Exercises to the lecture
Concurrency Theory
Sheet 4

Roland Meyer, Viktor Vafeiadis Delivery until 20.05.2014 at 12h

Exercise 4.1
Consider the following procedure CAS ("Compare-And-Swap"), where $r = \text{CAS}(x, e, n)$ means that the procedure returns the final value of r.

$r = \text{CAS}(x, e, n)$:
  atomic (
    $r = [x]$
    if $r = e$ **then** $([x] = n;\ r = 1)$
    else$(r = 0)$
)

a) Complete the following rule so that it specifies the behaviour of CAS:

$$\frac{\cdots}{r = \text{CAS}(x, e, n)\ \textbf{sat}\ (\boxed{\text{P}}, R, G, \boxed{\text{Q}})}$$

b) Implement a lock using the procedure CAS.
Do not use "assume", but you can use "if" and "while".

c) Prove that your lock implementation satisfies

$$\frac{\begin{array}{cc}\text{sem\_stable}(P, R) & \text{sem\_stable}(Q, R) \\ P \Rightarrow p \mapsto - * F & p \mapsto \text{TID} * F \Rightarrow Q \\ \multicolumn{2}{c}{\{p \mapsto 0 \rightsquigarrow p \mapsto \text{TID}\} \subseteq G}\end{array}}{\text{lock}(p, \text{TID})\ \textbf{sat}\ (\boxed{\text{P}}, R, G, \boxed{\text{Q}})}$$

Exercise 4.2
Let $\text{ls}(x, z)$ be defined as $x = z \wedge \textbf{emp} \vee \exists y.\ x \mapsto y * \text{ls}(y, z)$.

a) Prove $(x \mapsto y \mathrel{-\!\circledast} (P * Q)) \Rightarrow ((x \mapsto y \mathrel{-\!\circledast} P) * Q) \vee (P * (x \mapsto y \mathrel{-\!\circledast} Q))$.

b) Prove $((x \mapsto y) \mathrel{-\!\circledast} \text{ls}(k, m)) \Rightarrow (\text{ls}(k, x) * \text{ls}(y, m))$.

c) Conclude $\text{sem\_stable}(\text{ls}(k, m), R)$ with

$$R := \{x \mapsto y \rightsquigarrow (x \mapsto z * z \mapsto y), (x \mapsto z * z \mapsto y) \rightsquigarrow x \mapsto y\}.$$

Exercise 4.3
Prove the soundness of the PRIM rule:

$$\frac{\vdash_{\text{SL}} \{P\}\, c\, \{Q\} \qquad \text{writes}(c) \cap \text{fv}(R, G) = \emptyset}{c\ \textbf{sat}\ (P, R, G, Q)}$$

Delivery until 20.05.2014 at 12h into the box next to 34-401.4