Mini Project: Alternating Bit Protocol*

June 30, 2011

Introduction to Concurrency and Verification, Winter 2007/2008

`www.cs.uni-salzburg.at/~anas`

The deadline for delivering this mini-project is Monday, 25 February, 2008. Deliver your solution via email. (You should receive a confirmation email within two days. If not then resend your solutions or contact me.) Pay special attention to the fact that the solutions must be your own solutions and solutions of different groups must be independent.

In this mini-project you are asked to model the alternating bit protocol in the CCS language and verify it using CWB. The alternating bit protocol is a simple yet effective protocol for managing the retransmission of lost messages. Consider a sender S and a receiver R, and assume that the communication medium from S to R is initialized so that there are no messages in transit. The alternating bit protocol works like this:

- Each message sent by S contains an additional protocol bit, 0 or 1.
- When S sends a message, it sends it repeatedly (with its corresponding bit) until receiving an acknowledgment (ACK) from R that contains the same protocol bit as the message being sent.
- When R receives a message, it sends an ACK to S and includes the protocol bit of the message received. When a message is received for the first time, the receiver delivers it for processing, while subsequent messages with the same bit are simply acknowledged.
- When S receives an acknowledgment containing the same bit as the message it is currently transmitting, it stops transmitting that message, flips the protocol bit, and repeats the protocol for the next message.

There is no direct communication between the sender and the receiver; all messages must travel through the medium.

*This text is adopted (and adapted) from the project description at the web page of Luca Aceto.

Your tasks are as follows:

- Implement the alternating bit protocol in CWB. (Abstract away from the content of the messages and focus only on the additional control bit. To model the decision when the sender retransmits the message, use either nondeterminism or ever better a special process called Timer which will communicate with the sender on a channel called timeout and thus signal when a message should be retransmitted. You can also try to model the checksum verification - see the link below - using nondeterminism.)
- Suggest a specification of the protocol and check whether it is equivalent to your implementation (use a suitable equivalence notion available in CWB). In particular, consider the following degrees of reliability of the communication medium and answer this question for all of these choices:
  - perfect channels (all received messages are delivered)
  - lossy channels (received messages can get lost without any warning)
  - lossy and duplicating (in addition the received message can be delivered several times).
- Check for possible deadlocks (stuck configurations) and livelocks (a possibility of an infinite sequence of tau actions) by formulating the properties as recursive formulae and by verifying whether the implementation satisfies these formulae.

How to deliver the mini-project?

Create a short report. The report must be in pdf format and should contain:

- Full names and emails of all people that worked in your group (maximum 3 students per group).
- Your commented implementation and specification of the protocol (suggestion: use * for the comments in the protocol description and then copy/paste it directly into the report).
- A short conclusion about verification of the protocol using equivalence checking (including perfect, lossy, and lossy and duplicating channels). What equivalence did you choose and why?
- Formulae in CWB syntax for deadlock and livelock and whether your implementation does or does not satisfy these formulae.
- A short conclusion (e.g. Does the protocol satisfy the desired properties? What are the possible extensions of the protocol or do you have any suggestions for a more advanced modelling of some features of the protocol? What was your experience when working with CWB? ...). The report does not have to be long at all but it should contain (at least to some extent) all the points mentioned above. Send the report together with the CWB

source file containing the implementation and specification of the protocol via email to me (ana.sokolova@cs.uni-salzburg.at).

Useful links:

- A brief description of the protocol including checksum is at `http://www.answers.com/topic/alternating-bit-protocol`.
- A graphical simulation of the protocol is available at `http://www.cs.stir.ac.uk/ kjt/software/comms/jasper/ABP.html`. (Note that the control bits in the acknowledgment of the messages are switched.)