Homework assignment 1

T-106.420 Concurrent Programming

Family name: **Chaparro González**
First name: **Diego**
Student number: **59881P**
`dchaparro@acm.org`

14th October 2002

Contents

1 Exercise **3**

1.1 a) S_1; S_2; S_3; 3

1.2 b) **co** $< S_1; > || < S_2; > || < S_3; >$ **oc** 3

1.3 c) **co** $< await(x > y)S_1; S_2)|| < S_3; >$ **oc** 4

2 Exercise **5**

2.1 a) 5

2.2 b) 6

3 Exercise **7**

1 Exercise

Consider the following three statements:

$S_1 : x = x + y;$

$S_2 : y = x - y;$

$S_3 : x = x - y;$

Assume that x is initially 2 and that y is initially 5. For each of the following, what are the possible final values of x and y? Explain your answers.

1.1 a) S_1; S_2; S_3;

The final values of x and y are:

- $x = 5$
- $y = 2$

Because in this case the execution of the three statements is executed in a sequential order. After the execution of the first statement the values are:

- $x = 7$ and $y = 5$

After the execution of the second statement the values are:

- $x = 7$ and $y = 2$

And then after the execution of the third statement the values are:

- $x = 5$ and $y = 2$

1.2 b) co $< S_1; > \mid\mid < S_2; > \mid\mid < S_3; >$ oc

In this case, the final values depends on the order execution of the three statements. As the three statements are concurrent, the possible order execution and the final results of the variables can be:

- $S1; S2; S3;$
  $x = 5$ and $y = 2$
- $S1; S3; S2;$
  $x = 2$ and $y = -3$
- $S2; S1; S3;$
  $x = 2$ and $y = -3$
- $S2; S3; S1;$
  $x = 2$ and $y = -3$
- $S3; S1; S2;$
  $x = 2$ and $y = -3$
- $S3; S2; S1;$
  $x = -11$ and $y = -8$

1.3 c) **co** $< await(x > y)S_1; S_2)|| < S_3; >$ **oc**

In this case the program doesn't finish never, because the process that execute the first arm doesn't finish never. The reason is that the condition of the await instruction never will be true, because x always will be minor than y.

At first x is 2 and y is 5. The first process has to wait until x is greater than y, then at first the process has to wait. The second process can execute in any moment, and after his execution the value of x will be -3 and the value of y will be 5.

And the first process will continue waiting forever, because before and after the execution of the second process the conditional expression always will be false.

2 Exercise

Consider the following program:
$intx = 0, y = 0;$
co $x = x + 1; x = x + 2; || x = x + 2; y = y - x;$ **oc**

2.1 a)

Suppose each assignment statement is implemented by a single machine instruction and hence is atomic. How many possible histories are there? What are the possible final values of x and y?

The number of possible histories is the result of $(n * m)!/(m!^n)$, where n is the number of process and that each executes a sequence of m atomic actions.

In this case the first process execute two assignment statement:
$x = x + 1; x = x + 2;$
It means 2 atomic actions. The second process execute two assignment statements:
$x = x + 2; y = y - x;$
It means another two atomic actions.

Here are two processes, and each process executes two atomic actions. The number of possible histories is 6.
$(2 * 2)!/(2!^2) = 6$

In order to show the possible values I'm going to label the statements:
co
$S_1 : x = x + 1;$
$S_2 : x = x + 2; ||$
$S_3 : x = x + 2;$
$S_4 : y = y - x;$
oc

The possible values of the variables depends on the instruction order execution:

- $S_1; S_2; S_3; S_4;$
  $x = 5$ and $y = -5$
- $S_1; S_3; S_2; S_4;$
  $x = 5$ and $y = -5$
- $S_1; S_3; S_4; S_2;$
  $x = 5$ and $y = -3$
- $S_3; S_4; S_1; S_2;$
  $x = 5$ and $y = -2$
- $S_3; S_1; S_4; S_2;$
  $x = 5$ and $y = -3$
- $S_3; S_1; S_2; S_4;$
  $x = 5$ and $y = -5$

2.2 b)

Suppose each assignment statement is implemented by three atomic actions: the load register, add or substract a value from that register, then store the result. How many possible histories are there now? What are the possible final values of x and y?

Each process executes 2 assignments, which means that each process executes 6 atomic actions. I suppose that the instruction: $y = y - x$ only take three atomic actions as well, because the load register action loads the two registers at the same time (x and y). And the number of processes is two, then the number of histories is 924:

$(2 * 6)!/(6!^2) = 924$

The possible values of x can be: 2, 3, 4, 5. And the possible values of y can be: -2, -3, -4, -5

3 Exercise

Consider the following program:

$varx, y : int$

$S :$ **co** $x = x - 1; x = x + 1 || y = y + 1; y = y - 1$ **oc**

Show that { $x == y$ **}S{**$x == y$**} is a theorem.**

First, as x is equal to y, I suppose that both are equal to z, and z can be anything:

If x==y then x==z and y==z.

It's a proof outline in which is shown that the the expression above is a theorem:

```
{ x == z ^ y == z ^ x == y}
co  {x == z}
    <x = x - 1;>
    {x == z - 1}
    <x = x + 1;>
    {x == z}
||  {y == z}
    <y = y + 1;>
    {y == z + 1}
    <y = y - 1;>
    {y == z}
oc
{x == z ^ y ==z ^ x==y}
```