Parallel Programming WS 2014/2015 - Solution 2

Kastens, Pfahler
Institut für Informatik, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn
Nov 10, 2014

Solution for Exercise 1

a) Consider a second process `p2`:

```
{ P2: moneyBag = x }
t2 = moneyBag;
moneyBag = t2 - 5;
{ Q2: moneyBag = x - 5 }
```

Assuming the statement sequences in both `p1` and `p2` are executed as single atomic actions, we have:

```
{ P1: moneyBag = x } S1: moneyBag = moneyBag + 10; { Q1: moneyBag = x + 10}
{ P2: moneyBag = x } S2: moneyBag = moneyBag - 5; { Q2: moneyBag = x - 5}
```

The processes interfer. Proof by showing e.g. that

```
{P1 && P2} S2 {P1}
```

does not hold.

b) Weaken the preconditions to preconditions P1’ and P2’ with P1 => P1’ and P2 => P2’, such that

```
{ P1’ } S1 { Q1’ }
{ P2’ } S2 { Q2’ }
```

can be proven:

```
{ P1’: moneyBag = x || moneyBag = x - 5} S1: moneyBag = moneyBag + 10; { Q1’: moneyBag = x + 10 || moneyBag = x + 5}
{ P2’: moneyBag = x || moneyBag = x + 10} S2: moneyBag = moneyBag - 5; { Q2’: moneyBag = x - 5 || moneyBag = x + 5}
```

c) We show non-interference using the new pre- and postconditions:

```
{P1’ && P2’} S2 {P1’}
{P2’ && P1’} S1 {P2’}
{Q1’ && P2’} S2 {Q1’}
{Q2’ && P1’} S1 {Q2’}
```

d) Since the processes do not interfere we can apply the concurrency rule (PPJ-17f) to prove the result of the concurrent execution:

```
{ P1’ && P2’ } co S1 // S2 oc {Q1’ && Q2’}
```

which yields

```
{ moneyBag = x }
co moneyBag = moneyBag + 10 //
   moneyBag = moneyBag - 5 oc
{moneyBag = x + 5}
```

Solution for Exercise 2

The files `Counter.java` and `Counters.java` contain the Java sources of the concurrent counter simulation.

Solution for Exercise 3

The following system of assertions is suitable for the proof. Each assertion takes all possible interleavings with atomic actions in the other process into account:

```
{a1: y = 1 or y = 0 or y = 4} <s1: y = y + 2;> {a2: y = 3 or y = 2 or y = 6}
{a3: y = 1 or y = 3} <s2: y = y - 1;> {a4: y = 0 or y = 2} <s3: y = y + 4;> {a5: y = 4 or y = 6}
```

For non-interference we have to prove:

```
{a1 and pre(s2)} s2 {a1}
{a1 and pre(s3)} s3 {a1}
{a2 and pre(s2)} s2 {a2}
{a2 and pre(s3)} s3 {a2}
{a3 and pre(s1)} s1 {a3}
{a4 and pre(s1)} s1 {a4}
{a5 and pre(s1)} s1 {a5}
```

The concurrence rule then implies that the conjunction of a2 and a5 is valid:

```
(y = 3 or y = 2 or y = 6) and (y = 4 or y = 6)
```

which yields

```
y = 6
```

as desired.