Concurrency Theory (WS 2010/11) Out: Thu, Dec 22 Due: Mon, Jan 10

Exercise Sheet 9

Jun.-Prof. Roland Meyer, Georgel Călin Technische Universität Kaiserslautern

Problem 1: Acceleration of Control Loops

Let $\preceq^*_{swap}$ and $\preceq^*_m$ the word orderings for SREs given in class for proving the lemma:

Let p be a product and ops a sequence of operations. There is a natural number n (linear in the size of p) and a product p' such that either $\mathcal{L}(p \oplus ops^n) = \emptyset$ or $\mathcal{L}(p') = \bigcup_{j \geq n} \mathcal{L}(p \oplus ops^j)$.

Find n and p' when $p = (a+b)^*(c+\epsilon)b^*$ and ops is each of:

$$?a\,!b\,?c$$
$$!a\,!b\,?c\,?a$$
$$!a\,?c\,!b\,?a\,!c$$
$$?c\,!c\,!a\,?a\,!b\,!c\,!a$$

Don't forget to specify to which of the four cases discussed in class each sequence belongs and argue the correctness of your findings.

Problem 2: SRE Inclusion

Use the algorithm given in the lecture to check whether the following SRE inclusions hold:

(a) $(a+n+s)^*(t+a+n)^* \subseteq (s+a+n+t+a)^*$
(b) $(r+\epsilon)(p+\epsilon)(n+t)^* \subseteq p^*(r+\epsilon)(s+\epsilon)(n+t)^* + (p+\epsilon)r^*(n+e+t)^*$
(b) $(r+\epsilon)(p+\epsilon)(n+t)^* \subseteq (p+r+e)^*(s+\epsilon)(n+t)^*$

Problem 3: Symbolic Verification Algorithm

Using the introduced theory on SREs, checking whether a configuration (q, w) is reachable can be performed by computing the possible symbolic configurations of the system.

A symbolic configuration (q, r) of an lcs consists of the control state q and SRE r describing the possible contents of the channel(s) for control state q.

The set of symbolic configuration initially consists only of the start configuration and is enlarged by computing the effects of single transitions and control loops.

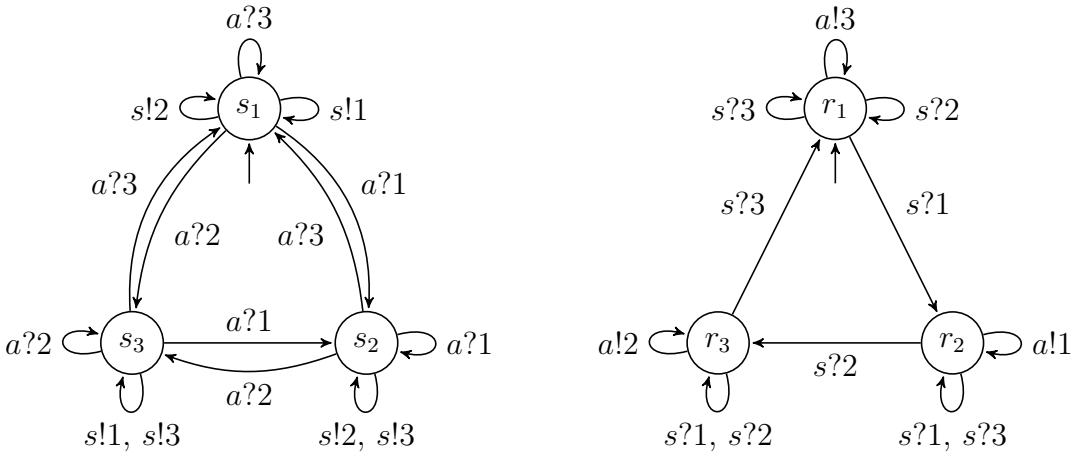
The algorithm stops in either of the cases:

- $w \in \mathcal{L}(r)$ for some already generated symbolic configuration (q, r)
- for every possible new (q, r) there is an already computed (q, r') such that $\mathcal{L}(r) \subseteq \mathcal{L}(r')$.

Give a (pseudo)code *implementation* of the described algorithm.

Problem 4: Sliding Windows Protocol

Consider the simplified sliding window protocol depicted by the lcs in the figure below.



Use the previous algorithm to determine all reachable configurations of the protocol.

[0]We wish you a *Merry Christmas!* and a *Happy New Year!* ... See you again in 2☺11.