CSE647: Testing and Verification (Fall 2000)

Scott Stoller

Homework 5 (ver 6 Oct, 13:30). Due 13 Oct.

For every problem, you must justify your answer.

Background: We discussed the following version of Overman's Alg in lecture on 5 Oct. It computes a set of transitions that is persistent in state s.

1. Select one transition $t \in \mathit{enabled}(s)$. Let $T = \{\mathit{thread}(t)\}$, where $\mathit{thread}(t)$ is the thread that can perform transition t.

2. For each thread $\theta \in T$, for each transition $t \in \mathit{pending}(s,\theta)$, for each thread $\theta' \in \Theta \setminus T$, if $(\exists op \in \mathit{used}(t), op' \in \mathit{used}(t') : op$ can-be-dependent-with $op')$, then insert θ' in T.

3. Repeat step 2 until no more processes can be added. Return $\{\mathit{enabled}(s,\theta) \mid \theta \in T\}$.

op can-be-dependent-with op' holds if op and op' are operations on the same object o and $\langle op, op'\rangle \in D_o$, where D_o is a valid unconditional dependency relation for o.

An improved version of Overman's algorithm is obtained by replacing "op can-be-dependent-with op'" with "$op \rhd_s op'$", defined below. Let $\mathit{used}(t,o)$ be the set of operations on object o used by transition t.

$op \rhd_s op'$ ("op' might be the first operation to interfere with op from s") holds if op and op' are operations on the same object o and there exists a sequence $s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} s_3 \cdots \xrightarrow{t_{n-1}} s_n \xrightarrow{t_n} s_{n+1}$ such that $s_1 = s$ and $(\forall i \in [1..n-1] : \forall op_i \in \mathit{used}(t_i,o) : \langle op, op_i, s_i\rangle \notin D^c_o)$ and $op' \in \mathit{used}(t_n,o)$ and $\langle op, op', s_n\rangle \in D^c_o$, where D^c_o is a valid conditional dependency relation for o.

Problem 1. Consider a concurrent system with exactly two threads, θ_0 and θ_1, that synchronize using only a semaphore sem. The value of sem in state s_0 is 1. Also, $\mathit{enabled}(s_0,\theta_0) = \{t_0\}$ and $\mathit{enabled}(s_0,\theta_1) = \{t_1\}$. t_0 has guard true and command "sem.Up()". t_1 has guard true and command sem.Dn(). Make no other assumptions about the future behavior of the threads.

a. Is $\{t_0\}$ persistent in s_0?

b. Can the first version of Overman's Algorithm return $\{t_0\}$? Use the best relation D_o that you can justify.

c. Can the second version of Overman's Algorithm return $\{t_1\}$? Use the best relations D^c_o and $\rhd_s$ that you can justify.

Problem 2. Same as problem 1, except for $\{t_1\}$.

Problem 3. Does the "running time" of state-less selective search depend on the (unspecified) order in which transitions are selected from T? For simplicity, assume the "running time" is directly proportional to the number N_t of executed transitions (a transition that is executed multiple times is counted multiple times in N_t). Assume the state space is finite and acyclic.