Universitetet i Oslo
Institutt for Informatikk

PMA

Einar Broch Johnsen, Martin Steffen

INF 4140: Models of Concurrency

Høst 2014

Series 6

12. 10. 2014

Topic Program Analysis II

Issued: 12. 10. 2014

Exercise 1 (While program) Consider the following program S:

```
S: x = 0; y = b;
    while (x < y) { x = x + 2; y = y + 1 }
```

Prove that the following triple is a theorem in PL:

$$\{b \geq 0\}\ \mathtt{S}\ \{x == 2 * b\}$$

You may use the following predicate I as loop invariant:

$$I: \ x \leq y \ \wedge \ x == 2(y - b)$$

Exercise 2 (Factorial function) Consider the following program S:

```
S: i = 0; x = 1;
    while (i < n) {
      i = i + 1;
      x = x * i;
    }
```

Prove the following triple using PL:

$$\{n \geq 0\}\ S\ \{x == n!\}$$

As a loop invariant I, you may use: $I : x == i! \wedge i \leq n$.

You may assume the following when reasoning about the factorial function:
1) $0! == 1$
2) $(j + 1)! == j! * (j + 1)$ for any integer $j \geq 0$

Exercise 3 (Monitor verification) Consider the monitor for Shortest-Job-Next allocation in the book (section 5.2.3). Use Programming Logic, extended with rules for `signal` and `wait` (lecture slides, week 6), to prove that this monitor satisfies the second part of the SJN invariant:

$$\texttt{free} \Rightarrow (\#\texttt{turn} == 0)$$

(You may use the rule for `wait(cv)` to reason about `wait(cv,rank)`).

Hint. When arriving at an implication, it is enough to argue for the truth of it. However, we may use the following rules when reasoning about implications.

$$\frac{}{\textbf{false} \Rightarrow A} \qquad \frac{(A \wedge B) \Rightarrow C}{A \Rightarrow (B \Rightarrow C)} \qquad \frac{(A \wedge B) \Rightarrow C}{((A \Rightarrow B) \wedge A) \Rightarrow C} \qquad \frac{(\neg A) \vee B}{A \Rightarrow B}$$

Exercise 4 Further exercises from the textbook:

2.22, 2.16, 2.24, 2.31, (2.28a, 2.29a)