Concurrent and Distributed Systems

Mitre Flavia Antonia

November 17, 2019

Third Year
Group: C.E.N. 3.1
Lecturer: Costin Bădică
Teaching assistant: Cristinel Ungureanu

1 Problem statement

1.1 Second problem

What values can n have at the end of the concurrent counting algorithm execution?
Assignment: Implement this algorithm in Promela. What values do you get at the end for n?
Assignment: Prove the above by describing in the technical reports a few test scenarios and different states as shown in the first course.
Assignment: Describe the output and different scenarios (in multiple executions).

Concurrent counting algorithm	
integer n ← 0	
p	q
integer temp	integer temp
p1: do 10 times	q1: do 10 times
p2: temp ← n	q2: temp ← n
p3: n ← temp + 1	q3: n ← temp + 1

2 Implementation

2.1 Second problem

For solving the second problem I've created:

- the variable n
- two active proctypes: P and X.

Each proctype contains the same code, I've did this because it was easier to see which one of the proctypes occurs in the output, insted of using a vector (Ex: active[2] proctype P()).

1. $n \leftarrow 0$
2. **active proctype P()**
3. $i \leftarrow 1$
4. **if** i>10 **break**
5. **else**
6. $temp \leftarrow n$
7. $n \leftarrow temp + 1$
8. $i \leftarrow i + 1$
9. **end if**
10. **end proctype**
11. **active proctype X()**
12. $i \leftarrow 1$
13. **if** i>10 **break**
14. **else**
15. $temp \leftarrow n$
16. $n \leftarrow temp + 1$
17. $i \leftarrow i + 1$
18. **end if**
19. **end proctype**

3 Experimental Data

3.1 Second problem

```
  0:    proc  - (:root:) creates proc  0 (P)
  0:    proc  - (:root:) creates proc  1 (X)
1 X     17  else
0 P     6   else
1 X     19  temp = n
Process Statement         X(1):temp
1 X     20  n = (temp+1)  0
Process Statement         X(1):temp  n
0 P     8   temp = n      0          1
Process Statement         P(0):temp  X(1):temp  n
0 P     9   n = (temp+1)  1          0          1
1 X     21  i = (i+1)     1          0          2
Process Statement         P(0):temp  X(1):i     X(1):temp  n
1 X     17  else          1          2          0          2
1 X     19  temp = n      1          2          0          2
-------------
depth-limit (-u10 steps) reached
#processes: 2
 10:    proc  1 (X) count.pml:20 (state 5)
 10:    proc  0 (P) count.pml:10 (state 6)
2 processes created
```

Figure 1: First output

In this case the value for n is 2. None of the two proctypes ends its execution, because the number of steps the program runs is 10. We can see that because of interthreading the result of n is not what we wanted in the first place.
The code lines are processed in a random order: 17, 6, 18, 20, 8, 9, 21, 17, 19. We can observe that n equals 2 (X: temp = 0, n = 1, P: temp = 1, n = 2, i = 2, X: temp = 2) when the variable i wasn't even incremented yet. This can provide a wrong result.
Explanation of "1 X 19 temp = n":
The line 19 from the active proctype X is executed, and X is the second proctype (the 1 at the beggining indicates this). The variable temp takes the value of the variable n.

```
0 P     8   temp = n        7           10          8           11          12
0 P     9   n = (temp+1)    7           12          8           11          12
1 X     17  else            7           12          8           11          13
0 P     10  i = (i+1)       7           12          8           11          13
1 X     19  temp = n        8           12          8           11          13
0 P     6   else            8           12          8           13          13
0 P     8   temp = n        8           12          8           13          13
0 P     9   n = (temp+1)    8           13          8           13          13
1 X     20  n = (temp+1)    8           13          8           13          14
0 P     10  i = (i+1)       8           13          8           13          14
1 X     21  i = (i+1)       9           13          8           13          14
0 P     6   else            9           13          9           13          14
1 X     17  else            9           13          9           13          14
1 X     19  temp = n        9           13          9           13          14
1 X     20  n = (temp+1)    9           13          9           14          14
1 X     21  i = (i+1)       9           13          9           14          15
Process Statement           P(0):i      P(0):temp   X(1):i      X(1):temp   n
0 P     8   temp = n        9           13          10          14          15
0 P     9   n = (temp+1)    9           15          10          14          15
1 X     17  else            9           15          10          14          16
1 X     19  temp = n        9           15          10          14          16
0 P     10  i = (i+1)       9           15          10          16          16
1 X     20  n = (temp+1)    10          15          10          16          16
1 X     21  i = (i+1)       10          15          10          16          17
0 P     6   else            10          15          11          16          17
1 X     17  i>10            10          15          11          16          17
 98:    proc  1 (X) terminates
0 P     8   temp = n        10          15          11          16          17
0 P     9   n = (temp+1)    10          17          11          16          17
-------------
depth-limit (-u100 steps) reached
#processes: 1
100:    proc  0 (P) count.pml:10 (state 6)
2 processes created
```

Figure 2: Second output

In this case the value for n is 16. The proctype X ends its execution. The number of steps is 100. We can see that because of interthreading the result of n is not what we wanted in the first place (16 <20).

```
1 X     20  n = (temp+1)   7          9          7          10         10
1 X     21  i = (i+1)      7          9          7          10         11
0 P     10  i = (i+1)      7          9          8          10         11
1 X     17  else           8          9          8          10         11
1 X     19  temp = n       8          9          8          10         11
1 X     20  n = (temp+1)   8          9          8          11         11
1 X     21  i = (i+1)      8          9          8          11         12
0 P     6   else           8          9          9          11         12
1 X     17  else           8          9          9          11         12
0 P     8   temp = n       8          9          9          11         12
0 P     9   n = (temp+1)   8          12         9          11         12
0 P     10  i = (i+1)      8          12         9          11         13
1 X     19  temp = n       9          12         9          11         13
1 X     20  n = (temp+1)   9          12         9          13         13
1 X     21  i = (i+1)      9          12         9          13         14
0 P     6   else           9          12         10         13         14
0 P     8   temp = n       9          12         10         13         14
Process Statement          P(0):i     P(0):temp  X(1):i     X(1):temp  n
1 X     17  else           9          14         10         13         14
0 P     9   n = (temp+1)   9          14         10         13         14
1 X     19  temp = n       9          14         10         13         15
1 X     20  n = (temp+1)   9          14         10         15         15
1 X     21  i = (i+1)      9          14         10         15         16
1 X     17  i>10           9          14         11         15         16
0 P     10  i = (i+1)      9          14         11         15         16
0 P     6   else           10         14         11         15         16
0 P     8   temp = n       10         14         11         15         16
0 P     9   n = (temp+1)   10         16         11         15         16
------------
depth-limit (-u100 steps) reached
#processes: 2
100:    proc  1 (X) count.pml:23 (state 10)
100:    proc  0 (P) count.pml:10 (state 6)
2 processes created
```

Figure 3: Third output

In this case the value for n is 17. The number of steps is 100. We can see that because of interthreading the result of n is not what we wanted in the first place (17 <20).

```
0 P     6   else            7          8          8          8          9
0 P     8   temp = n        7          8          8          8          9
1 X     19  temp = n        7          9          8          8          9
1 X     20  n = (temp+1)    7          9          8          9          9
0 P     9   n = (temp+1)    7          9          8          9          10
0 P     10  i = (i+1)       7          9          8          9          10
1 X     21  i = (i+1)       8          9          8          9          10
1 X     17  else            8          9          9          9          10
1 X     19  temp = n        8          9          9          9          10
0 P     6   else            8          9          9          10         10
0 P     8   temp = n        8          9          9          10         10
1 X     20  n = (temp+1)    8          10         9          10         10
0 P     9   n = (temp+1)    8          10         9          10         11
0 P     10  i = (i+1)       8          10         9          10         11
1 X     21  i = (i+1)       9          10         9          10         11
0 P     6   else            9          10         10         10         11
Process Statement           P(0):i     P(0):temp  X(1):i     X(1):temp  n
0 P     8   temp = n        9          10         10         10         11
1 X     17  else            9          11         10         10         11
1 X     19  temp = n        9          11         10         10         11
1 X     20  n = (temp+1)    9          11         10         11         11
0 P     9   n = (temp+1)    9          11         10         11         12
1 X     21  i = (i+1)       9          11         10         11         12
0 P     10  i = (i+1)       9          11         11         11         12
1 X     17  i>10            10         11         11         11         12
0 P     6   else            10         11         11         11         12
 98:    proc  1 (X) terminates
0 P     8   temp = n        10         11         11         11         12
0 P     9   n = (temp+1)    10         12         11         11         12
-------------
depth-limit (-u100 steps) reached
#processes: 1
100:    proc  0 (P) count.pml:10 (state 6)
2 processes created
```

Figure 4: Fourth output

In this case the value for n is 12. The proctype X ends its execution. The number of steps is 100. We can see that because of interthreading the result of n is not what we wanted in the first place (12<20).

```
1 X     20  n = (temp+1)   7          11         7          12         12
0 P     10  i = (i+1)      7          11         7          12         13
1 X     21  i = (i+1)      8          11         7          12         13
0 P     6   else           8          11         8          12         13
0 P     8   temp = n       8          11         8          12         13
0 P     9   n = (temp+1)   8          13         8          12         13
1 X     17  else           8          13         8          12         14
0 P     10  i = (i+1)      8          13         8          12         14
1 X     19  temp = n       9          13         8          12         14
1 X     20  n = (temp+1)   9          13         8          14         14
1 X     21  i = (i+1)      9          13         8          14         15
1 X     17  else           9          13         9          14         15
0 P     6   else           9          13         9          14         15
1 X     19  temp = n       9          13         9          14         15
1 X     20  n = (temp+1)   9          13         9          15         15
1 X     21  i = (i+1)      9          13         9          15         16
0 P     8   temp = n       9          13         10         15         16
0 P     9   n = (temp+1)   9          16         10         15         16
0 P     10  i = (i+1)      9          16         10         15         17
Process Statement          P(0):i     P(0):temp  X(1):i     X(1):temp  n
0 P     6   else           10         16         10         15         17
0 P     8   temp = n       10         16         10         15         17
0 P     9   n = (temp+1)   10         17         10         15         17
0 P     10  i = (i+1)      10         17         10         15         18
0 P     6   i>10           11         17         10         15         18
1 X     17  else           11         17         10         15         18
1 X     19  temp = n       11         17         10         15         18
1 X     20  n = (temp+1)   11         17         10         18         18
-------------
depth-limit (-u100 steps) reached
#processes: 2
100:    proc  1 (X) count.pml:21 (state 6)
100:    proc  0 (P) count.pml:12 (state 10)
2 processes created
```

Figure 5: Fifth output

In this case the value for n is 18. The number of steps is 100. We can see that because of interthreading the result of n is not what we wanted in the first place (18 <20).

4 Conclusions

This problem is a good example for interthreading, and how it works. I understood better this concept after elaborating this assignment, because Promela and jspin help describing better how the instructions are executed and what happens for real with the variables and the threads. one of the conclusions is that the variable n can take any value in the interval 1 to 20 depending on the order in which the code lines are executed.