Prof. Bernd Finkbeiner, Ph.D.
Martin Zimmermann, Ph.D.
Leander Tentrup, B.Sc.

Summer term 2013
Problem Set 12
September 24, 2013

Verification

Problem 1: Stuttering [4 Points]

Consider the transition system TS shown in the following figure.

(a) Give the smallest transition system that is stutter-trace equivalent to TS. [2 Points]

(b) Is the resulting transition system from (a) stutter bisimilar to TS? Justify your answer. [2 Points]

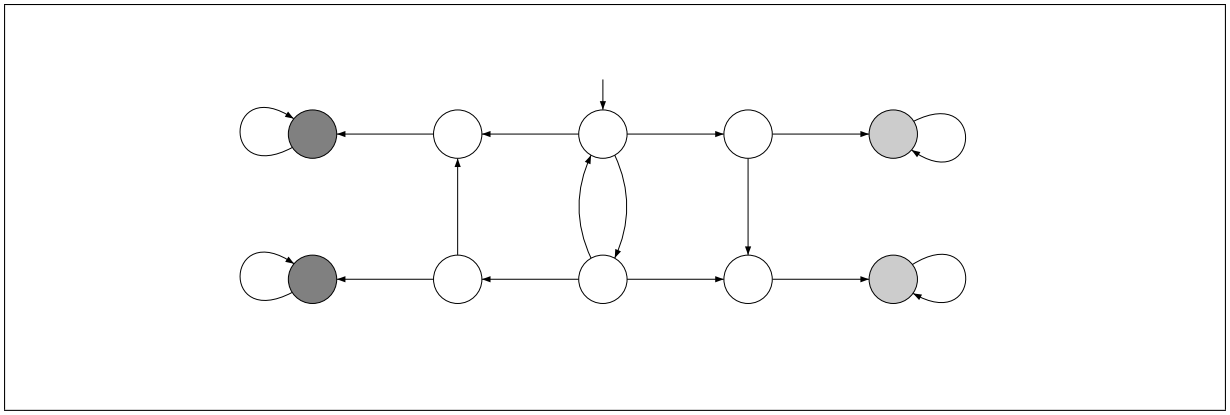
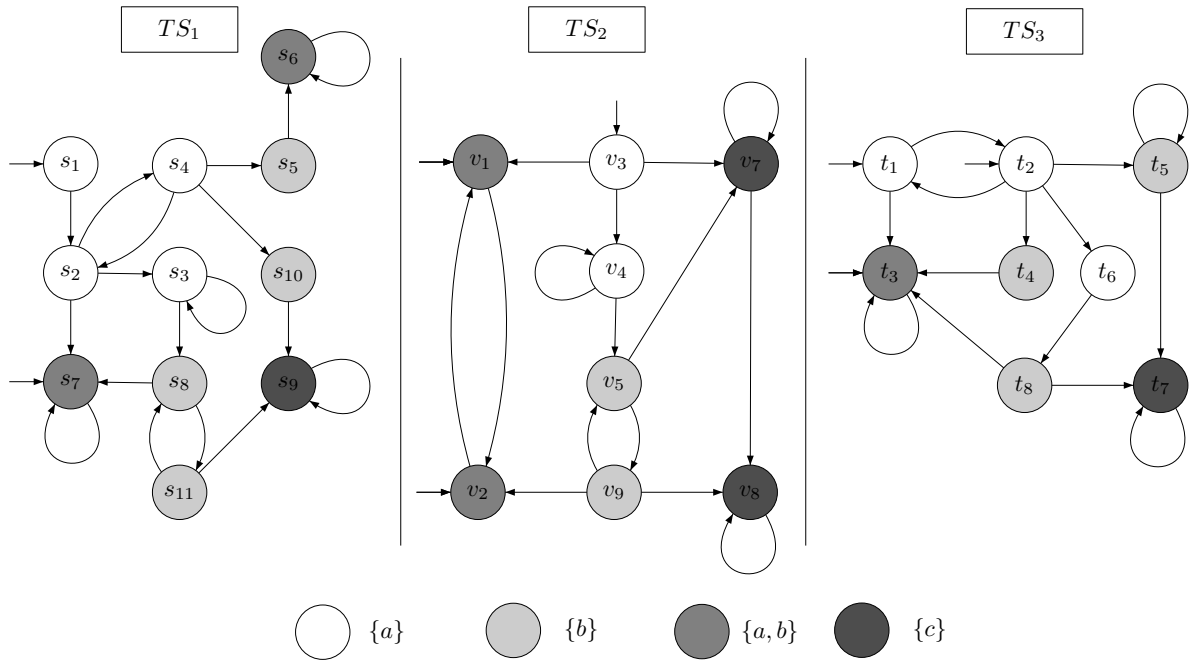


Figure 1: Transition system TS

Problem 2: Stuttering Bisimulation [6 Points]

Consider three transitions systems given on the next Figure:



For each $i, j \in \{1 \dots 3\} \times \{1 \dots 3\}$, $i \neq j$, determine whether $TS_i \approx TS_j$ or $TS_i \not\approx TS_j$. Justify your answer.

The following exercises belong to the afternoon session.

Problem 3: The CSMA/CD Protocol [4+8 Points]

In this exercise you will study the Media Access Control (Mac) sub layer of the Carrier Sense, Multiple Access with Control Detection (CSMA/CD) communication protocol. The protocol specification consists of two MAC entities, **MAC1** and **MAC2**, interconnected by a bi-directional medium **M**. The MAC entities are identical and can both transmit and receive messages over the medium. This means that collisions may occur on the medium (if the two MAC's transmit simultaneously). It is assumed that collisions will be detected in the medium and signaled to both MAC1 and MAC2.

A model of the protocol can be downloaded from the lecture website (the specification is taken from "Verifying a CSMA/CD Protocol with CCS" by Joachim Parrow). The specification uses the following synchronization actions to describe the protocol events:

- *send* - service provided by Mac which reacts by transmitting a message,
- *rec* - (**rec**eive) service provided by Mac, indicates that a message is ready to be received,
- *b* - (**b**egin) Mac begins message transmission to M,
- *e* - (**e**nd) Mac terminates message transmission to M,

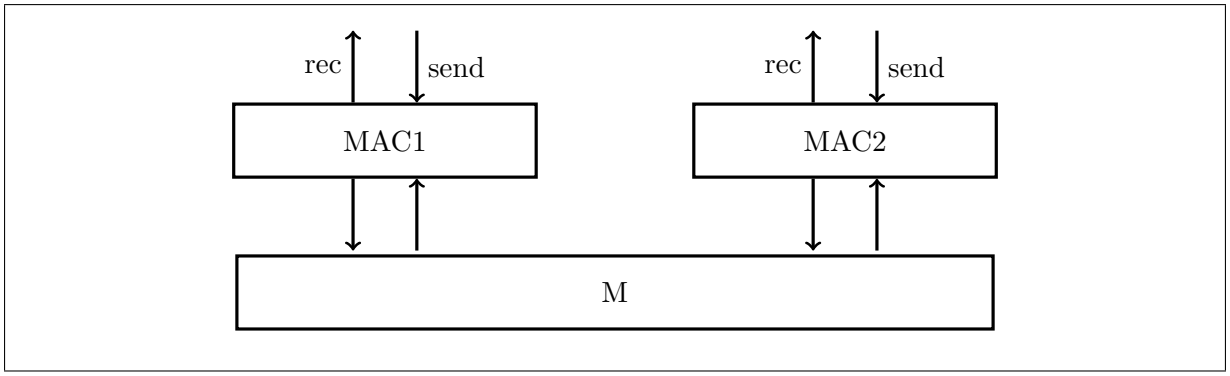


Figure 2: Overview of the MAC sub layer.

- *br* - (**b**egin **r**eceive) M begins message delivery to Mac,
- *er* - (**e**nd **r**eceive) M terminates message delivery to Mac,
- *c* - (**c**ollision) Mac is notified that a collision has occurred on M.

Note that a message transmission is not modeled by a single action. Instead the start of a transmission and the end of a transmission are modeled by two separate actions (the actions *b(r)* and *e(r)*). This is needed as there may be collisions detected in the middle of a transmission. Note also that we use indexes on all actions as there are two identical MAC entities.

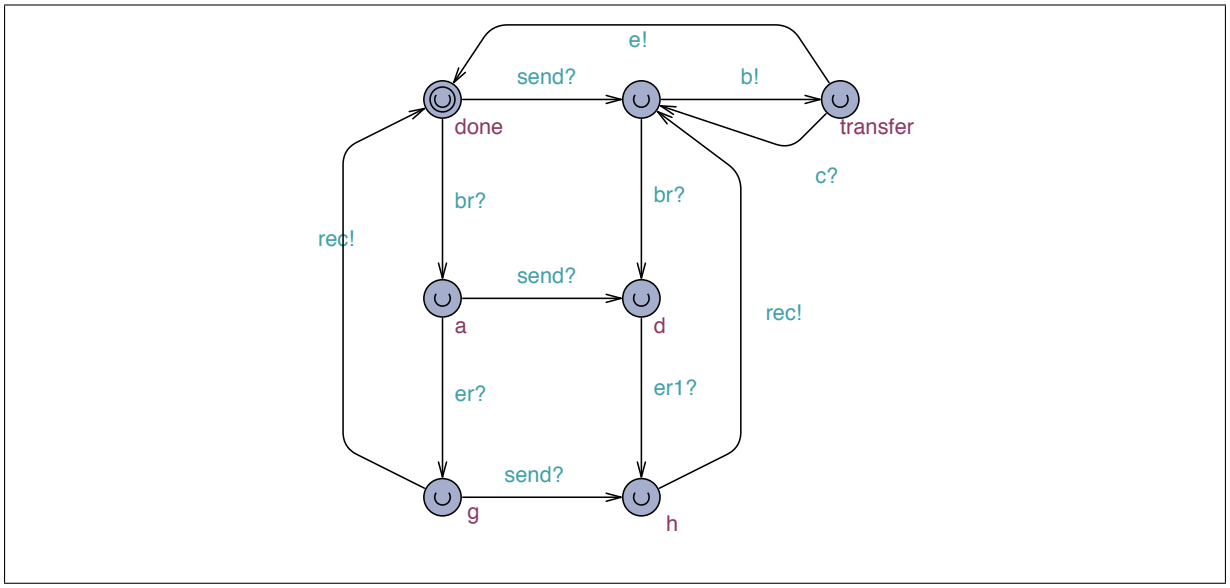


Figure 3: MAC.

Initially, MAC accepts a service call (*send?*). The MAC initiates transmission (*b!*), unless a message is in the process of being received. If the transmission is successfully terminated (*e!*) new messages can be transmitted and the process is repeated. If a collision occurs (*c?*) the MAC attempts re-transmission of the message. In all states (except when a message is being transmitted) the MAC is willing to start receiving. A message may be received (*br?*) after which the MAC may not begin message transmission before the end of message (*er?*) has been received and the MAC has signaled that the message is ready to be delivered (*rec!*). However, the MAC may receive a send request (*send?*) if there is not already another request waiting.

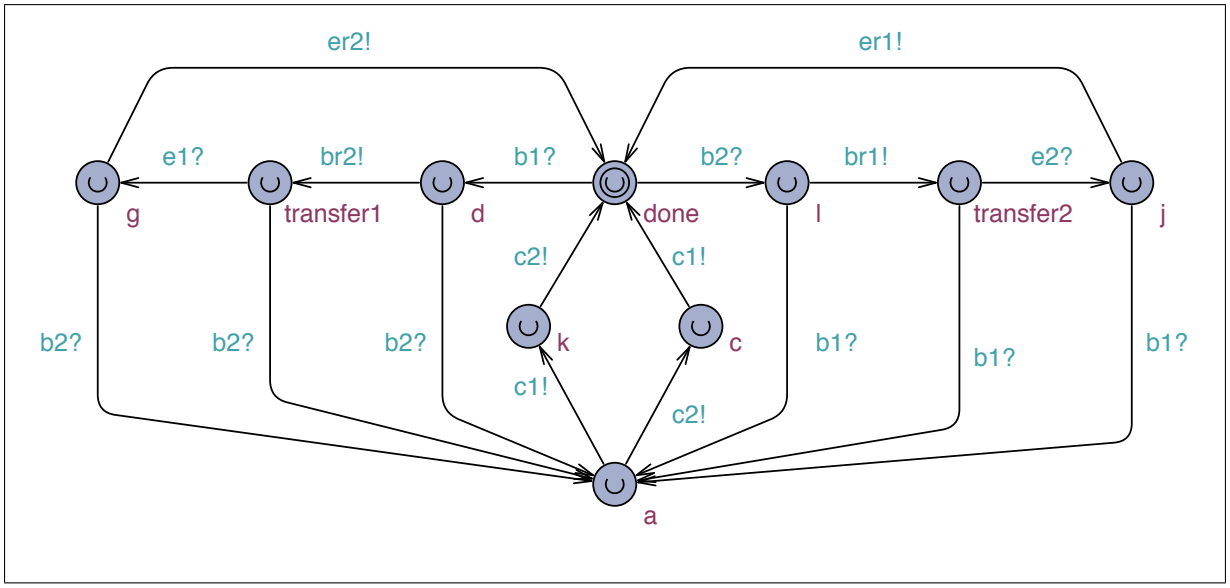


Figure 4: The medium M.

Initially, the medium accepts transmission from one of the MAC's (*b1?* or *b2?*). We assume MAC1 (*b1?*) starts transmitting first (the case for *b2?* is symmetric). The medium is assumed to be "half-duplex" meaning that a full message must be transmitted (*br2!*, *e1?*, *er2!*) before the next message can be accepted. If the receiving MAC (i.e. MAC2) starts transmission (*b2?*) the medium interrupts the transmission and signals collision (*c1!*, *c2!*) to both MAC1 and MAC1 (in any order).

a) Open the protocol specification and add a test environment that "uses" the protocol. Validate that your system is functionally correct using UPPAAL's simulator. [2 Points]

b) Check (by verification) if the system is correct in the sense that sent messages are received. How many messages can be in transfer at the same time? Is it 1, 2 or more messages? [2 Points]

The model of the MAC is slightly inaccurate. In reality, a MAC would be two processes: one sender performing the actions *send!*, *b!*, *e!*, *c?* and one receiver performing the actions *rec!*, *br?* and *er?*

c) Refine the protocol by letting each MAC consist of two processes, a sender (S) and a receiver (R). The idea is to let S perform all "horizontal" transitions and R all "vertical" transitions. Replace MAC with S and R. Use UPPAAL to validate and verify that the protocol no longer is correct. [4 Points]

c) Redefine the protocol again so that it works, while still keeping the S and R separate. This is a bit tricky. You may need to add some synchronization channels (or data variables) to achieve synchronization between S and R. [4 Points]

Part c)-d) belong to the morning session on Wednesday.

The exercise was taken from the Quantitative Model Checking lecture by Kim G. Larsen. You can find more problems to exercise at http://people.cs.aau.dk/~kgl/QMC2010/exercises/.