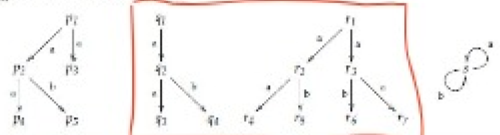
3.5 Definition (Alternative Definition für Bisimulation)
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} \subseteq \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *(starke) Bisimulation*,
wenn sowohl $\mathcal{R}$ als auch $\mathcal{R}^{-1}$ eine (starke) Simulation ist.

$R^{-1} = \{(q_1, p_1), (q_2, p_2),$ ist keine starke Simulation, da $q_2 \xrightarrow{a} q_3$ und $(q_2, p_3) \in R^{-1}$

$p_3 \xrightarrow{a}$ gilt nicht

$\Rightarrow p_1 \not\sim q_1$ und $p_1 \lesssim q_1$, $q_1 \gtrsim p_1$

Aufgabe 2: Starke Bisimulation I



Gib für die folgenden Paare an, ob sie stark bisimilar sind, sich gegenseitig stark simulieren und/oder einer den anderen stark simuliert. Gib die jeweilige Relation an.
Beweise exemplarisch für Aufgabe 2.b) die angegebene Eigenschaft.

2.a) p_1 und q_1
2.b) q_1 und r_1
2.c) p_1 und r_1
2.d) p_1 und s
2.e) q_1 und s
2.f) r_1 und s

$\mathcal{R} = \{(p_1, q_1), (p_2, q_2), (p_3, q_2), (p_4, q_3), (p_5, q_4)\}$

$p_1 \xrightarrow{a} p_2$
$q_1 \xrightarrow{a} q_2$

R ist starke Simulation

$\Rightarrow p_1$ wird von q_1 stark simuliert, d.h. $p_1 \lesssim q_1$

Starke Bi-/Simulation

3.4 Definition (Starke Simulation) [6]
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} \subseteq \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *(starke) Simulation*, wenn gilt:
für alle $(s_1, s_2) \in \mathcal{R}$,
für alle $\alpha \in \mathsf{Act}$,
für alle $s_1' \in \mathsf{Proc}$
wenn $s_1 \xrightarrow{\alpha} s_1'$,
dann gibt es s_2' mit $s_2 \xrightarrow{\alpha} s_2'$ mit $(s_1', s_2') \in \mathcal{R}$.
Seien $s_1, s_2 \in \mathsf{Proc}$.

- s_1 *wird von* s_2 *(stark) simuliert*, geschrieben $s_1 \lesssim s_2$, falls es eine (starke) Simulation $\mathcal{R}$ gibt mit $(s_1, s_2) \in \mathcal{R}$.
- s_1 *und* s_2 *simulieren sich (stark) gegenseitig*, geschrieben $s_1 \simeq s_2$, falls es zwei (starke) Simulationen $\mathcal{R}_1$ und ~~$\mathcal{R}_2$~~ $\mathcal{R}_1^{-1}$ gibt mit $(s_1, s_2) \in \mathcal{R}_1$ und $(s_2, s_1) \in \mathcal{R}_2$.

2b) $R_1 = \{(q_1, r_1), (q_2, r_2), (q_2, r_3), (q_3, r_7), (q_3, r_4), (q_4, r_5)\}$ $(q_3,$

$\Rightarrow q_1 \lesssim r_1$

$R_2 = \{(r_1, q_1), (r_2, q_2), (r_3, q_2), (r_4, q_3), (r_7, q_3), (r_5, q_4), (r_6, q_4)\}$

$\Rightarrow r_1 \lesssim q_1$

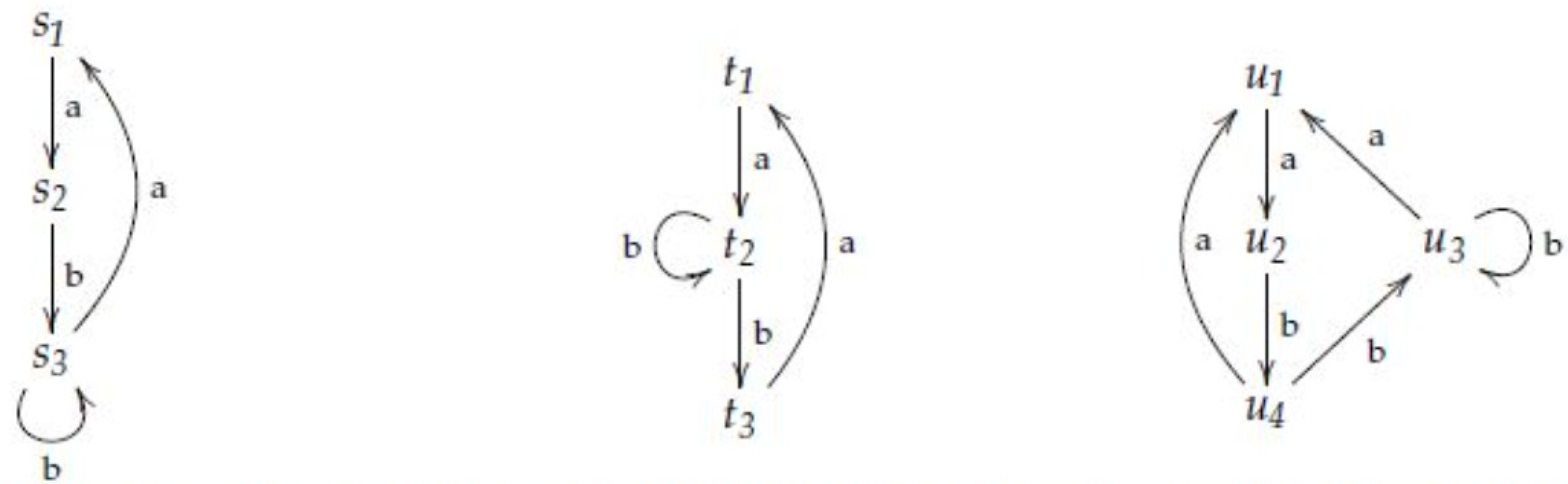
$R_1 \neq R_2^{-1}$

$\Rightarrow r_1 \simeq q_1$

R_2^{-1} ist ~~nicht~~/auch starke Simulation $\Rightarrow R_2$ starke Bisimulation

$\Rightarrow r_1 \sim q_1$

Aufgabe 1: Starkes Bisimulationspiel



Gib für die folgenden Paare an, ob sie stark bisimilar sind. Wenn sie nicht stark bisimilar sind, gib eine Gewinnstrategie für den Angriff an. Wenn sie stark bisimilar sind, gib die jeweilige Relation an.

1.a) $s_1 \sim t_1$

Angriff	Verteidiger
$s_1 \xrightarrow{a} s_2$	$t_1 \xrightarrow{a} t_2$
$t_2 \xrightarrow{b} t_2$	$s_2 \xrightarrow{b} s_3$
$s_3 \xrightarrow{a} s_1$	

b

$v_3 \to v_5$

$v_3 \xrightarrow{b}$

$u_2 \to$ v_3

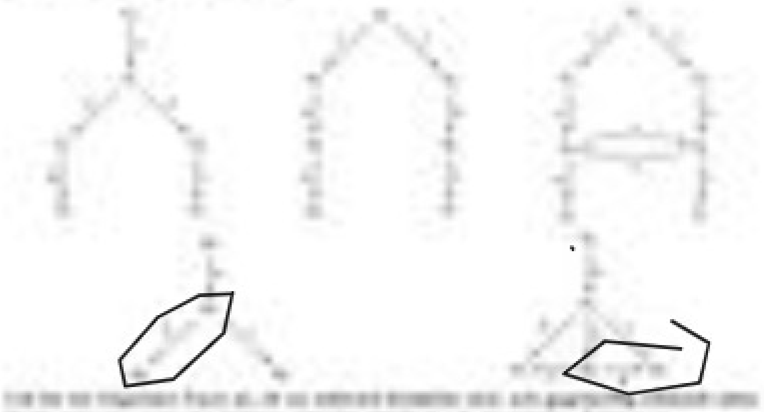
$\langle u_3, v_5 \rangle$

$v_4 \to$ v_3, u_4

2b)

$R_1 = \{ (u_1, v_1),$
$(u_2, v_2),$
$(u_3, v_3),$
$\Rightarrow (u_3, v_5),$
(u_4, v_5)
$\Rightarrow (u_4, v_3) \}$
(u_2, v_4)
$(\ , \ )$

R_1 ist schwache Bisimulation $u_1 \approx v_1$



$R_1 = \{ (r_1, s_1), (r_2, s_2), (r_2, s_3)$
$(r_3, s_4),$
$(r_4, s_5),$
$(r_5, s_6),$
$(r_6, s_7) \}$ ...

R_1 ist schwache Bisim...
$\Rightarrow r_1 \approx s_1$

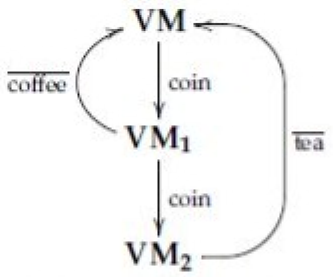
Seien $s_1, s_2 \in$ Proc.

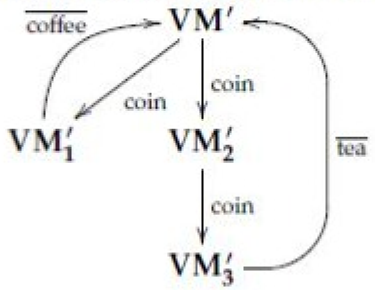
- s_1 *und* s_2 *sind schwach bisimilar*, geschrieben $s_1 \approx s_2$, falls es eine schwache Bisimulation $\mathcal{R}$ gibt mit $(s_1, s_2) \in \mathcal{R}$.

Die Relation $\approx$ heißt *schwache Bisimulationsäquivalenz* bzw. *schwache Bisimilarität*; $\approx$ heißt auch *Beobachtungsäquivalenz*.

Aufgabe 1: Trace Equivalence

Gegeben seien die folgenden zwei Lösungen zum Getränkeautomaten aus dem letzten Tutorium.





1.a) Gib Traces(**VM**) und Traces(**VM′**) an.

Traces (VM) =

$$\left(\begin{array}{l} \text{coin} \cdot \overline{\text{coffee}} \;\Big|\; \\ \text{coin} \cdot \text{coin} \cdot \overline{\text{tea}} \end{array} \right)^*$$

$(\varepsilon \mid \text{coin} \mid \text{coin coin})$

Traces (VM)

Spuren

3.2 Definition (Traces)

Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.

Seien

- $k \in \mathbb{N}$
- $s_0, \ldots, s_k \in \mathsf{Proc}$
- $\alpha_1, \ldots, \alpha_k \in \mathsf{Act}$

so dass für alle $1 \leqslant j \leqslant k$ eine Transition $s_{j-1} \xrightarrow{\alpha_j} s_j$ existiert.

Dann heißt $\alpha_1 \cdot \ldots \cdot \alpha_k \in \mathsf{Act}^*$ eine *Spur* bzw. ein *Trace* in von s_0 in T.

$$\mathrm{Traces}(s) \triangleq \{\, w \in \mathsf{Act}^* \mid w \text{ ist Trace von } s \text{ in } T \,\}$$

$\alpha . P$