Exercises to the lecture
Concurrency Theory
Sheet 14

Roland Meyer, Viktor Vafeiadis Optional sheet, no delivery

Exercise 14.1
Consider the following lock implementation **lock**(l) from the lecture:

do
  $T = l.CAS_{acq}(0, 1);$
  if $\neg t$ **then**
    while $(l.load(rlx) \neq 0);$
while $(\neg t)$

Prove that $\{\text{Lock}(l, P)\}$ **lock** $\{\text{Lock}(l, P) * P\}$ holds.

Exercise 14.2
Construct a proof in relaxed separation logic that the following program is data race free.

$*a = 7;$
$*b = 8;$
$y.\text{store}(1, rel);$
	if $(y.\text{load}(acq))$ **then**
	  $t_1 = *a;$
		if $(y.\text{load}(acq))$ **then**
		  $t_2 = *b;$
		  $*b = t_2 + 1;$

Exercise 14.3
Use the following program prog to prove that the rules for relaxed memory accesses are unsound if there is a dependency cycle.

$$x = y = 0$$

if $(x.\text{load}(rlx) == 1)$ **then**
  $y.\text{store}(1, rlx);$
	if $(y.\text{load}(rlx) == 1)$ **then**
	  $x.\text{store}(1, rlx);$

$$t = x.\text{load}(rlx);$$

Hint: Show that $\{\textbf{true}\}\ \text{prog}\ \{\text{t} = 0\}$ is derivable using RSL.

Exercise 14.4
Prove that in the following program, m always contains an even number:

$$x = y = 0$$

$$\begin{pmatrix} t = x.\text{load}(rlx) \\ x.\text{store}(t + 2, rlx) \end{pmatrix}^* \Bigg\| \begin{pmatrix} u = x.\text{load}(rlx) \\ x.\text{store}(u \times 2, rlx) \end{pmatrix}^*$$

$$m = x.\text{load}(rlx)$$