Generalizing Redundancy Elimination in Checking Sequences

K. Tuncay Tekle[1], Hasan Ural[2], M. Cihan Yalcin[1], and Husnu Yenigun[1]

[1] Faculty of Engineering and Natural Sciences, Sabanci University, Tuzla 34956, Istanbul, Turkey

[2] School of Information Technology and Engineering, University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada

Abstract. Based on a distinguishing sequence for a Finite State Machine (FSM), an efficient checking sequence may be produced from the elements of a set $E_{\alpha'}$ of α'–sequences and a set E_T of T–sequences, that both recognize the states, and elements of E_C which represents the transitions in the FSM. An optimization algorithm may then be used to produce a reduced length checking sequence by connecting the elements of $E_{\alpha'}$, E_T, and E_C using transitions taken from an acyclic set E''. It is known that only a subset E'_C of E_C is sufficient to form a checking sequence. This paper improves this result by reducing the number of elements in E'_C that must be included in the generated checking sequence.

1 Introduction

Finite state machine (FSM) model has been widely used to specify behaviour of various types of systems [1]. An FSM M models the externally observable behaviour of a system under test (SUT) N in terms of the sequences of inputs and outputs exchanged between a "black box" representing N and its environment. When testing N to ensure its correct functionality with respect to M, a *checking sequence* (i.e., a sequence of inputs constructed from M) is applied to N to determine whether N is a correct or faulty implementation of M [2, 3]. Often, N is considered to have the same input and output alphabets of M and to have no more states than M.

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 s to state s' under input x, firstly, N must be transferred to the state recognized as state s of M; secondly, when the input x is applied, the output produced by N in response to x must be as specified in M; i.e., there must not be an output fault; and thirdly, the state reached by N after the application of x must be recognized as state s' of M; i.e., there must not be 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 [3], a characterization set [3]

P. Yolum et al.(Eds.): ISCIS 2005, LNCS 3733, pp. 915–926, 2005.

or a unique input-output (UIO) sequence [4]. It is known that a distinguishing sequence may not exist for every minimal FSM [5], and that determining the existence of a distinguishing sequence for an FSM is PSPACE-complete [6].

Nevertheless, based on distinguishing sequences, various methods have been proposed for FSM based testing (for example, [3, 7, 8]). Some of these methods aim in generating reduced length checking sequences [8, 9, 10]. A representative example of these methods is [9] which shows that an efficient checking sequence may be produced by combining the elements in some predefined set $E_{\alpha'}$ of α'–sequences that recognize subsets of states, the elements of a set E_T of T–sequences which recognize individual states, and the elements of a set E_C of subsequences that represent individual transitions, using an acyclic set E'' of transitions from M. An optimization algorithm is then used in order to produce a shortest checking sequence by connecting the elements of $E_{\alpha'}$, E_T, and E_C using transitions drawn from E''.

Recently it is shown in [10] that the length of checking sequences can be reduced even further by eliminating some elements of E_C. Those transitions in E_C, that *correspond to the last transitions traversed* when a T–sequence is applied in an α'–sequence, are taken to be the candidate transitions for which transition tests can be eliminated. A dependency relation is derived on these candidate transitions, and only an acyclic subset of them (which does not depend on each other – directly or indirectly – with respect to this dependency relation) is considered to be eliminated.

In this paper, we generalize the condition for a transition to be considered as a candidate for transition test exemption. The candidate transitions are again among the transitions traversed when a T–sequence is applied in an α'–sequence. However, they do not have to be the last transitions traversed. The condition given in this paper trivially holds for the last transitions, hence the approach of [10] is a special case of the approach given in this paper.

Besides the theoretical novelty of providing a more general condition, our approach also has the following practical implication. Since we identify more candidate transitions, the dependency relation between these candidate transitions is more relaxed. This allows us to find acyclic subsets of candidate transitions with greater cardinality, hence we can eliminate more transition tests than the approach of [10].

The rest of the paper is organized as follows. Section 2 gives an overview of the concepts used in constructing checking sequences based on distinguishing sequences, and Section 3 explains an existing approach for the construction of checking sequences. Section 4 presents the proposed method for eliminating redundant transition tests and shows the application of the method to an example. Section 5 gives the concluding remarks.

2 Preliminaries

A deterministic FSM M is defined by a tuple $(S, s_1, X, Y, \delta, \lambda)$ in which S is a finite set of *states*, $s_1 \in S$ is the *initial state*, X is the finite *input alphabet*,

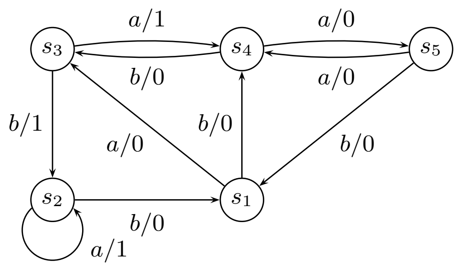

Fig. 1. The FSM M_0

Y is the finite *output alphabet*, $\delta : S \times X \to S$ is the *next state function* and $\lambda : S \times X \to Y$ is the *output function*. The functions δ and λ can be extended to input sequences in a straightforward manner. The number of states of M is denoted n and the states of M are enumerated, giving $S = \{s_1, \ldots, s_n\}$. An FSM is *completely specified* if the functions λ and δ are total.

An FSM, that will be denoted M_0 throughout this paper, is described in Figure 1. Here, $S = \{s_1, s_2, s_3, s_4, s_5\}$, $X = \{a, b\}$ and $Y = \{0, 1\}$.

Throughout the paper, we use barred symbols (e.g. $\bar{x}, \bar{P}, \ldots$) to denote sequences, and juxtaposition to denote concatenation. In an FSM M, $s_i \in S$ and $s_j \in S$, $s_i \neq s_j$, are *equivalent* if, $\forall \bar{x} \in X^*$, $\lambda(s_i, \bar{x}) = \lambda(s_j, \bar{x})$. If $\exists \bar{x} \in X^*$ such that $\lambda(s_i, \bar{x}) \neq \lambda(s_j, \bar{x})$ then $\bar{x}$ is said to *distinguish* s_i and s_j. An FSM M is said to be *minimal* if none of its states are equivalent. A *distinguishing sequence* for an FSM M is an input sequence $\bar{D}$ for which each state of M produces a distinct output. More formally, for all $s_i, s_j \in S$ if $s_i \neq s_j$ then $\lambda(s_i, \bar{D}) \neq \lambda(s_j, \bar{D})$. Thus, for example, M_0 has distinguishing sequence abb.

The shortest prefix of a distinguishing sequence $\bar{D}$ that distinguishes a state in M can actually be used as a special distinguishing sequence for that state [11]. Based on this observation, we use prefixes of distinguishing sequences, in order to further reduce the length of checking sequences. We will use $\bar{D}_i$ to denote the shortest prefix of a distinguishing sequence $\bar{D}$ that is sufficient to distinguish a state s_i from the other states. Formally, given a distinguishing sequence $\bar{D}$ and a state s_i, $\bar{D}_i$ is the shortest prefix of $\bar{D}$ such that for any state s_j, if $s_i \neq s_j$ then $\lambda(s_i, \bar{D}_i) \neq \lambda(s_j, \bar{D}_i)$. For example, M_0 has $\bar{D}_1 = ab$, $\bar{D}_2 = \bar{D}_3 = \bar{D}_4 = \bar{D}_5 = abb$. Below we call $\bar{D}_i$'s as *prefix distinguishing sequences.*

An FSM M can be represented by a directed graph (*digraph*) $G = (V, E)$ where a set of vertices V represents the set S of states of M, and a set of directed edges E represents all transitions of M. Each edge $e = (v_j, v_k, x/y) \in E$ represents a transition $t = (s_j, s_k, x/y)$ of M from state s_j to state s_k with input x and output y where $s_j, s_k \in S$, $x \in X$, and $y \in Y$ such that $\delta(s_j, x) = s_k$, $\lambda(s_j, x) = y$. For a vertex $v \in V$, $indegree_{E'}(v)$ denotes the number of edges from E' that enter v and $outdegree_{E'}(v)$ denotes the number of edges from E' that leave v, where $E' \subseteq E$.

A sequence $\bar{P} = (n_1, n_2, x_1/y_1)(n_2, n_3, x_2/y_2) \ldots (n_{k-1}, n_k, x_{k-1}/y_{k-1})$ of pairwise adjacent edges from G forms a *path* in which each *node* n_i represents a vertex from V and thus, ultimately, a state from S. Here *initial*$(\bar{P})$ denotes n_1, which is the *initial node* of $\bar{P}$, and $final(\bar{P})$ denotes n_k, which is the *final node* of $\bar{P}$. The sequence $\bar{Q} = (x_1/y_1)(x_2/y_2)\ldots(x_{k-1}/y_{k-1})$ is the *label* of $\bar{P}$ and is denoted *label*$(\bar{P})$. In this case, $\bar{Q}$ is said to *label* the path $\bar{P}$. $\bar{Q}$ is said to be a *transfer sequence* from n_1 to n_k. The path $\bar{P}$ can be represented by the tuple $(n_1, n_k, \bar{Q})$ or by the tuple $(n_1, n_k, \bar{x}/\bar{y})$ in which $\bar{x} = x_1x_2\ldots x_{k-1}$ is the *input portion* of $\bar{Q}$ and $\bar{y} = y_1y_2\ldots y_{k-1}$ is the *output portion* of $\bar{Q}$. Two paths $\bar{P}_1$ and $\bar{P}_2$ can be concatenated as $\bar{P}_1\bar{P}_2$ only if $final(\bar{P}_1) = initial(\bar{P}_2)$.

A *tour* is a path whose initial and final nodes are the same. Given a tour $\bar{\Gamma} = e_1e_2\ldots e_k$, $\bar{P} = e_je_{j+1}\ldots e_ke_1e_2\ldots e_{j-1}$ is a path formed by *starting* $\bar{\Gamma}$ with edge e_j, and hence by *ending* $\bar{\Gamma}$ with edge e_{j-1}. An *Euler Tour* is a tour that contains each edge exactly once. A set E' of edges from G is *acyclic* if no tour can be formed using the edges in E'.

A digraph is *strongly connected* if for any ordered pair of vertices (v_i, v_j) there is a path from v_i to v_j. An FSM is *strongly connected* if the digraph that represents it is strongly connected. It will be assumed that any FSM considered in this paper is deterministic, minimal, completely specified, and strongly connected.

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 alphabets 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. 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/faults. 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 reduced length checking sequences (for example, [8, 9]). These notions are defined below in terms of a given distinguishing sequence $\bar{D}$ (more precisely the prefix distinguishing sequences) for FSM M.

3 An Existing Approach

3.1 Basics

Consider the digraph $G = (V, E)$ representing M and let $\bar{Q}$ be the label of a path $\bar{P}$ in G. A vertex v of $\bar{P}$ is said to be *recognized* (in $\bar{Q}$) as a state s_i of M, if the label $\bar{T}$ of a subpath $\bar{R}$ of $\bar{P}$ starting at v has a prefix $\bar{D}_i/\lambda(s_i, \bar{D}_i)$. This rule says that $initial(\bar{R})$ is recognized as state s_i if $label(\bar{R})$ has a prefix $\bar{D}_i/\lambda(s_i, \bar{D}_i)$. Alternatively, if $\bar{P}_1 = (v_i, v_j, \bar{T})$ and $\bar{P}_2 = (v_k, v, \bar{T})$ are two subpaths of $\bar{P}$ such that v_i and v_k are recognized as state s' of M and v_j is recognized as state s of M, then v is said to be recognized (in $\bar{Q}$) as state s of M. This rule says that if $\bar{P}_1$ and $\bar{P}_2$ are labeled by the same input/output sequence at their starting vertices which are recognized as the same state s' of M, then their terminating vertices correspond to the same state s of M. An edge $(v, v', x/y)$ of $\bar{P}$ is said to be *verified* (in $\bar{Q}$) as a transition $(s_i, s_j, x_i/y_i)$ of M if v is recognized as state s_i, v' is recognized as state s_j, $x = x_i$, and $y = y_i$; i.e., v is recognized as state s_i of M and there is a subpath $\bar{P}'$ of $\bar{P}$ starting at v whose label is $x\bar{D}_j/\lambda(s_i, x\bar{D}_j)$. The subpath $\bar{P}'$ is called the *transition test* for the transition $(s_i, s_j, x_i/y_i)$; i.e., $\bar{P}'$ is the transition sequence labeled by (the application of) the input sequence $x\bar{D}_j$ at state s_i. Accordingly, the following result will form the basis of the checking sequence construction method proposed in this paper.

Theorem 1. *(Theorem 1, [8]) Let $\bar{P}$ be a path of G representing an FSM M that starts at s_1 and $\bar{Q} = label(\bar{P})$. If every edge of G is verified in $\bar{Q}$, then the input portion of $\bar{Q}$ is a checking sequence of M.*

Let $\bar{Q}$ be the label of a path $\bar{P}$ in G starting at v_1 such that $\bar{Q}$ contains n subsequences of the form $\bar{D}_i/\lambda(s_i, \bar{D}_i)$, $(1 \leq i \leq n)$. Since $\bar{D}_i$'s are prefix distinguishing sequences for M, each of these subsequences of the form $\bar{D}_i/\lambda(s_i, \bar{D}_i)$, $(1 \leq i \leq n)$, is unique. If $\bar{Q}$ labels a path starting at the initial state of N from $\Phi(M)$ then, since N has at most n states, $\bar{D}_i$'s must also be prefix distinguishing sequences for N. This says that if n different expected responses to $\bar{D}_i$'s are observed in N, then $\bar{D}_i$'s define a one-to-one correspondence between the states of N and M. In this case, we say that the uniqueness of the response of each of the n states of N to $\bar{D}_i$'s are verified and hence N has n distinct states.

Let $DS(s_i)$ denote the transition sequence labeled by $\bar{D}_i/\lambda(s_i, \bar{D}_i)$ at state s_i and let $\bar{T}_i$, called henceforth T–sequence, be $label(\bar{R}_i)$ where $\bar{R}_i = DS(s_i)\bar{B}_i$ and $\bar{B}_i$ is a (possibly empty) sequence of transitions of G starting at $final(DS(s_i))$, $(1 \leq i \leq n)$. Since a T–sequence $\bar{T}_i$ is a sequence of input/output pairs with a prefix $label(DS(s_i)) = \bar{D}_i/\lambda(s_i, \bar{D}_i)$, it may be used to recognize the ending state of any transition terminating at state s_i [8]. $\bar{R}_i$'s can be connected to each other in a succinct manner to form the elements of an α'–set $= \{\bar{\alpha}'_1, \bar{\alpha}'_2, \ldots \bar{\alpha}'_q\}$ where each $\bar{\alpha}'_k$ $(1 \leq k \leq q)$ is called an α'–sequence [9]. An α'–sequence $\bar{\alpha}'_k$ is the label of an α'–path $\bar{\rho}_k = \bar{R}_{k_1}\bar{R}_{k_2}\ldots\bar{R}_{k_{r_k}}$, $1 \leq k_1, k_2, \ldots, k_{r_k} \leq n$, such that (a) $\exists$ an α'–path $\bar{\rho}_j = \bar{R}_{j_1}\bar{R}_{j_2}\ldots\bar{R}_{j_{r_j}}$, $1 \leq j \leq q$ and $1 \leq j_1, j_2, \ldots, j_{r_j} \leq n$, such that for some i, $1 \leq i < r_j$, $k_{r_k} = j_i$; and (b) No other $\bar{R}_{k_i}$, $1 \leq i < r_k$, in $\bar{\rho}_k$ satisfies (b). In other words for every α'–path $\bar{\rho}_k$, the last component and only the last component $\bar{R}_{k_{r_k}}$ in $\bar{\rho}_k$ appears in the same or in some other α'–path $\bar{\rho}_j$ before the last component in $\bar{\rho}_j$. Since $\bar{\alpha}'_k = label(\bar{\rho}_k)$, $\bar{\alpha}'_k$ will be concatenation of T–sequences. A set of α'–sequences is called an α'–set only if $\forall \bar{T}_i$, $1 \leq i \leq n$, $\exists\ \bar{\alpha}'_k$, $1 \leq k \leq q$, such that $\bar{T}_i$ is a subsequence of $\bar{\alpha}'_k$.

Let $A = \{\bar{\alpha}'_1, \bar{\alpha}'_2, \ldots, \bar{\alpha}'_q\}$ be an α'–set with the corresponding set of α'–paths $\{\bar{\rho}_1, \bar{\rho}_2, \ldots, \bar{\rho}_q\}$, and $\bar{Q}$ be the label of a path $\bar{P}$ such that each $\bar{\alpha}'_k \in A$ is a subsequence of $\bar{Q}$. Then we have the following properties:

1) Since $\bar{\alpha}'_k$ starts with a $\bar{T}_i$ that has a prefix $\bar{D}_i/\lambda(s_i, \bar{D}_i)$, $initial(\bar{\rho}_k)$ is recognized in $\bar{Q}$
2) Since every α'–sequence $\bar{\alpha}'_k$ is a subsequence of $\bar{Q}$, $final(\bar{\rho}_k)$ is recognized in $\bar{Q}$

3) Since every $\bar{T}_i$ is in some $\bar{\alpha}'_k$, $initial(\bar{R}_i)$ is recognized in $\bar{Q}$
4) Since every $\bar{T}_i$ is followed by a $\bar{T}_j$ in some $\bar{\alpha}'_k$, $final(\bar{R}_i)$ is recognized in $\bar{Q}$
5) Since $\bar{\alpha}'_k$ starts with a $\bar{T}_i$ that has a prefix $\bar{D}_i/\lambda(s_i, \bar{D}_i)$, $\bar{\alpha}'_k$ may be used to recognize the ending state of any transition terminating at state s_i [9].

3.2 Checking Sequence Construction

The checking sequence construction method given in [9] first builds a digraph $G' = (V', E')$ by augmenting the digraph $G = (V, E)$ representing an FSM where $V' = V \cup U'$, $E' = E_C \cup E_{\alpha'} \cup E_T \cup E''$ by:

- replicating each vertex v in V as a vertex v' in U' to represent the "recognized" version of v
- replacing each edge $(v_i, v_j, x/y)$ of E by an edge $(v'_i, v_j, x/y)$ in E_C so that the transition to be verified starts at the recognized vertex v'_i
- inserting an edge $(v_i, v'_j, \bar{\alpha}'_k)$ in $E_{\alpha'}$ for each $\bar{\rho}_k = (v_i, v_j, \bar{\alpha}'_k)$, $(1 \leq k \leq q)$ so that $\bar{\rho}_k$ ends at the recognized vertex v'_j
- inserting an edge $(v_i, v'_j, \bar{T}_m)$ in E_T for each $\bar{R}_m = (v_i, v_j, \bar{T}_m)$, $(1 \leq m \leq n)$ so that $\bar{R}_m$ ends at the recognized vertex v'_j
- inserting an edge $(v'_i, v'_j, x/y)$ in E'' for each edge $(v_i, v_j, x/y)$ in a subset of edges of E such that $G'' = (U', E'')$ does not have a tour and G' is strongly connected.

Note that in G' each edge in E_C is followed by an edge from $E_{\alpha'} \cup E_T$ to form a transition test for the transition corresponding to that edge of E_C. Then, the approach in [9] forms a minimal symmetric augmentation G^* of the digraph induced by $E_{\alpha'} \cup E_C$ by adding replications of edges from E' . If G^*, with its isolated vertices removed, is connected, then G^* has an Euler tour. Otherwise, a heuristic such as the one given in [8] is applied to make G^* connected and an Euler tour of this new digraph is formed. On the basis of Theorem 1, it is argued in [9] that the input portion of the label of the Euler tour of G^* starting at vertex v_1 which is followed by $\bar{D}_1$ is a checking sequence of M.

4 An Enhancement on the Existing Approach

This section explains how, given an α'–set A, we can produce a checking sequence without considering some of the edges in E_C. In the following, we first define a set of edges $L \subset E$, then show that transition tests for the edges in L are redundant, and finally explain how we can modify the algorithm to generate the checking sequence in order not to include these redundant transition tests.

4.1 Transition Test Exemption

In this section, we consistently use $\bar{P}$ to denote a path in G, and $\bar{Q}$ to denote $label(\bar{P})$. Similar to showing an edge being verified as given in Section 3.1, in order to show a sequence of edges being verified we first introduce the notion of a sequence of edges being traced.

Definition 1. *Let $\bar{P}' = e_1 e_2 \ldots e_h$ be a sequence of edges in G, where $e_m = (v_{i_m}, v_{i_{m+1}}, x_m/y_m)$ for $1 \le m \le h$. $\bar{P}'$ is* traced *in $\bar{Q}$ if there exists a subpath $(n_1, n_{h+1}, x'_1 x'_2 \ldots x'_h / y'_1 y'_2 \ldots y'_h)$ in $\bar{P}$ such that n_1 is recognized as v_{i_1}, n_{h+1} is recognized as $v_{i_{h+1}}$, and $x_m/y_m = x'_m/y'_m$ for $1 \le m \le h$.*

Lemma 1. *Let $\bar{P}' = e_1 e_2 \ldots e_h$ $(h \ge 1)$ be a sequence of edges in G traced in $\bar{Q}$, where $e_m = (v_{i_m}, v_{i_{m+1}}, x_m/y_m)$ for $1 \le m \le h$. Assume that $e_1, e_2, \ldots, e_l$ for some $1 \le l < h$ are all verified in $\bar{Q}$. Then, $\bar{P}'' = e_{l+1} e_{l+2} \ldots e_h$ is also traced in $\bar{Q}$.*

Proof. The proof is by induction on l. Let's assume $l = 1$. If e_1 is verified in $\bar{Q}$, then $\bar{P}$ includes a subpath $\bar{P}_1 = (n_j, n_k, x'_1/y'_1)$ where n_j is recognized as v_{i_1}, n_k is recognized as v_{i_2} and $x_1/y_1 = x'_1/y'_1$. Since $\bar{P}'$ is traced in $\bar{Q}$, there must exist a subpath $\bar{P}_2 = (n_q, n_s, x''_1 x''_2 \ldots x''_h / y''_1 y''_2 \ldots y''_h)$ in $\bar{P}$ where n_q is recognized as v_{i_1}, n_s is recognized as $v_{i_{h+1}}$, and $x''_m/y''_m = x_m/y_m$ for $1 \le m \le h$. Let us divide the path $\bar{P}_2$ into two as $\bar{P}_{21} = (n_q, n_i, x''_1/y''_1)$ and $\bar{P}_{22} = (n_i, n_s, x''_2 x''_3 \ldots x''_h / y''_2 y''_3 \ldots y''_h)$. According to the definition of a recognized vertex given in Section 3.1, the paths $\bar{P}_1$ and $\bar{P}_{21}$ recognize n_i as v_{i_2}. Then, the existence of $\bar{P}_{22}$ in $\bar{P}$ implies that $\bar{P}'' = e_2 e_3 \ldots e_h$ is traced in $\bar{Q}$.

For the inductive step, we can again use the arguments given above to conclude that $\bar{P}'' = e_2 e_3 \ldots e_h$ is traced in $\bar{Q}$. However, we have $l - 1$ verified transitions at the beginning of $\bar{P}''$, hence the proof is completed by using the induction hypothesis. □

Definition 2. *An edge $(v, v', x_v/y_v)$ in G is said to be a* nonconverging edge *[12] if $\forall (u, u', x_u/y_u)$, $u \ne v$ and $x_u = x_v$ implies $u' \ne v'$ or $y_u \ne y_v$.*

Lemma 2. *Let $(v, v', x_v/y_v)$ be a nonconverging edge in G, and $(n_p, n_q, x/y)$ be a subpath of $\bar{P}$ such that n_q is recognized as v' and $x/y = x_v/y_v$. If all the edges $(u, u', x_u/y_u)$ in G, where $x_u = x_v$ are verified in $\bar{Q}$, then n_p is recognized as v.*

Proof. Since all the edges in G corresponding to the x_v transitions of the states in M are verified, and since the state corresponding to the node v is the only state that produces y_v and moves into the state corresponding to the node v' when x_v is applied, we can conclude that whenever x_v/y_v is seen in $\bar{Q}$ and the ending node is recognized as v', then the previous node must be the node v. □

Lemma 3. *Let $\bar{P}' = e_1 e_2 \ldots e_h$ $(h \ge 1)$ be a sequence of edges in G traced in $\bar{Q}$, where $e_m = (v_{i_m}, v_{i_{m+1}}, x_m/y_m)$ for $1 \le m \le h$. Assume that $e_l, e_{l+1}, \ldots, e_h$ for some $1 < l \le h$ are all verified in $\bar{Q}$. If for all $l \le r \le h$*
(i) $e_r = (v_{i_r}, v_{i_{r+1}}, x_r/y_r)$ is a nonconverging edge, and
(ii) For each vertex v in G, all the edges of the form $(v, v', x/y)$ (where $x = x_r$) are verified in $\bar{Q}$ then $\bar{P}'' = e_1 e_2 \ldots e_{l-1}$ is also traced in $\bar{Q}$.

Proof. The proof is by induction on $h-l+1$. Let's assume $h-l+1 = 1$, i.e $l = h$. Since $\bar{P}'$ is traced in $\bar{Q}$, $\bar{P}'$ must be a subpath $(n_q, n_s, x'_1 x'_2 \ldots x'_h / y'_1 y'_2 \ldots y'_h)$ of $\bar{P}$ where n_q is recognized as v_{i_1}, n_s is recognized as $v_{i_{h+1}}$, and $x'_m/y'_m = x_m/y_m$ for $1 \le m \le h$. Let us divide $\bar{P}'$ into two as $\bar{P}_1 = (n_q, n_i, x'_1 x'_2 \ldots x'_{h-1} / y'_1 y'_2 \ldots y'_{h-1})$

and $\bar{P}_2 = (n_i, n_s, x'_h/y'_h)$. Note that $\bar{P}_2$ corresponds to e_h which is a nonconverging edge. Since all the edges of the form $(v, v', x_h/y)$ are verified in $\bar{Q}$, n_i is recognized as v_{i_h} by using Lemma 2. Then $\bar{P}_1 = e_1e_2 \ldots e_{h-1}$ is traced in $\bar{Q}$.

For the inductive step, we can again use the arguments given above to conclude that $\bar{P}'' = e_1e_2 \ldots e_{h-1}$ is traced in $\bar{Q}$. However, we have $h - l$ verified transitions at the end of $\bar{P}''$, hence the proof is completed by using the induction hypothesis. □

Lemma 4. *Let $\bar{P}' = e_1e_2 \ldots e_h$ $(h \geq 1)$ be a sequence of edges in G traced in $\bar{Q}$, where $e_m = (v_{i_m}, v_{i_{m+1}}, x_m/y_m)$ for $1 \leq m \leq h$. Let e_l be an edge in $\bar{P}'$, where $1 \leq l \leq h$. If*
(i) $\forall r, 1 \leq r \leq h, r \neq l$ implies e_r is verified in $\bar{Q}$, and
(ii) $\forall r$, $l < r \leq h$
(ii.a) $e_r = (v_{i_r}, v_{i_{r+1}}, x_r/y_r)$ is a nonconverging edge; and
(ii.b) For each vertex v in G, all the edges of the form $(v, v', x/y)$ (where $x = x_r$) are verified in $\bar{Q}$ then e_l is also verified in $\bar{Q}$.

Proof. Since $\bar{P}'$ is traced in $\bar{Q}$, $\bar{P}'$ is a subpath $(n_q, n_s, x'_1x'_2 \ldots x'_h/y'_1y'_2 \ldots y'_h)$ of $\bar{P}$ where n_q is recognized as v_{i_1}, n_s is recognized as $v_{i_{h+1}}$, and $x'_m/y'_m = x_m/y_m$ for $1 \leq m \leq h$. Let us divide the path $\bar{P}'$ into three as follows: $\bar{P}_1 = (n_q, n_i, x'_1x'_2 \ldots x'_{l-1}/y'_1y'_2 \ldots y'_{l-1})$, and $\bar{P}_2 = (n_i, n_s, x'_l/y'_l)$, and finally $\bar{P}_3 = (n_s, n_t, x'_{l+1}x'_{l+2} \ldots x'_h/y'_{l+1}y'_{l+2} \ldots y'_h)$. By using Lemma 1, $\bar{P}_2\bar{P}_3$ is traced in $\bar{Q}$ and n_i is therefore recognized as v_{i_l}. By using Lemma 3, $\bar{P}_1\bar{P}_2$ is traced in $\bar{Q}$ and n_s is therefore recognized as $v_{i_{l+1}}$. Since both n_i and n_s are recognized in $\bar{P}_2 = e_l$, e_l is verified. □

Lemma 4 suggests that if there is a sequence of edges which is traced in the label $\bar{Q}$ of a path, then $\bar{Q}$ already includes what it takes to verify an edge e_l in the sequence, provided that the conditions (i), (ii.a) and (ii.b) given in the premises of Lemma 4 hold. Therefore, we can pick a transition e_l in a sequence of edges which is known to be traced, and do not include the transition test for e_l, provided that the conditions are satisfied for e_l. Note that, one can always pick e_h as e_l (the last transition in the sequence of edges) according to the conditions of Lemma 4. This is what has been proposed in [10], and therefore the approach given in [10] is a special case of our approach.

In fact, inclusion of α'–sequences in the checking sequences guarantee that there are some sequences of edges which are traced, as shown by the following lemma.

Lemma 5. *Let A be an α'–set, and $\bar{Q}$ include all the α'–sequences in A. Then $\forall i, 1 \leq i \leq n$, $\bar{R}_i = DS(s_i)\bar{B}_i$ is traced in $\bar{Q}$.*

Proof. Note that $\exists \bar{\alpha}'_k$, $1 \leq k \leq q$, with the subsequence $label(\bar{R}_i)label(\bar{R}_j)$ for some j, $1 \leq j \leq n$. Since $label(\bar{R}_i)$ starts with $\bar{D}_i/\lambda(s_i, \bar{D}_i)$, $initial(\bar{R}_i)$ is recognized. Since $label(\bar{R}_j)$ starts with $\bar{D}_j/\lambda(s_j, \bar{D}_j)$, $initial(\bar{R}_j)$, hence $final(\bar{R}_i)$ is also recognized. □

Lemma 6. *Let A be an α'–set, and $\bar{Q}$ include all the α'–sequences in A, and $\bar{R}_i = e_{j_1}e_{j_2}\ldots e_{j_h}$ be the sequence of edges corresponding to the application of the T–sequence $\bar{T}_i$ at a state s_i. Let $e_{j_l} = (v_{j_l}, v_{j_{l+1}}, x_{j_l}/y_{j_l})$ be an edge in $\bar{R}_i$. If*
(i) $\forall r, 1 \leq r \leq h, r \neq l$ implies e_{j_r} is verified in $\bar{Q}$, and
(ii) $\forall r, l < r \leq h$
(ii.a) e_{j_r} is a nonconverging edge; and
(ii.b) For each vertex v in G, all the edges of the form $(v, v', x/y)$ (where $x = x_{j_l}$) are verified in $\bar{Q}$ then e_{j_l} is also verified in $\bar{Q}$.

Proof. The result follows from Lemma 4 and Lemma 5. □

Lemma 6 suggests that one can identify an edge per state to be excluded from the transition tests. However, if we identify some edge e for a state s, exclusion of e depends on some other transitions being verified, as given in the premises of Lemma 6. We may identify another edge e' for another state s'. Nevertheless, exclusion of e may depend on e' being verified, and exclusion of e' may depend on e being verified (either directly or indirectly). The following procedure shows a possible way to calculate a set of edges that can be excluded from the transition tests without having such a cyclic dependency.

For an $\bar{R}_i = DS(s_i)\bar{B}_i = e_1e_2\ldots e_h$, $1 \leq i \leq n$, an edge e_l $(1 \leq l \leq h)$ is a *candidate edge of* $\bar{R}_i$ if $\forall r, l < r \leq h$, e_r is a nonconverging edge. Note that e_h is always a candidate edge of $\bar{R}_i$ according to this definition. Let $L_0 = \{e \mid e \text{ is a candidate edge of } \bar{R}_i, 1 \leq i \leq n\}$.

Note that, the generated checking sequence must start from s_1, the initial state of M. Therefore at least one incoming transition of s_1 must be tested, so that the generated tour passes over v_1. Therefore let L_1 be a maximal subset of L_0 such that, $indegree_{L_1}(v_1) < indegree_E(v_1)$.

Further note that according to Lemma 6, the test for a transition can be exempted only if some other transitions are tested. In order to avoid cyclic dependencies, the following algorithm can be used:

Construct a digraph $G_S = (V_S, E_S)$ where V_S contains one vertex for each $e \in L_1$. $(v_1, v_2) \in E_S$ if and only if $v_1 \neq v_2$, and for some $\bar{R}_i$, the edges e_1 and e_2 corresponding the vertices v_1 and v_2 appear in $\bar{R}_i$. Find a maximal subgraph $G'_S = (V'_S, E'_S)$ of G_S by removing vertices from G_S (and the edges connected to the removed vertices) such that E'_S is acyclic. Let L be the set of edges that correspond to the vertices in V'_S.

Finding G'_S is an instance of Feedback Vertex Set problem [13], which is NP–complete. However certain heuristic approaches exist for this problem [14, 15]. Note that for an $\bar{R}_i$, there will always be a cyclic dependency between the candidate edges of $\bar{R}_i$. Therefore only one of the edges in $\bar{R}_i$ will survive in G'_S. Hence, at most n transition tests can be removed from the checking sequence.

4.2 Improved Checking Sequence Construction

Now using L, we can improve on the algorithm in [9] for the checking sequence generation, by reducing the set of edges that must be included in the checking sequence. First the digraph $G' = (V', E')$ is obtained as explained in Section 3.2. E'' can be constructed similarly as discussed in [8].

Theorem 2. *Let E'_C be defined as $E'_C = \{(v'_i, v_j, x/y) : (v_i, v_j, x/y) \in E - L\}$. Let $\bar{\Gamma}$ be a tour of G' that contains all edges in $E_{\alpha'} \cup E'_C$ which is found in the same manner as in [9]. Let $e = (v'_i, v_1, x/y) \in E'_C$ be an edge in $\bar{\Gamma}$ ending at v_1 that corresponds to the initial state s_1 of M. Let $\bar{P}$ be a path of G' that is formed by ending $\bar{\Gamma}$ with edge e, and $\bar{Q} = label(\bar{P})\bar{D}_1/\lambda(s_1, \bar{D}_1)$. Then the input portion of $\bar{Q}$ is a checking sequence of M.*

Proof. All edges in $E - L$ are verified in $\bar{Q} = label(\bar{P})\bar{D}_1/\lambda(s_1, \bar{D}_1)$. According to Lemma 6 and the way L is constructed, if all edges in $E - L$ are verified in $\bar{Q}$, then all edges in L are verified in $\bar{Q}$. Thus, all edges of G are verified in $\bar{Q}$, and by Theorem 1, the input portion of $\bar{Q}$ is a checking sequence of M. □

4.3 Application

Let us consider FSM M_0 given in Figure 1. A distinguishing sequence for M_0 is $\bar{D} = abb$. The shortest prefixes of $\bar{D}$ that are sufficient to distinguish each state are: $\bar{D}_1 = ab$, $\bar{D}_2 = \bar{D}_3 = \bar{D}_4 = \bar{D}_5 = abb$. In this example, we will use $\bar{B}_i$'s in $\bar{R}_i = DS(s_i)\bar{B}_i$, as empty sequences. Hence $\bar{T}_i = \bar{D}_i/\lambda(s_i, \bar{D}_i)$, $1 \leq i \leq n$. Using these $\bar{T}_i$'s, an α'–set for M_0 is $\{\bar{\alpha}'_1 = \bar{T}_1\bar{T}_2\bar{T}_4\bar{T}_4, \bar{\alpha}'_2 = \bar{T}_3\bar{T}_2, \bar{\alpha}'_3 = \bar{T}_5\bar{T}_4\}$, with the following corresponding α'–paths: $\bar{\rho}_1 = (v_1, v_4, \bar{\alpha}'_1)$, $\bar{\rho}_2 = (v_3, v_4, \bar{\alpha}'_2)$, $\bar{\rho}_3 = (v_5, v_4, \bar{\alpha}'_3)$.

Note that in FSM M_0, all the edges except $(v_2, v_1, b/0)$ and $(v_5, v_1, b/0)$ are nonconverging edges. According to the definition of candidate edges given in Section 4.1, the set L_0 can be found as $\{(v_1, v_3, a/0), (v_1, v_4, b/0), (v_2, v_1, b/0), (v_3, v_4, a/1), (v_3, v_2, b/1), (v_4, v_3, b/0), (v_5, v_4, a/0), (v_5, v_1, b/0)\}$. Note that $\forall e \in L_0$, $\exists \bar{R}_i$, $1 \leq i \leq n$, such that e occurs in $\bar{R}_i$, and all the edges that come after e in $\bar{R}_i$ are nonconverging edges.

Since all the incoming edges of v_1 are in L_0, we need to exclude one of the incoming edges of v_1 from L_0 to get L_1. Let $L_1 = L_0 \setminus \{(v_2, v_1, b/0)\}$.

A maximal acyclic subgraph G'_S of G_S for L_1 includes the vertices corresponding to the following edges: $L = \{(v_1, v_3, a/0), (v_1, v_4, b/0), (v_2, v_1, b/0), (v_3, v_4, a/1), (v_5, v_4, a/0)\}$.

The graph $G' = (V', E')$ is given in Figure 2.

A tour $\bar{\Gamma}$ over G' that contains all the edges in $E_{\alpha'} \cup E'_C$ is

$$(v_1, v'_4, \bar{\alpha}'_1), (v'_4, v_5, a/0), (v_5, v'_4, \bar{\alpha}'_3), (v'_4, v_3, b/0), (v_3, v'_4, \bar{\alpha}'_2), (v'_4, v'_3, b/0),$$
$$(v'_3, v_2, b/1), (v_2, v'_4, \bar{T}_2), (v'_4, v'_3, b/0), (v'_3, v'_2, b/1), (v'_2, v_2, a/1), (v_2, v'_4, \bar{T}_2),$$
$$(v'_4, v'_5, a/0), (v'_5, v_1, b/0)$$

Note that $\bar{\Gamma}$ already starts at v_1. Hence when we consider the path $\bar{P}$ corresponding to $\bar{\Gamma}$ given above, the input portion of $\bar{Q} = label(\bar{P})\bar{D}_1/\lambda(s_1, \bar{D}_1)$ forms a checking sequence of length 40. Using the approach of [10], only the transition tests for the edges $(v_1, v_4, b/0)$ and $(v_3, v_2, b/1)$ are found to be redundant, since these are the only edges that occur as the last edges in $\bar{R}_i$'s. The

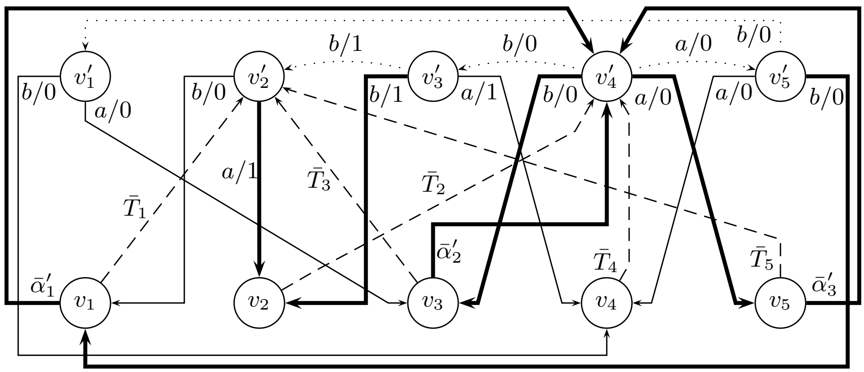

Fig. 2. $G' = (V', E')$ for M_0. The nodes in V and U' are at the bottom, and at the top respectively. The dashed lines are the edges in E_T, and the dotted lines are the edges in E''. The edges in $E_{\alpha'} \cup E_C$ are given in solid lines. The bold solid lines are the edges in $E_{\alpha'} \cup E_C'$, and the remaining solid lines are the edges in L.

checking sequence in this case is found to be of length 52, which is still shorter than the checking sequence of length 63 that would be found by applying the general method proposed in [9].

5 Conclusion

We have shown that, when α'–sequences are used in constructing a checking sequence, some transitions tests can be identified as redundant. Such tests are then eliminated by the optimization algorithm used to construct a shorter checking sequence, and hence a further reduction is obtained in the length of a resulting checking sequence. We have also shown that our approach can identify more redundant transition tests than the approach of a similar work given in [10].

The approach proposed in this paper starts with a given set of α'–sequences. We believe that selecting α'–sequences judiciously will result in further reductions in the length of a checking sequence. A recent study by Hierons and Ural [16] show how α'–sequences can be chosen so that their use minimizes the sum of the lengths of the subsequences to be combined in checking sequence generation. The related checking sequence generation algorithm then produces the set of connecting transitions *during* the optimization phase. Our proposed approach can also be incorporated to the method given in [16].

Acknowledgment

This work was supported in part by "Natural Sciences and Engineering Research Council of Canada under grant RGPIN 976".

References

1. Tanenbaum, A.S.: Computer Networks. 3rd edn. Prentice Hall International Editions, Prentice Hall (1996)
2. Gill, A.: Introduction to the Theory of Finite–State Machines. McGraw–Hill, New York (1962)
3. Hennie, F.C.: Fault–detecting experiments for sequential circuits. In: Proceedings of Fifth Annual Symposium on Switching Circuit Theory and Logical Design, Princeton, New Jersey (1964) 95–110
4. Sabnani, K., Dahbura, A.: A protocol test generation procedure. Computer Networks **15** (1988) 285–297
5. Kohavi, Z.: Switching and Finite Automata Theory. McGraw–Hill, New York (1978)
6. Lee, D., Yannakakis, M.: Testing finite state machines: state identification and verification. IEEE Trans. Computers **43** (1994) 306–320
7. Gonenc, G.: A method for the design of fault detection experiments. IEEE Transactions on Computers **19** (1970) 551–558
8. Ural, H., Wu, X., Zhang, F.: On minimizing the lengths of checking sequences. IEEE Transactions on Computers **46** (1997) 93–99
9. Hierons, R.M., Ural, H.: Reduced length checking sequences. IEEE Transactions on Computers **51** (2002) 1111–1117
10. Chen, J., Hierons, R., Ural, H., Yenigun, H.: Eliminating redundant tests in a checking sequence. In: 17th IFIP International Conference on Testing of Communicating Systems. Volume 3502 of Lecture Notes in Computer Science., Montreal, Canada (2005) 146–158
11. Lee, D., Yannakakis, M.: Principles and methods of testing finite–state machines – a survey. Proceedings of the IEEE **84** (1996) 1089–1123
12. Miller, R.E., Paul, S.: On the generation of minimal–length conformance tests for communication protocols. IEEE/ACM Transactions on Networking **1** (1993) 116–129
13. Garey, M.R., Johnson, D.S.: Computers and Intractability. W. H. Freeman and Company, New York (1979)
14. Bar-Yehuda, R., Geiger, D., Naor, J., Roth, R.: Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and bayesian inference. In: Proceedings of Fifth ACM-SIAM Symposium on Discrete Algorithms. (1994) 344–354
15. Fujito, T.: A note on approximation of the vertex cover and feedback vertex set problems. Information Processing Letters **59** (1996) 59–63
16. Hierons, R.M., Ural, H.: Optimizing the length of checking sequences. IEEE Transactions on Computers (2004) submitted.