Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Exercise Sheet 13: Modelling Solutions

Submission deadline: June 7th, 2010

Assignment 1

(a) We use the following Promela model.

```
#define initX 3
#define initY 7

int x = initX, y = initY;

inline s() {
  y = 0;
  do
  :: x > 0 -> y = y + x; x = x - 2;
  :: else -> break
  od
}

init {
  printf("Starting in state where x = %d\n", x);
  s();
  assert y == 4;
  printf("Finishing in state where y = %d\n", y);
}
```

What changes do we need to make to the model if we want to use `proctype s()` instead of `inline s()`?

(b) The model is as follows.

```
init {
  int x;
  if
  :: x = 1
  :: x = 2; x = x + 2
  fi;

  assert (x == 1 || x == 4);
  printf("Value of x is %d\n", x);
}
```

(c) The model is as follows.

```
int x;

init {

  run Left();
  run Right();

  /* wait for processes to terminate */
  _nr_pr == 1;

  printf("Value of x is %d\n", x);
  assert x == 1 || x == 3 || x == 4;
}

proctype Left() {
  x = 1;
}

proctype Right() {
  x = 2;
  x = x + 2
}
```

(d) The model is as follows.

```
int x, y;

proctype foo () {
  do
  :: x > 1 && y < 5 ->
       x = x - y
  :: x > 1 && y < 5 ->
       y = y + 1
  :: else ->
       break
  od
}
```

```
proctype goo () {
  do
  :: x > 0 ->
       y = y + 1;
       x = x - 1
  :: else ->
       break
  od
}

init {
  x = 5;
  y = 1;

  atomic {
    run foo ();
    run goo ()
  };

  printf ("x=%d, y=%d\n", x, y);

  assert (x > -10)
}
```

We use the assert command at the end to determine the least value of the variable x.

Assignment 2

You find a sample solution at the course webpage at `philosopher.pr`.

Assignment 3

You find a sample solution at the course webpage at `needham.pr`.

Assignment 4

You find a sample solution at the course webpage at `leader.pr`.

Assignment 5 - Headache of the week

You find a sample solution at the course webpage at `knights.pr`.