Telecooperation Lab
Prof. Dr. Max Mühlhäuser

# Telekooperation 1: Exercise WS15/16
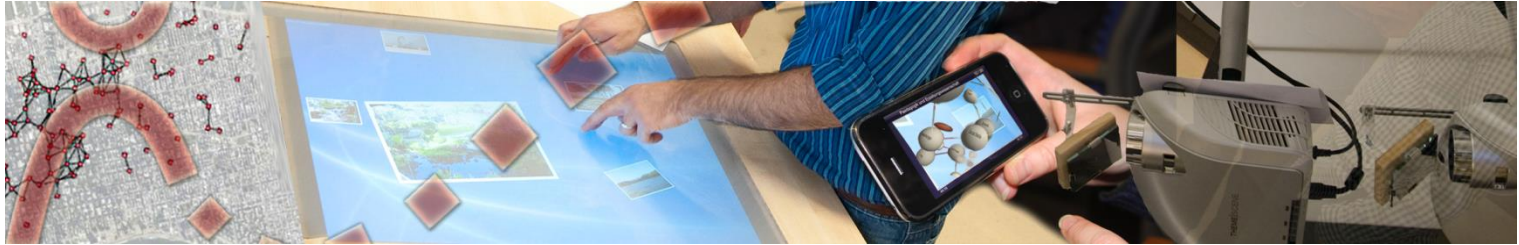
Michael Stein, MSc.

michael.stein@tk.informatik.tu-darmstadt.de

Jens Heuschkel, MSc.

jens.heuschkel@tk.informatik.tu-darmstadt.de

# TK1 – EXERCISE

- Solution Extra Exercise

- 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.

**Processes (for Task 1):**

$P_1 := b.0 + c.0$

$P_2 := a.(b.0 + c.0)$

$P_3 := a.\tau.(\tau.b.0 + c.0)$

$P_4 := (d, e)(a.d.(e.b.0 + c.0)|'d.'e.0)$

$P_5 := ((a.'b.0 + c.'b.0)|'a.b.0)$

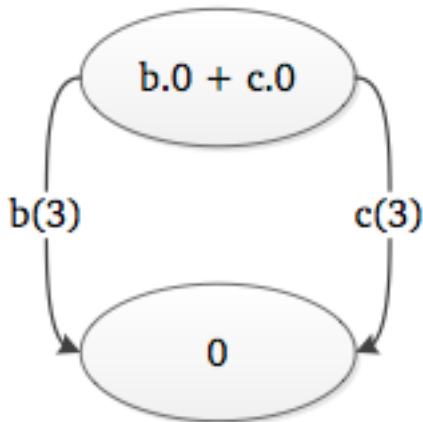$P_6 := (b)((a.'b.0 + c.'b.0)|('a.b.0+'c.b.0))$

$P_7 := (a, d)(c.d.0[c/d][d/a]|'a.0|'b.0)$

- P_1 := b.0 + c.0

P1

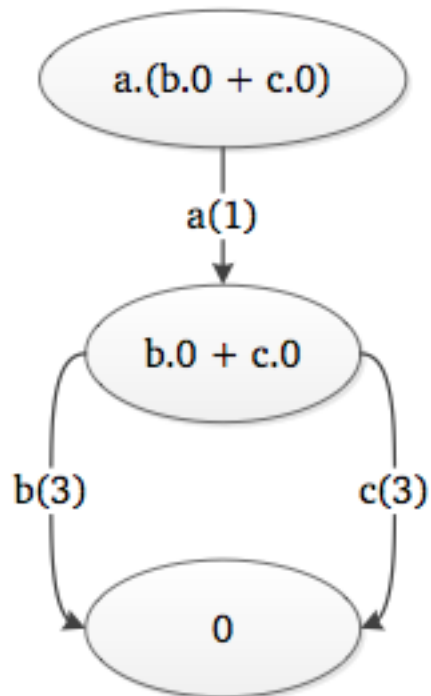- P_2 := a.(b.0 + c.0)

P2

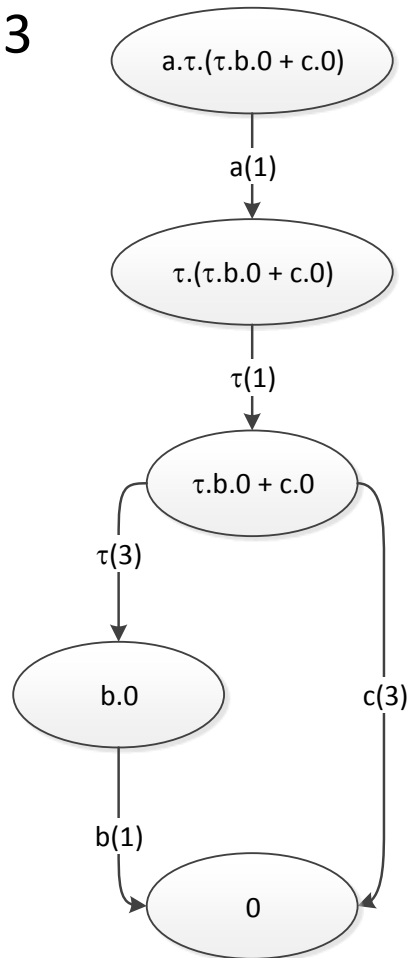

a.(b.0 + c.0)

a(1)

b.0 + c.0

b(3)          c(3)

0

- P_3 := a.tau.(tau.b.0 + c.0)

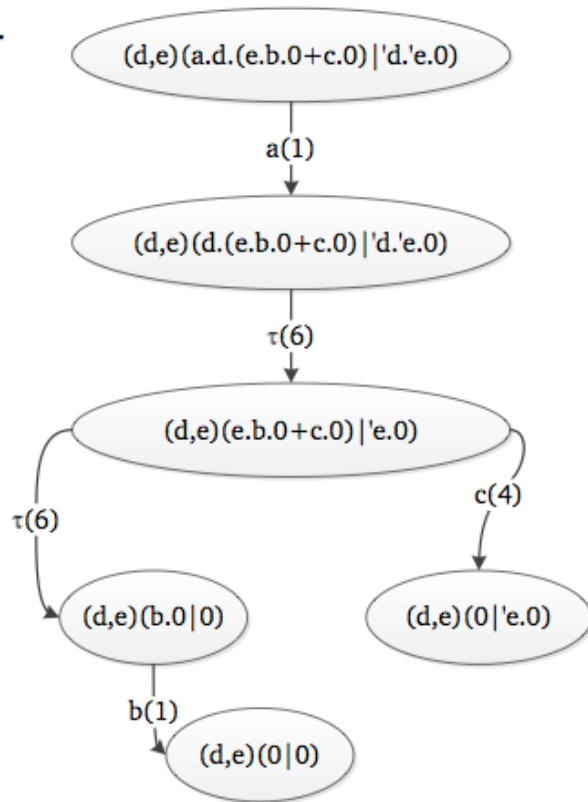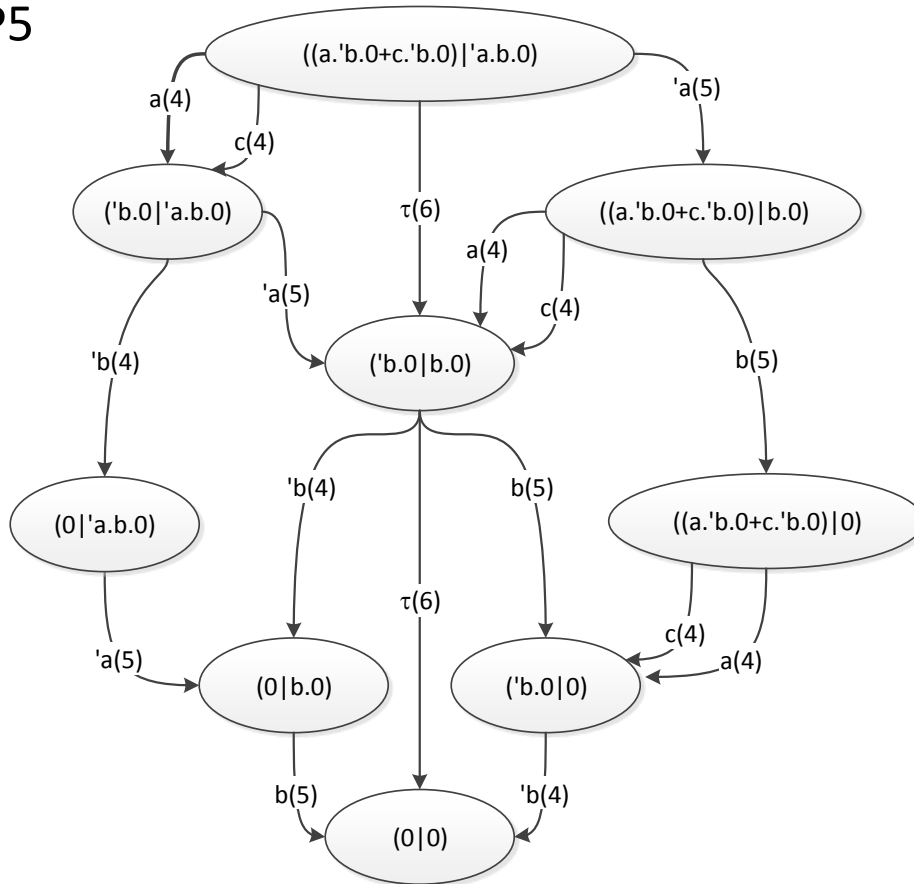P3

- P_4 := (d, e) (a.d.(e.b.0 + c.0) | 'd.'e.0)

- P_5 := ((a.'b.0 + c.'b.0) | 'a.b.0)

P5

- P_6 := (b)((a.'b.0 + c.'b.0) | ('a.b.0 + 'c.b.0))



P6

(b)((a.'b.0+c.'b.0)|('a.b.0+'c.b.0) )

a(4),c(4)

'a(5),'c(5)

(b)(('b.0|('a.b.0+'c.b.0) )

τ(6)

(b)((a.'b.0+c.'b.0)|(b.0) )
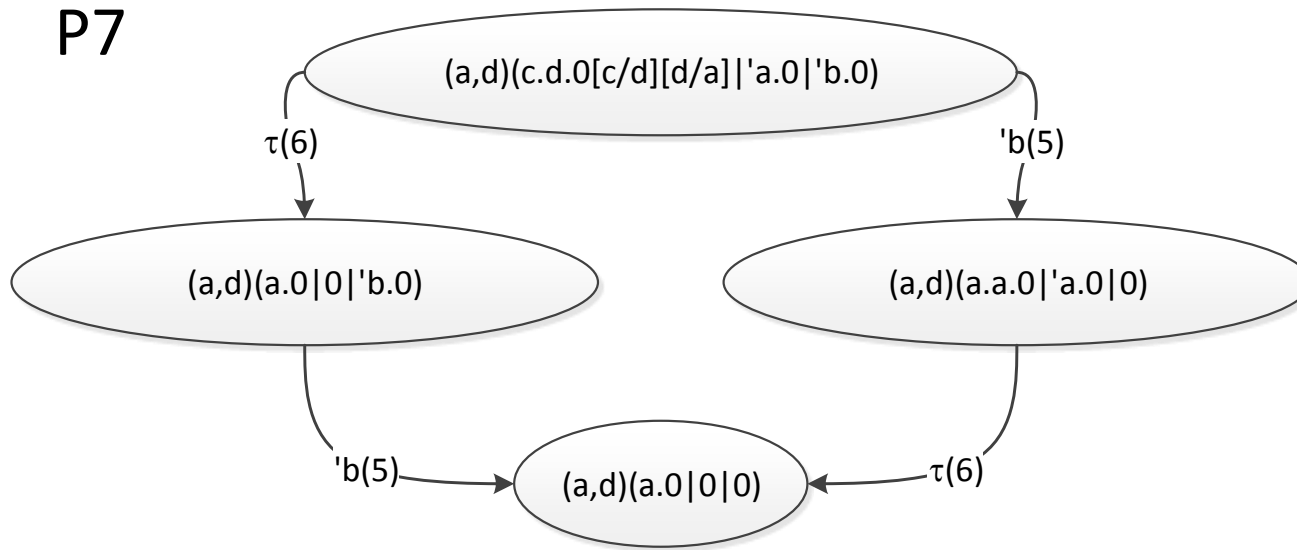
'a(5),'c(5)

(b)('b.0|b.0)

a(4),c(4)

τ(6)

(b)(0|0)
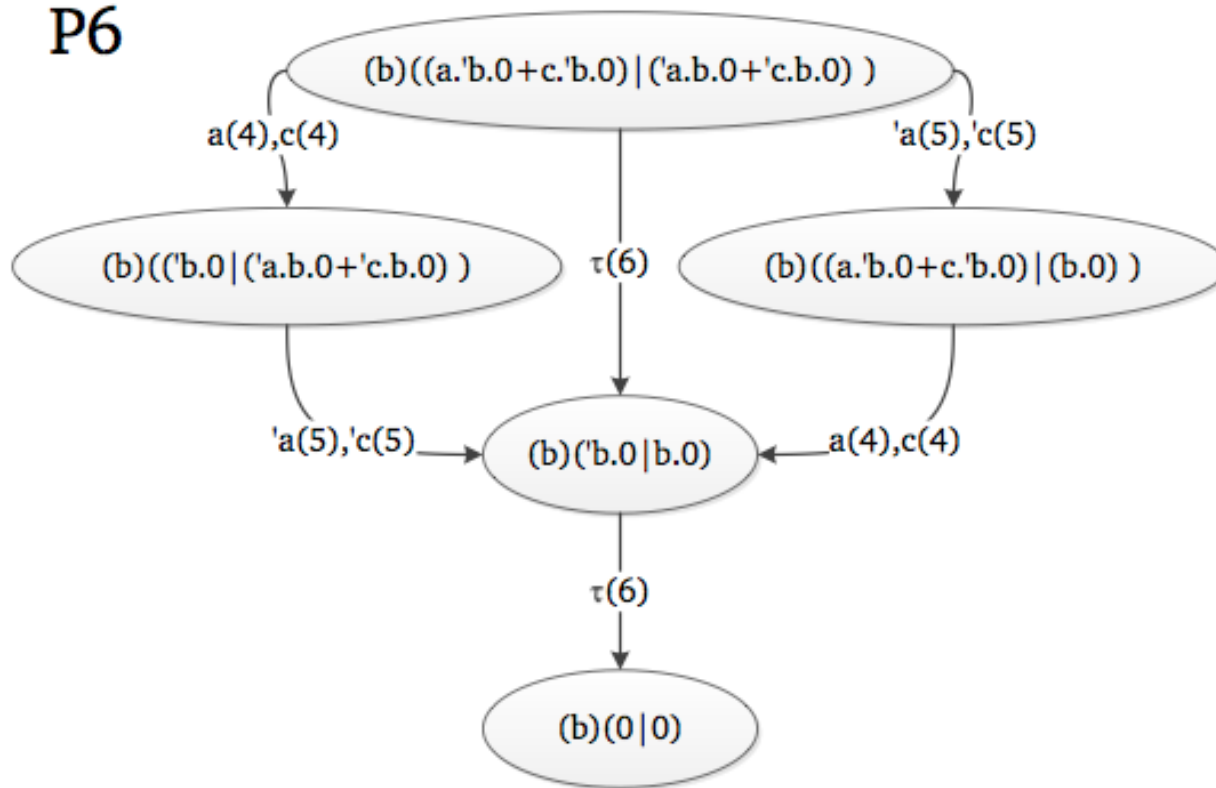
- P_7 := (a, d)(c.d.0[c/d][d/a] | 'a.0 | 'b.0)

P7

# Weak Bisimulation

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

$$P := a.(b.P + c.P)$$

$$Q := a.(\tau.b.Q + \tau.c.Q)$$

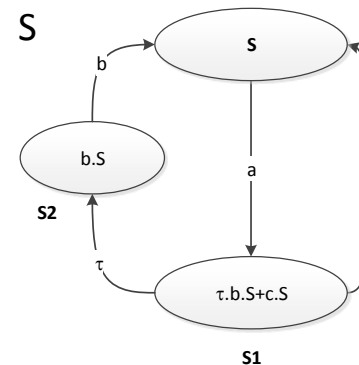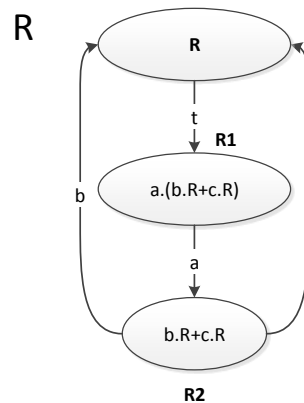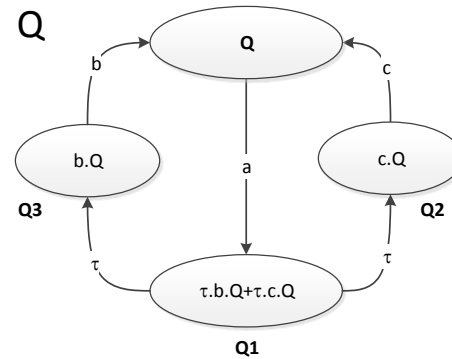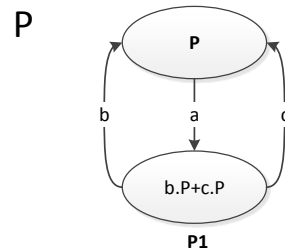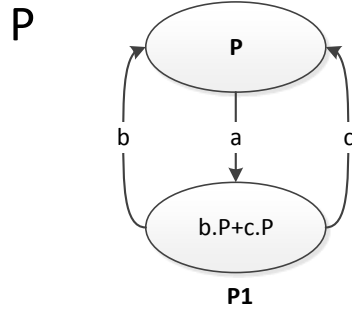$$R := \tau.a.(b.R + c.R)$$

$$S := a.(\tau.b.S + c.S)$$

$$P := a.(b.P + c.P)$$

$$Q := a.(\tau.b.Q + \tau.c.Q)$$

$$R := \tau.a.(b.R + c.R)$$

$$S := a.(\tau.b.S + c.S)$$

P

Q

P1

Q3

Q2

Q1

$P \approx R, R=\{(P, R), (P, R1), (P1, R2)\}$

R

S

R1

S2

R2

S1

# Switch
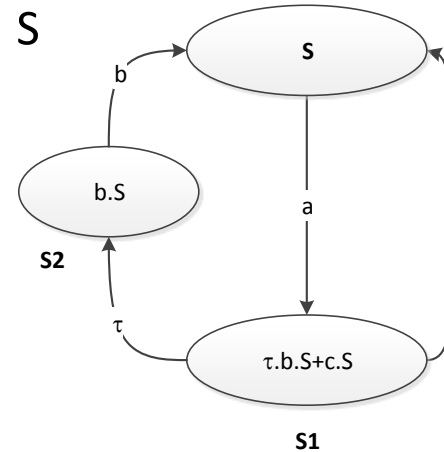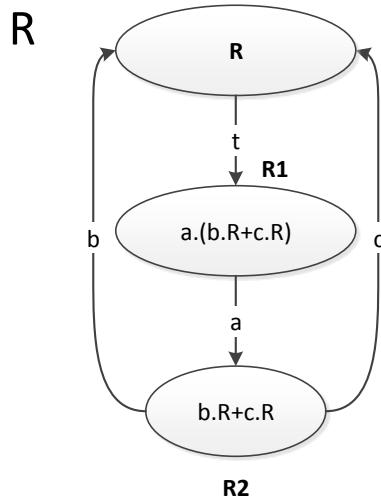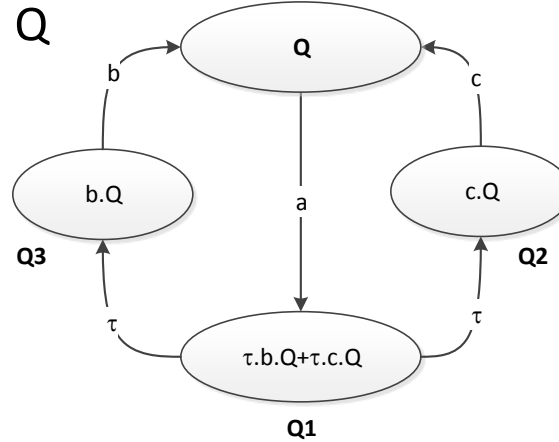
- Initially, every incoming *a* is channeled through. After an *s* is received, the following *a'*s are discarded.

- S1 := a.'a.S1 + s.S2

- S2 := a. S2

- 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.

- S1:= a. S1 + s.S2

- S2:= a.'b.S2 + s.S1

# Switch

- 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.


- S1 := a.'b.S1 + s.S2
- S2 := a.'a.S3 +  s.S2
- S3 := a.'a.S1 + s.S2