those elements of $P_\alpha \cup P_C$ not contained in any D-overlapping path. Let E_π be a set of edges, whose end vertices are in V of G, that represent the paths of Π.

Theorem 2: Let P be an RCP path of $G_0 = (V, E \cup E_\pi)$ such that P starts at v_1 and contains every edge of E_π and D is the prefix of the input portion of its label Q. Let E_1 denote the set of edges of P after excluding E_π, i.e., $E_1 = E(P) \cap E$. If $G_1 = (V, E_1)$ does not contain a cycle, then the input portion of Q forms a checking sequence of M.
Proof: Suppose that $P = (n_1, n_2; L_1)\ldots(n_r, n_{r+1}; L_r)$ and its label $Q = L_1L_2\ldots L_r$, where each $(n_j, n_{j+1}; L_j)$, $1 \leq j \leq r$, is an edge (of G_0) representing either a single edge or a subpath of $G = (V, E)$. First we claim that every n_j of P, $1 \leq j \leq r$, is recognized (in Q).

Suppose the claim is not true. Since $G_1 = (V, E_1)$ does not contain a cycle, it is well known [2] that the vertices of V can be assigned an order "$\propto$" such that $u \propto v$ if there exists a path from u to v in G_1. Let n_i be a node corresponding to the smallest member of V (with respect to "$\propto$") such that n_i is not recognized. Note that $i > 1$ as n_1 is d-recognized. We consider $(n_{i-1}, n_i; L_{i-1})$ of P and derive a contradiction for each of all possible cases below.

If $(n_{i-1}, n_i; L_{i-1})$ corresponds to an edge of E_π, then n_i corresponds to the end of either an α-sequence or a test segment, which must be t-recognized, a contradiction. If $(n_{i-1}, n_i; L_{i-1}) \in E_1$, i.e., $(n_{i-1}, n_i; L_{i-1}) = (u, v; x/y) \in E$, then n_{i-1} is recognized as $u \propto n_i$. Note that P contains the test segment for this edge of E, say $(n_j, n_{j+1}; L_j)$ where $L_j = (xD_v)/\lambda(u, xD_v)$. As n_j corresponds to $u \propto n_i$, n_j is recognized. Also the node adjacent to n_j in the subpath $(n_j, n_{j+1}; L_j)$, $\delta(n_j, x)$, is d-recognized. Thus, n_i is t-recognized as $\delta(u, x)$ of M, another contradiction. Therefore, every n_i is recognized.

For every transition t of M, its test segment t' is contained in a subpath P_i (of P) represented by an edge $(n_{i-1}, n_i; L_{i-1})$ of P. If the start state of t' is n_{i-1}, from the argument above, n_{i-1} recognized; otherwise, t' is contained through D-overlapping, its start state is d-recognized. Hence, t' is verified in Q, and by Theorem 1, the input portion of Q is a checking sequence of M. □

Correctness of the proposed algorithm is a direct consequence of Theorem 2. Notice that E_π is formed naturally in the process of solving for an RCP path P of G^*. The RCP path P of G^* can be viewed as a path of $G_0 = (V, E \cup E_\pi)$, by mapping the nodes of P into the corresponding vertices of G. Thus, as long as the premise of Theorem 2 holds, the correctness of the proposed algorithm is guaranteed.

Up to this point, we have presented the proposed optimization model with a simplification for ease of presentation. This simplification is in the formation of α-elements, that is, instead of using a more general form DT_iDT_j, where T_i, T_j are transfer sequences, we used DD (equivalently, assumed $T_i = T_j = \varepsilon$). In the previous models [11, 16] for constructing reduced-length checking sequences, a given set of transfer sequences $T_i = I_i/O_i$ starting at state $\delta(s_i, D)$, $i = 1, \ldots, n$, is used for two main purposes. First, it is used to redefine a set of test segments as $P_C = \{(s_i, s_j; x/y)(s_j, \lambda(s_j, DI_j); DI_j/\lambda(s_j, DI_j))$: for all $t = (s_i, s_j; x/y)$ of $M\}$, so that every state can be reached by at least one of these test segments. Second, it is used to increase the flexibility of the models to obtain a possible further reduction in the lengths of checking sequences.

Now we present a generalization of the optimization model that incorporates a given set of transfer sequences $T_i = I_i/O_i$ starting at state $\delta(s_i, D)$, $i = 1, \ldots, n$ through an adjustment of α-elements and test segments as follows.

Let $P_\alpha = \{(s_i, s_j; DI_i/\lambda(s_i, DI_i))(s_j, \delta(s_j, DI_j); DI_j/\lambda(s_j, DI_j))$: $i = 1, \ldots, n\}$ and $P_C = \{(s_i, s_j; x/y)(s_j, \lambda(s_j, DI_j); DI_j/\lambda(s_j, DI_j))$: for all $t = (s_i, s_j; x/y)$ of $M\}$.

This adjustment amounts to replacing subsequences of the form $DD/\lambda(s_i, DD)$ with subsequences of the form $DI_iDI_j/\lambda(s_i, DI_iDI_j)$, $i = 1, \ldots, n$, $1 \leq j \leq n$ as the labels of α-elements; and replacing subsequences of the form $xD/\lambda(s_i, xD)$ with subsequences of the form $xDI_j/\lambda(s_i, xDI_j)$, $i = 1, \ldots, n$, $1 \leq j \leq n$ as the labels of test segments. As such, the adjustment does not alter the validity of the Propositions 1 and 2, and Theorem 2: Their proofs are similar to the proofs of those given for the optimization model presented in the previous section. With these new P_α and P_C, we can apply the proposed algorithm to solve the same optimization problem as the one given earlier.

Example: For FSM M_0, D = "*aa*", and a given set of transfer sequences $T_1 = a/1$, $T_2 = T_3 = \varepsilon$, the set P_α of α-elements and the set P_C of test segments are listed in Table 5 and Table 6 below. Figure 3 shows the general optimization model for FSM M_0 with the given T_i, $i = 1, 2, 3$. (Note that t_1' and t_3' are prefixes of α-elements, and thus, they are eliminated from the model.) We obtain the optimal solution to the general model as an RCP path P of G^* starting at 1* (and ending at state 2"), which is:

$P = (1^*,1'; \varepsilon)t_1'\text{-}\alpha_1\text{-}\alpha_2\text{-}\alpha_3\text{-}t_3'\text{-}t_5'(1'',1; \varepsilon)(1,1'; \varepsilon)t_2'(2'',2; \varepsilon)(2,2'; \varepsilon)t_4'(1'',1; \varepsilon)t_2(3,3'; \varepsilon)t_6'$

where each hyphen sign indicates an occurrence of D-overlapping. Its corresponding input sequence (without overlapping) is:

εaaa-aaaaaa-aaaaa-aaaa-aaa-aaaεεbaaεεbaaεbεbaa

The checking sequence obtained from P with D-overlapping starting at state 1 is

aaa-~~aaa~~aaa-~~aaaaa-aaaa-aa~~a-~~aa~~a baa baa b baa

whose length is: 3+3+1+1+3+3+1+3 = 18.

This result is longer than the checking sequence produced by the optimization model presented earlier where all $T_i = \varepsilon$, $i = 1, 2, 3$. Indeed, an experimental study reported in [12] confirms the intuitive hypothesis that using empty transfer sequences results in shorter checking sequences based on distinguishing sequences.

Table 5. α-elements for FSM M_0 (with D = "*aa*", $T_1 = a/1$, $T_2 = T_3 = \varepsilon$)

start s_i	$\lambda(s_i, DI_i)$	$s_j = \delta(s_i,DI_i)$	$\lambda(s_j, DI_j)$	end $\delta(s_j,DI_j)$
1	001	1	001	1
2	01	1	001	1
3	10	2	01	1

Table 6. Test Segments of FSM M_0 (with D = "*aa*", $T_1 = a/1$, $T_2 = T_3 = \varepsilon$)

k	1	2	3	4	5	6
t_k	(1,2; a/0)	(1,3; b/1)	(2,3; a/0)	(2,1; b/0)	(3,1; a/1)	(3,3; b/1)
t_k'	(1,1;aD/ 001)	(1,2;bD/ 110)	(2,2;aD/ 010)	(2,1;bDa/ 0001)	(3,1;aDa/ 1001)	(3,2;bD/ 110)

Theorem 3. Given an FSM M represented by graph G, a DS D for M and a set of TS $T_i = I_i/O_i$ starting at state $\delta(s_i, D)$, $i = 1, \ldots, n$, the minimum-length checking sequence constructed by our general model is at least as short as the ones constructed by the previous optimization models of [11, 16].

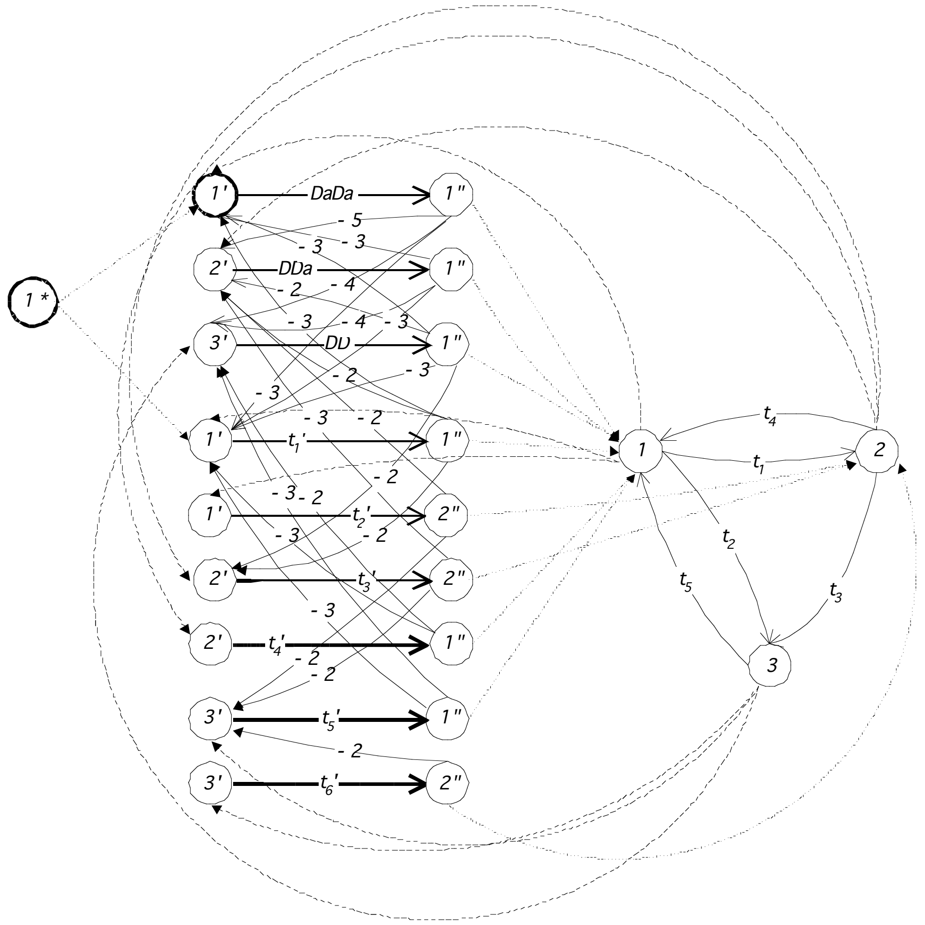

Fig. 3. The General Model for FSM M_0

Proof: In the previous models [11, 16], a set of I/O sequences of M is first generated, where each I/O sequence, called an *α-sequence* α_k in [16], *α'-sequence* α_k' in [11], is the label of a path P_k (of G, $k = 1,\ldots, q$) that is formed by concatenating some of the paths $\{(s_i, \delta(s_i, DI_i); D/\lambda(s_i, DI_i)): i = 1,\ldots, n\}$, such that, by including all $P_1, \ldots, P_q$ in a path P of G, the following requirements are satisfied:

- For each $D/\lambda(s_i, D)T_i$ in label(P), $1 \le i \le n$, its start node is *d*-recognized;
- For each $D/\lambda(s_i, D)T_i$ in label(P), $1 \le i \le n$, its end node is *t*-(or *d*-)recognized.

α_k (or α'_k) = label(P_k), $1 \le k \le q$ which is formed as follows: Let $V=\{v_1, v_2, ..., v_n\}$ and let $V_1, V_2, ..., V_q$, $q \ge 1$, be subsets of V, i.e., $V_k \subseteq V$, $1 \le k \le q$, whose union is V. Without loss of generality, assume that $V_k = \{v_1^k, v_2^k, ..., v_{m_k}^k\}$, $1 \le k \le q$, and for V_k, define α_k [16] as: $\alpha_k = D/\lambda(v_1^k, D)T_1^k\ D/\lambda(v_2^k, D)T_2^k \ldots D/\lambda(v_{m_k}^k, D)T_{m_k}^k\ D/\lambda(v_w^k, D)T_w^k$ where $T_j^k = (I_j^k / O_j^k)$ is a transfer sequence from $\delta(v_j^k, D)$ to v_{j+1}^k for $j = 1, 2, ..., m_{k\text{-}1}$

$T^k_{m_k}=(I^k_{m_k}/O^k_{m_k})$ is a transfer sequence from $\delta(v^k_{m_k}, D)$ to v^k_w, $w \in \{1, 2, ..., m_k\}$

$T^k_w=(I^k_w/O^k_w)$ is some T^k_j, $1\leq j\leq m_k$, where T^k_j, $T^k_{m_k}$, and T^k_w may be empty sequences; or define α'_k [11] as above except that v^k_w may not necessarily be in V_k.

In both models [11, 16], the set of paths P_1, ..., P_q is included in the augmented digraph $G^* = (V^*, E^*)$ as edges in $E_\alpha \subset E^*$. The set of test segments, which is $\{(v_i, (\delta(v_i, xDI^k_j)); (xDI^k_j)/\lambda(v_i, xDI^k_j))$: for every $(v_i, v_j; x/y) \in E\}$ in [16], is included as edges in $E_C \subset E^*$. The set of test segments is not explicitly formed in [11]. However, it is implicitly formed since each element of $E_C = \{(v_i, v_j; x/y)$: $(v_i, v_j; x/y) \in E\}$ is followed by a $(DI^k_j)/\lambda(v_j, D\, I^k_j))$ or an α'_k. An RCP path P is sought in G^* over $E_\alpha \cup E_C$ (without considering their overlapping) and the input portion of label(P) obtained is used as a checking sequence. On the other hand, the general model makes use of the best available combination of α-elements and makes use of overlapping not only among P_α but also among $P_\alpha \cup P_C$, and thus generates a checking sequence that is *at least as short* as the one generated by [11, 16].

4. Conclusions

We have presented an optimization model (and its generalization) that allow any possible overlapping of α-elements and test segments to construct a minimal-length checking sequence of a given deterministic, completely specified, minimal and strongly connected FSM M. The optimal solution of the model is based on all possible combinations rather than a single a priori selection of a set of state recognition sequences and on all possible overlapping between the α-elements and test segments. This model generates a checking sequence that is at least as short as the ones generated by the previous models [11, 16].

Potential simplifications of the proposed optimization model include the following:

(i) In the model, if a vertex v of G^* has only one incoming edge (s, v) and one outgoing edge (v, s'), it may be merged into another vertex, which simplifies the model. This is particularly useful when a test segment is not involved in D-overlapping. This simplification is clearly applicable to both the proposed model and its generalization.

(ii) Further simplification may also be achieved by eliminating some test segments from the model as they are part of the α-elements. The elimination of test segments that are related to the transitions traversed as the last transition of a path induced by D in an α-sequence, formed by concatenated $DI_i/\lambda(s_i, DI_i)$'s where $I_i = \varepsilon$, has been proposed in [3]. Such elimination can be incorporated into the model as follows: First, identify the test segments that are covered by α-elements and do not include them in P_C. Then, find an RCP P in the simplified model that contains all α-elements and those remaining test segments, and use the input portion of label(P) as a checking sequence. Since the generalization of the proposed model utilizes $DI_i/\lambda(s_i, DI_i)$'s where $I_i \neq \varepsilon$, incorporation of this simplification requires further study.

It must be noted that the RCP problem is NP-complete [7]. However, for some systems, the given FSM M has a reset feature, i.e. there is input r such that $\delta(s_i, r) = s_1$ for every state s_i. With this reset feature, an optimal solution of our proposed model can be found in polynomial-time and is guaranteed to be a checking sequence for M as follows. For each transition t of form $(s_i, s_1; r/\lambda(s_i, r))$, i.e., t is triggered by r, its test segment $t' = (s_i, \delta(s_1, DI_1); (rDI_1)/\lambda(s_i, rDI_1))$ is added to the graph G^* with both vertices in V (as no overlapping is involved). These edges consist of a connected spanning subgraph of G. In this case, the problem of finding an RP path with minimum cost is reduced to a min-cost flow problem as in [1, 16], which can be solved in polynomial-time [5].

In the generalized model, a given set of transfer sequences $\{T_i: i = 1, \ldots, n\}$ is used together with a DS in forming α-elements and test segments. Although it was shown experimentally that empty TS (i.e., $T_i = \varepsilon$, $i = 1, \ldots, n$) leads to shorter checking sequences [12], the best selection of such a set is unknown and worth further study.

References

1. A.V. Aho, A.T. Dahbura, D. Lee, and M.U. Uyar, "An optimization technique for protocol conformance test sequence generation based on UIO sequences and rural Chinese postman tours", *IEEE Trans. on Comm.* vol.39, pp.1604-1615, 1991.
2. J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, New York: Elsevier North Holland, Inc. 1976.
3. J. Chen, R.M. Hierons, H. Ural and H. Yenigun, "Eliminating redundant tests in a checking sequence", *Proc. of IFIP TestCom* 2005, May 2005, pp.146-158.
4. T. Chow, "Testing software design modeled by finite-state machines", *IEEE Trans. Software Eng.*, vol.SE-4, pp.178-187, 1978.
5. W.J. Cook, W.H. Cunningham, W.R. Pulleyblank and A. Schrijver, *Combinatorial Optimization*, John Wiley and Sons, New York, 1998.
6. A.T. Dahbura, K.K. Sabnani, and M.U. Uyar, "Formal methods for generating protocol conformance test sequences", *Proc. of IEEE*, vol.78, pp.1317-1325, 1990.
7. M.R. Garey and D.S. Johnson, *Computers and Intractability*, W.H. Freeman and Company, New York, 1979.
8. A. Gill, *Introduction to the Theory of Finite-State Machines*, NY: McGraw-Hill, 1962.
9. G. Gonenc, "A method for the design of fault detection experiments", *IEEE Trans. on Computer*, vol.19, pp.551-558, June 1970.
10. F.C. Hennie, "Fault detecting experiments for sequential circuits", *Proc. 5th. Symp. Switching Circuit Theory and Logical Design*, pp.95-110, Princeton, N.J.,1964.
11. R.M. Hierons and H. Ural, "Reduced length checking sequences", *IEEE Trans. on Computers,* vol.51(9), pp.1111-1117, 2002.
12. R. Hieron and H. Ural, "Optimizing the length of checking sequences", *submitted to IEEE Trans on Computers*, 2004.
13. K. Inan and H. Ural, "Efficient checking sequences for testing finite state machines", *Information and Software Technology*, vol.41, pp.799-812, 1999.
14. Z. Kohavi, *Switching and Finite State Automata Theory*, McGraw-Hill, 1978.
15. D. Lee and M. Yannakakis, "Testing finite state machines: state identification and verification", *IEEE Trans. on Computers,* vol.43, pp.306-320, 1994.
16. H. Ural, X. Wu and F. Zhang, "On minimizing the length of checking sequence", *IEEE Trans. on Computers*, vol.46, pp.93-99, 1997.
17. M.P. Vasilevskii, "Failure diagnosis of automata", *Kibernetika*, vol.4, pp.98-108, 1973.