Reduced Length Checking Sequences

Robert M. Hierons[1] and Hasan Ural[2]

1 Department of Information Systems and Computing, Brunel University, Middlesex, UB8 3PH, United Kingdom

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

Abstract -- Here the method proposed in [13] for constructing minimal-length checking sequences based on distinguishing sequences is improved. The improvement is based on optimizations of the state recognition sequences and their use in constructing test segments. It is shown that the proposed improvement further reduces the length of checking sequences produced from minimal, completely specified, and deterministic finite state machines.

Index Terms -- Finite State Machine, Checking Sequence, Test Minimization, Distinguishing Sequence.

1 INTRODUCTION

Finite state machines (FSMs) have been used to model many types of systems including control circuits [11], pattern matching systems, machine learning systems, and communication protocols [1, 4]. FSM based modelling techniques are also often employed in defining the control structure of a system specified using languages such as SDL [2], Estelle [3], and State Charts [7]. Principles of testing an implementation *I* of a system modelled as an FSM *M* can be found in sequential circuit and switching system testing literature [5], where determining, under certain assumptions, whether *I* is a correct implementation of *M* is referred to as a *fault detection experiment*. This experiment consists of applying an input sequence (derived from *M*) to *I*, observing the actual output sequence produced by *I* in response to the application of the input sequence, and comparing the actual output sequence with the expected output sequence. The applied input sequence and the expected output sequence form a *checking sequence*.

Given an FSM *M*, that models the required behaviour of an implementation *I*, it is common to assume that *I* behaves like some unknown FSM (i.e., a "black box") with the same input and output sets as *M*, and that the faults in *I* do not increase the number of states in *I* but may alter the output and destinations of transitions in *I*. A further common assumption is that *M* is minimal, completely specified, and represented by a strongly connected digraph [8].

During the construction of a checking sequence from *M*, the following steps must be carried out in order to verify the correct implementation of each state transition of *M* by *I*, (say, from state s_j to state s_k under input x),

a) before the application of x, *I* must be transferred to the state recognized as state s_j,

b) the output produced by *I* in response to the application of x must be as specified in *M*,

c) the state reached by *I* after the application of x must be recognized as s_k.

Steps b) and c) are collectively called a *test segment* for the transition.

Clearly, a crucial part of testing the correct implementation of each transition is recognizing the starting and terminating states of the transition. The recognition of a state of an FSM M can be achieved by a distinguishing sequence which is an input sequence for which the output sequence produced by M in response to this input sequence is different for each state of M [5]. It is known that a distinguishing sequence may not exist for every minimal FSM [5], and determining the existence of a distinguishing sequence of an FSM is PSPACE-complete [9]. Nevertheless, based on distinguishing sequences, a variety of methods for the construction of checking sequences have been proposed in the literature [5, 6, 8, 12, 13, 14]. An excellent survey on testing FSMs is given by Lee and Yannakakis [10]. However, the particular problem of constructing minimal length checking sequences remains open [6, 13].

This paper considers the problem of generating a minimal length checking sequence in the presence of a distinguishing sequence and improves the work of Ural et al. [13] by modifying it in two ways, each contributing to a reduction in the length of the checking sequence produced. These two modifications are related to extending the definition and the use of α-sequences which are state recognition sequences such that each α-sequence recognizes a subset of states of a given FSM. Firstly, the notion of α-sequences [13] is extended to α'-sequences. The essential difference between α-sequences and α'-sequences is that an α-sequence α_k must end in a section from within its own body while an α'-sequence α'_k can end in a section from within the body of some other α'-sequence $\alpha'_{k'}$. It is shown in this paper that α'-sequences have two main advantages: α'-sequences may be shorter than α-sequences and the use of α'-sequences increases the set of checking sequences over which optimization occurs.

A second improvement is the use of α'-sequences in forming test segments for some transitions. Ural et al. form a test segment for each transition of the given FSM by appending explicitly a distinguishing sequence at the end of the transition to verify the state reached by the transition. Since an α'-sequence starts with a distinguishing sequence, its use in a checking sequence for state recognition eliminates the need for the explicit use of distinguishing sequences for state recognition and hence further reduces the length of the resulting checking sequence.

The rest of the paper is structured as follows. Section 2 provides an overview of related material. Section 3 then describes the new approach, contrasting it with that in [13], applies both approaches to an example, and compares these approaches. Section 4 gives the conclusions.

2 PRELIMINARIES

A *finite state machine* (FSM) is a quintuple $M = (S, X, Y, \delta, \lambda)$, where S is a finite set of states, 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. Functions δ and λ can be extended to take input sequences in the normal way [13]. $s_1 \in S$ is considered as the *initial state* of M. States s_i, $s_j \in S$, $i \neq j$, are *equivalent* if, for every input sequence $I \in X^*$, $\lambda(s_i, I) = \lambda(s_j, I)$. M is *minimal* if there is no pair of states s_i, $s_j \in S$, $i \neq j$, that are equivalent.

M can be represented by a digraph $G = (V, E)$ where vertex set $V = \{v_1, v_2, ..., v_n\}$ represents the set S of states of M, $|S| = n$, and an edge $e = (v_j, v_k; x/y) \in E$ represents a transition from state s_j to state s_k with input $x \in X$ and output $y \in Y$. Here v_j and v_k are the *head* and *tail* of e, denoted *head*(e) and *tail*(e), respectively and input/output (i/o pair) x/y is the *label* of e, denoted *label*(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 is a finite sequence of (not necessarily distinct) adjacent edges in E, where each node n_i represents a vertex from V; n_1 and n_r are the *head* and *tail* of P, denoted *head*(P) and *tail*(P), respectively; and $(x_1/y_1)(x_2/y_2) ... (x_{r-1}/y_{r-1})$ is the *label* of P, denoted *label*(P). For convenience, P will be represented by $(n_1, n_r; I/O)$ where *label*(P) = I/O is the *IO-sequence* $(x_1/y_1)(x_2/y_2) ... (x_{r-1}/y_{r-1})$, input sequence $I = (x_1x_2 ... x_{r-1})$ is the *input portion* of I/O, and output sequence $O = (y_1y_2 ... y_{r-1})$ is the *output portion* of I/O. G is *strongly connected* if for all $v_i, v_j \in V$, there is a path from v_i to v_j. The *cost* (or *length*) of an edge is the number of i/o pairs in the label of the edge. The *cost* (or *length*) of path P is the sum of the costs of edges in P. The concatenation of two sequences (or paths) P and Q is denoted by PQ.

Digraph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. Digraph $G = (V, E)$ is *symmetric* if every vertex $v \in V$ has the same number of edges from E entering it as leaving it. A *rural postman path* (*RPP*) from v_i to v_j over $E' \subseteq E$ in $G = (V, E)$ is a path from v_i to v_j that includes all edges of E'. A *rural Chinese postman path* (*RCPP*) from v_i to v_j over $E' \subseteq E$ in $G = (V, E)$ is a minimum-cost RPP. A *tour* is a path that starts and terminates at the same vertex. An *Euler tour* of $G = (V, E)$ is a tour that contains every edge in E exactly once.

Consider a minimal FSM $M = (S, X, Y, \delta, \lambda)$ represented by strongly connected digraph $G = (V, E)$. A *transfer sequence* T of M from state s_i to state s_j is the label of a path from s_i to s_j. A *distinguishing sequence* of M is an input sequence D for which the output sequence produced by M in response to D identifies the state of M: for all s_i, $s_j \in S$, $i \neq j$, $\lambda(s_i, D) \neq \lambda(s_j, D)$. Let $\Phi(M)$ be the set of FSMs that have at most n states and the same input and output sets as M and suppose $M^* \in \Phi(M)$. M and M^* are *equivalent* if and only if for every state in M there is a corresponding equivalent state in M^*, and vice versa. A *checking sequence* of M is an IO-sequence I/O starting at a specific state of M that distinguishes M from any $M^* \in \Phi(M)$ that is not equivalent to M.

Let D denote a distinguishing sequence of M and let an IO-sequence Q of M be the label of a path $P = (n_1, n_r; Q) = (n_1, n_2; L_1)(n_2, n_3; L_2) ... (n_{r-1}, n_r; L_{r-1})$ of G (cf. Fig. 1a). Hence, Q = $L_1 L_2 ... L_{r-1}$ where $r > 1$, $L_j = x_j/y_j$, $1 \leq j \leq r$-1. It is shown in [13] that if every edge of G is verified in the IO-sequence Q, then Q is a checking sequence of M that starts at v_1.

In the following, we recall the definitions of *recognition* of a node n_i of P in Q as some state of M and *verification* of an edge $e = (a, b; x/y)$ of G in Q given in [13].

Definition 1 A node n_i of P is *d-recognized* in Q as some state a of M if n_i is the head of a subpath of P whose label is an IO-sequence $D/\lambda(a, D)$.

Definition 2 Suppose that $(n_q, n_i; T)$ and $(n_j, n_k; T)$ are subpaths of P and $D/\lambda(a, D)$ is a prefix of T, and thus nodes n_q and n_j are d-recognized in Q as state a of M. Suppose also that node n_k is d-recognized in Q as some state a' of M. Then, node n_i is *t-recognized* in Q as state a' of M.

Definition 3 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 either d-recognized or t-recognized in Q as some state a of M, and node n_k is either d-recognized or t-recognized in Q as some state a' of M. Then, node n_i is *t-recognized* in Q as state a' of M.

If node n_i of P is d-recognized or t-recognized in Q as some state a of M, then it is said to be *recognized* as state a. A node of P is said to be *recognized* if it is recognized as some state a.

Definition 4 An edge $e = (a, b; x/y)$ of G is *verified* in Q if there is a subpath $(n_i, n_{i+1}; x_i/y_i)$ of P such that n_i and n_{i+1} are recognized in Q as states a and b of M, and $x_i/y_i = x/y$.

Thus, for edge e to be verified in Q, it is sufficient for P to contain a subpath $(n_i, n_j; (xD)/\lambda(a, xD))$ with head$((n_i, n_j; (xD)/\lambda(a, xD)))$ recognized in Q as a.

Definition 5 The subpath $(n_i, n_j; (xD)/\lambda(a, xD))$ of P used to verify e is called the *test segment* for e.

Figure 1 depicts the notions captured by the definitions above.

3 CHECKING SEQUENCE CONSTRUCTION

The problem studied in this paper is defined as follows: Given a strongly connected digraph $G = (V, E)$ representing a minimal FSM M with distinguishing sequence D, find a minimum-length path P of G such that every edge of G is verified in label$(P) = Q$. By definition, for an edge $e = (a, b; x/y)$ of G to be verified in label$(P) = Q$ it is sufficient for the following conditions to be satisfied: 1) P contains a test segment $(n_i, n_j; (xD)/\lambda(a, xD))$ for e; and 2) head$((n_i, n_j; (xD)/\lambda(a, xD)))$ is recognized in Q as state a of M. If condition 1) and 2) hold for every edge of G, then every transition of M is verified in Q. Thus, Q is a checking sequence of M that starts at v_1 (Theorem 1, [13]).

3.1 An Existing Solution

The proposed solution to this problem is an enhancement of the solution given in [13] where first a digraph $G' = (V', E')$ is obtained by augmenting the given digraph $G = (V, E)$, representing an

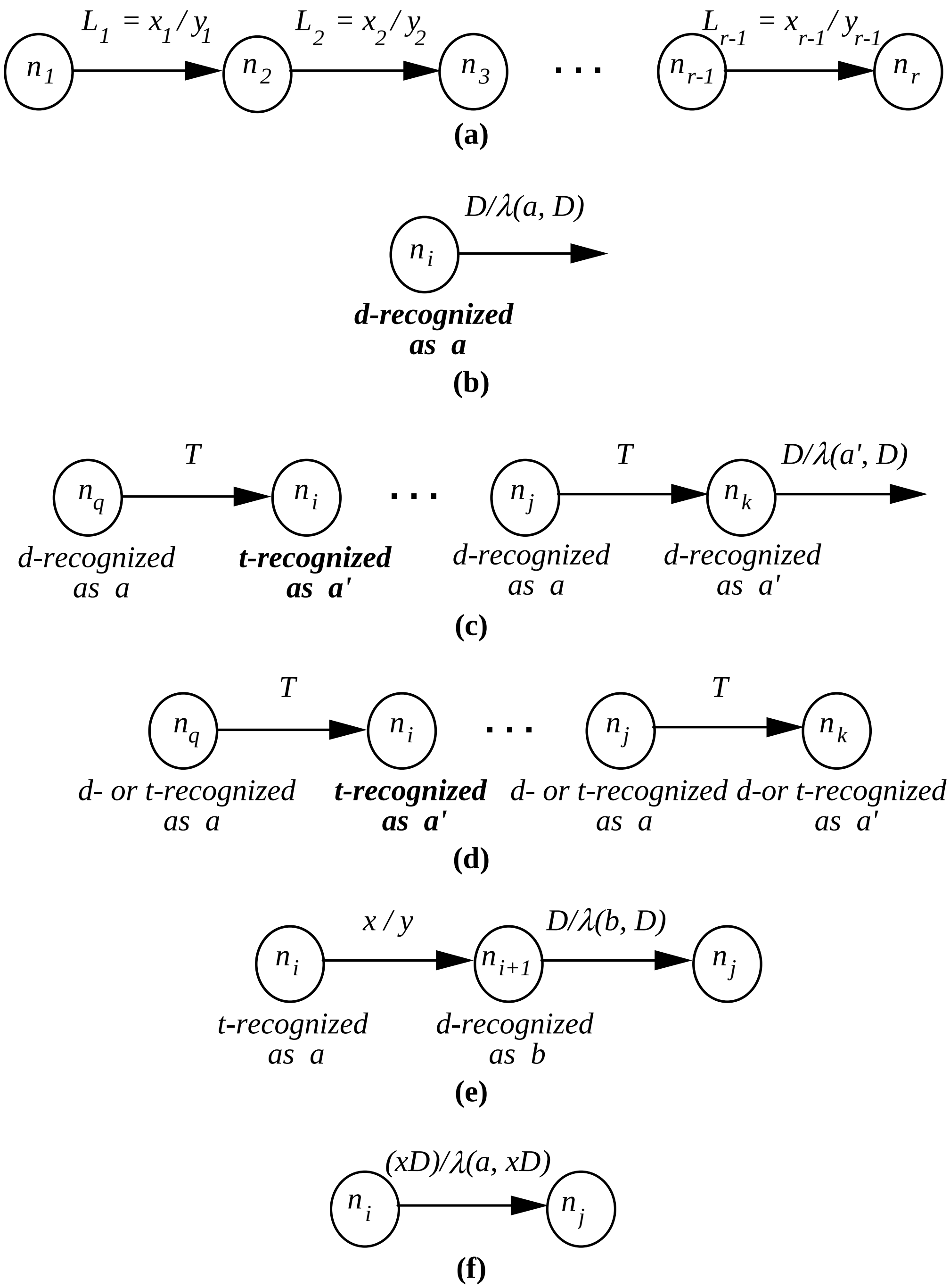

Fig.1 Path $P = (n_1, n_r; Q)$, (b) *d*-recognition, (c) *t*-recognition, (d) *t*-recognition, (e) edge verification, (f) test segment for edge $e = (a, b; x/y)$

FSM M, with a set of edges (E_α) that recognize each state, and a set of edges (E_C) that verifies each transition. A checking sequence is then derived from $G' = (V', E')$ as the label of a path P constructed by combining elements of these two sets of edges in a judicious manner [13]. The enhancements to the solution in [13] will be given after the steps of the solution in [13] are outlined as follows.

Edges in E_α are constructed such that the label of each edge (an α-sequence) recognizes a subset of the states of M and that each state of M is recognized at least once by the labels of the edges in E_α. The construction of E_α is facilitated by forming a set of paths $P_1, ..., P_q$ of G where each path P_k induces an edge of E_α whose label is label(P_k) = α-sequence α_k, $1 \le k \le q$. That is,

a) the set of vertices $V_k \subseteq V$ covered by P_k, $1 \le k \le q$, is $\{ v_1^k, v_2^k, \ldots, v_{m_k}^k \}$;

b) the union of the V_k is V; and

c) the label of P_k, α-sequence α_k, is $D/\lambda(v_1^k, D)T_1^k \;\; D/\lambda(v_2^k, D)T_2^k \;\; \ldots \;\; D/\lambda(v_{m_k}^k, D)T_{m_k}^k$ $D/\lambda(v_w^k, D)T_w^k$ where for $1 \le j \le m_k$, $T_j^k = (I_j^k / O_j^k)$ is a transfer sequence from $\delta(v_j^k, D)$ to v_{j+1}^k, $v_{m_k+1}^k = v_w^k$, and v_w^k is any member of V_k.

So, when P_k, a path whose label is α_k, $1 \le k \le q$, is contained in the solution P of G then

a) v_j^k, $1 \le j \le m_k$, is d-recognized in α_k;

b) $\delta(v_j^k, DI_j^k)$, $1 \le j \le m_k$, is d-recognized in α_k; and

c) tail(P_k) is recognized in α_k.

The labels $\alpha_1, ..., \alpha_q$ of paths $P_1, ..., P_q$ form an *α–set*. From the elements of the α–set, a set of transfer sequences, called *T-set*, is formed as a set of labels of subpaths $R_1, ..., R_p$ of paths $P_1, ..., P_q$, such that each element T_i of T-set is label(R_i) where $\{R_i$: $i = 1, 2, ..., p\}=\{(v_j^k, \delta(v_j^k, DI_j^k)$; $D/\lambda(v_j^k, D)T_j^k$): $1 \le k \le q$ and $1 \le j \le m_k\}$. Thus, head(R_i) is recognized in some α_k because D is applied to head(R_i) and tail(R_i) is recognized in some α_k because tail(R_i) is $\delta(v_j^k, DI_j^k)$ to which D is applied. The set of paths $P_1, ..., P_q$ and the set of subpaths $R_1, ..., R_p$ are included in G' as edges in $E_\alpha \subset E'$ and in $E_T \subset E'$, respectively, in order to facilitate the recognition of vertices in the label Q of the solution P. Moreover, a test segment for each edge of G is included in G' as edges in $E_C \subset E'$ in order to verify every transition of M in label(P) = Q. Furthermore, two more sets of edges are included in G' as edges in $E_\varepsilon \subset E'$ and in $E'' \subset E'$ to increase the connectivity of the vertices in G'.

Formally, $G' = (V', E')$ is obtained from $G = (V, E)$ as follows:

$V' = V \cup U'$ where $U' = \{v'_i$: for every $v_i \in V\}$ and $E' = E \cup E_\alpha \cup E_T \cup E_C \cup E_\varepsilon \cup E''$,

$E_\alpha=\{$(head(P_k), (tail(P_k))$'$; α_k): $1 \le k \le q\}$: for every α_k, (tail(P_k))$'$ is recognized in α_k;

$E_T=\{$(head(R_i), (tail(R_i))$'$; T_i): $1 \le I \le p\}$: for every R_i, (tail(R_i))$'$ is recognized in some α_k;

$E_C=\{(v'_i, (\delta(v_i, xDI_j^k))'; (xDI_j^k)/\lambda(v_i, xDI_j^k)): (v_i, v_j; x/y) \in E\}$: $(\delta(v_i, xDI_j^k))'$ is recognized;

$E_\varepsilon=\{(v'_i, v_i; \varepsilon): v_i \in V\}$;

E'' is a subset of $\{(v'_i, v'_j; x/y): (v_i, v_j; x/y) \in E\}$ such that $G'' = (U', E'')$ has no tour and G' is strongly connected.

Once G' is formed, an RPP P' of G' is found that contains all edges in $E_\alpha \cup E_C$. Since G' is obtained from G, P' represents a path P of G. It is proven in [13] that, for each edge of G, P

satisfies conditions 1) and 2) above, and thus Q = label(P) is a checking sequence of M that starts at v_1. In [13] an RPP P is found through two steps. First, the minimal symmetric augmentation G'' of (V', $E_\alpha \cup E_C$), that may be produced by adding edges from E', is found. If G'', with its isolated vertices removed, is connected, G'' has an Euler tour and this forms P. Otherwise, a heuristic is applied to make G'' connected and an Euler tour is formed. If G'' is connected, P is an RCPP over $E_\alpha \cup E_C$ [13].

3.2 The Proposed Enhancement

Our enhancements to the solution in [13] are based on modifying the definition of G'.

Modification 1

The first modification is on the formation of the elements of the α–set. We observed that if the final section of an α-sequence is not required, unlike in [13], to end in a section within its own body, then the lengths of some α-sequences can be reduced which may reduce the overall length of a checking sequence. We call an α-sequence that does not necessarily end in a section within its own body an α'-sequence. The following is an outline of a procedure that constructs the α'-sequence label(P_k), $1 \leq k \leq q$, called α'_k as opposed to α_k in [13], which can be used to form the P_k of G: Choose subsets $V_k \subseteq V$ ($1 \leq k \leq q$) of V whose union is V and order the elements in each V_k, giving $V_k = \{v_1^k, v_2^k, \ldots, v_{m_k}^k\}$, $1 \leq k \leq q$. Given a V_k, obtain α'_k 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 (possibly empty) transfer sequence from $\delta(v_j^k, D)$ to v_{j+1}^k for $1 \leq j \leq m_k$, $v_{m_k+1}^k = v_w^{k'}$, and $v_w^{k'}$ is contained in any $V_{k'}$, $1 \leq k' \leq q$ and $1 \leq w \leq m_{k'}$. This definition differs from that, for α_k, in [13] in one important way: unlike α_k, the final section of an α'_k need not be contained in this α'_k but could be contained in any $\alpha'_{k'}$. Thus, every α_k is an α'_k but the converse is not true.

Using the definition of α'_k, the set of labels α'_1, ..., α'_q of paths P_1, ..., P_q, called an *α'–set*, can be formed. From the definition of α'_k, it follows that, given an α'_k,

a) v_j^k, $1 \leq j \leq m_k$, is d-recognized in α'_k,

b) $\delta(v_j^k, D I_j^k)$, $1 \leq j \leq m_k$, is d-recognized in α'_k, and

c) tail(P_k) is recognized in some $\alpha'_{k'}$, $1 \leq k' \leq q$.

Example 1

Consider the α–set and α'-set for FSM M_0, in Fig. 2, where D = *aba* and empty transfer sequences are used in forming every α_k and α'_k. The α-set for M_0 is $\{\alpha_1, \alpha_2\}$ where α_1, the label of $P_1 = (s_5, s_4; \alpha_1)$, is $D/\lambda(s_5, D)$ $D/\lambda(s_2, D)$ $D/\lambda(s_4, D)$ $D/\lambda(s_1, D)$ $D/\lambda(s_2, D)$ and α_2, the label of $P_2 = (s_3, s_2; \alpha_2)$, is $D/\lambda(s_3, D)$ $D/\lambda(s_1, D)$ $D/\lambda(s_2, D)$ $D/\lambda(s_4, D)$ $D/\lambda(s_1, D)$. The α'-set for M_0 is

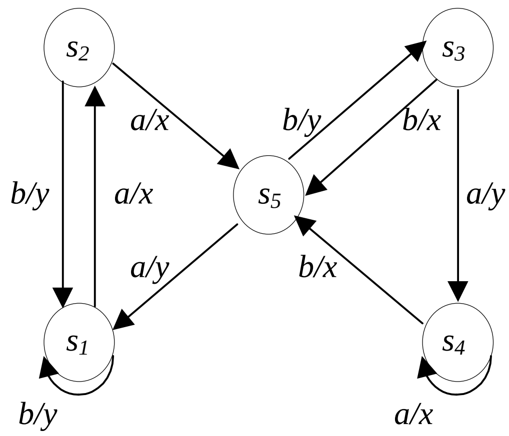

Fig. 2: FSM M_0 represented by $G = (V, E)$

$\{\alpha'_1, \alpha'_2\}$ where α'_1, the label of $P'_1 = (s_5, s_4; \alpha'_1)$, is $D/\lambda(s_5, D)$ $D/\lambda(s_2, D)$ $D/\lambda(s_4, D)$ $D/\lambda(s_1, D)$ $D/\lambda(s_2, D)$ and α'_2, the label of $P'_2 = (s_3, s_2; \alpha'_2)$, is $D/\lambda(s_3, D)$ $D/\lambda(s_1, D)$. It is observed that the final section of α'_2 (the application of D at s_1) is contained in α'_1 but not α'_2. Thus α'_2 is *not* an α-sequence and the α'-set contains 7 instances of D while the α-set contains 10 instances of D. As we shall show later, the difference between α'_k and α_k may have a significant impact on the length of a checking sequence. □

Modification 2

The second modification is in the formation of the elements of the subset E_C of E' and stems from the following two observations. Since label(P_k), $1 \leq k \leq q$, starts with the application of D, the head of P_k is recognized and since label(R_i) = T_i, $1 \leq i \leq p$, starts with the application of D, the head of R_i is recognized. Thus, an α'_k or T_i can be used to verify the end state of a transition in forming a test segment for that transition. These properties of α_k or T_i were not utilized in [13] and their use will also contribute to the reduction in the length of the checking sequence.

These two modifications give rise to the following changes in the definition of $G' = (V', E')$:

(1) replace all occurrences of α_k by α'_k

(2) replace E_C in [13] by $E_C = \{(v'_i, v_j; x/y): (v_i, v_j; x/y) \in E\}$

(3) eliminate E and E_ε

(1) ensures that α'_k is used rather than α_k; (2) stands for the test segments for all edges of G since each edge in E_C terminates at a vertex in V and is to be followed by an edge leaving a vertex in V whose label is either an α'_k or T_i; and (3) eliminates a precautionary measure in the previous definition of $G' = (V', E')$ in [13] to provide connectivity that is now guaranteed without these edge sets. Since these changes do not alter the semantics of the definition of $G' = (V', E')$, a path P' of G' that contains all edges in $E_\alpha \cup E_C$ is an RPP of G' over $E_\alpha \cup E_C$. It is proven in [13] that this path is in fact a path P of G and for each edge of G, P of G satisfies the conditions 1) and 2).

Thus, it follows that the label Q of P is a checking sequence of M that starts at v_1.

Example 2

Consider now the problem of generating a checking sequence for FSM M_0 in Fig. 2 using the algorithm from [13], the α-set $\{\alpha_1, \alpha_2\}$ given earlier, and the test segments in Table 1.

Table 1 Edges of E_C

$(xD)/\lambda(v_i, xD) = L_{ijk}$	$(v'_i, v'_k; L_{ijk})$
$(aD)/(xxyy) = L_{124}$	$(v'_1, v'_4; L_{124})$
$(bD)/(yxyx) = L_{112}$	$(v'_1, v'_2; L_{112})$
$(aD)/(xyyx) = L_{252}$	$(v'_2, v'_2; L_{252})$
$(bD)/(yxyx) = L_{212}$	$(v'_2, v'_2; L_{212})$
$(aD)/(yxxy) = L_{341}$	$(v'_3, v'_1; L_{341})$
$(bD)/(xyyx) = L_{352}$	$(v'_3, v'_2; L_{352})$
$(aD)/(xxxy) = L_{441}$	$(v'_4, v'_4; L_{441})$
$(bD)/(xyyx) = L_{452}$	$(v'_4, v'_2; L_{452})$
$(aD)/(yxyx) = L_{512}$	$(v'_5, v'_2; L_{512})$
$(bD)/(yyxy) = L_{531}$	$(v'_5, v'_1; L_{531})$

In Table 1 a label of the form L_{ijr} represents a test segment, that ends at s_r, for a transition from s_i to s_j. This leads to the digraph shown in Fig. 3, in which the edges from E and E_ε (which are used for connectivity) are not shown and dashed lines are used for the edges that are not in $E_\alpha \cup E_C$.

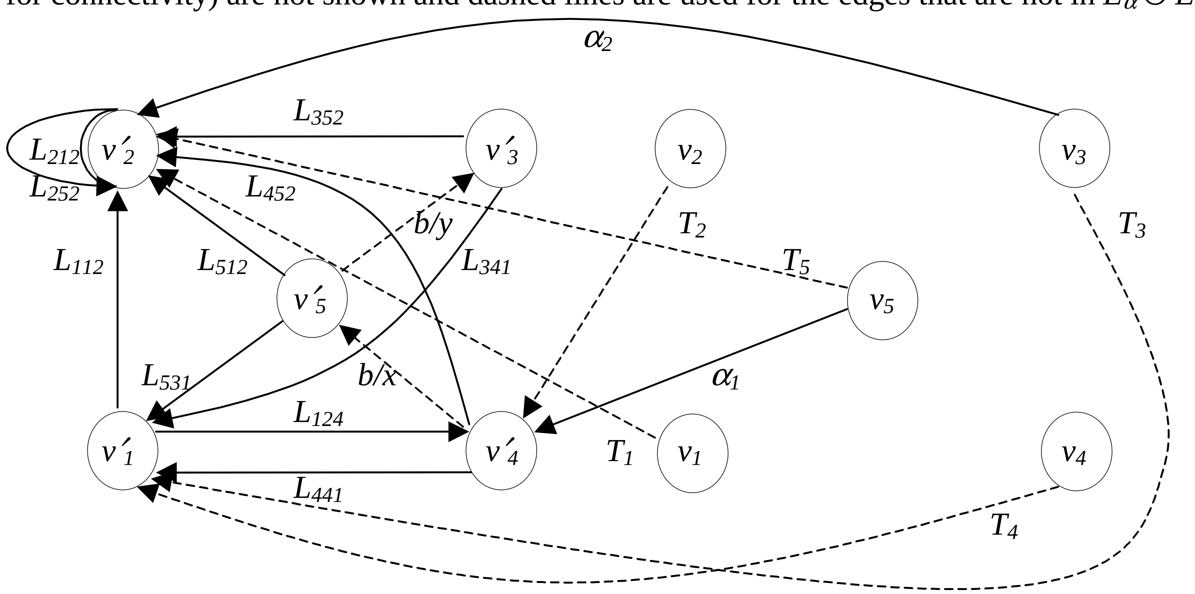

Fig. 3: $G' = (V', E')$ with α-sequences

Here E_α is $\{(v_5, v'_4; \alpha_1), (v_3, v'_2; \alpha_2)\}$ and E_T is $\{(v_1, v'_2; T_1), (v_2, v'_4; T_2), (v_3, v'_1; T_3), (v_4, v'_1; T_4), (v_5, v'_2; T_5)\}$. The minimal symmetric augmentation, of the edge set $E_\alpha \cup E_C$ of G', is now produced: this is the smallest symmetric digraph G'' that can be formed from $E_\alpha \cup E_C$ by adding edges from G'. Digraph G'' is shown in Fig. 4. Since G'', with its isolated vertices removed, is connected an Euler tour P of G'' exists and the label of P forms a checking sequence [13].

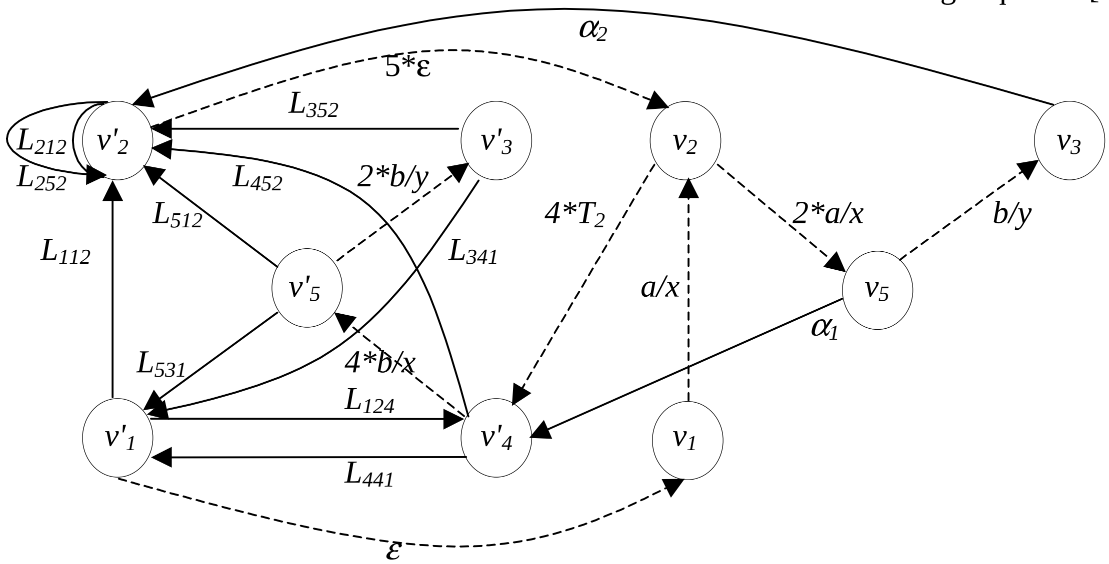

Fig. 4: $G'' = (V'', E'')$ with α-sequences

This leads to the checking sequence, of length 92, represented by the following:
L_{112}, L_{212}, L_{252}, T_2, L_{441}, L_{124}, b/x, L_{531}, a/x, a/x, α_1, b/x, L_{512}, a/x, b/y, α_2, T_2, L_{452}, T_2, b/x, b/y, L_{352}, T_2, b/x, b/y, L_{341}

Consider now the use of the α'-set $\{\alpha'_1, \alpha'_2\}$ and the modification proposed in this paper. The digraph G' is shown in Fig. 5, in which all the edges except the edges in $E_\alpha \cup E_C$ are represented by dashed-lines. Here the set E_α is $\{(v_5, v'_4; \alpha'_1), (v_3, v'_2; \alpha'_2)\}$ and E_T is as above.

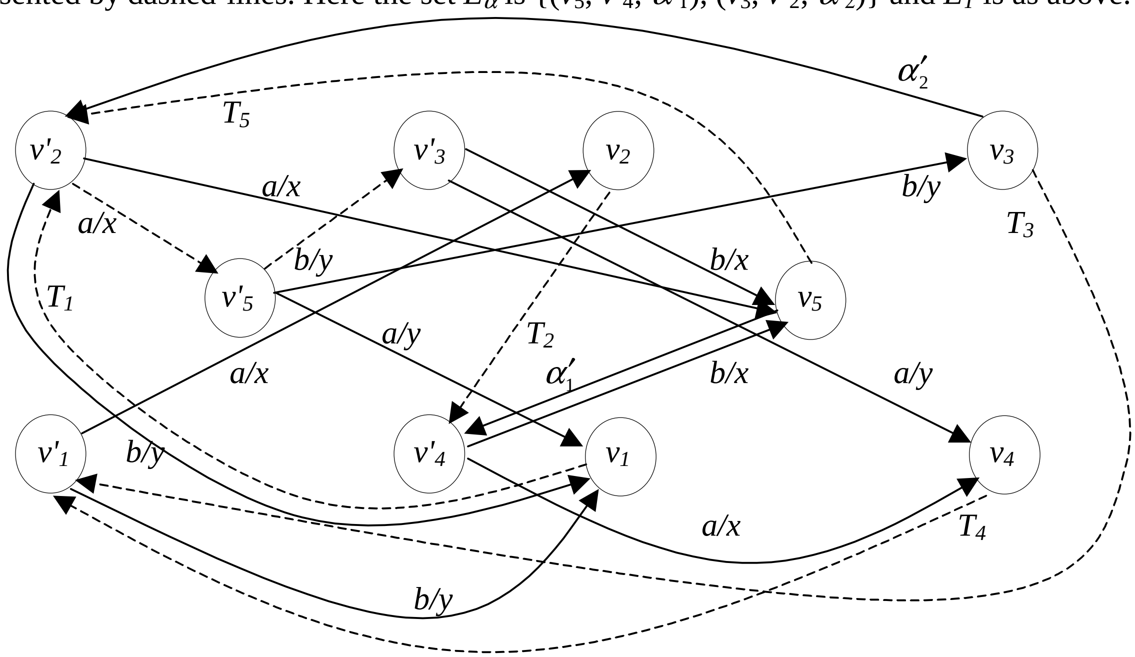

Fig. 5: $G' = (V', E')$ with α'-sequences

Note that, as mentioned earlier, each edge from V represents an α'_k or T_i and thus a sequence that recognizes its initial node. It follows that the inclusion, in a tour, of an edge e from E_C leads to the inclusion of a test segment for e. Thus a tour that includes every edge in E_C must include a test segment for every transition. The minimal symmetric augmentation of the edge set $E_\alpha \cup E_C$, formed by adding edges from G', is G'' which is shown in Fig. 6. G'' is connected and thus has an Euler tour which leads to the following checking sequence, of length 61, and thus to a reduction of one third in the checking sequence length:

b/y, $D/\lambda(s_1,D)$, b/y, $D/\lambda(s_1,D)$, a/x, α'_1, a/x, $D/\lambda(s_4,D)$, a/x, $D/\lambda(s_2,D)$, b/x, $D/\lambda(s_5,D)$, a/x, b/y, α'_2, a/x, a/y, $D/\lambda(s_1,D)$, a/x, b/y, b/x, $D/\lambda(s_5,D)$, a/x, b/y, a/y, $D/\lambda(s_4,D)$ □

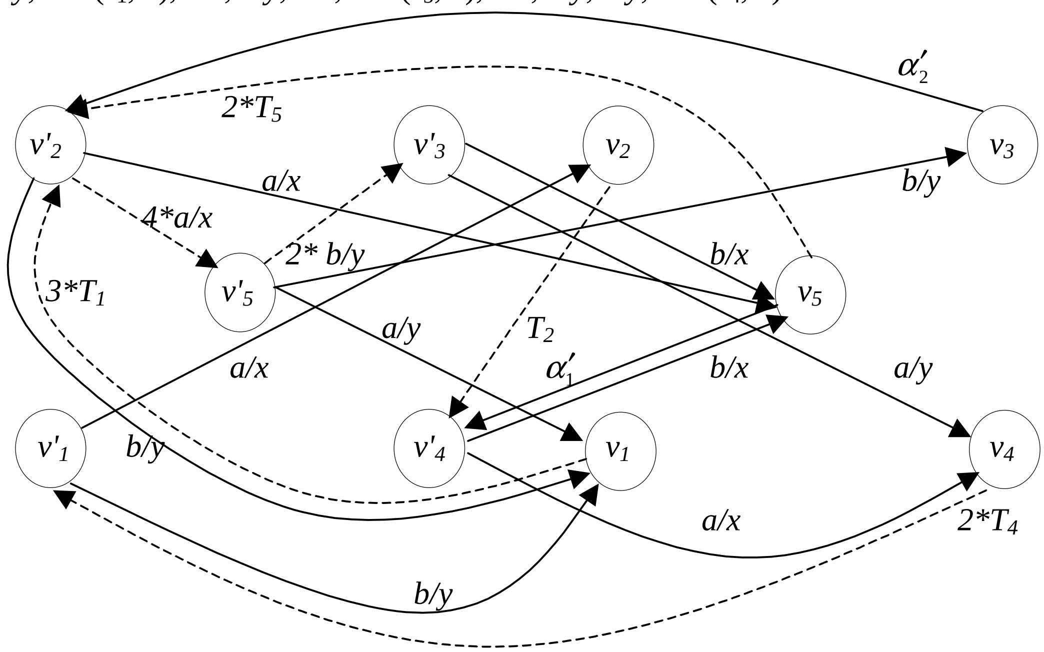

Fig. 6: $G'' = (V'', E'')$ with α'-sequences

3.3 Comparison Between Two Approaches

First we note that the method proposed in this paper and that given in [13] involve solving the RCPP for digraphs of the same order. They thus have the same algorithmic complexity. Then, we compare the relative lengths of the sequences that are constructed by the two methods. For this, we will first consider an infinite class of FSMs and focus on our claim that: α'-sets yield shorter sequences than α-sets. It will transpire that the elements of this class have α'-sets that are significantly smaller than the corresponding α-sets. This shows that the proposed improvements are significant for a range of FSMs. After this analytical comparison, we will give the results of an empirical study performed on the lengths of sequences that corroborates the analytical comparison.

The α-set and α'-set produced for an FSM M are defined by the digraph $G^T=(V, E^T)$ in which $E^T = \{head(R_i), tail(R_i); T_i\}$: each α-sequence and α'-sequence is formed from a path in G^T. Given G^T, derived from an FSM with n states, there is always an α'-set formed from no more than $2n$ edges of G^T. We will now consider a class of such digraphs, with $n=2m$, for which any α–set is significantly larger than this. Given m, consider $G_m^T = (V_m, E_m^T)$ where $V_m=\{v_1, \ldots, v_{2m}\}$; for all i, $1 \le i \le m$, there is an edge in E_m^T from v_i to $v_{i+1 \bmod m}$; and for all i, $m < i \le 2m$, there is an edge in E_m^T from v_i to an element of $\{v_1, \ldots, v_m\}$. This is illustrated in Fig. 7. It is straightforward to show that each G_m^T may arise from a real FSM.

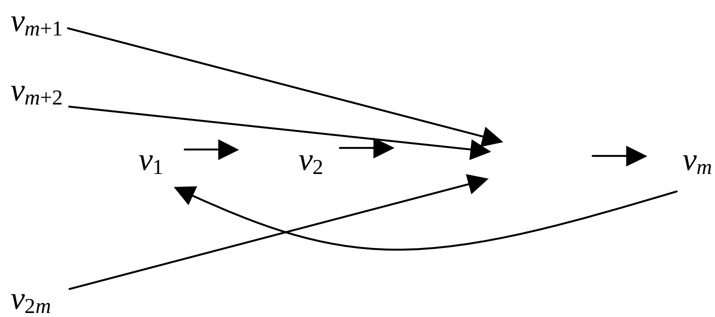

Fig. 7: The digraph G_m^T

Consider a minimal α-set that may be produced from G_m^T. Each $v_j \in \{v_{m+1}, \ldots, v_{2m}\}$ leads to the inclusion of the α-sequence: an initial edge to some v_i, the cycle back to v_i, and one further edge. Thus the α-set contains m sequences, each comprising of $m + 2$ edges from G_m^T, and so $O(m^2) = O(n^2)$ edges from G_m^T. As noted above, there are α'-sets formed from $O(n)$ edges of G_m^T. Here such an α'-set may be formed in the following way. Create one α'-sequence α'_1 in the form of an edge from v_{m+1} to some v_i, then the cycle followed by one further edge. For each vertex $v_j \in \{v_{m+2}, \ldots, v_{2m}\}$ there is a further α'-sequence generated from the path of length 2 from v_j, since the second edge is contained in α'_1. Thus, if each edge from G_m^T has cost at most c ($c \ge |D|$) then an α-set generated from G_m^T must have size of $O(cn^2)$ while there is an α'-set with size $O(cn)$. Further, the costs of the test segments is of $O(c|X|n)$ and thus this difference, in the sizes of the α-set and α'-set, is significant and grows more significant as the number of states increases. This class of examples also shows that in general α-sets have size $O(cn^2)$ while α'-sets have size $O(cn)$. Moreover, since every α-set is an α'-set, any checking sequence allowed by the method of [13] is allowed by the method proposed in this paper, that is reduction in the lengths of checking sequences achieved by the method of [13] occurs over a larger set of checking sequences when our modifications are applied.

In order to further investigate the differences in sizes of α'-sets and α-sets, 10 digraphs representing G^T were randomly generated for each FSM in the set of: FSMs with 10 states; FSMs with 20 states; FSMs with 30 states; and FSMs with 50 states. In each case the number of edges used from G^T was recorded. The results, which are summarized in Table 2, suggest that α'-sets are significantly smaller than α-sets and that this difference increases as the number of states increases. This observation is consistent with the analytical comparison given above.

Table 2 Mean sizes of randomly generated sets

n	Mean size of α-set	Mean size of α'-set	Saving
10	20.7	14.4	30%
20	54.8	28.4	48%
30	98.3	41.7	58%
50	214.5	69.4	68%

4 CONCLUSIONS

This paper has introduced a method, for generating checking sequences, that enhances that given in [13] in two ways. Firstly, the notion of α-sequences has been generalized to α'-sequences. Essentially, an α-sequence α_k must end in a section from within its own body while an α'-sequence α'_k can end in a section from within the body of some other α'-sequence $\alpha'_{k'}$. Thus, while every α-sequence is an α'-sequence, the converse is not the case. The use of α'-sequences, as opposed to α-sequences, allows two main advantages: α'-sequences may be shorter than α-sequences; and using α'-sequences increases the set of checking sequences over which optimization occurs.

The second improvement upon [13] is based upon the observation that an α'-sequence may be used to check the final state of a transition. This property is utilized, in the generation of checking sequences, to allow overlap between the α'-sequences and the test segments. This further contributes to a reduction in the length of the checking sequence.

The method given in this paper might be further enhanced in two ways. Firstly, the connecting transitions might be chosen from the set of transitions of the given FSM M during optimization, rather than being drawn from a cycle-free subset (E'') found prior to optimization. This may be achieved by including a copy of each transition and relying upon properties of the optimization algorithm, that starts with the production of a minimal symmetric augmentation, that guarantee that the set chosen is cycle free. Secondly, prefixes of the distinguishing sequence may be used to recognize states.

ACKNOWLEDGEMENTS

This work is supported in part by the Natural Sciences and Engineering Research Council of Canada under grant OGP0000976. The authors wish to thank the anonymous referees for their comments and suggestions.

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 Communications*, vol. 39, pp.1604-1615, 1991.

[2] F. Belina and D. Hogrefe, "The CCITT-Specification and Description Language SDL", *Computer Networks and ISDN Systems* , vol. 16, pp. 311-341, 1989.

[3] S. Budkowski and P. Dembinski, “An introduction to ESTELLE: A specification language for distributed systems”, *Computer Networks and ISDN Systems*, vol. 14, pp. 3-23, 1987.

[4] A.T. Dahbura, K.K. Sabnani, and M.U. Uyar, “Formal methods for generating protocol conformance test sequences”, *Proceedings of the IEEE*, vol. 78, pp. 1317-1325, 1990.

[5] A. Gill, *Intro. to the Theory of Finite-State Machines*, New York: McGraw-Hill, 1962.

[6] G. Gonenc, “A method for the design of fault detection experiments”, *IEEE Trans. on Computers*, vol. 19, pp. 551-558, 1970.

[7] D. Harel, “Statecharts: A visual formalism for complex systems", *Science of Computer Programming*, vol. 8, pp.231-274, 1987.

[8] F.C. Hennie, "Fault detecting experiments for sequential circuits", *Proc. Fifth Ann. Symp. Switching Circuit Theory and Logical Design*, pp. 95-110, Princeton, N.J., 1964.

[9] D. Lee and M. Yannakakis, "Testing finite state machines: state identification and verification", *IEEE Trans. on Computers*, vol. 43, pp. 306-320, 1994.

[10] D. Lee and M. Yannakakis, "Principles and methods of testing FSMs: a survey", *Proceedings of the IEEE*, vol. 84, pp. 1089-1123, 1996.

[11] I. Pomeranz and S. M. Reddy, "Test generation for multiple state-table faults in finite-state machines", *IEEE Trans. on Computers*, vol. 46, pp. 783-794, 1997.

[12] D.P. Sidhu and T.K. Leung, “Formal methods for protocol testing: a detailed study”, *IEEE Trans. on Software Engineering*, vol. 15, pp. 413-426, 1989.

[13] H. Ural, X. Wu, and F. Zhang, "On minimizing the lengths of checking sequences", *IEEE Trans. on Computers*, vol. 46, pp. 93-99, 1997.

[14] M. Yannakakis and D. Lee, "Testing finite state machines: fault detection", *Journal of Computer and System Sciences*, vol. 50, pp. 209-227, 1995.