T–79.4301 **Autumn 2008**
Parallel and Distributed Systems
Tutorial 3 - Thu Oct 9, 14:15

1. In the lectures the following simplified alternating bit protocol (ABP) Promela model example was presented:

```
mtype = { msg0, msg1, ack0, ack1 };

chan to_sndr = [2] of { mtype };
chan to_rcvr = [2] of { mtype };

active proctype Sender()
{
again:
  to_rcvr!msg1;
  to_sndr?ack1;
  to_rcvr!msg0;
  to_sndr?ack0;
  goto again
}

active proctype Receiver()
{
again:
  to_rcvr?msg1;
  to_sndr!ack1;
  to_rcvr?msg0;
  to_sndr!ack0;
  goto again
}
```

   a) Modify the model of the ABP protocol to not use the `goto` statement while still retaining the protocol behaviour.
   b) Modify the model of the protocol further to transport messages of data type `byte` from the sender to the receiver. The sender obtains the data to be sent to the receiver through a channel of capacity 0 called `indata`, and the receiver sends the data it received to a channel of capacity 0 called `outdata`.
   c) Add a data source process that first sends to the channel `indata` a finite (possibly empty) sequence of 0's followed by a 1, and then enters an infinite loop in which it keeps sending 2's to this channel.

d) Add a data sink process that receives messages from the receiver through the channel `outdata`, and checks using one or more assertions that the sequence formed by the incoming messages received can be extended into a word in the language characterised by the regular expression $(0)^*1(2)^*$. (The parentheses are just to show grouping and are not part of the language specified by the expression.)

e) Check with Spin that the assertion(s) added in the previous step are not violated. What does this result tell about the model's behaviour as regards the possibility of losing or duplicating messages generated by the data source process?

f) Add to the model the possibility of losing messages in the message queues `to_sndr` and `to_rcvr`.

g) Repeat the check in step e) in the model with message loss. Explain why it is possible for the model to reach a deadlock state.

h) Fix the model to ensure the correct transmission of messages to the data sink process without deadlocking, or losing or duplicating messages generated by the data source process. (Hint: Use the `timeout` construct.) Use Spin to check that the assertions added in step d) are not violated.