UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

Comments on "On minimizing the lengths of checking sequences"

Adenilso da Silva Simão

№ 307

RELATÓRIOS TÉCNICOS

São Carlos - SP
Agosto/2007

Comments on "On minimizing the lengths of checking sequences"

Adenilso Simão

[1] Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo

São Carlos, São Paulo, Brazil

adenilso@icmc.usp.br

Abstract. In [1], Ural et al. propose a sufficient condition for a sequence to be a checking sequence for a complete deterministic FSM. The condition is then used in [1] to elaborate a method for generating checking sequences and it is also used in subsequent improved versions of this method published in [2] and [3]. In this correspondence, we present a counter-example which demonstrates that the condition of the theorem is not sufficient, as it does not address all initialization faults, and show what can be added to make it sufficient.

We use definitions from [1]. Let $M = (S, X, Y, \delta, \lambda)$ be a complete deterministic reduced Mealy machine for which there exists a distinguishing sequence D. s_1 is the initial state of M. M is represented as a graph $G = (V, E)$, whose vertices are the states of M (v_1 represents s_1) and whose edges are its transitions. A *path* $P = (n_1, n_2; x_1/y_1)(n_2, n_3; x_2/y_2)\ldots(n_{r-1}, n_r; x_{r-1}/y_{r-1})$ is a finite sequence of adjacent edges in E. The label of P, denoted *label*(P), is the IO-sequence $(x_1/y_1)(x_2/y_2)\ldots(x_{r-1}/y_{r-1})$. P can also be represented by $(n_1, n_r; x_1x_2\ldots x_{r-1}/y_1y_2\ldots y_{r-1})$, and, accordingly, its label can be represented by $x_1x_2\ldots x_{r-1}/y_1y_2\ldots y_{r-1}$.

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. Let Q = *label*(P). A *checking* sequence of M is an IO-sequence starting at a specific (initial) state of M that distinguishes M from any FSM M^* that has at most as many states as M, has the same input and output sets and is not equivalent to M (in the initial state). To formulate a condition for an input sequence to be checking, auxiliary notions of recognition of nodes of P in Q and verification of edges of G in Q are used.

(i) n_i is *d-recognized* in Q as some state a of M if $(n_i, n_k; D/\lambda(a, D))$ is a subpath of P;

(ii) if $(n_q, n_i; T)$ and $(n_j, n_k; T)$ are subpaths of P and $D/\lambda(a, D)$ is a prefix of T (and, thus, n_q and n_j are d-recognized in Q as state a of M) and n_k is d-recognized in Q as some state a' of M, then n_i is *t-recognized* in Q as state a' of M;

(iii) if $(n_q, n_i; T)$ and $(n_j, n_k; T)$ are subpaths of P such that n_q, and n_j are either d-recognized or t-recognized 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 n_i is also *t-recognized* in Q as state a' of M;

(iv) if n_i of P is d-recognized or t-recognized in Q (as some state a of M), then n_i is *recognized* in Q (as state a of M).

(v) 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, respectively, and $x_i/y_i = x/y$.

Ural et al.'s theorem then states that a sufficient condition for Q to be a checking sequence of M that starts at v_1 is that every edge of G is verified in Q.

Theorem 1 [1]. Let Q be an IO-sequence of M starting at v_1 (i.e., the label of a path $P = (n_1, n_r; Q)$ of G) and let D be a distinguishing sequence of M. If every edge $(v_i, v_j; x/y)$ of G is verified in Q, then Q is a checking sequence of M.

This condition is not sufficient, though.

Theorem 2. Given M and an IO-sequence Q that satisfies the condition of Theorem 1, let $(n_1, n_k; I/O)$ be the longest prefix of Q, such that no n_i is recognized in Q, for $1 \leq i \leq k - 1$, and let n_k be recognized in Q as state s. If there exists a state s' of M, $s' \neq s_1$, such that $\lambda(s', I) = \lambda(s_1, I) = O$ and $\delta(s', I) = \delta(s_1, I) = s$, then Q is not a checking sequence.

Proof. Consider the FSM M in Fig.1, taken from [2]. Consider also two sequences derived in that paper according to Theorem 1. Both sequences start with b/y, followed by the distinguishing sequence D. b/y causes in M a looping transition at state s_1 as well as a transition from state s_2 to s_1. Thus, an FSM M^* which produces the same output sequence in response to the input sequence of Q is the FSM M which starts not in s_1, but in state s_2. States s_1 and s_2 are not equivalent, thus M^* is not equivalent to M. QED

Actually, the problem with the condition of the theorem is that it does not require the starting node of Q be recognized as the initial state s_1 of M. This additional requirement fixes the problem.

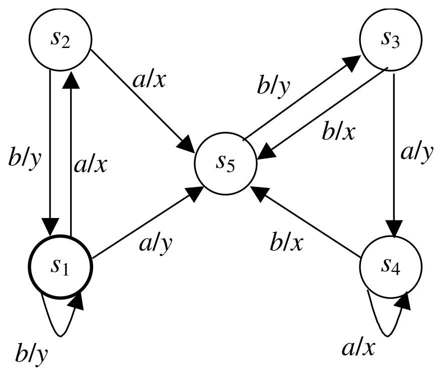

Figure 1. Fig. 2 of [2].

Theorem 3. Let Q be an IO-sequence of M starting at v_1 (i.e., the label of a path $P = (n_1, n_r; Q)$ of G) and let D be a distinguishing sequence of M. If every edge $(v_i, v_j; x/y)$ of G is verified in Q **and n_1 is d-recognized as s_1 in Q**, then Q is a checking sequence of M.

Notice that now the longest prefix of Q which has no recognized nodes, as required by Theorem 2, becomes empty. This implies that no state except s_1 could be the initial state of M. Therefore, any initialization fault can be exposed.

Based on Theorem 1, Ural et al. elaborate a method for generating minimized checking sequences for complete strongly-connect deterministic FSMs. They reduce the problem of finding a checking sequence into a Rural Chinese Postman problem by deriving a suitable auxiliary graph G' whose paths can be induced in G. These paths are not required to start with the distinguishing sequence, though. As a result, the obtained sequence may not be a checking sequence. The method (and subsequent improvements in [2] and [3]) can be fixed by constraining these paths to start with $D/\lambda(s_1, D)$.

References

[1] H. Ural, X. Wu, and F. Zhang. On minimizing the lengths of checking sequences. IEEE Transactions on Computers, 46(1):93-99, 1997.

[2] R. M. Hierons and H. Ural. Reduced length checking sequences. IEEE Transactions on Computers, 51(9):1111-1117, 2002.

[3] R. M. Hierons and H. Ural. Optimizing the length of checking sequences. IEEE Transactions on Computers, 55(5):618-629, 2006.