| | |
|---|---|
| **Concurrency Theory** (WS 2016) | Out: Mon, 19 Dec    Due: Wed, 04 Jan |

# Exercise Sheet 8

D'Osualdo, Lederer, Schneider    Technische Universität Kaiserslautern

Please submit your solutions before Wed, 04 Jan at 10am. E-mail submissions are encouraged.

## Prelude: Ring Systems

We want to model a ring of processes, in which the processes can communicate with their left/right neighbours. Each process has a control state and budget to spend, i.e. a natural number which must remain non-negative at all times. Coins from the budget can be transferred to neighbours or received from neighbours. New processes can be created dynamically, and any process in the ring can fail at any point in time. We call this model of computation a *ring system*.

More formally, a ring system is specified by an automaton $(Q, \Delta)$ with finite control states $Q$ and transition relation $\Delta \subseteq Q \times Act \times Q$. Let the set of directions be $Dir = \{\mathsf{left}, \mathsf{right}\}$, then the set $Act$ of actions is defined as below:

$$\begin{aligned} Act := & \{d \,!\, n \mid d \in Dir, n \in \mathbb{N}\} \\ & \cup \{d\,? \mid d \in Dir\} \\ & \cup \{(q, n) \mathbin{@} d \mid d \in Dir, q \in Q, n \in \mathbb{N}\} \end{aligned}$$

Intuitively, an action $d \,!\, n$ means that the process is ready to transfer $n$ coins of its budget to its $d$-neighbour; an action $d\,?$ means that the process is ready to receive coins from the $d$-neighbour; and executing an action $(q, n) \mathbin{@} d$ will insert a new process on the $d$ of the process executing the action, and this new process will have control state $q$ and initial budget $n$. There is no transfer of budget during creation of a new process. Figure 1 should provide some illustration of the intended semantics.

Transfer of coins between two neighbours is synchronous. Let $\overline{\mathsf{left}} := \mathsf{right}$ and $\overline{\mathsf{right}} := \mathsf{left}$. If a process has control state $q_1$ with $(q_1, d_1 \,!\, n, q_1')$ and budget greater or equal than $n$, and its $d_1$-neighbour has control state $q_2$ with $(q_2, d_2?, q_2')$ and $d_2 = \bar{d}_1$, then they can execute both transitions at the same time and transfer $n$ coins from one process the other.

At any point in time a processes can fail. When a process fails in one step the ring of $k$ processes becomes a ring of $k - 1$ processes. The two neighbours of the failed process become neighbours to each other.

The goal is to define an algorithm to verify if a ring containing a process with control state $q$ and budget greater than $B$ can be reached from a given initial ring. We will achieve this by reducing the problem to coverability of a WSTS and instantiating the Backwards and the Forwards Search algorithms.

# Problem 1: Backwards Search on ring processes

a) Formalize the configuration space $S$ and the semantics $\rightarrow \subseteq S \times S$ of the model.

[Hint: *A ring of arbitrary size is almost a word*]

b) Define an order $\trianglelefteq$ over configurations. Prove that $(S, \rightarrow, \trianglelefteq)$ is an effective WSTS and argue why the problem above reduces to coverability.

c) Give an algorithm to compute minpre for ring processes.

# Problem 2: Forwards Search for ring processes

a) Describe $\mathrm{Ideals}(S)$ for the configuration space $S$ of problem 1, as a recursively enumerable set.

b) Argue whether inclusion between ideals of $S$ is decidable.

c) Show how to compute $\widehat{\mathrm{post}}$ for transitions executing $(q, n) \mathbin{@} d$.
**OPTIONAL:** Consider also the process of transferring coins.

[Hint: *Use the definition of* $\widehat{\mathrm{post}}$ *for LCS as source of ideas*]

# Problem 3: Ideals

a) Show that $\mathcal{L}(r)$ is downward closed for every $r \in \mathrm{SRE}(\Sigma)$ where $\Sigma$ is a finite alphabet. (This completes the proof seen in class that the downward closed languages are exactly the simple regular languages)

b) Prove that the products over $\Sigma$ are exactly the ideals of $\Sigma^*$.

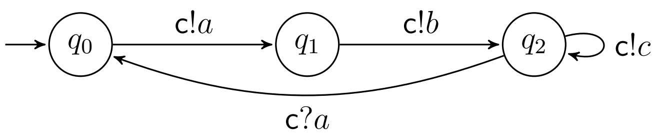
[Hint: *Use the fact that* $\mathrm{SRE}$ *are exactly the downward closed subsets of* $\Sigma^*$]

c) **OPTIONAL:** Let $(S, \leq)$ be a wqo with effective ideal completion (i.e. $\mathrm{Ideals}(S)$ is recursively enumerable and inclusion between ideals is decidable). Show inclusion of $\mathrm{SRE}(S)$ languages is decidable, by extending the algorithm seen for inclusion of products for simple regular languages over finite alphabets.

[Hint: *This may be useful for solving problem 2b*]

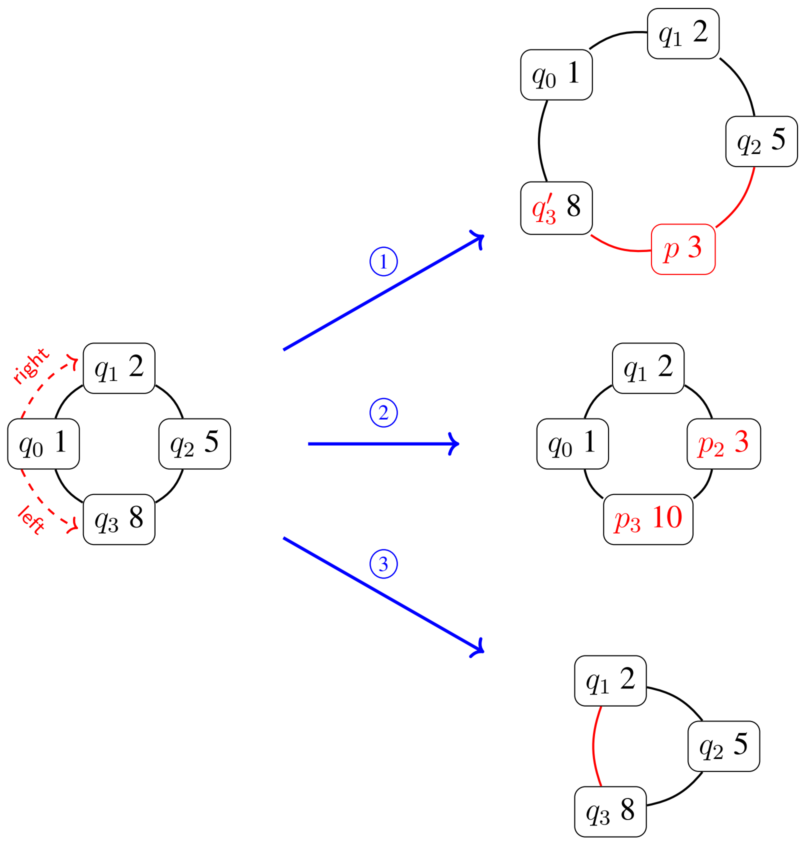
# Problem 4: Forward Uncoverability for LCS

Consider the following LCS with a single channel c and messages $M = \{a, b, c\}$:




Prove unreachability of $\gamma = (q_0, \mathsf{c} \mapsto bb)$ by defining a downward closed set of configurations $D$, represented as a finite union of ideals, such that $D \supseteq \widehat{\mathrm{post}}(D)$ and $(q_0, \varepsilon) \in D$, but $\gamma \notin D$.

Figure 1: The picture shows a ring of processes on the left and three possible transitions.




The changes introduced by the transitions are highlighted in red in the three resulting rings. The three transitions are justified as follows:

① $(q_3, (p, 3)\ @\ \mathsf{left}, q_3') \in \Delta$
② $(q_2, \mathsf{right}\ !\ 2, p_2), (q_3, \mathsf{left}?, p_3) \in \Delta$
③ the process with control state $q_0$ fails