SS 2014 25.06.2014

Exercises to the lecture
Concurrency Theory
Sheet 10

Roland Meyer, Viktor Vafeiadis Delivery until 01.07.2014 at 12h

**Exercise 10.1**
Prove the following lemma from the lecture:

a) If $\mathrm{Tr}_{\mathrm{TSO}}(P) = \mathrm{Tr}_{\mathrm{SC}}(P)$ then $\mathrm{Reach}_{\mathrm{TSO}}(\mathrm{P}) = \mathrm{Reach}_{\mathrm{SC}}(\mathrm{P})$.

b) The opposite implication does not hold.

**Exercise 10.2**
Prove the lemma by Shasha and Snir:

A trace $\mathrm{T}(\tau)$ is in $\mathrm{Tr}_{\mathrm{SC}}(P)$ if and only if $\rightarrow_{\mathrm{hb}}$ is acyclic.

**Exercise 10.3**
Consider two traces $\tau = \alpha.a.b.\gamma$ and $\tau' = \alpha'.a.\beta.b.\gamma'$ where $\mathrm{thread}(c) \neq \mathrm{thread}(a)$ and $\mathrm{thread}(c) \neq \mathrm{thread}(b)$ for all $c \in \beta$. Prove the following:

$$\text{If } a \rightarrow_{\mathrm{hb}} b \text{ in } \mathrm{Tr}(\tau) \text{ then } a \rightarrow_{\mathrm{hb}}^{+} b \text{ in } \mathrm{Tr}(\tau')$$

**Exercise 10.4**
Consider the following program implementing an instance of the *non-blocking write* protocol by H. Kopetz and J. Reisinger:

| | | | |
|---|---|---|---|
| $l_1$ : | $h := \mathrm{mem}[g];$ goto $l_2$ | $l_9$ : | $h := \mathrm{mem}[g];$ goto $l_{10}$ |
| $l_2$ : | $\mathrm{mem}[g] := h + 1;$ goto $l_3$ | $l_{10}$ : | $\mathrm{mem}[g] := h + 1;$ goto $l_{11}$ |
| $l_3$ : | $\mathrm{mem}[x] := 42;$ goto $l_4$ | $l_{11}$ : | $\mathrm{mem}[x] := 43;$ goto $l_{12}$ |
| $l_4$ : | $\mathrm{mem}[g] := h + 2;$ goto $l_5$ | $l_{12}$ : | $\mathrm{mem}[g] := h + 2;$ |
| $l_5$ : | $r := \mathrm{mem}[g];$ goto $l_6$ | | |
| $l_6$ : | $v := \mathrm{mem}[x];$ goto $l_7$ | | |
| $l_7$ : | $s := \mathrm{mem}[g];$ goto $l_8$ | | |
| $l_8$ : | assert($r \neq s \vee r$ is odd); goto $l_5$ | | |
| $l_8$ : | assert($r = s \wedge r$ is even); | | |

Prove that the program is not robust under TSO. Initially assume $\mathrm{mem}[g] = 0$ and $g \neq x$.

**Delivery until 01.07.2014 at 12h into the box next to 34-401.4**