Temporal Logics for the Specification of Performance and Reliability

ERRATA CORRIGE

Luca de Alfaro

June 24, 1997

Department of Computer Science

Stanford University

Given a set C of states of a TPNS, Algorithm 8 of [1] correctly computes the maximal stable set $B \subseteq C$. However, the computation of the set L of maximal SCSS in a given set C of states is incorrect: it is necessary to iterate the step of computing the maximal strongly connected components of the graph (B, ρ_B). A revised algorithm for this latter problem is presented below. No other results of the paper are affected by this error.

Algorithm 1. (computation of maximal SCSS)

Input: A set $C \subseteq S$ of states.

Output: The set L of maximal SCSSs in C.

Define: Given a set $D \subseteq S$ of states, let $SCC(D)$ be the set of strongly connected components of the graph (D, ρ_D) (thus, $SCC(D)$ is a set of subsets of D).

Initialization: $L := \{C\}$.

Repeat: $L := \bigcup_{B \in L} SCC(B)$;

Until: L is not changed by the above iteration. □

The correctness of the revised algorithm can be stated and proved as follows.

Theorem 2 (correctness of Algorithm 1). *Algorithm 1 correctly computes the set of maximal SCSSs of a given set of states.*

Proof. Let $C \subseteq S$ be the set given as input to the algorithm.

First, note that if the algorithm stops with output L, then every $B \in L$ is an SCSS subset of C. To see this, note that when the algorithm terminates we have $SCC(B) = \{B\}$ for every $B \in L$, since L is not changed by the iteration. Thus, (B, ρ_B) is strongly connected, which means that B is an SCSS.

Next, consider an arbitrary SCSS $D \subseteq C$. By induction on the number of iterations of the algorithm, we prove that the following invariant holds:

For all $B \in L$, it is either (i) $D \subseteq B$ or (ii) $D \cap B = \emptyset$; furthermore, (i) holds for at least one $B \in L$.

The invariant trivially holds before the first iteration, when $L = \{C\}$. For the induction step, assume that the invariant holds before an iteration, and consider how each $B \in L$ is modified by the iteration. There are two cases.

1. If $D \subseteq B$, then (D, ρ_D) is a subgraph of (B, ρ_B), and since (D, ρ_D) is strongly connected there will be exactly one $B' \in SCC(B)$ such that $D \subseteq B'$. The conclusion follows form the observation that the sets in $SCC(B)$ are mutually disjoint.
2. If $D \cap B = \emptyset$, then for all $B' \in SCC(B)$ it is $D \cap B' = \emptyset$.

Finally, to prove that upon termination L is the set of maximal SCSSs in C, we reason as follows.

1. Let D be a maximal SCSS of C. By the above argument there must be $B \in L$ such that $D \subseteq B$, and B must be an SCSS in C, as shown earlier. By the maximality of D in C, we conclude $D = B$, so that $D \in L$.
2. Conversely, consider upon termination of the algorithm a set $B \in L$; again, B is an SCSS. To see that B is maximal in C, assume towards the contradiction that there is another SCSS $D \subseteq C$ with $B \subset D$. Then, the SCSS D would not satisfy the inductive invariant proved earlier, since it is neither $D \subseteq B$ nor $D \cap B = \emptyset$. □

References

1. L. de Alfaro. Temporal logics for the specification of performance and reliability. In *Proc. of Symp. on Theor. Asp. of Comp. Sci.*, volume 1200 of *Lect. Notes in Comp. Sci.*, pages 165–176. Springer-Verlag, February 1997.