Theorie kommunizierender Systeme Sequentielle Prozesse und Bisimulation

Ehsan Gholamsaghaee
Betreuer: Andreas Rossberg

[1]Department 6.2 - Computer Science

[2]Fakultät für Naturwissenschaft und Technologie I

4. Oktober 2004

Ziel

Beschreibung von sequentiellen Prozessausdrücken und dazu gehörigen Äquivalenzrelation

Labelled Transition Systems

Sequentielle Prozessausdrücke können durch beschriftetes Transitionssystem (*LTS* Labelled Transition System) beschrieben werden.

$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, \ldots$

$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, \ldots$
$\overline{\mathcal{N}}$	co-Namen	$\{\overline{n} \mid n \in \mathcal{N}\}$

$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, \ldots$
$\overline{\mathcal{N}}$	co-Namen	$\{\overline{n} \mid n \in \mathcal{N}\}$
$\mathcal{L}$	Labels	$\mathcal{N} \bigcup \overline{\mathcal{N}}$

$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, \ldots$
$\overline{\mathcal{N}}$	co-Namen	$\{\overline{n} \mid n \in \mathcal{N}\}$
$\mathcal{L}$	Labels	$\mathcal{N} \bigcup \overline{\mathcal{N}}$
Act	Actionen	$Act = \mathcal{L}$

$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, \ldots$
$\overline{\mathcal{N}}$	co-Namen	$\{\overline{n} \mid n \in \mathcal{N}\}$
$\mathcal{L}$	Labels	$\mathcal{N} \bigcup \overline{\mathcal{N}}$
Act	Actionen	$Act = \mathcal{L}$

Definition *LTS*

Ein *LTS* über *Act* besteht aus

- $\mathcal{Q}$ Menge der Zustände;
- $\mathcal{T} \subseteq (\mathcal{Q} \times Act \times \mathcal{Q})$ ternäre Relation

$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, \ldots$
$\overline{\mathcal{N}}$	co-Namen	$\{\overline{n} \mid n \in \mathcal{N}\}$
$\mathcal{L}$	Labels	$\mathcal{N} \bigcup \overline{\mathcal{N}}$
Act	Actionen	$Act = \mathcal{L}$

Definition LTS

Ein LTS über Act besteht aus

- $\mathcal{Q}$ Menge der Zustände;
- $\mathcal{T} \subseteq (\mathcal{Q} \times Act \times \mathcal{Q})$ ternäre Relation

Für $(q,\ \alpha,\ q') \in\ \mathcal{T}$, schreiben wir $q \xrightarrow{\alpha} q'$

$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, \ldots$
$\overline{\mathcal{N}}$	co-Namen	$\{\overline{n} \mid n \in \mathcal{N}\}$
$\mathcal{L}$	Labels	$\mathcal{N} \bigcup \overline{\mathcal{N}}$
Act	Actionen	$Act = \mathcal{L}$

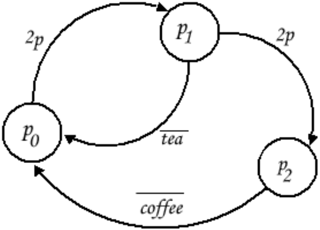
Definition *LTS*

Ein *LTS* über *Act* besteht aus

- $\mathcal{Q}$ Menge der Zustände;
- $\mathcal{T} \subseteq (\mathcal{Q} \times Act \times \mathcal{Q})$ ternäre Relation

Für $(q,\ \alpha,\ q') \in\ \mathcal{T}$, schreiben wir $q \xrightarrow{\alpha} q'$
... ein „endlicher Automat ohne Anfangs- und Endzustände“

Beispiel

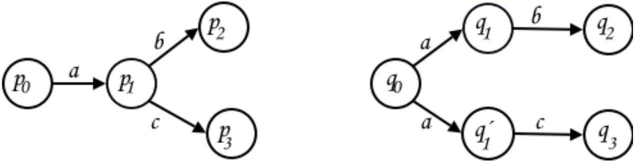


$\mathcal{Q} = \{p_0, p_1, p_2\}$
$\mathcal{T} = \{(p_0, 2p, p_1), (p_1, \overline{tea}, p_0), (p_1, 2p, p_2), (p_2, \overline{coffee}, p_0)\}$

Starke Simulation

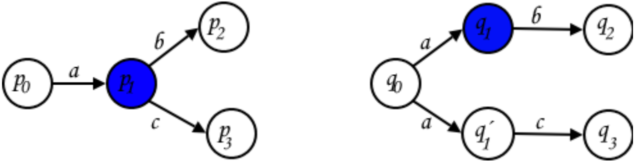
Wann sollen zwei Zustände in einem *LTS* als äquivalent betrachtet werden?

Beispiel



Der zentrale Unterschied:

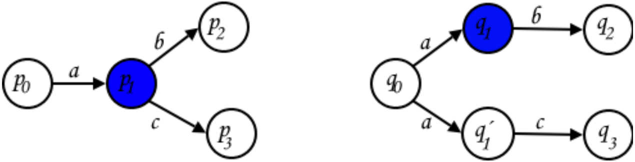
Beispiel



Der zentrale Unterschied:

Die Übergänge, die q_1 kann, kann p_1 auch
Die Übergänge, die p_1 kann, kann q_1 nicht
und auch sonst kein q_i

Beispiel



Unterscheidbar durch Simulation
p_1 simuliert q_1, aber q_1 simuliert p_1 nicht

Definition starke Simulation

$(\mathcal{Q}, \mathcal{T})$ ein *LTS*
$S \subseteq (\mathcal{Q} \times \mathcal{Q})$
S ist eine starke Simulation über $(\mathcal{Q}, \mathcal{T})$, wenn für alle pSq gilt:

$$p \xrightarrow{\alpha} p' \; \Rightarrow \exists q' \in \mathcal{Q} : \; q \xrightarrow{\alpha} q' \wedge p' S q'$$

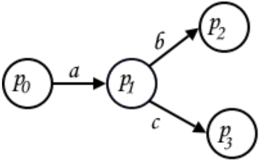
Definition starke Simulation

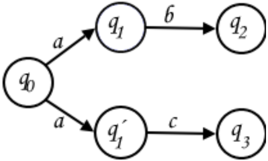
$(\mathcal{Q}, \mathcal{T})$ ein LTS
$S \subseteq (\mathcal{Q} \times \mathcal{Q})$
S ist eine starke Simulation über $(\mathcal{Q}, \mathcal{T})$, wenn für alle pSq gilt:

$$p \xrightarrow{\alpha} p' \ \Rightarrow \exists q' \in \mathcal{Q} : \ q \xrightarrow{\alpha} q' \wedge p' S q'$$

q simuliert p stark, wenn es eine starke Simulation S gibt mit pSq.

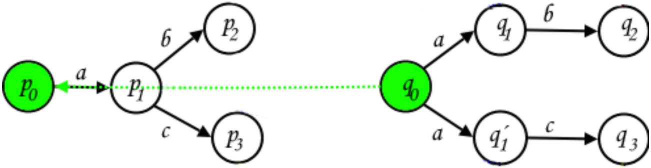
Beispiel





Fortsetzung des obiges Beispiels:
Behauptung: p_0 simuliert stark q_0

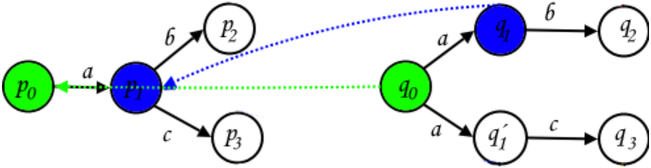
Beispiel



Fortsetzung des obiges Beispiels:
Behauptung: p_0 simuliert stark q_0

$$S = \{(q_0, p_0)$$

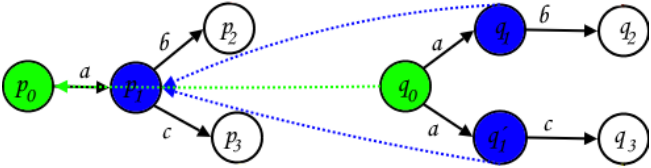
Beispiel



Fortsetzung des obiges Beispiels:
Behauptung: p_0 simuliert stark q_0

$$S = \{(q_0, p_0), (q_1, p_1)$$

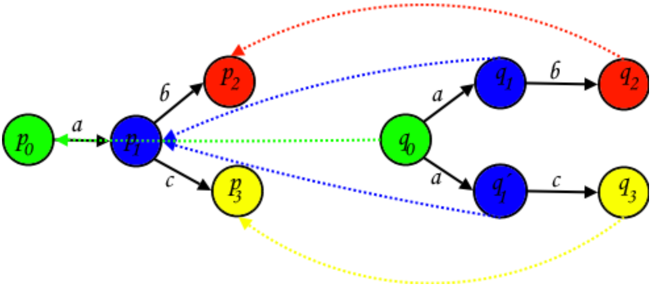
Beispiel



Fortsetzung des obiges Beispiels:
Behauptung: p_0 simuliert stark q_0

$$S \;=\; \{(q_0, p_0), (q_1, p_1), (q_1', p_1)$$

Beispiel



Fortsetzung des obiges Beispiels:
Behauptung: p_0 simuliert stark q_0

$$S = \{(q_0, p_0), (q_1, p_1), (q_1', p_1), (q_2, p_2), (q_3, p_3)\}$$

Starke Bisimulation

Mit Simulation haben wir ein Kriterium gefunden, um zwischen unseren beiden Automaten zu unterscheiden. Wir sollen jetzt eine Äquivalentzrelation definieren, in der unsere zwei Getränkeautomaten nicht äquivalent sind.

Definition

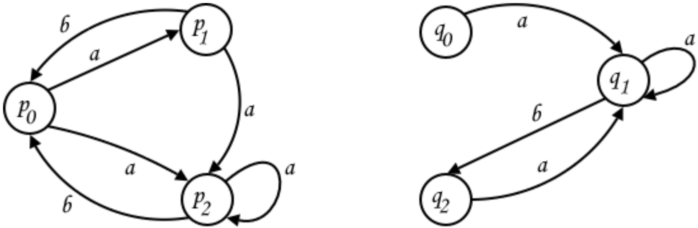
$(\mathcal{Q}, \mathcal{T})$ ein *LTS*
$S \subseteq (\mathcal{Q} \times \mathcal{Q})$
S ist eine starke Bisimulation, wenn S und S^{-1} starke Simulationen sind.

Definition

$(\mathcal{Q}, \mathcal{T})$ ein *LTS*
$S \subseteq (\mathcal{Q} \times \mathcal{Q})$
S ist eine starke Bisimulation, wenn S und S^{-1} starke Simulationen sind.

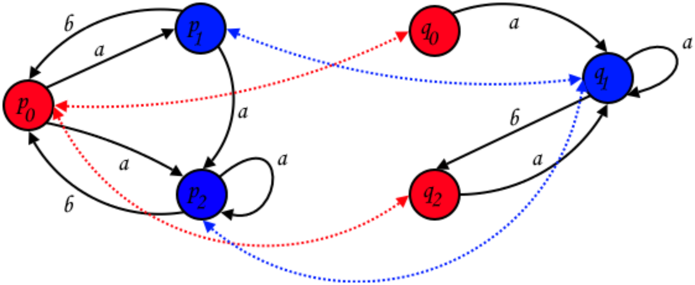
p und q sind stark bisimular, wenn es eine starke Bisimulation S gibt mit pSq.
Schreibweise: $p \sim q$

Beispiel



Behauptung: $p_0 \sim q_0$

Beispiel



Behauptung: $p_0 \sim q_0$

$$S = \{(p_0, q_0), (p_0, q_2), (p_1, q_1), (p_2, q_1)\}$$

Proposition

$$\sim \textit{ ist eine Äquivalenzrelation}$$

Proposition

$$\sim \ \textit{ist eine Äquivalenzrelation}$$

d.h.

1. $p \sim p$	(reflexiv)
2. $p \sim q \Rightarrow q \sim p$	(symmetrisch)
3. $p \sim q \wedge q \sim r \Rightarrow p \sim r$	(transitiv)

Proposition

$$\sim \ \ \textit{ist eine Äquivalenzrelation}$$

d.h.

$1. p \sim p$	(reflexiv)
$2. p \sim q \Rightarrow q \sim p$	(symmetrisch)
$3. p \sim q \wedge q \sim r \Rightarrow p \sim r$	(transitiv)

Bew

1.zeige: $\{(q, q) | q \in \mathcal{Q}\}$ ist eine Bisimulation.
ist trivial.

Proposition

$$\sim \quad \textit{ist eine Äquivalenzrelation}$$

d.h.

1.$p \sim p$ (reflexiv)
2.$p \sim q \Rightarrow q \sim p$ (symmetrisch)
3.$p \sim q \wedge q \sim r \Rightarrow p \sim r$ (transitiv)

Bew

1.zeige: $\{(q,q) | q \in \mathcal{Q}\}$ ist eine Bisimulation.
ist trivial.
2.wegen Def. von Bisimulation

3.wenn $p \sim q$ und $q \sim r$ dann $p \sim r$

Bew

S_1, S_2 Bisimulationen

Zeige: $S := \{(p, r) | \exists q.\ pS_1q \ und \ qS_2r\}$ ist auch eine Bisimulation.

3.wenn $p \sim q$ und $q \sim r$ dann $p \sim r$

Bew
S_1, S_2 Bisimulationen
Zeige: $S := \{(p, r) | \exists q.\ pS_1q\ und\ qS_2r\}$ ist auch eine Bisimulation.

Reicht zu zeigen: S ist Simulation (S^{-1} folgt analog).

3.wenn $p \sim q$ und $q \sim r$ dann $p \sim r$

Bew
S_1, S_2 Bisimulationen
Zeige: $S := \{(p, r) | \exists q.\ pS_1q\ und\ qS_2r\}$ ist auch eine Bisimulation.

Reicht zu zeigen: S ist Simulation (S^{-1} folgt analog).
Sei also $(p, r) \in S$ und $p \xrightarrow{\alpha} p'$.

3.wenn $p \sim q$ und $q \sim r$ dann $p \sim r$

Bew

S_1, S_2 Bisimulationen
Zeige: $S := \{(p,r)|\exists q.\ pS_1q\ und\ qS_2r\}$ ist auch eine Bisimulation.

Reicht zu zeigen: S ist Simulation (S^{-1} folgt analog).
Sei also $(p,r) \in S$ und $p \xrightarrow{\alpha} p'$.
$\exists q : pS_1q\ und\ qS_2r$; Def. S

3.wenn $p \sim q$ und $q \sim r$ dann $p \sim r$

Bew

S_1, S_2 Bisimulationen
Zeige: $S := \{(p, r) | \exists q.\ pS_1q \textit{ und } qS_2r\}$ ist auch eine Bisimulation.

Reicht zu zeigen: S ist Simulation (S^{-1} folgt analog).
Sei also $(p, r) \in S$ und $p \xrightarrow{\alpha} p'$.

$\exists q : pS_1q \textit{ und } qS_2r;$ Def. S

und $\exists q' : q \xrightarrow{\alpha} q' \textit{ und } p'S_1q';$ Def. S_1

3.wenn $p \sim q$ und $q \sim r$ dann $p \sim r$

Bew

S_1, S_2 Bisimulationen
Zeige: $S := \{(p, r) | \exists q.\ pS_1q\ und\ qS_2r\}$ ist auch eine Bisimulation.

Reicht zu zeigen: S ist Simulation (S^{-1} folgt analog).
Sei also $(p, r) \in S$ und $p \xrightarrow{\alpha} p'$.

	$\exists q : pS_1q\ und\ qS_2r;$	Def. S
und	$\exists q' : q \xrightarrow{\alpha} q'\ und\ p'S_1q';$	Def. S_1
und	$\exists r' : r \xrightarrow{\alpha} r'\ und\ q'S_2r'.$	Def. S_2

3.wenn $p \sim q$ und $q \sim r$ dann $p \sim r$

Bew

S_1, S_2 Bisimulationen
Zeige: $S := \{(p, r) | \exists q.\ pS_1q \ und \ qS_2r\}$ ist auch eine Bisimulation.

Reicht zu zeigen: S ist Simulation (S^{-1} folgt analog).
Sei also $(p, r) \in S$ und $p \xrightarrow{\alpha} p'$.

	$\exists q : pS_1q \ und \ qS_2r;$	Def. S
und	$\exists q' : q \xrightarrow{\alpha} q' \ und \ p'S_1q';$	Def. S_1
und	$\exists r' : r \xrightarrow{\alpha} r' \ und \ q'S_2r'.$	Def. S_2

Damit: $(p', r') \in S$. Def. S

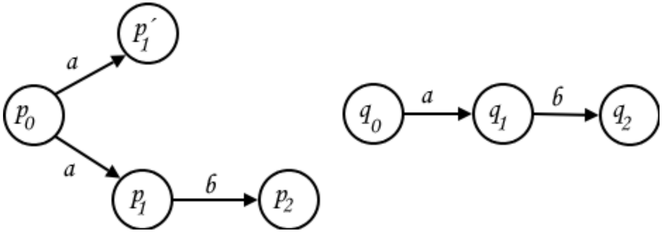
Bemerkung

$p \sim q$ **heisst nicht** „p simuliert stark q und q simuliert stark p “.

Bemerkung

$p \sim q$ **heisst nicht** „p simuliert stark q und q simuliert stark p “.

Beispiel

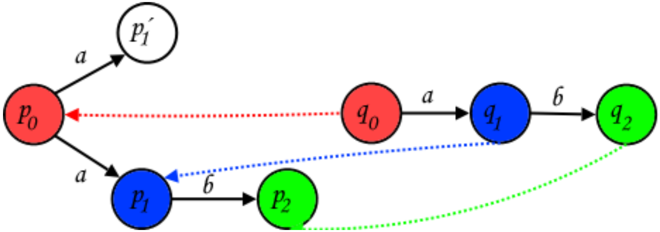


Beh.: p_0 simuliert stark q_0 und umgekehrt; aber: $p_0 \nsim q_0$

Bemerkung

$p \sim q$ **heisst nicht** „p simuliert stark q und q simuliert stark p “.

Beispiel

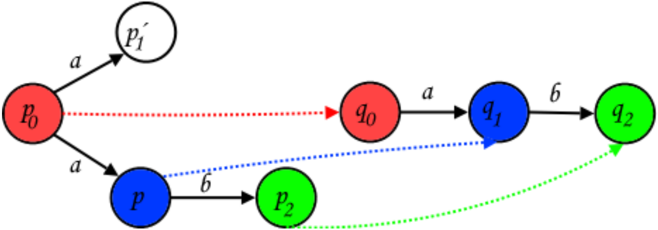


Mit S gilt:
p_0 simuliert stark q_0

Bemerkung

$p \sim q$ **heisst nicht** „p simuliert stark q und q simuliert stark p “.

Beispiel

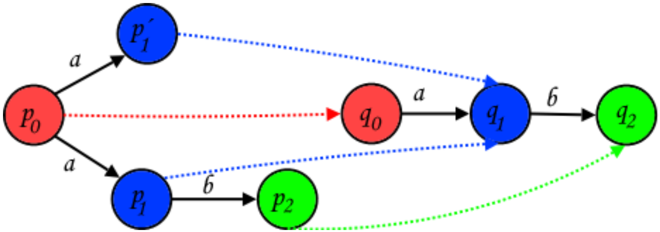


Mit S^{-1} gilt nicht:
q_0 simuliert stark p_0
Also: S ist keine Bisimulation

Bemerkung

$p \sim q$ **heisst nicht** „p simuliert stark q und q simuliert stark p “.

Beispiel

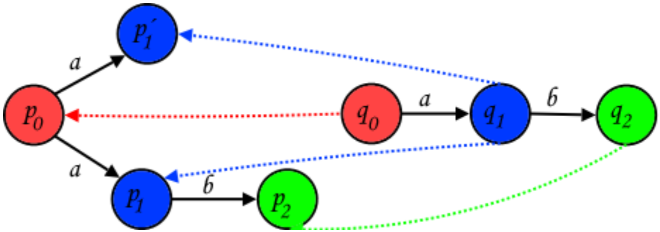


Mit S' gilt:
q_0 simuliert stark p_0

Bemerkung

$p \sim q$ **heisst nicht** „p simuliert stark q und q simuliert stark p “.

Beispiel



Mit S'^{-1} gilt nicht:
p_0 simuliert stark q_0
Also: S' ist auch keine Bisimulation

Beispiel

S starke Simulation mit: $p_0 S q_0$.
Da $p_0 \xrightarrow{a} p_1'$, muss es ein q_i geben nämlich q_1 mit $q_0 \xrightarrow{a} q_1$ und $p_1' S q_1$.
So gilt $q_1 S^{-1} p_1'$.
Aber für den Übergang $q_1 \xrightarrow{b} q_2$ gibt es keinen passenden Übergang von p_1'.
Daraus folgt die zweite Behauptung.

Sequentielle Prozessausdrücke

hilfreich...
... jeden Zustand eines Systems durch ein Prozessausdruck darzustellen

Prozessausdrücke...
... tragen Informationen über das Verhalten und die Struktur eines Systems, die im einfachen Fall sequentieller Prozessausdrücke die mögliche Übergänge eines Zustandes anzeigen.

Id	Prozessbezeichner	$A, B, Buffer, \ldots$

Id	Prozessbezeichner	$A, B, Buffer, \ldots$
$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, \ldots$
$\overline{\mathcal{N}}$	co-Namen	$\{\overline{n} \mid n \in \mathcal{N}\}$
$\mathcal{L}$	Labels	$\mathcal{N} \bigcup \overline{\mathcal{N}}$
Act	Actionen	$Act = \mathcal{L}$

Id	Prozessbezeichner	$A, B, Buffer, ...$
$\mathcal{N}$	Namen	$a, b, c, tea, 2p, coffee, ...$
$\overline{\mathcal{N}}$	co-Namen	$\{\overline{n} \mid n \in \mathcal{N}\}$
$\mathcal{L}$	Labels	$\mathcal{N} \bigcup \overline{\mathcal{N}}$
Act	Actionen	$Act = \mathcal{L}$

Definition

Die Menge $\mathcal{P}^{seq}$ wird wie folgt definiert:

$$P ::= A\langle a_1, ..., a_n\rangle \mid \sum_{i \in I} \alpha_i.P_i$$

- Jeder Prozessbezeichner A hat folgende Gleichungsdefinition:

$$A(\vec{a}) =_{def} P_A$$

  P_A ist eine Summe, $\vec{a}$ enthält $fn(P_A)$.
- $A\langle\vec{b}\rangle$ bedeutet das Gleiche wie $\{\vec{b}/\vec{a}\}P$

Definition **Strukturelle Kongruenz**

Zwei seq. Prozessausdrücke P und Q sind strukturell kongruent ($P \equiv Q$), wenn sie durch Substitution ineinander umwandelbar sind.

Definition Strukturelle Kongruenz

Zwei seq. Prozessausdrücke P und Q sind strukturell kongruent ($P \equiv Q$), wenn sie durch Substitution ineinander umwandelbar sind.

Beispiel

Sei

$$B(c, d) \;=_{def}\; c.d.0$$

dann gilt:

$$B\langle a, a\rangle \;\equiv\; a.a.0$$

Ein *LTS*, das alle sequentielle Prozessausdrücke enthält, definieren wir wie folgt:

Ein *LTS*, das alle sequentielle Prozessausdrücke enthält, definieren wir wie folgt:

Definition *LTS* **von sequentiellen Prozessen**

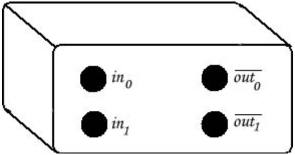
Ein *LTS* von sequentiellen Prozessen über einer Menge *Act* hat die Menge $\mathcal{P}^{seq}$ als Zuständen und folgende Übergänge:

$$wenn\ P \equiv \sum_{i \in I} \alpha_i.P_i,\ dann\ gibt\ es\ f\ddot{u}r\ alle\ j \in I, P \xrightarrow{\alpha_i} P_j.$$

Beispiele

Beispiele

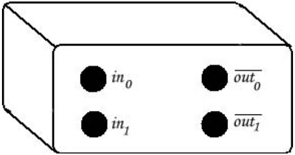
Boolescher Puffer



Buff

Beispiele

Boolescher Puffer

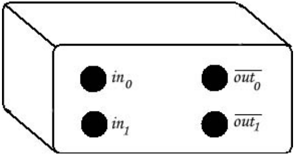


Buff

$\mathcal{N} := \{in_i, out_i | i \in \{0,1\}\}$

Beispiele

Boolescher Puffer

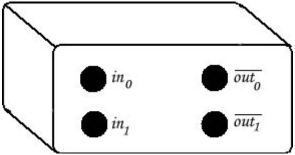


Buff

$\mathcal{N} := \{in_i, out_i | i \in \{0, 1\}\}$
$s \in \{0, 1, 00, 01, 10, 11\} \quad (Buff_s)$

Beispiele

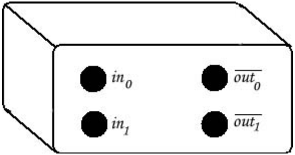
Boolescher Puffer



Buff

$\mathcal{N} := \{in_i, out_i | i \in \{0, 1\}\}$
$s \in \{0, 1, 00, 01, 10, 11\} \quad (Buff_s)$
Prozessbezeichner: $Buff$, $Buff_0$, $Buff_1$, $Buff_{00}$, ...

Boolescher Puffer

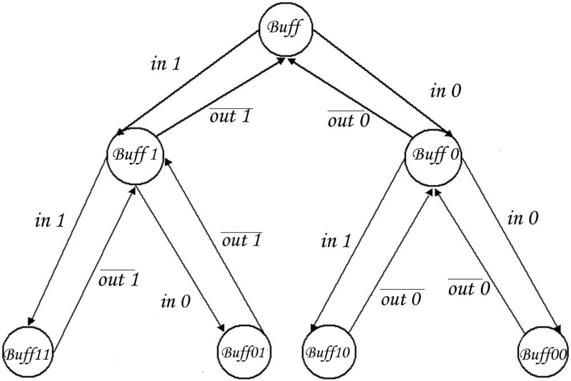


$$Buff =_{def} \sum_{i \in \{0,1\}} in_i . Buff_i$$

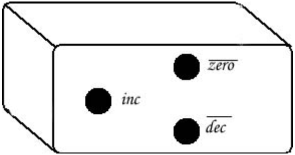
$$Buff_i =_{def} \overline{out_i} . Buff + \sum_{j \in \{0,1\}} in_j . Buff_{ji}$$

$$Buff_{ij} =_{def} \overline{out_j} . Buff_i$$

Boolescher Puffer

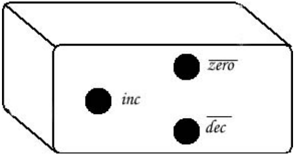


Zähler



Count

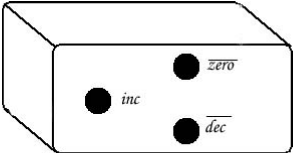
Zähler



Count

- Erhöhen durch Bestätigen von *inc* (immer möglich).

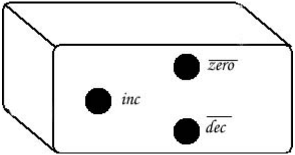
Zähler



Count

- Erhöhen durch Bestätigen von *inc* (immer möglich).
- Erniedrigen durch Bestätigen von $\overline{dec}$ (falls > 0).

Zähler



Count

- Erhöhen durch Bestätigen von *inc* (immer möglich).
- Erniedrigen durch Bestätigen von $\overline{dec}$ (falls > 0).
- Test auf 0 mit $\overline{zero}$.

Zähler

Zustände: $Count_n$ $(n \in \{0, 1, ...\})$

Zähler

Zustände: $Count_n$ $(n \in \{0, 1, ...\})$

$$Count_0 =_{def} inc.Count_1 + \overline{zero}.Count_0$$

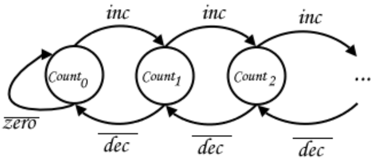
$$Count_{n+1} =_{def} inc.Count_{n+2} + \overline{dec}.Count_n$$

Zähler

Zustände: $Count_n$ $(n \in \{0, 1, ...\})$

$$Count_0 =_{def} inc.Count_1 + \overline{zero}.Count_0$$

$$Count_{n+1} =_{def} inc.Count_{n+2} + \overline{dec}.Count_n$$



Zusammenfassung

Zusammenfassung

- *LTS*

Zusammenfassung

- *LTS*
- starke Simulation

Zusammenfassung

- *LTS*
- starke Simulation
- starke Bisimulation $\quad p \sim q$

Zusammenfassung

- *LTS*
- starke Simulation
- starke Bisimulation $p \sim q$
- seq. Prozessausdrücke P^{seq}

Zusammenfassung

- *LTS*
- starke Simulation
- starke Bisimulation $p \sim q$
- seq. Prozessausdrücke P^{seq}
  - strukturelle Kongruenz $(P \equiv Q)$

Zusammenfassung

- *LTS*
- starke Simulation
- starke Bisimulation $p \sim q$
- seq. Prozessausdrücke P^{seq}
  - strukturelle Kongruenz $(P \equiv Q)$
  - *LTS* von seq. Proessausdrücke

Zusammenfassung

- *LTS*
- starke Simulation
- starke Bisimulation $p \sim q$
- seq. Prozessausdrücke P^{seq}
  - strukturelle Kongruenz $(P \equiv Q)$
  - *LTS* von seq. Proessausdrücke
- Beispiele