First Steps with SPIN and PROMELA

David Kendall

1 Objectives

- Take first steps with SPIN and PROMELA in modelling and analysing some simple concurrent systems
- Test understanding of basic PROMELA constructs

2 Configuring the environment

In order to complete the work in these labs, you will need to ensure that the verification tools have been correctly configured for your Linux environment.

You can check this by entering the command `ispin &` in a command terminal, e.g.

```
student@c123456:~$ ispin &
```

Note that `student@c123456:~$` is a prompt provided by the OS. The command that you enter is `ispin &`. In future examples, the prompt will be abbreviated to `$`.

If the ispin program does not start then your verification environment needs to be configured. Please ask your lab tutor for help with this.

You'll be working initially with the example models that are distributed with the Spin tool. Use your browser to download the example models then execute the following commands to unpack them:

```
$ cd
$ tar zxvf ~/Downloads/SpinExamples.tgz
```

The `leader.pml` example required for the introductory session can be found in `SpinExamples`.

3 First steps

Work through the official SPIN Getting Started introduction. Then complete the exercises below.

4 Exercises

1. Consider the following program then answer the questions below.

```
1  proctype A(chan c) {
2    c!3
3  }
4
5  proctype B(chan d) {
6    byte x,y,z;
7
8    d?x
9  }
10
11 init {
12   chan channel = [0] of {byte};
13
14   atomic {run A(channel); run B(channel)}
15 }
```

   (a) Is this program syntactically correct? If it is, what does it do? Given that the name of the channel in line 2 is ‘c’ and the name of the channel in line 8 is ‘d’, say whether process A will succeed in communicating with process B. Explain your answer. Assume that you change line 5 to `proctype B(chan c) {` will this affect the behaviour of the program? Explain your answer.

   (b) Does it matter that there are 3 variables declared at line 6?

   (c) If we change line 2 to `c!3,4,5` and line 8 to `d?x,y,z` is the program still correct?

   (d) Assume we change line 2 to `c!3,4,5` and line 12 to `chan channel = [3] of {byte}` is the program correct now?

   (e) If we change line 2 to `c!3,4,5` and line 12 to `chan channel = [0] of {byte, byte, byte}` is the program correct now? Explain your answers to 1 (d) and 1 (e).

2. Are the following programs (a and b) syntactically correct? If yes, how do they behave? Suggest how they can be improved by the use of `mtype` declarations.

   (a)

```
proctype A(chan c) {
  c!3(4);
  c!4(5)
}

proctype B(chan c) {
  byte x,y,z;

  c?x(z);
  c?y(z)
}

init {
  chan channel = [1] of {byte, byte};

  atomic {run A(channel); run B(channel)}
}
```

(b)

```
proctype A(chan c) {
  c!3(4,5);
  c!4(5,6)
}

proctype B(chan c) {
  byte x,y,z;

  c?x(y,z);
  c?y(x,z)
}

init {
  chan channel = [1] of {byte, byte, byte};

  atomic {run A(channel); run B(channel)}
}
```

3. Consider the following program then answer the questions below.

```
proctype A(chan c) {
  c!3;
  c!4
}

proctype B(chan c) {
  byte x,y,z;

  c?x;
  c?y
}

init {
```

```
14   chan channel = [0] of {byte};
15
16   atomic {run A(channel); run B(channel)}
17 }
```

(a) Is it possible for line 3 to be executed before line 9? Explain your answer.

(b) Assume we change line 14 to `chan channel = [1] of {byte}` now is it possible for line 3 to be executed before line 9? Explain your answer.

4. Consider the following program then answer the questions below.

```
1  #define true 1
2  #define false 0
3
4  bool sentmsg2 = false;
5
6  proctype A(chan c) {
7    c!3;
8    c!4;
9    sentmsg2 = true
10 }
11
12 proctype B(chan c) {
13   byte x,y,z;
14
15   assert (!sentmsg2);
16   c?x;
17   c?y
18 }
19
20 init {
21   chan channel = [2] of {byte};
22
23   atomic {run A(channel); run B(channel)}
24 }
```

(a) Explain how this program allows us to check that it is not possible for process A to send its second message (i.e. to execute line 8) before process B has received the first message (i.e. has executed line 16).

(b) Use ispin to check the assertion. Explain the answer.

(c) Assume we change line 21 to `chan channel = [1] of {byte};` Does this change the behaviour of the program? Use ispin to check your conclusion. Explain the answer.

(d) Try to reach the same conclusions by using simulation rather than verification. (i.e. Use "Simulate" with a variety of seed values instead of using "Verify"). What does your experience suggest to you about the pros and cons of simulation/verification?

5. Attempt to verify the following program. What does the result tell you about the granularity of execution of Promela's assignment statement? What possible result could a finer granularity lead to?

```
#define false 0
#define true 1

byte x = 0;
bool Adone = false;
bool Bdone = false;

proctype A() {
  x = x + 1;
  Adone = true
}

proctype B() {
  x = x + 1;
  Bdone = true
}

init {
  atomic {run A(); run B()};
  Adone; Bdone;
  assert (x == 2)
}
```

6. Consider now the following program. Attempt to verify it. Explain the answer.

```
#define false 0
#define true 1

byte x = 0;
bool Adone = false;
bool Bdone = false;

proctype A() {
  byte v = 0;

  v = x;
  x = v + 1;
  Adone = true
}

proctype B() {
  byte v = 0;

  v = x;
  x = v + 1;
  Bdone = true
}

init {
  atomic {run A(); run B()};
  Adone; Bdone;
  assert (x == 2)
}
```

7. What difference does it make to the program from 6) if we replace `v = x; x = v + 1` by `atomic {v + x; x = v + 1}`? Explain your answer.