Introduction to specification and verification, exercise 1,

November 5, 2014

1. Design a data structure for transition systems. Describe the structure using diagrams and natural language. Then implement the structure using Java or C. (This means writing the type definitions; it is not necessary to define and implement operations.)

2. Design an algorithm which finds deadlocks in transition systems. Use the data structure from the task 1 in your algorithm.

3. Suppose that the channel in the AB protocol is two-way (so messages can travel in both directions). Moreover, messages may disappear, duplicate and the order of the messages in the channel may change. The channel may contain several messages at the same time. Show, with the help of a scenario, that the AB protocol does not work correctly with this kind of a channel.

   Assume next that the channel is as before but the order of the messages does not change in the channel. Is the AB protocol working now properly? If not, give a scenario. If it works, give reasons.

4. In our AB protocol, the timer may timeout too early without waiting for acknowledgements. Modify the specification of the AB protocol without channels so that only the lost acknowledgements cause the timer to timeout.

5. Draw the global state graph of your modified AB protocol. Using the global state graph, can you deduce that the modified version is correct?