Concurrency Theory
Exercise Sheet 6

Franks, Billy Joe
b_franks12@cs.uni-kl.de

Kahraman, Kerem
keremkahraman@hotmail.de

Bergsträßer, Pascal
p_bergstra15@cs.uni-kl.de

December 3, 2018

1 Event-Driven Asynchronous Programs

Claim: Asynchronous programs are WSTS.

Proof: Let (D, τ, d_0, t_0) be an asynchronous program and (D, T) the state of the system.

- $((D, T), \rightarrow)$ is TS obviously.
- $((D, T), \leqslant)$ with

$$(D, T) \leqslant (D', T') \Leftrightarrow T \leqslant T'$$

  is WQO because it is:

  - reflexive: $(D, T) \leqslant (D, T)$ because $T \leqslant T'$
  - transitive: Let $(D, T) \leqslant (D', T') \bigwedge (D', T') \leqslant (D'', T'')$
    $\Rightarrow T \leqslant T' \bigwedge T' \leqslant T'' \Rightarrow T \leqslant T''$
  - Let (D_n, T_n) be a infinite sequence of system states. Therefore there exists a $k \in \mathbb{N}$ with $T_k = min\{T_n | n \in \mathbb{N}\}$ and an $m \in \mathbb{N}, m > k$ with $T_m \geqslant T_n$.

- Compatibility: Let $(D_1, T_1), (D_2, T_2), (D'_1, T'_1)$ be system states such that $(D_1, T_1) \rightarrow (D_2, T_2) \bigwedge (D_1, T_1) \leqslant (D'_1, T'_1)$. Set all $(D'_2, T_2[i]')$ with the maximum value of $T_2[i]$ and $T'_1[i]$ and their respective D values. It follows that $(D'_2, T'_2) \geqslant (D_2, T_2)$ and the transition from $(D'_1, T'_1) \rightarrow (D'_2, T'_2)$ exists by construction.

2 Comparing models of Communicating State Machines

2.1 Task 1

The transition rules for point-to-point communication with bags are as follows:

Sending a message

$$\frac{M[i] \xrightarrow{j!a} s \quad C' = C[(i,j) \leftarrow C[i,j] \cup \{a\}] \quad M' = M[i \leftarrow s]}{(M,C) \rightarrow (M',C')}$$

Receiving a message

$$\frac{M[i] \xrightarrow{?a} s \quad a \in C[j,i] \quad C' = C[(j,i) \leftarrow C[j,i] \setminus \{a\}] \quad M' = M[i \leftarrow s]}{(M,C) \rightarrow (M',C')}$$

2.2 Task 2

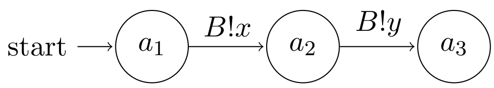
Since a communicating state machine has no mechanism to receive messages dependent on the sender or channel, both the p2p bags and the mailbox bag can be seen as one multiset of received messages. Therefore, every bag+p2p trace is also a bag+mailbox trace and vice versa.

2.3 Task 3

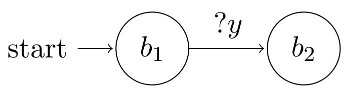
Since FIFO+mailbox+lookahead only extends the receive rule, it is still possible to not make use of the lookahead by choosing w as empty. Thus, every FIFO+mailbox trace is still a possible FIFO+mailbox+lookahead trace.

Conversely, we can give a FIFO+mailbox+lookahead trace that is not a possible FIFO+mailbox trace in the following example:

A:



B:



The following trace is possible with FIFO+mailbox+lookahead:

- initial state
  A: state $a1$, messages: ϵ, B: state $b1$, messages: ϵ
- A sends to B
  A: state $a2$, messages: ϵ, B: state $b1$, messages: x

- A sends to B
  A: state $a3$, messages: ϵ, B: state $b1$, messages: $x \cdot y$
- B skips x and receives y
  A: state $a2$, messages: ϵ, B: state $b2$, messages: ϵ

This trace is not possible with FIFO+mailbox, since B can not skip x to receive y. So, B can not apply any transitions.

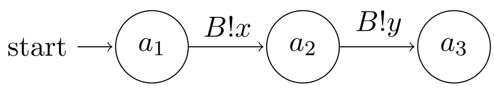
2.4 Task 4

We suppose that bag+mailbox is meant, since FIFO+bag does not make sense.

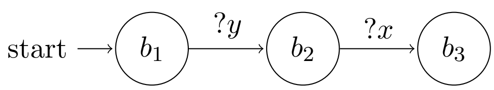
Every FIFO+mailbox+lookahead trace is also a bag+mailbox trace, since with a bag it is not necessary to skip messages as the multiset semantics allows us to receive the messages in an arbitrary order. Moreover, in the bag+mailbox setting we can just ignore the messages that are skipped by FIFO+mailbox+lookahead.

To give a bag+mailbox trace that is not possible with FIFO+mailbox+lookahead, we extend the example above as follows:

A:



B:



The following trace is possible with bag+mailbox:

- initial state
  A: state $a1$, messages: $\varnothing$, B: state $b1$, messages: $\varnothing$
- A sends to B
  A: state $a2$, messages: $\varnothing$, B: state $b1$, messages: $\{x\}$
- A sends to B
  A: state $a3$, messages: $\varnothing$, B: state $b1$, messages: $\{x, y\}$
- B receives y
  A: state $a2$, messages: $\varnothing$, B: state $b2$, messages: $\{x\}$
- B receives x
  A: state $a2$, messages: $\varnothing$, B: state $b3$, messages: $\varnothing$

This trace is not possible with FIFO+mailbox+lookahead, since to receive y, B has to skip x. Therefore it can not receive x after y as skipped messages are dropped.