Heriot-Watt University

School of Mathematical and Computer Sciences

# Distributed Systems Programming F29NM1

*SPIN Exercise Sheet 2*

Andrew Ireland

### Exercise 1

In the lecture entitled "SPIN: Formal Analysis I", the following Promela model was used to illustrate SPIN's local assertion verification mechanism:

```
byte value1 = 1, value2 = 2, value3 = 3;
proctype A() { value3 = value3 + value2;
               assert( value3 == 5 )
}
proctype B() { value2 = value2 + value1;
               assert( value3 == 5 )
}
init { atomic{ run A(); run B()}}
```

(see `http://www.macs.hw.ac.uk/~air/spin/handouts/value`). The verification effort, however, failed. Rerun the failed verification for yourself using XSPIN. In addition, formulate and verify a global assertion about `value3`, *i.e.* a property that is always true with respect to the value of `value3`.

### Exercise 2

Attempt the verification of `TrainWare` with respect to the following global assertion:

```
nfull(TunnelAB) && nfull(TunnelBC) && nfull(TunnelCD) && nfull(TunnelDA)
```

### Exercise 3

Consider again exercise 2, but instead of using a monitor process as the basis for your verification effort, try using SPIN's temporal reasoning capabilities.

### Exercise 4

Consider your solution to the vending machine modelling exercise from sheet 1 (exercise 4). Using LTL, specify a response property that asserts:

> *If a request for chocolate is sent to the vending machine then chocolate will eventually be delivered.*

Using XSPIN, verify that your model is correct with respect to your response property.

**Exercise 5**

Consider again your answer to exercise 5 (sheet 1), *i.e.* your refined model of the vending machine that includes a coin box and a finite supply of chocolates. Using LTL, specify an invariant property that relates the contents of the coin box with the number of chocolate bars within the vending machine. Verify your model with respect to your invariant using XSPIN.