TECHNISCHE UNIVERSITÄT DARMSTADT

Prof. Dr. Max Mühlhäuser
Dr. Immanuel Schweizer

Jens Heuschkel, MSc.
Michael Stein, MSc.

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

TK1: Distributed Systems - Programming & Algorithms

Extra Theory Assignment
Submission Date: -

Task 1: Applying SOS-rules (3P)

The semantics of CCS (**C**alculus of **C**ommunicating **S**ystems) is defined on the basis of **S**tructural **O**perational **S**emantics (SOS)-rules. By applying these rules on CCS expressions the corresponding **L**abelled **T**ransition **S**ystem (LTS) can be built, which shows the system as a directed graph.
Apply the SOS-rules on the CCS processes P_1 to P_7 and sketch the corresponding LTSs as shown in the following example (Figure 1). Please write the number of the SOS-rule that you used for solving next to the corresponding edge. You can find the number of the SOS-rule in the lecture slides.

$Example := (b)\big((a.'b.0|'a.b.0)\big)$

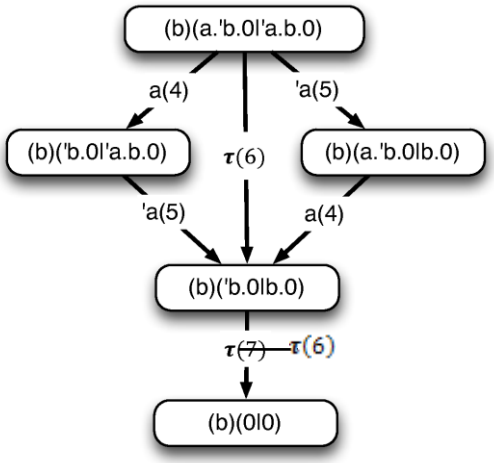

Figure 1: LTS of the process "Example"

Processes (for Task 1):

$P_1 := b.0 + c.0$
$P_2 := a.(b.0 + c.0)$
$P_3 := a.\tau.(\tau.b.0 + c.0)$
$P_4 := (d,e)(a.d.(e.b.0 + c.0)|'d.'e.0)$
$P_5 := \big((a.'b.0 + c.'b.0)|'a.b.0\big)$
$P_6 := (b)\big((a.'b.0 + c.'b.0)|('a.b.0 + 'c.b.0)\big)$
$P_7 := (a,d)(c.d.0[c/d][d/a]|'a.0|'b.0)$

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

Task 2: Weak Bisimulation (3P)

Four processes are given

$P \coloneqq a.(b.P + c.P)$ $\quad\quad$ $R \coloneqq \tau.a.(b.R + c.R)$

$Q \coloneqq a.(\tau.b.Q + \tau.c.Q)$ $\quad\quad$ $S \coloneqq a.(\tau.b.S + c.S)$

Sketch the LTS for each process. Then test each process to each other process on weak Bisimulation equivalence ($P \approx Q$): Which processes are equivalent to each other? Provide the equivalence relation R for the equivalent processes.

Task 3: Switch (4P)

The process in figure 2 models a switch. Model different behavior in CCS:

1. Initially, every incoming *a* is channeled through. After an *s* is received, the following *a*'s are discarded.
2. Initially, every incoming *a* is discarded. After an s has been received, incoming a's are transformed into b's. Another *s* changes the behavior back again, etc.
3. Initially, every incoming *a* is transformed into *b*. If an s is received, the next two *a*'s are channeled through. The following *a*'s are again transformed into *b*'s.

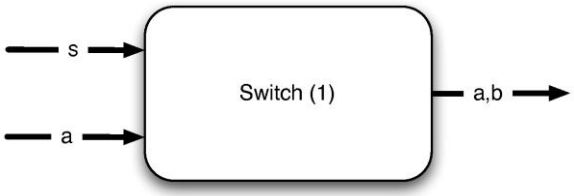

Figure 2: Depending on the internal behavior of the agent and depending on the reception of s, the message a is either channeled through, discarded or transformed into b.