Software Modeling and Verification
Lehrstuhl für Informatik 2
RWTH Aachen University

Prof. Dr. Ir. J.-P. Katoen
Priv.-Doz. Dr. T. Noll
T. Han, M. Neuhäußer

Modeling Concurrent and Probabilistic Systems

Winter Term 07/08

– Series 7 –

Hand in until December 07 before the exercise class.

Exercise 1 (3+3+1+1 points)

Modify the ABP and replace the failure situations $ack_\perp$ and $trans_\perp$ by a timeout handling to model lossy channels. The modified version of the ABP should behave as follows:

- If *Sender* sends a message, it starts a timer. If a *timeout* occurs before the acknowledgement is received, the message is retransmitted. Messages can get lost.
- If *Receiver* sends an acknowledgement, it starts a timer. If a *timeout* occurs before the next message is received, it retransmits the acknowledgement. Acknowledgements can get lost.

a) Give the modified process definition for the alternating bit protocol. Use the following timer process:

$$Timer = start.(\overline{timeout}.Timer + stop.Timer).$$

Compose your new *Sender* and *Receiver* each with a *Timer* process!

b) Minimize the LTS of the *Sender* and its *Timer* by computing its quotient under weak bisimulation! Use the partitioning algorithm from the lecture!

c) Do the same for the LTS of the *Receiver* and its *Timer*! You may do this directly, i.e. without applying the partitioning algorithm.

d) To prove the new protocol correct, one could replace the *Sender* $||$ *Timer* and *Receiver* $||$ *Timer* components by their quotients under weak bisimulation to obtain a smaller LTS. Why is this approach incorrect in general? Why can it still be applied in our setting?

Exercise 2 (4 points)

Show that the following simple communication protocol works correctly.
To this aim, prove that $Protocol(a, f)$ is observationally congruent to a one-place buffer:

$$\begin{aligned}
Protocol(a,f) &= \mathsf{new}\ b,c,d,e\ \big(Sender(a,b,d,e) \mid\mid Medium(b,c,d) \mid\mid Receiver(c,e,f)\big)\\
Sender(a,b,d,e) &= a.Sender'(a,b,d,e)\\
Sender'(a,b,d,e) &= \overline{b}.\big(d.Sender'(a,b,d,e) + e.Sender(a,b,d,e)\big)\\
Medium(b,c,d) &= b.\big(\overline{c}.Medium(b,c,d) + \overline{d}.Medium(b,c,d)\big)\\
Receiver(c,e,f) &= c.\overline{f}.\overline{e}.Receiver(c,e,f)
\end{aligned}$$

Here the single actions can be interpreted as follows:

a *Sender* is requested to transmit data

b *Sender* sends data along *Medium*

c *Medium* transmits data correctly

d *Medium* transmits data incorrectly

e *Receiver* acknowledges transmission

f *Receiver* delivers data

Reminder: the one-place buffer is defined by $B(a, f) = a.\overline{f}.B(a, f)$.