Universitetet i Oslo
Institutt for Informatikk

PMA

Olaf Owe, Martin Steffen

INF 4140: Models of Concurrency

Høst 2015

Series 6

14. 10. 2015

Topic: Program Analysis II

Issued: 14. 10. 2015

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

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


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


Prove that the following triple is a theorem in PL, i.e., that it is *derivable*:

$$\{ \ b \geq 0 \ \} \ S \ \{ \ x = 2 * b \ \} \ . \tag{1}$$

You may use the following predicate I as loop invariant:

$$I : \ \ x \leq y \wedge x = 2(y - b) \ . \tag{2}$$

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

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

Prove the following triple using PL:

$$\{n \geq 0\} \ S \ \{x = n!\} \tag{3}$$

As a loop invariant I, you may use:

$$I : x = i! \land i \leq n \tag{4}$$

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:

$$\texttt{free} \Rightarrow (\#\texttt{turn} = 0) \tag{5}$$

(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.

$$\frac{}{\textbf{false} \Rightarrow A} \qquad \frac{(A \land B) \Rightarrow C}{A \Rightarrow (B \Rightarrow C)} \qquad \frac{(A \land B) \Rightarrow C}{((A \Rightarrow B) \land A) \Rightarrow C} \qquad \frac{(\neg A) \lor B}{A \Rightarrow B}$$

Exercise 4 Further exercises from the textbook:

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

Exercise 5 (For-loop) Design a rule for *for-loops.* ([?, Exercise 2.22])

Exercise 6 (Verification of a parallel program ([?, Exercise 2.16])) Consider the following parallel program.

```
int x := 0; { x = 0 }
co
   Process1: < await (x≠0) x := x − 2>
||
   Process2: < await (x≠0) x := x − 3>
||
   Process3: < await (x=0) x := x + 5>
oc
```

Prove that the final value is 0.

Exercise 7 (Interference freedom ([?, Exercise 2.24])) Consider the following statement together with a pre-condition:

$$\{\ x \geq 4\ \}\ \texttt{< x := x - 4>} \tag{6}$$

Then consider, whether the given statements *interfere* with it.

1) $\{\ x \geq 0\ \}\ \ \langle x := x + 5 \rangle\ \ \{\ x \geq 5\ \}$
2) $\{\ x \geq 0\ \}\ \ \langle x := x + 5 \rangle\ \ \{\ x \geq 0\ \}$
3) $\{\ x \geq 10\ \}\ \ \langle x := x + 5 \rangle\ \ \{\ x \geq 11\ \}$
4) $\{\ x \geq 10\ \}\ \ \langle x := x + 5 \rangle\ \ \{\ x \geq 12\ \}$
5) $\{\ x \text{ is odd}\ \}\ \ \langle x := x + 5 \rangle\ \ \{\ x \text{ is even}\ \}$
6) $\{\ x \text{ is odd}\ \}\ \ \langle y := x + 1 \rangle\ \ \{\ x \text{ is even}\ \}$
7) $\{\ x \text{ is odd}\ \}\ \ \langle y := y + 1 \rangle\ \ \{\ x \text{ is even}\ \}$
8) $\{\ x \text{ is a multiple of } 3\ \}\ \ \langle y := x \rangle\ \ \{\ y \text{ is a multiple of } 3\ \}$.

Exercise 8 (Interference freedom (Exercise [?, 2.31])) Assume two triples

$$\{ P_1 \}\ S_1\ \{ Q_1 \} \quad \text{and} \quad \{ P_2 \}\ S_2\ \{ Q_2 \}\ .$$

Assume they are *interference* free (according to the definition). Assume that S_1 contains an await-statement $\langle \texttt{await}(B)\ T \rangle$. Let then S_1' be the same as S_1, except that the await-statement is replaced by "corresponding" while-loop

1. Assume the triple $\{ P_1 \}\ S_1\ \{ Q_1 \}$ holds. Then: Is $\{ P_1 \}\ S_1'\ \{ Q_1 \}$ still true.
2. Are $\{ P_1 \}\ S_1\ \{ Q_1 \}$ and $\{ P_2 \}\ S_2\ \{ Q_2 \}$ still *interference-free?*

Exercise 9 (Parallel boolean check (2.28a)) Check whether all elements in an array are set 0. See [**?**, Exercise 2.28(a), page 89].

Exercise 10 (Maximum (2.29a)) Determine the max from an integer array, searching for the even and odd numbers in parallel. See [**?**, Exercise 2.29, page 89].