Kapitel 5: Prozessalgebra

Die Prozessalgebra wurde aus der Automatentheorie entwickelt, um nebenläufige und reaktive Prozesse und Systeme beschreiben, modellieren und verifizieren zu können. Dabei wurden wie bei endlichen Automaten Zustände festgelegt und für Aktionen oder Folgen von Aktionen spezifiziert, welches der Nachfolgezustand ist.

Die Prozessalgebra wurde aus der Automatentheorie entwickelt, um nebenläufige und reaktive Prozesse und Systeme beschreiben, modellieren und verifizieren zu können. Dabei wurden wie bei endlichen Automaten Zustände festgelegt und für Aktionen oder Folgen von Aktionen spezifiziert, welches der Nachfolgezustand ist.

Die Prozessalgebra wurde aus der Automatentheorie entwickelt, um nebenläufige und reaktive Prozesse und Systeme beschreiben, modellieren und verifizieren zu können. Dabei wurden wie bei endlichen Automaten Zustände festgelegt und für Aktionen oder Folgen von Aktionen spezifiziert, welches der Nachfolgezustand ist.

Die Prozessalgebra wurde aus der Automatentheorie entwickelt, um nebenläufige und reaktive Prozesse und Systeme beschreiben, modellieren und verifizieren zu können. Dabei wurden wie bei endlichen Automaten Zustände festgelegt und für Aktionen oder Folgen von Aktionen spezifiziert, welches der Nachfolgezustand ist.

Die Prozessalgebra wurde aus der Automatentheorie entwickelt, um nebenläufige und reaktive Prozesse und Systeme beschreiben, modellieren und verifizieren zu können. Dabei wurden wie bei endlichen Automaten Zustände festgelegt und für Aktionen oder Folgen von Aktionen spezifiziert, welches der Nachfolgezustand ist.

Die *elementare Prozessalgebra* entspricht der Modellierung eines einzigen Automaten. Im Unterschied zur traditionellen Automatentheorie wird aber eine algebraische Behandlung und der Aspekt der Beobachtbarkeit und Verhaltensäquivalenz reaktiver Systeme betont.

In der *Prozessalgebra* werden nebenläufige und kommunizierende Systeme durch mehrere Automaten beschrieben, die mittels Rendezvous-Synchronisation gekoppelt werden.

In der *Prozessalgebra* werden nebenläufige und kommunizierende Systeme durch mehrere Automaten beschrieben, die mittels Rendezvous-Synchronisation gekoppelt werden.

Prozessalgebra

Aktion
-orientiert

In der *Prozessalgebra* werden nebenläufige und kommunizierende Systeme durch mehrere Automaten beschrieben, die mittels Rendezvous-Synchronisation gekoppelt werden.

Prozessalgebra

Aktion
-orientiert

Harelgraphen
State-Charts

Zustands
-orientiert

In der *Prozessalgebra* werden nebenläufige und kommunizierende Systeme durch mehrere Automaten beschrieben, die mittels Rendezvous-Synchronisation gekoppelt werden.

Prozessalgebra	Petrinetze	Harelgraphen State-Charts
Aktion - orientiert		Zustands - orientiert

In der *Prozessalgebra* werden nebenläufige und kommunizierende Systeme durch mehrere Automaten beschrieben, die mittels Rendezvous-Synchronisation gekoppelt werden.

Prozessalgebra	Petrinetze	Harelgraphen State-Charts
Aktion - orientiert	Aktion/Zustands - orientiert	Zustands - orientiert

Dualität

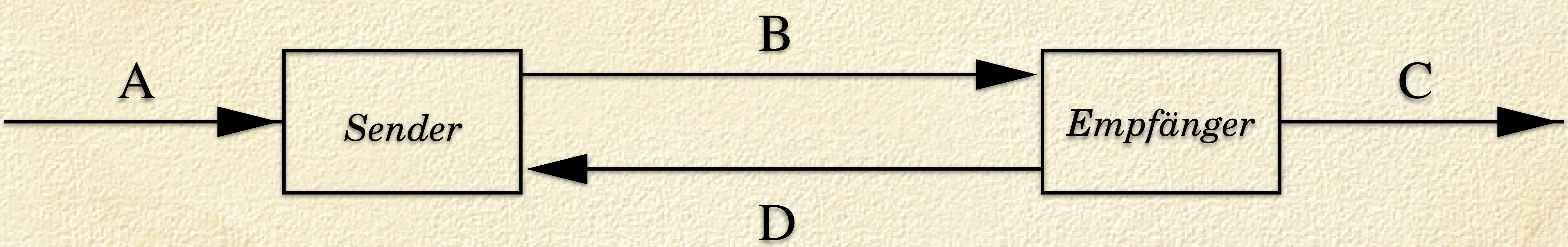
Literatur

die wichtigste Quelle:

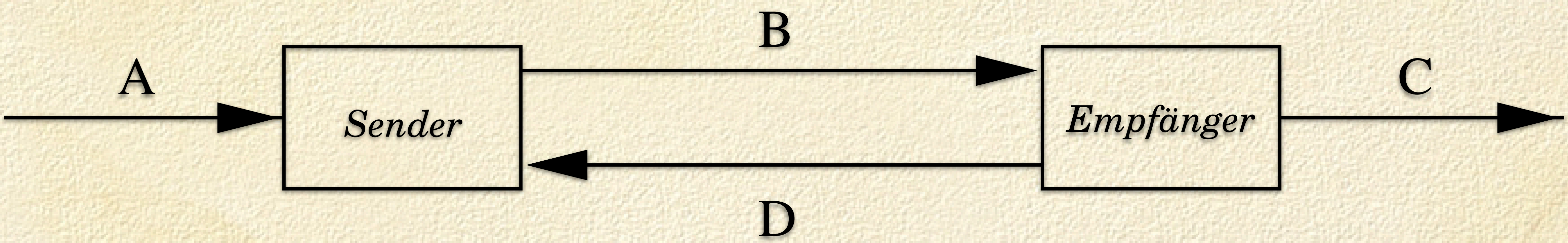
Wan Fokkink, *Introduction to Process Algebra*, Springer-Verlag, 1999.

R. Milner, *Communicating and Mobile Systems: the π-Calculus*, Cambridge University Press, 1999

Das Alternierbitprotokoll:



Das Alternierbitprotokoll:



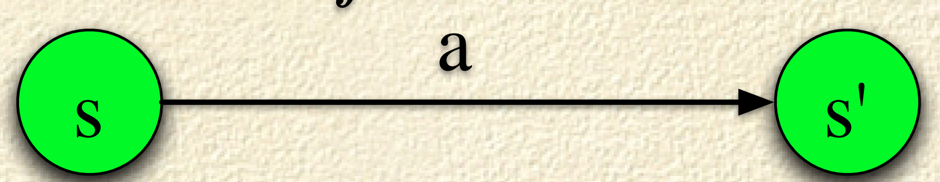
Beweis des Alternierbitprotokolls

Prozess-Graphen und elementare Prozessterme

Definition *Sei S eine Menge von Zuständen mit $\surd \in S$ und A eine Menge von (atomaren) Aktionen. Die Relation $tr \subseteq S \times A \times S$ heißt Transitionsrelation*

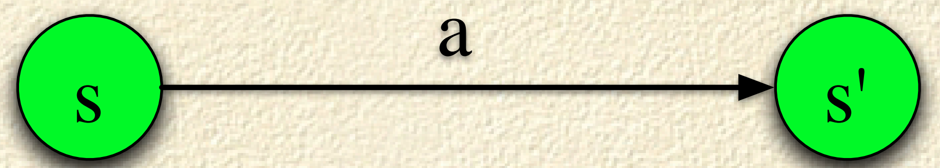
Prozess-Graphen und elementare Prozessterme

Definition *Sei S eine Menge von Zuständen mit $\surd \in S$ und A eine Menge von (atomaren) Aktionen. Die Relation $tr \subseteq S \times A \times S$ heißt* *Transitionsrelation*



Prozess-Graphen und elementare Prozessterme

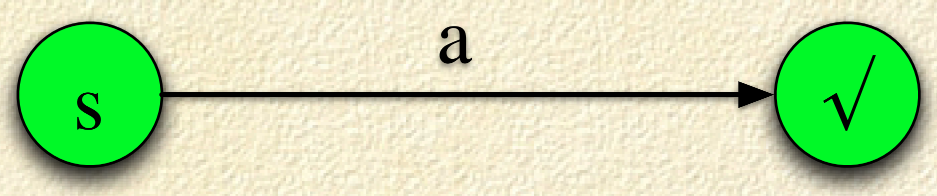
Definition *Sei S eine Menge von Zuständen mit $\surd \in S$ und A eine Menge von (atomaren) Aktionen. Die Relation $tr \subseteq S \times A \times S$ heißt Transitionsrelation*



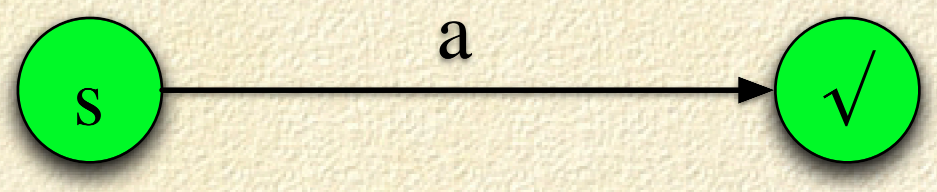
- *Eine Transition $(s, a, s') \in tr$ (geschrieben : $s \xrightarrow{a} s'$) drückt aus, dass der Zustand s durch die Aktion a in den Zustand s' wechseln kann.*

- *Eine Transition* $(s, a, \surd) \in tr$ *(geschrieben :* $s \xrightarrow{a} \surd$*) drückt aus, dass der Zustand s durch die Aktion a ordnungsgemäß terminieren kann.*

- *Eine Transition* $(s, a, \surd) \in tr$ *(geschrieben :* $s \xrightarrow{a} \surd$*) drückt aus, dass der Zustand s durch die Aktion a ordnungsgemäß terminieren kann.*



- *Eine Transition* $(s, a, \surd) \in tr$ *(geschrieben :* $s \xrightarrow{a} \surd$*) drückt aus, dass der Zustand* s *durch die Aktion* a *ordnungsgemäß terminieren kann.*



Ein Transitionssystem

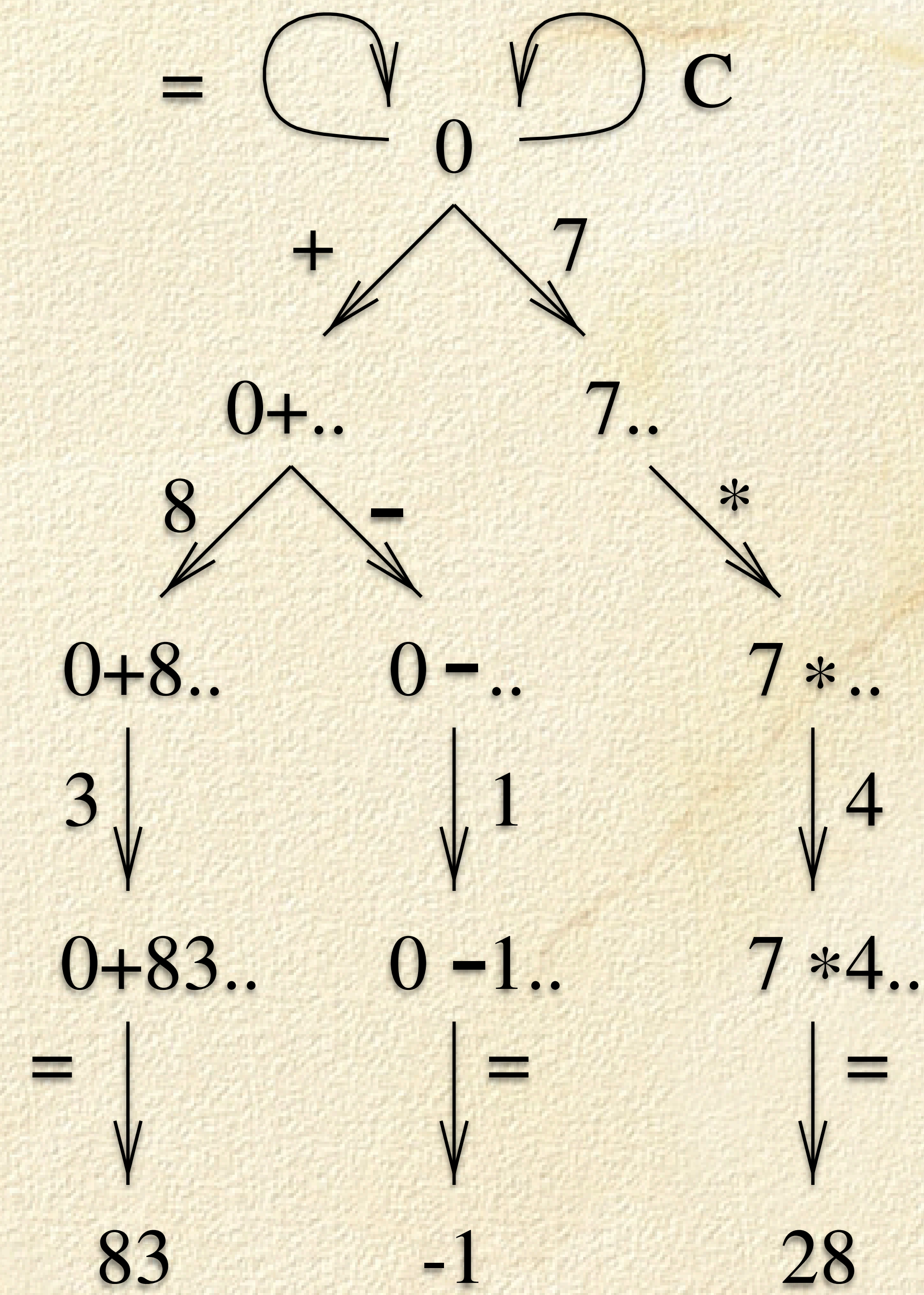
$$TS = (S, A, tr, s_0)$$

ist eine Menge von Transitionen, zusammen mit einem ausgezeichneten Zustand (Wurzelzustand, Anfangszustand).

Taschenrechner:

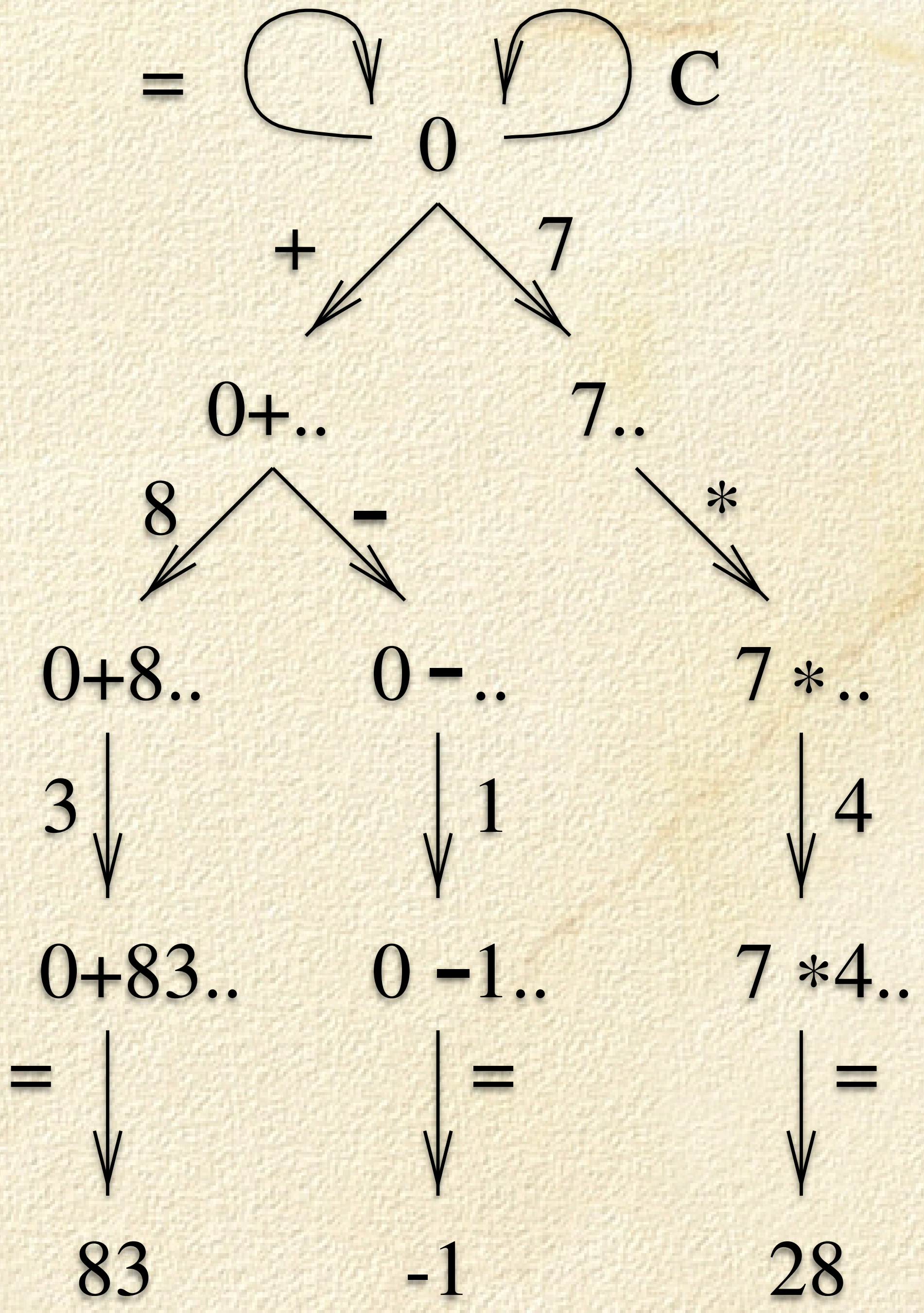
Taschenrechner:

Ausschnitt des Prozessgraphen des Taschenrechner(beispiel)s:



Taschenrechner:

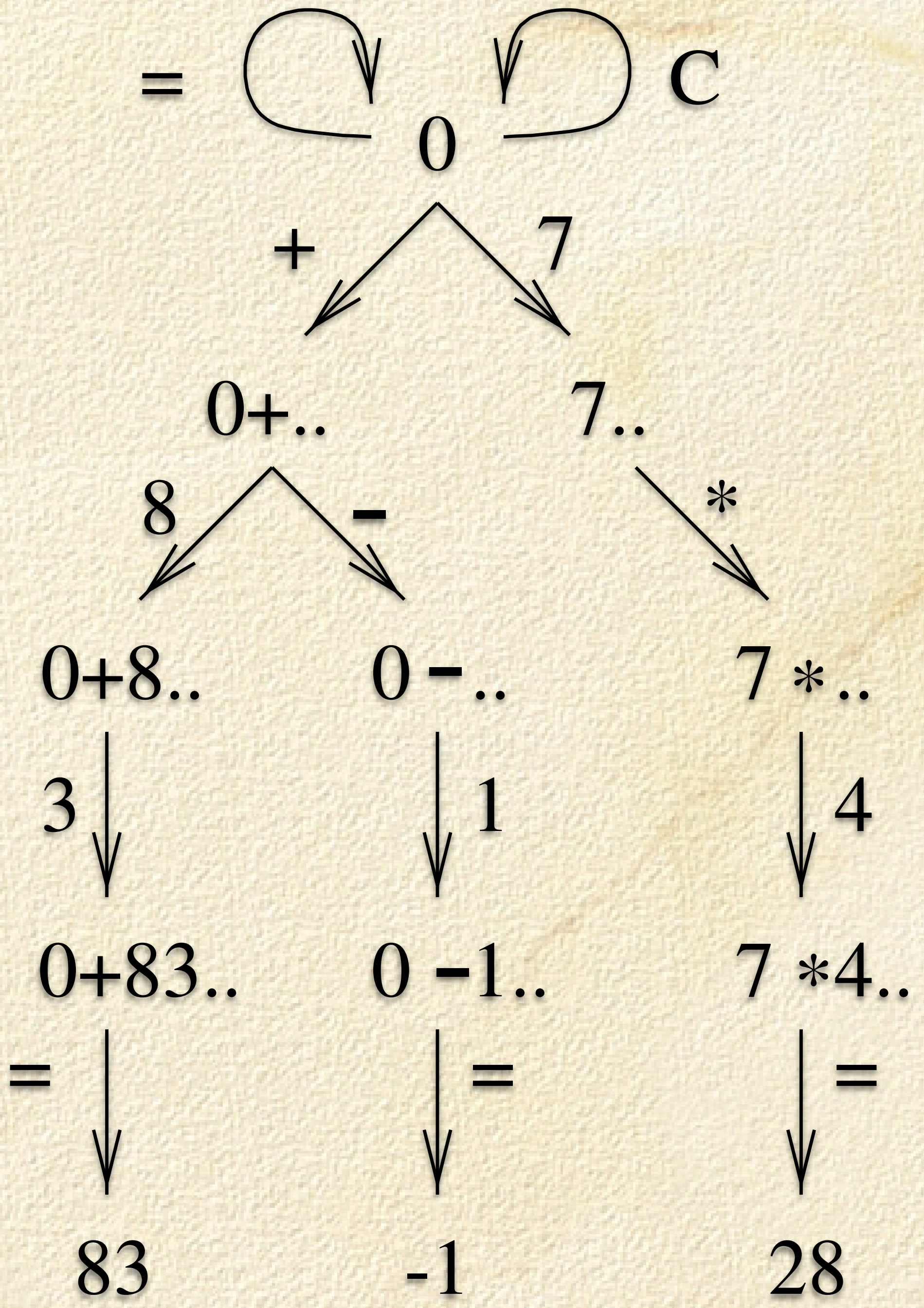
Ausschnitt des Prozessgraphen des Taschenrechner(beispiel)s:



hier: *Zustand beschreibt Vergangenheit*

Taschenrechner:

Ausschnitt des Prozessgraphen des Taschenrechner(beispiel)s:



hier: *Zustand beschreibt Vergangenheit*

in der Prozess-Algebra: *Zustand beschreibt Zukunft*

Elementare Prozess-Algebra-Terme (BPA)

(Basic Process Algebra Terms)

Definition 5.1 *Die Menge der BPA wird aus* atomaren Aktionen $a \in A$ *sowie der* Auswahl *und* Hintereinanderausführung *gebildet:*

Prozess-Ausdrücke

Elementare Prozess-Algebra-Terme (BPA)
(Basic Process Algebra Terms)

Definition 5.1 *Die Menge der BPA wird aus* atomaren Aktionen $a \in A$ *sowie der* Auswahl *und* Hintereinanderausführung *gebildet:*

- *Für jedes $a \in A$ ist ein BPA.*
  (**atomare Aktion** *: ordnungsgemäße Termination nach Ausführung von a)*

Prozess-Ausdrücke

- *Für alle* $t_1, t_2 \in BPA$ *ist* $(t_1 + t_2) \in BPA$.
  (**Auswahl** : *Es wird entweder* t_1 *oder* t_2 *ausgeführt.)*

Prozess-Ausdrücke

- *Für alle* $t_1, t_2 \in BPA$ *ist* $(t_1 + t_2) \in BPA$.
  (**Auswahl** : *Es wird entweder* t_1 *oder* t_2 *ausgeführt.)*

- *Für alle* $t_1, t_2 \in BPA$ *ist* $(t_1 \cdot t_2) \in BPA$.
  (**Hintereinanderausführung, Sequenz** : *Nach der ordnungsgemäßen Termination von* t_1 *wird* t_2 *ausgeführt.)*

Prozess-Ausdrücke

- *Für alle* $t_1, t_2 \in BPA$ *ist* $(t_1 + t_2) \in BPA$.
  (**Auswahl** : *Es wird entweder* t_1 *oder* t_2 *ausgeführt.)*

- *Für alle* $t_1, t_2 \in BPA$ *ist* $(t_1 \cdot t_2) \in BPA$.
  (**Hintereinanderausführung, Sequenz** : *Nach der ordnungsgemäßen Termination von* t_1 *wird* t_2 *ausgeführt.)*

- *Nur nach diesen Regeln gebildete Terme liegen in BPA.*

Prozess-Ausdrücke

Klammersparregeln durch Bindungsstärke der Operatoren:

Sequenz $(\cdot)$ bindet stärker als Auswahl $(+)$.

Also:
Statt $a + (b \cdot c)$ kann man $a + b \cdot c$ schreiben.

weitere Abkürzung, falls Mehrdeutigkeit ausgeschlossen:

$$ab \text{ statt } a \cdot b$$

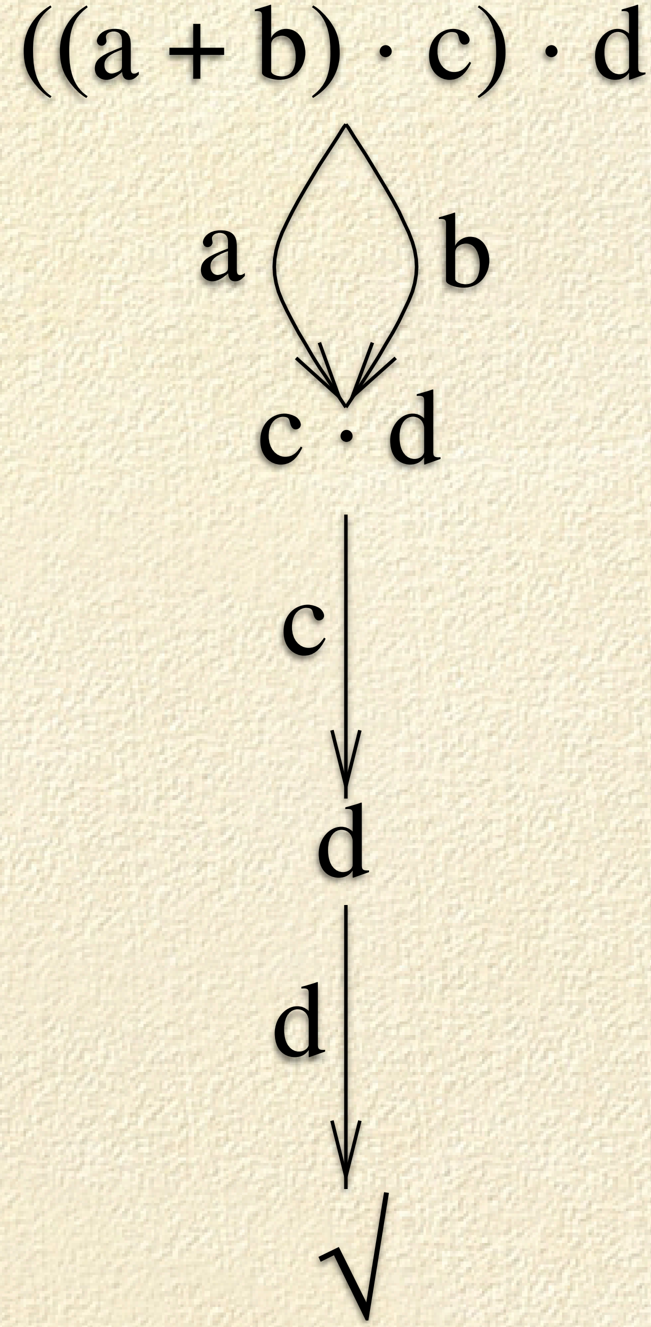
Prozess-Ausdrücke

Beispiel:

$((a+b)\cdot c)\cdot d \in BPA$ repräsentiert den folgenden Prozessgraphen:

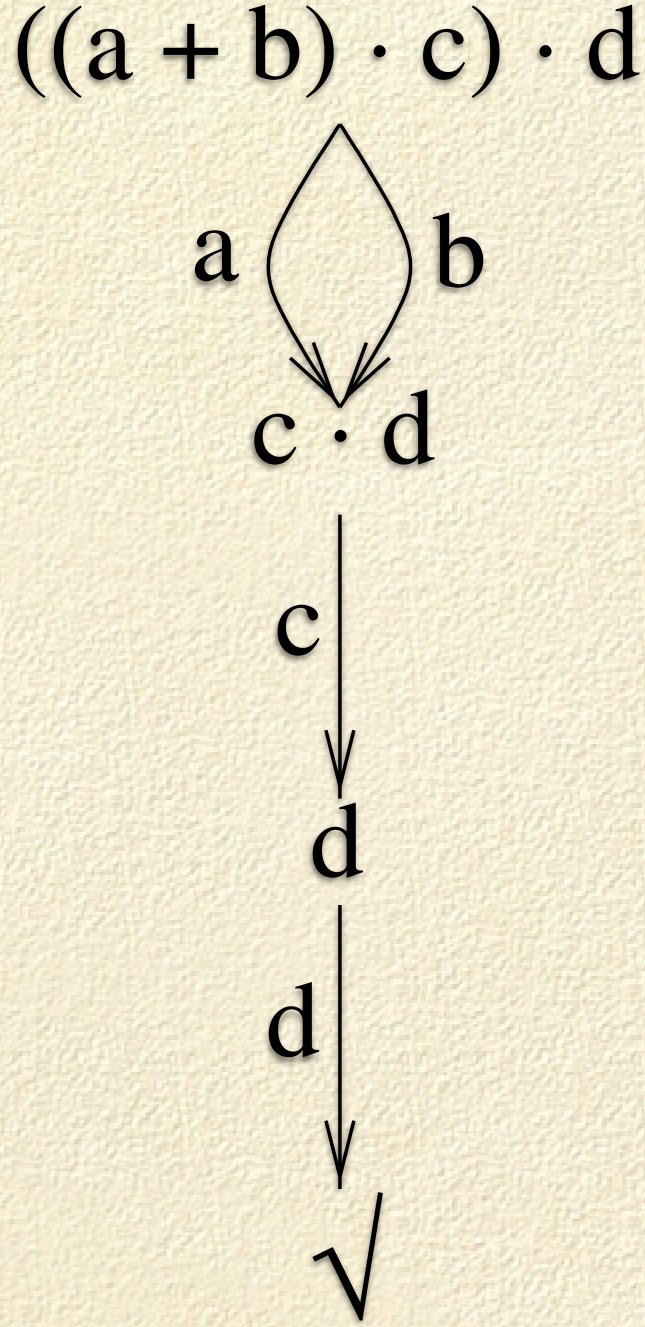
Beispiel:

$((a+b) \cdot c) \cdot d \in BPA$ repräsentiert den folgenden Prozessgraphen:



Beispiel:

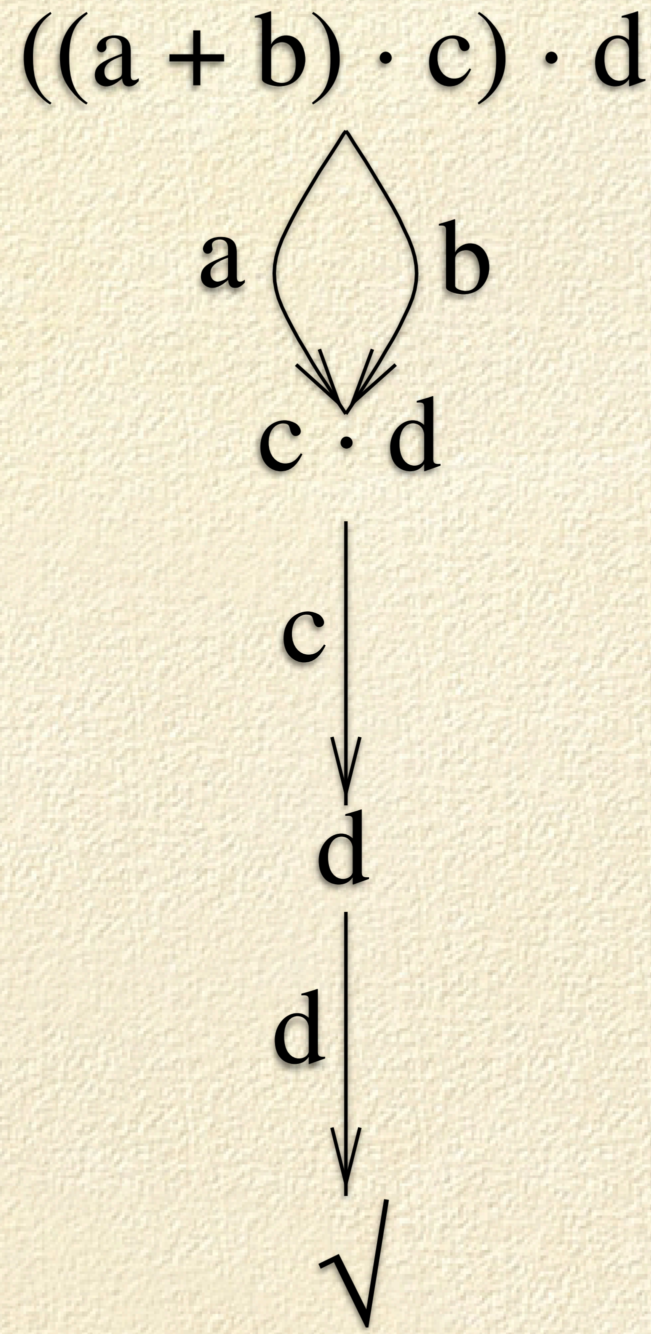
$((a+b)\cdot c)\cdot d \in BPA$ repräsentiert den folgenden $\boxed{\text{Prozessgraphen:}}$



informelle Semantik

Beispiel:

$((a+b)\cdot c)\cdot d \in BPA$ repräsentiert den folgenden Prozessgraphen:



informelle Semantik

es folgt:

formale Semantik

Definition eines **Prozessgraphen** für einen Prozessterm t_0 durch ein Transitionssystem:

Definition 5.4 $TS = (S, A, tr, s_0)$ *mit*

$S \subset BPA,$

$s_0 := t_0$ *und*

tr *durch ein Kalkül mit* **Axiom** (A_0) *und*

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

$$\frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

Schlussregeln

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

$$\frac{x \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

Schlussregeln

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

$$\frac{x \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

Schlussregeln

$$\frac{y \xrightarrow{v} y'}{x + y \xrightarrow{v} y'}$$

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

$$\frac{x \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

Schlussregeln

$$\frac{y \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{y \xrightarrow{v} y'}{x + y \xrightarrow{v} y'}$$

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

Axiom

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Schlussregeln

$$\frac{x \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

$$\frac{y \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{y \xrightarrow{v} y'}{x + y \xrightarrow{v} y'}$$

$$\frac{x \xrightarrow{v} x'}{x \cdot y \xrightarrow{v} x' \cdot y}$$

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

Axiom

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Schlussregeln

$$\frac{x \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

$$\frac{y \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{y \xrightarrow{v} y'}{x + y \xrightarrow{v} y'}$$

$$\frac{x \xrightarrow{v} \surd}{x \cdot y \xrightarrow{v} y} \qquad \frac{x \xrightarrow{v} x'}{x \cdot y \xrightarrow{v} x' \cdot y}$$

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

$$\frac{x \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

Schlussregeln

$$\frac{y \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{y \xrightarrow{v} y'}{x + y \xrightarrow{v} y'}$$

$$\frac{x \xrightarrow{v} \surd}{x \cdot y \xrightarrow{v} y} \qquad \frac{x \xrightarrow{v} x'}{x \cdot y \xrightarrow{v} x' \cdot y}$$

(strukturelle operationale Semantik)

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

$$\frac{x \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

Schlussregeln

$$\frac{y \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{y \xrightarrow{v} y'}{x + y \xrightarrow{v} y'}$$

$$\frac{x \xrightarrow{v} \surd}{x \cdot y \xrightarrow{v} y} \qquad \frac{x \xrightarrow{v} x'}{x \cdot y \xrightarrow{v} x' \cdot y}$$

da Transitionssystem

(strukturelle operationale Semantik)

Transitionsregeln :

$(v \in A, \quad x, y, \ldots \in BPA)$

$$\frac{}{v \xrightarrow{v} \surd} \qquad (A_0)$$

Axiom

$$\frac{x \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x'}{x + y \xrightarrow{v} x'}$$

Schlussregeln

$$\frac{y \xrightarrow{v} \surd}{x + y \xrightarrow{v} \surd} \qquad \frac{y \xrightarrow{v} y'}{x + y \xrightarrow{v} y'}$$

$$\frac{x \xrightarrow{v} \surd}{x \cdot y \xrightarrow{v} y} \qquad \frac{x \xrightarrow{v} x'}{x \cdot y \xrightarrow{v} x' \cdot y}$$

da über Formelaufbau

da Transitionssystem

(strukturelle operationale Semantik)

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c \cdot d$ aus den Transitionsregeln:

$$b \xrightarrow{b} \surd$$

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c\cdot d$ aus den Transitionsregeln:

$$\frac{}{b \xrightarrow{b} \surd}$$

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c\cdot d$ aus den Transitionsregeln:

$$\frac{}{b \xrightarrow{b} \surd} \qquad \frac{}{v \xrightarrow{v} \surd}$$

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c \cdot d$ aus den Transitionsregeln:

$$\frac{}{b \xrightarrow{b} \surd}$$

$$\frac{}{v \xrightarrow{v} \surd}$$

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c\cdot d$ aus den Transitionsregeln:

$$\frac{\qquad\qquad}{b \xrightarrow{b} \surd} \qquad\qquad \frac{}{v \xrightarrow{v} \surd}$$

$$\frac{\qquad\qquad}{a+b \xrightarrow{b} \surd} \qquad\qquad \frac{y \xrightarrow{v} \surd}{x+y \xrightarrow{v} \surd}$$

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c\cdot d$ aus den Transitionsregeln:

$$\frac{}{b \xrightarrow{b} \surd}$$

$$\frac{}{v \xrightarrow{v} \surd}$$

$$\frac{}{a+b \xrightarrow{b} \surd}$$

$$\frac{y \xrightarrow{v} \surd}{x+y \xrightarrow{v} \surd}$$

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c\cdot d$ aus den Transitionsregeln:

$$\frac{}{b \xrightarrow{b} \surd} \qquad \frac{}{v \xrightarrow{v} \surd}$$

$$\frac{}{a+b \xrightarrow{b} \surd} \qquad \frac{y \xrightarrow{v} \surd}{x+y \xrightarrow{v} \surd}$$

$$\frac{}{(a+b)\cdot c \xrightarrow{b} c} \qquad \frac{x \xrightarrow{v} \surd}{x\cdot y \xrightarrow{v} y}$$

Wednesday, January 5, 2011

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c \cdot d$ aus den Transitionsregeln:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{b \xrightarrow{b} \surd}}{a+b \xrightarrow{b} \surd}}{(a+b)\cdot c \xrightarrow{b} c}}{}$$

$$\dfrac{}{v \xrightarrow{v} \surd}$$

$$\dfrac{y \xrightarrow{v} \surd}{x+y \xrightarrow{v} \surd}$$

$$\dfrac{x \xrightarrow{v} \surd}{x \cdot y \xrightarrow{v} y}$$

Beispiel: Beweis von $((a+b)\cdot c)\cdot d \xrightarrow{b} c\cdot d$ aus den Transitionsregeln:

$$\frac{}{b \xrightarrow{b} \surd} \qquad \frac{}{v \xrightarrow{v} \surd}$$

$$\frac{}{a+b \xrightarrow{b} \surd} \qquad \frac{y \xrightarrow{v} \surd}{x+y \xrightarrow{v} \surd}$$

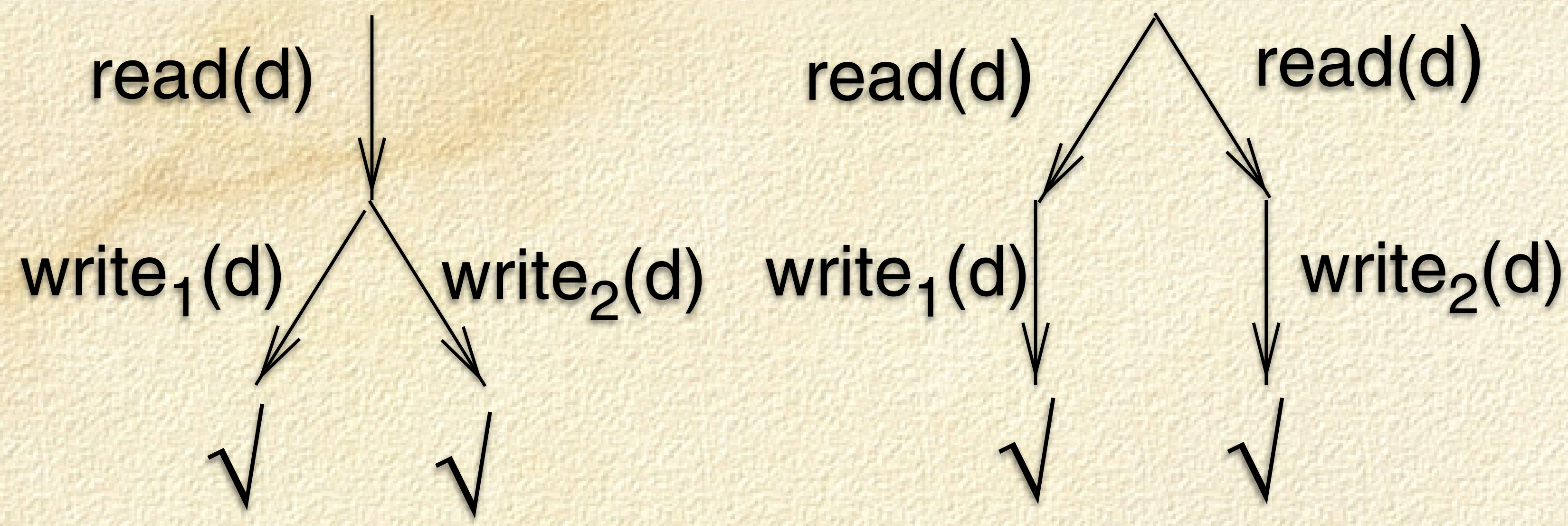
$$\frac{}{(a+b)\cdot c \xrightarrow{b} c} \qquad \frac{x \xrightarrow{v} \surd}{x\cdot y \xrightarrow{v} y}$$

$$\frac{}{((a+b)\cdot c)\cdot d \xrightarrow{b} c\cdot d} \qquad \frac{x \xrightarrow{v} x'}{x\cdot y \xrightarrow{v} x'\cdot y}$$

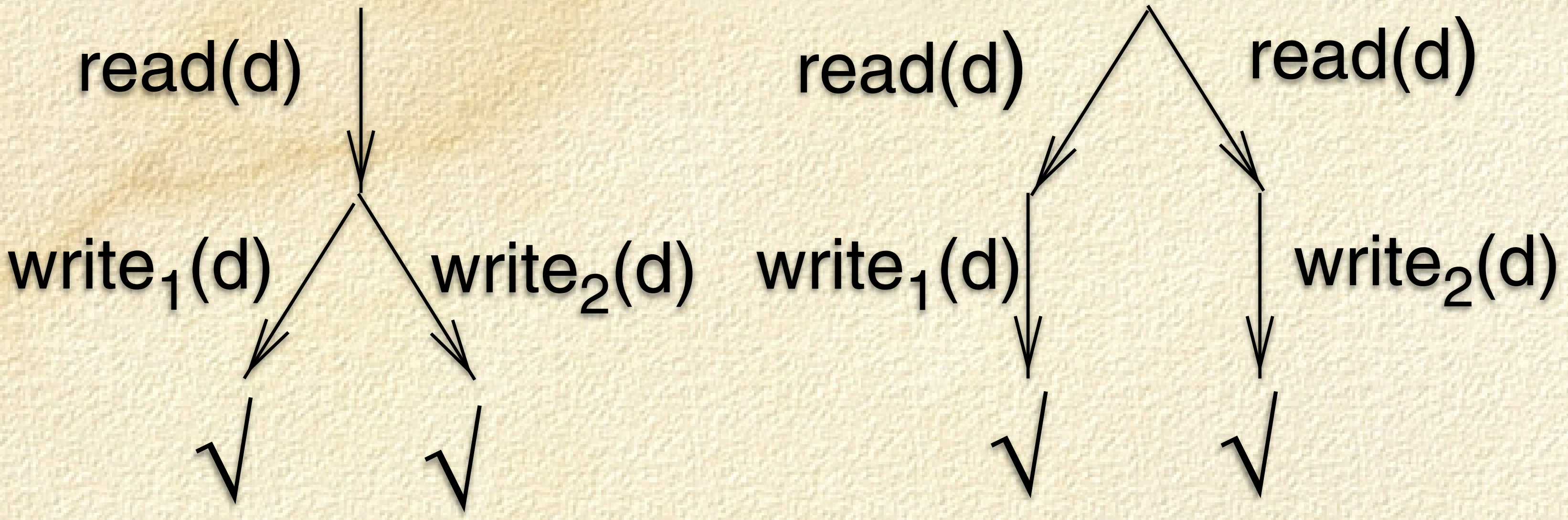
Bisimulation und Äquivalenz

Um das Verhalten eines Systems mit dem Verhalten eines anderen Systems oder einer Spezifikation zu vergleichen, wurde der Begriff der wechselseitigen Simulation oder Bisimulation eingeführt.

Beispiel:



Beispiel:



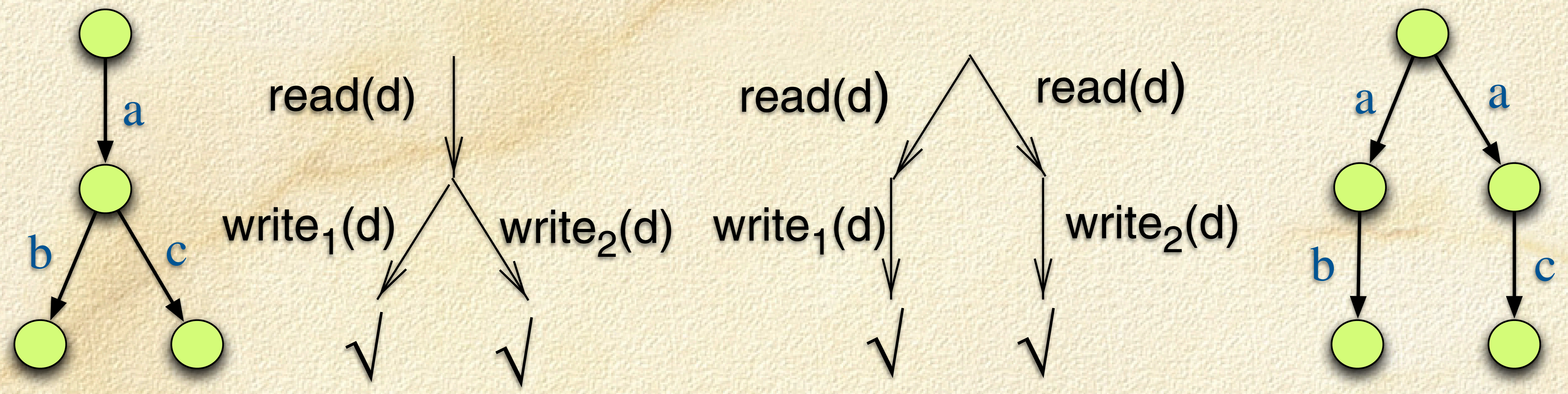
Der linke Prozeß liest d. Dann wird entschieden, ob d auf Platte 1 oder Platte 2 geschrieben wird. In dem anderen Prozeß wird die Entscheidung vor dem Lesen getroffen.

Beide Prozesse haben

$read(d)write_1(d)$ und $read(d)write_2(d)$

als Schaltfolgen und ...

Beispiel:



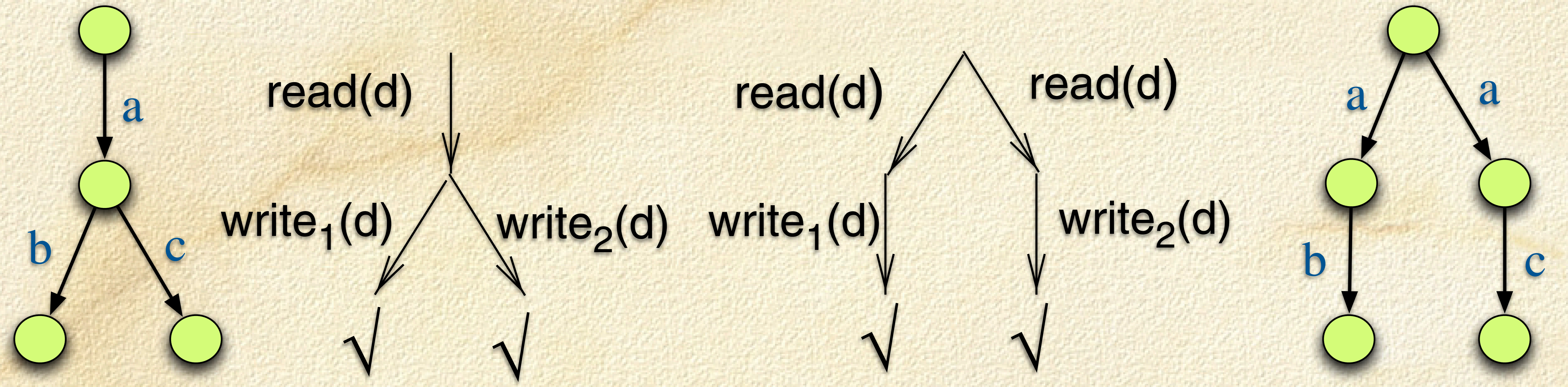
Der linke Prozeß liest d. Dann wird entschieden, ob d auf Platte 1 oder Platte 2 geschrieben wird. In dem anderen Prozeß wird die Entscheidung vor dem Lesen getroffen.

Beide Prozesse haben

$read(d)write_1(d)$ und $read(d)write_2(d)$

als Schaltfolgen und ...

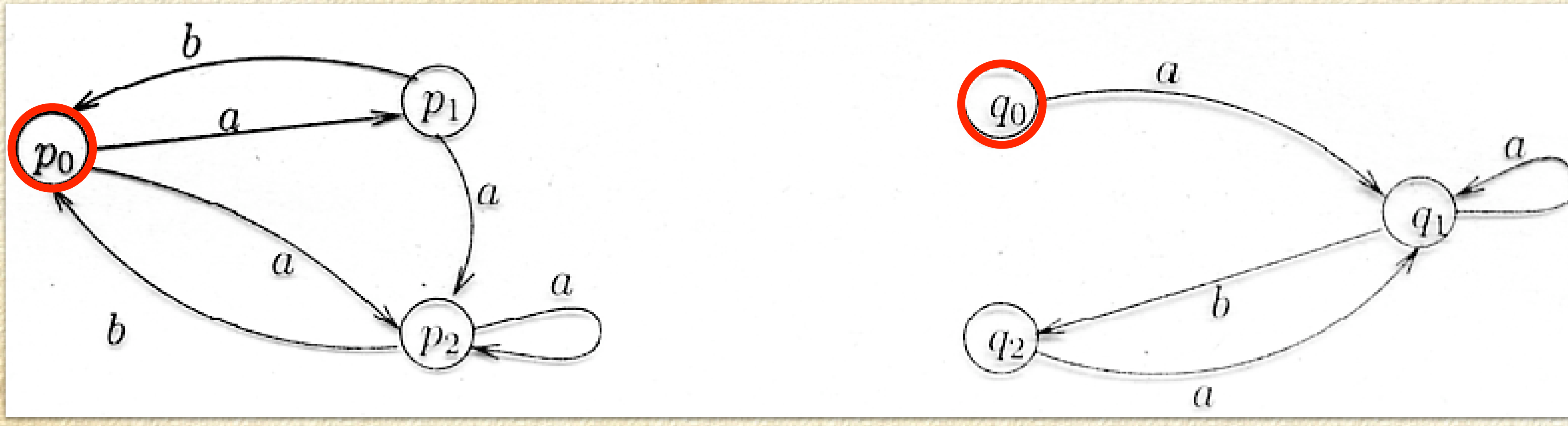
Beispiel:



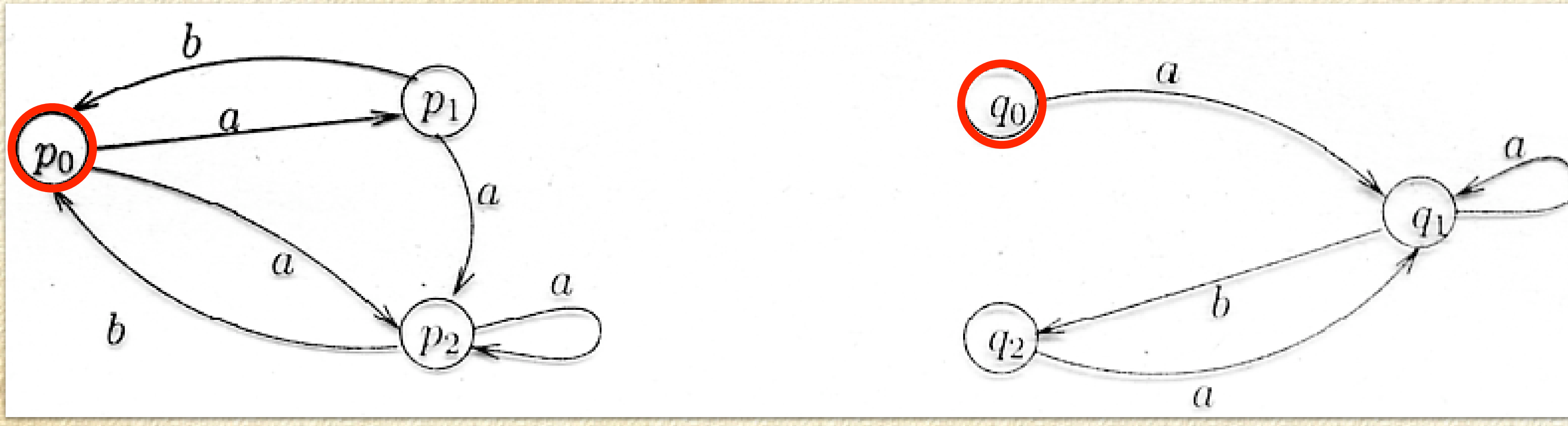
…sind daher “schaltfolgenäquivalent” (trace equivalent).

Diese Art der Äquivalenz ist jedoch häufig nicht angemessen, z.b. wenn die Platte 1 ausfällt. Dann würde der erste Prozeß d bei jedem Ablauf auf Platte 2 schreiben - im Gegensatz zum anderen Prozeß, der in eine Verklemmung geraten kann.

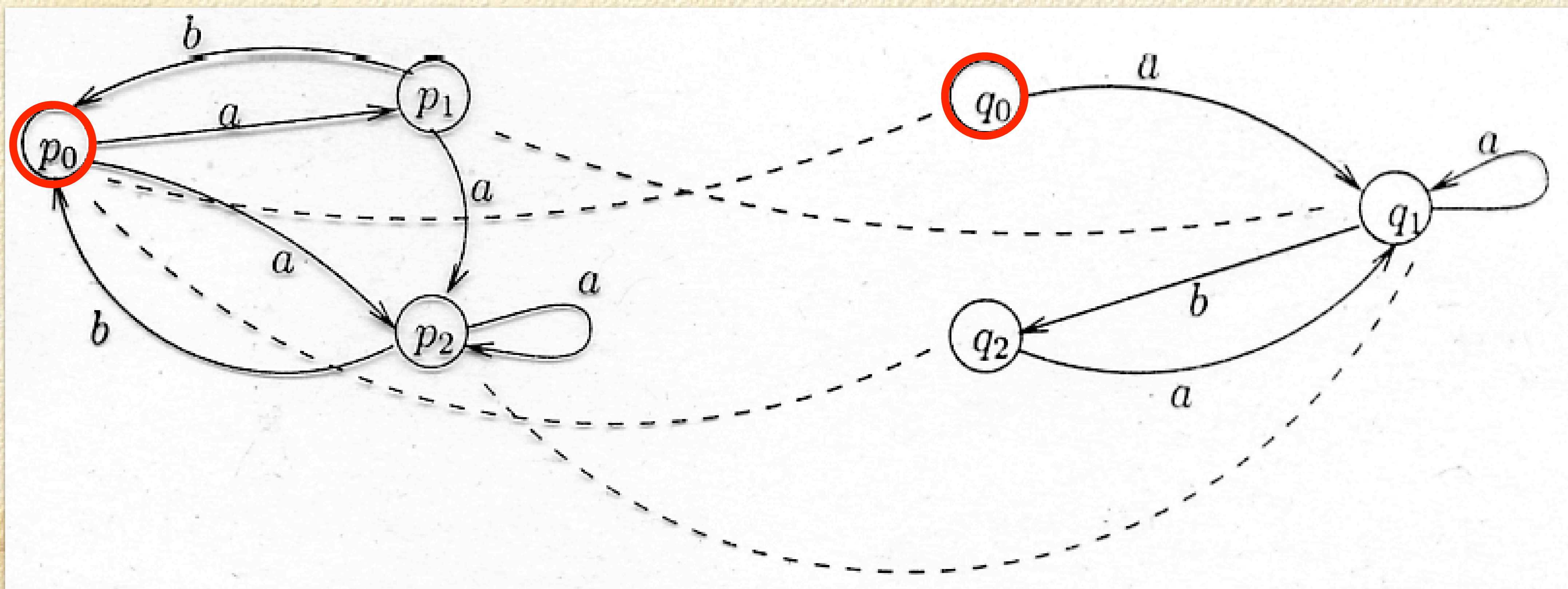
Dies ist die Motivation, für eine auf “Bisimulation” beruhende Äquivalenz.



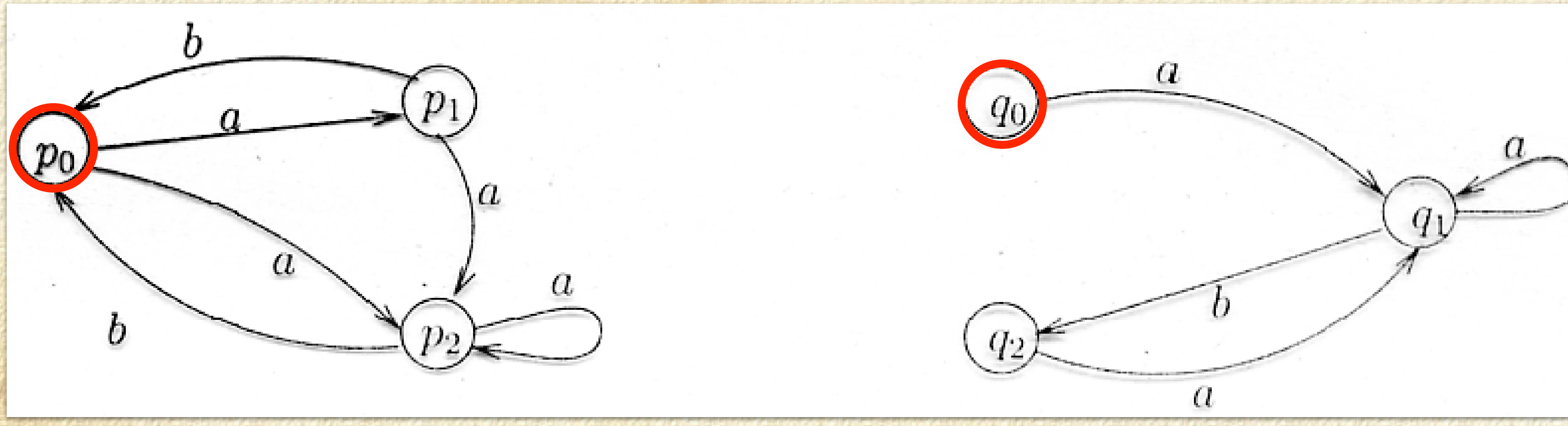
b
a
a
a
b
a
p_0
p_1
p_2
a
a
b
a
q_0
q_1
q_2

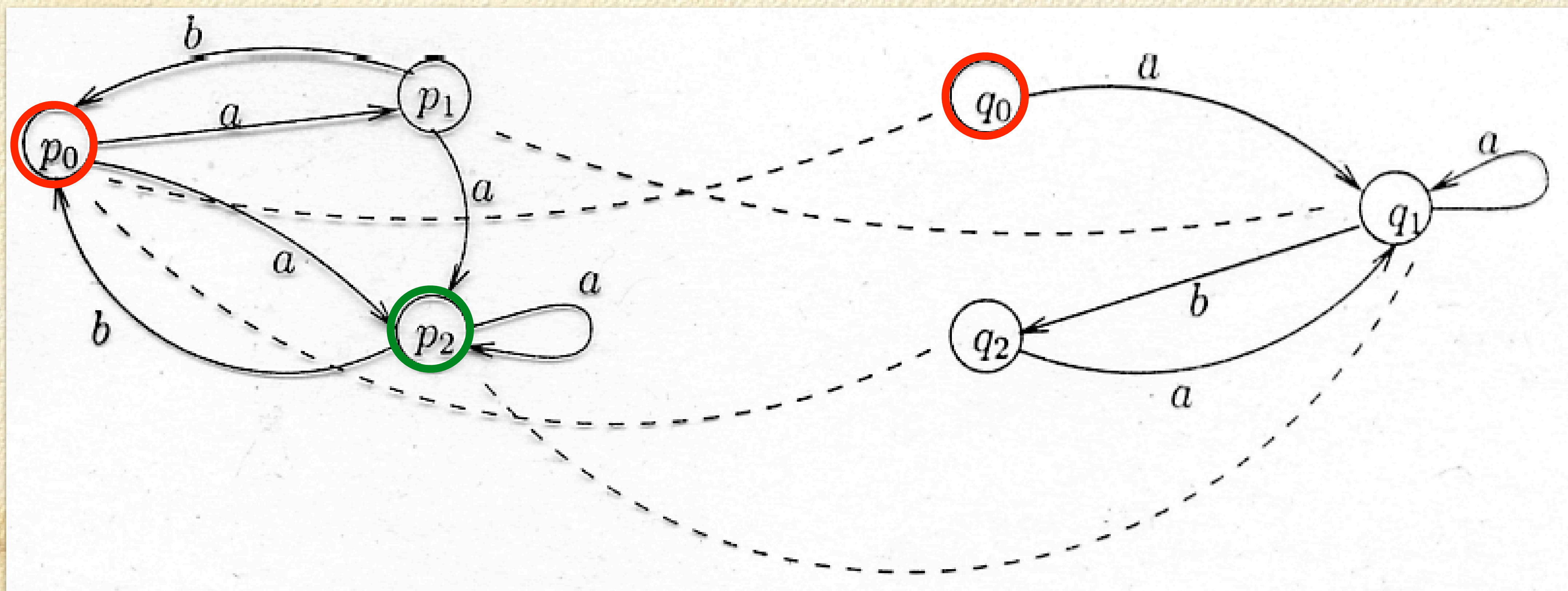


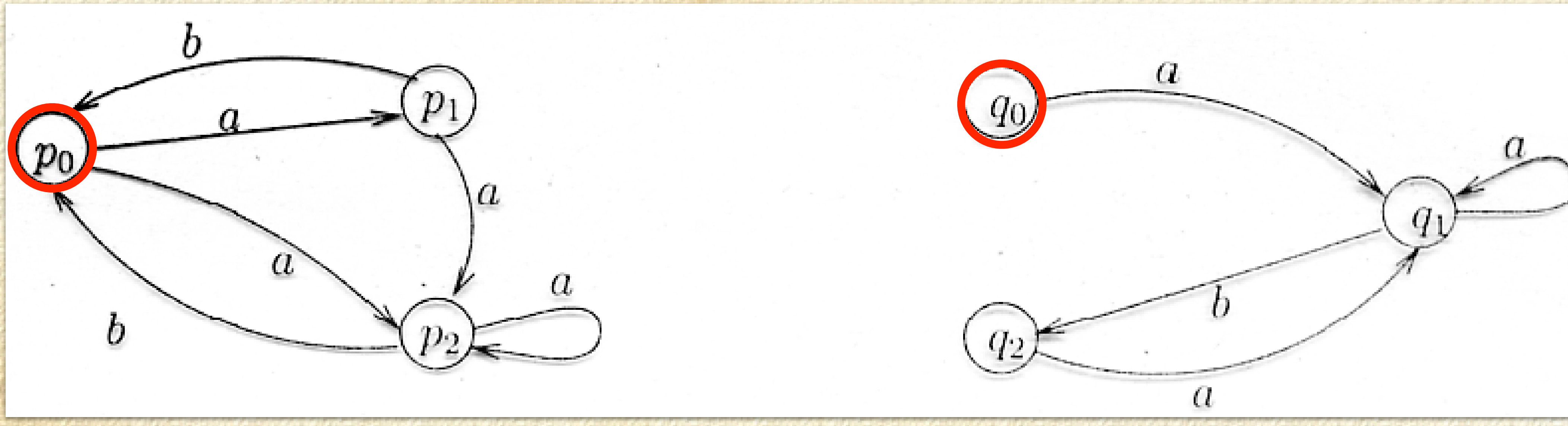
p_0
p_1
p_2
q_0
q_1
q_2
a
b



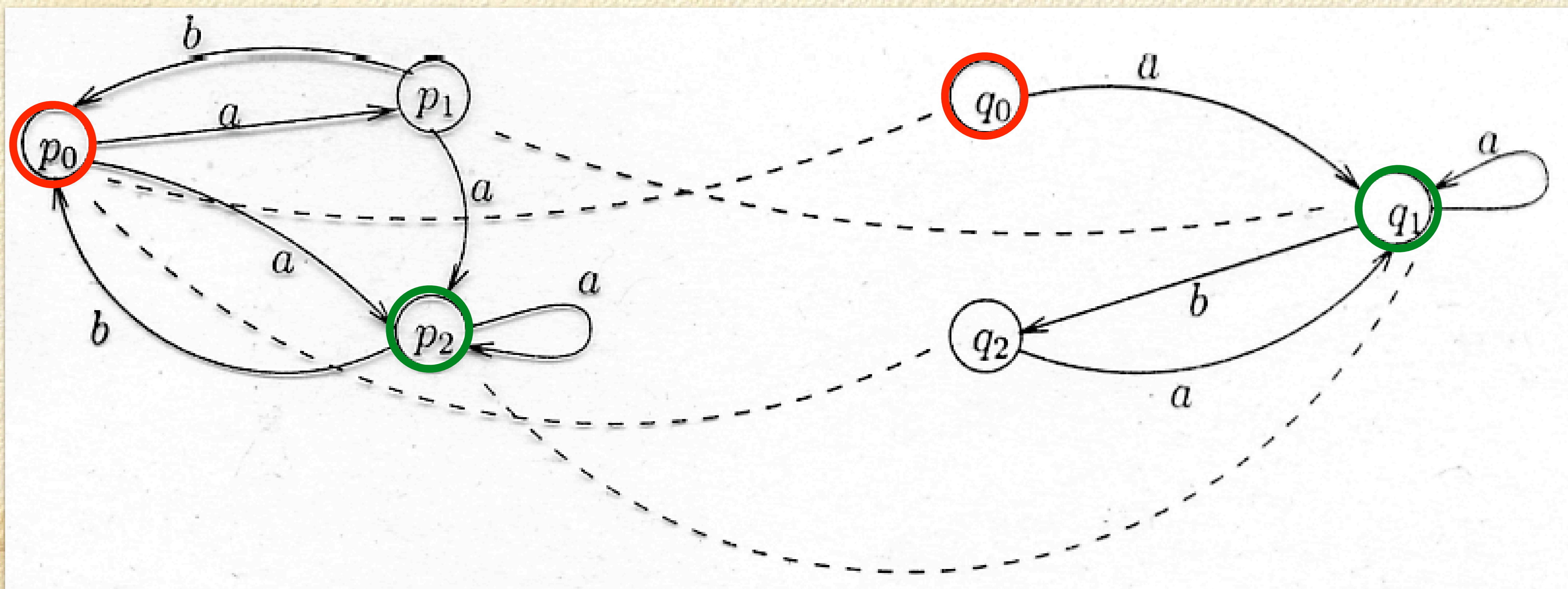
p_0
p_1
p_2
q_0
q_1
q_2
a
b


b
a
a
a
b
a
p0
p1
p2
a
a
a
b
a
q0
q1
q2

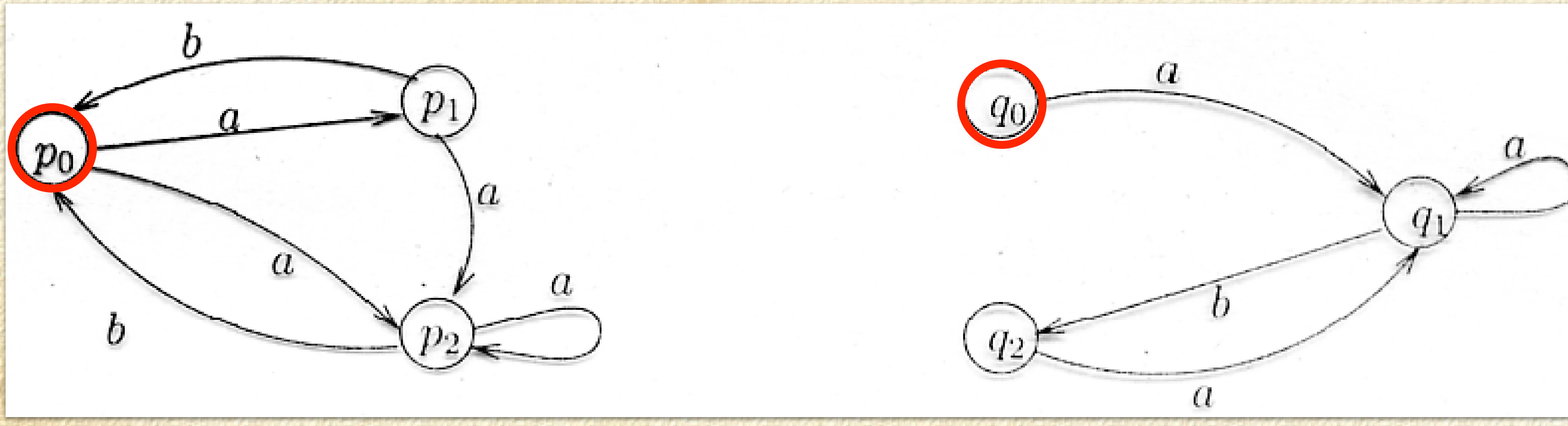

b
a
a
a
b
a
p0
p1
p2
a
a
b
a
q0
q1
q2

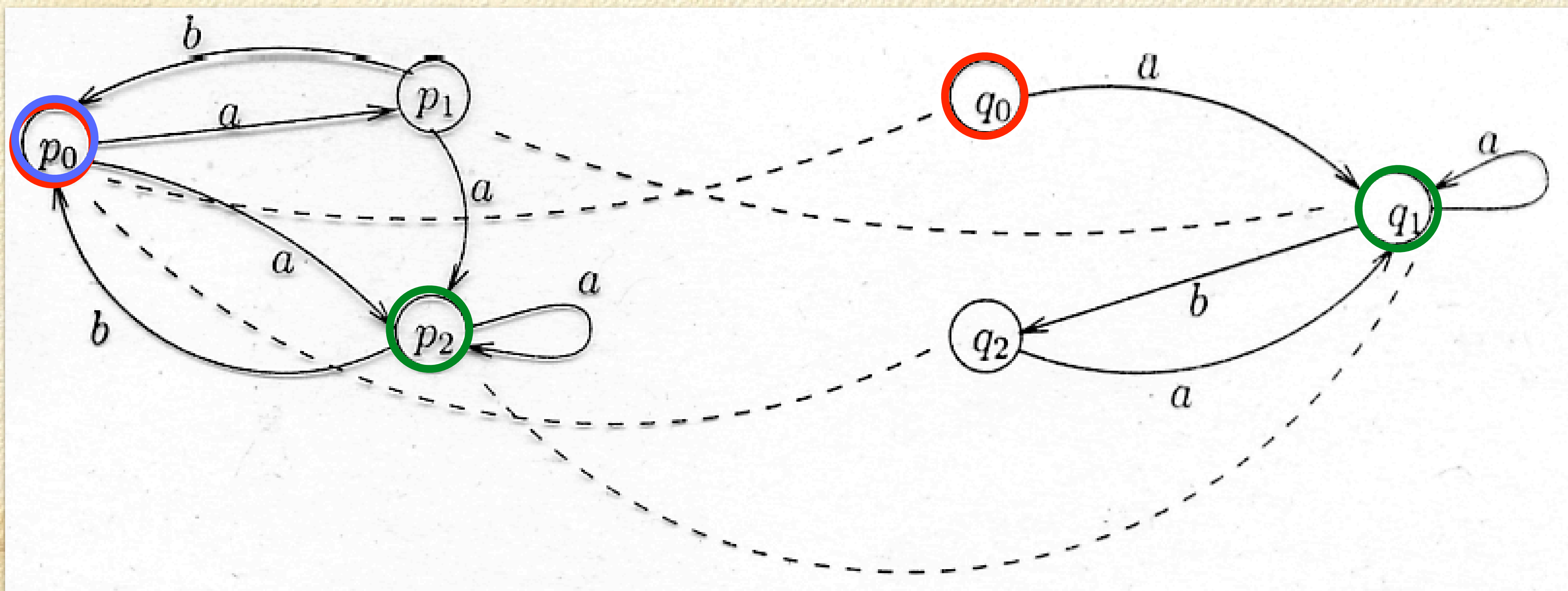


b
a
a
a
b
a
p_0
p_1
p_2
a
a
a
b
a
q_0
q_1
q_2

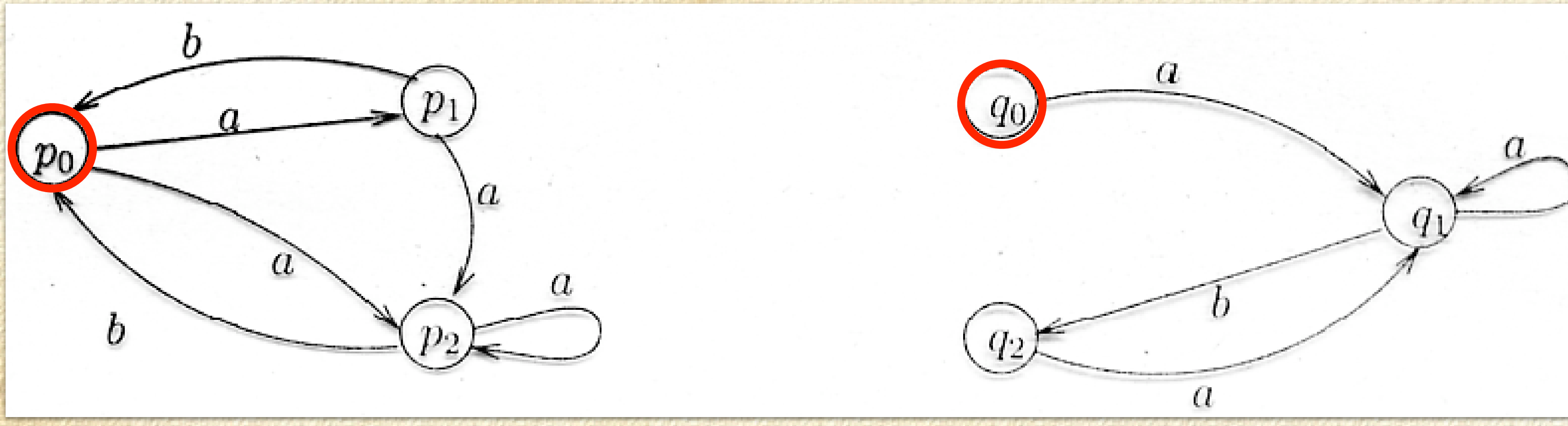


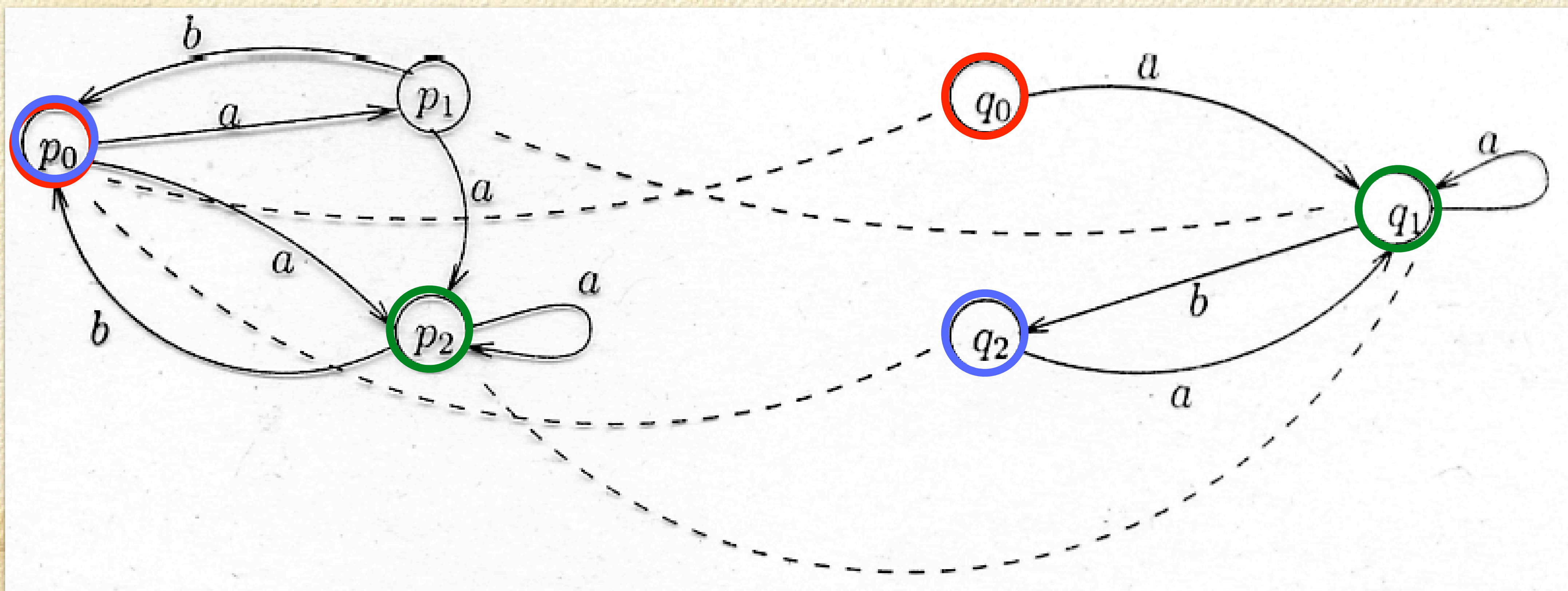
b
a
a
a
b
a
p_0
p_1
p_2
a
a
b
a
q_0
q_1
q_2

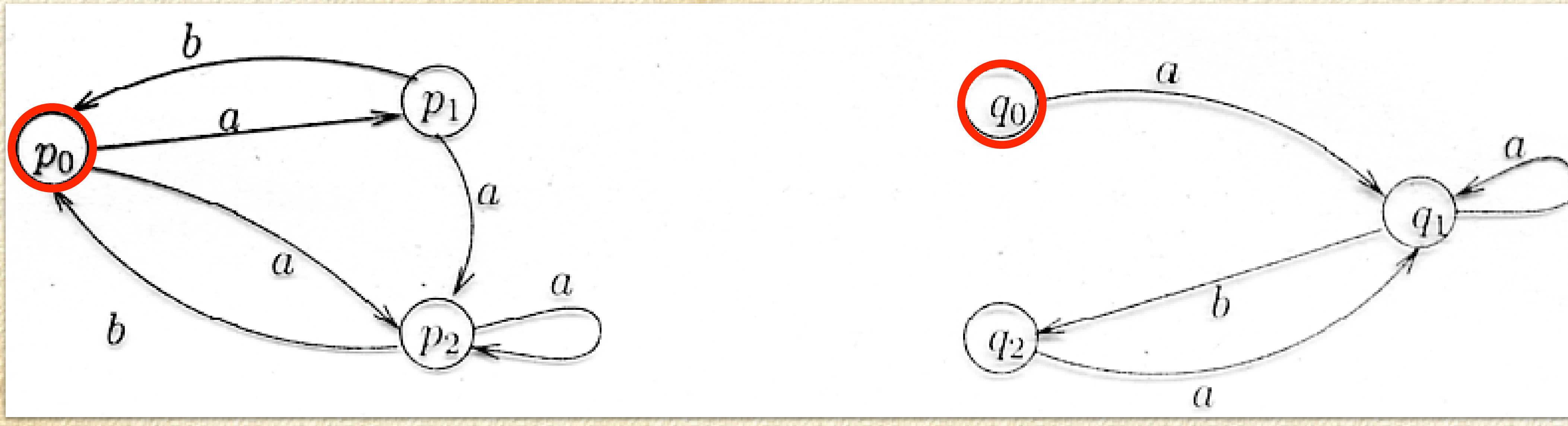




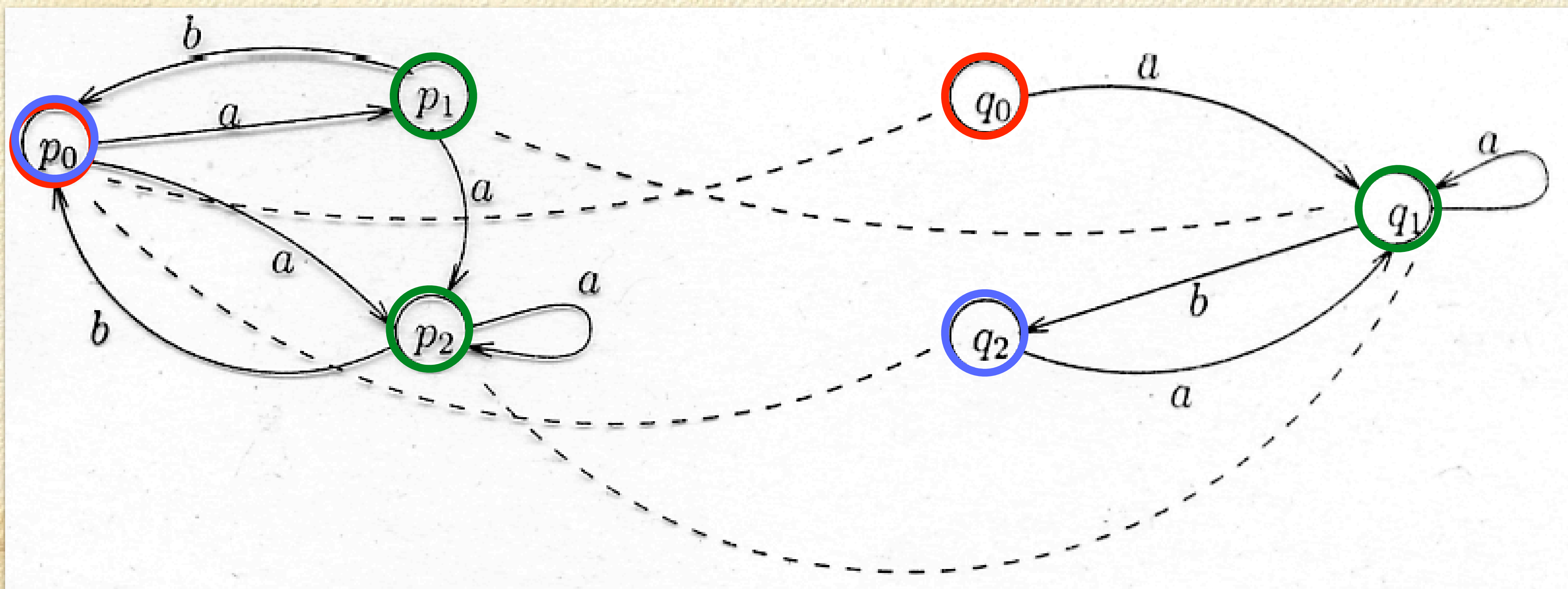
Wednesday, January 5, 2011


b
a
a
a
b
a
p_0
p_1
p_2
q_0
q_1
q_2
a
a
b
a


b
a
a
a
b
a
p_0
p_1
p_2
q_0
q_1
q_2
a
a
b
a



b
p_1
a
p_0
a
a
a
b
p_2
q_0
a
a
q_1
b
q_2
a



b
p_1
a
q_0
a
p_0
a
a
a
a
b
p_2
q_1
b
q_2
a

Äquivalenz durch Bisimulation

Eine *Bisimulation* ist eine binäre Relation $\mathcal{B}$ auf BPA (d.h. $\mathcal{B} \subseteq BPA \times BPA$) mit folgenden Eigenschaften:

Äquivalenz durch Bisimulation

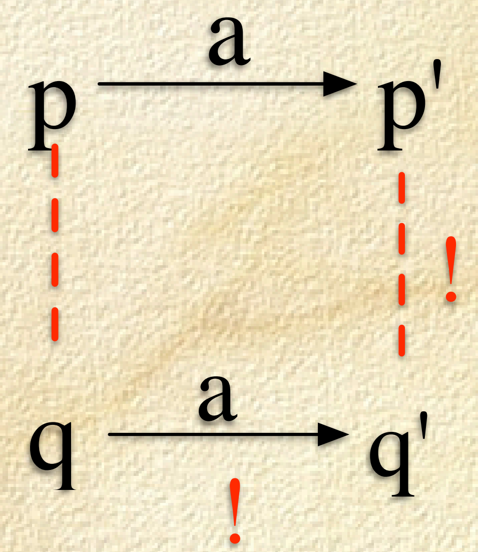
Eine *Bisimulation* ist eine binäre Relation $\mathcal{B}$ auf BPA (d.h. $\mathcal{B} \subseteq BPA \times BPA$) mit folgenden Eigenschaften:

1. falls $p\,\mathcal{B}\,q$ und $p \xrightarrow{a} p'$, dann $q \xrightarrow{a} q'$ mit $p'\,\mathcal{B}\,q'$
2. falls $p\,\mathcal{B}\,q$ und $q \xrightarrow{a} q'$, dann $p \xrightarrow{a} p'$ mit $p'\,\mathcal{B}\,q'$

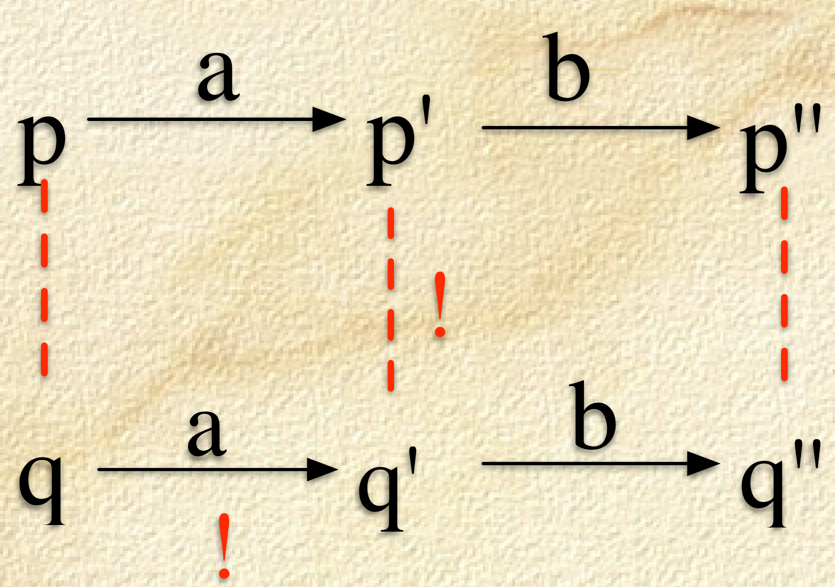
1. falls $p \, \mathcal{B} \, q$ und $p \xrightarrow{a} p'$, dann $q \xrightarrow{a} q'$ mit $p' \, \mathcal{B} \, q'$

$$\begin{tikzcd} p \arrow[r, "a"] \arrow[d, dash, dashed] & p' \\ q \end{tikzcd}$$

1. falls $p\,\mathcal{B}\,q$ und $p \xrightarrow{a} p'$, dann $q \xrightarrow{a} q'$ mit $p'\,\mathcal{B}\,q'$



1. falls $p\,\mathcal{B}\,q$ und $p \xrightarrow{a} p'$, dann $q \xrightarrow{a} q'$ mit $p'\,\mathcal{B}\,q'$



1. falls $p\,\mathcal{B}\,q$ und $p \xrightarrow{a} p'$, dann $q \xrightarrow{a} q'$ mit $p'\,\mathcal{B}\,q'$

$$
\begin{tikzcd}
p \arrow[r, "a"] \arrow[d, dash, dashed, red] & p' \arrow[r, "b"] \arrow[d, dash, dashed, red, "!"] & p'' \arrow[d, dash, dashed, red] \\
q \arrow[r, "a", "!"'] & q' \arrow[r, "b"] & q''
\end{tikzcd}
$$

1. falls $p \, \mathcal{B} \, q$ und $p \xrightarrow{a} p'$, dann $q \xrightarrow{a} q'$ mit $p' \, \mathcal{B} \, q'$
2. falls $p \, \mathcal{B} \, q$ und $q \xrightarrow{a} q'$, dann $p \xrightarrow{a} p'$ mit $p' \, \mathcal{B} \, q'$

$$
\begin{tikzcd}
p \arrow[r, "a"] \arrow[d, dash, dashed] & p' \arrow[r, "b"] \arrow[d, dash, dashed, "!"'] & p'' \arrow[d, dash, dashed] \\
q \arrow[r, "a", "!"'] & q' \arrow[r, "b"] & q''
\end{tikzcd}
$$

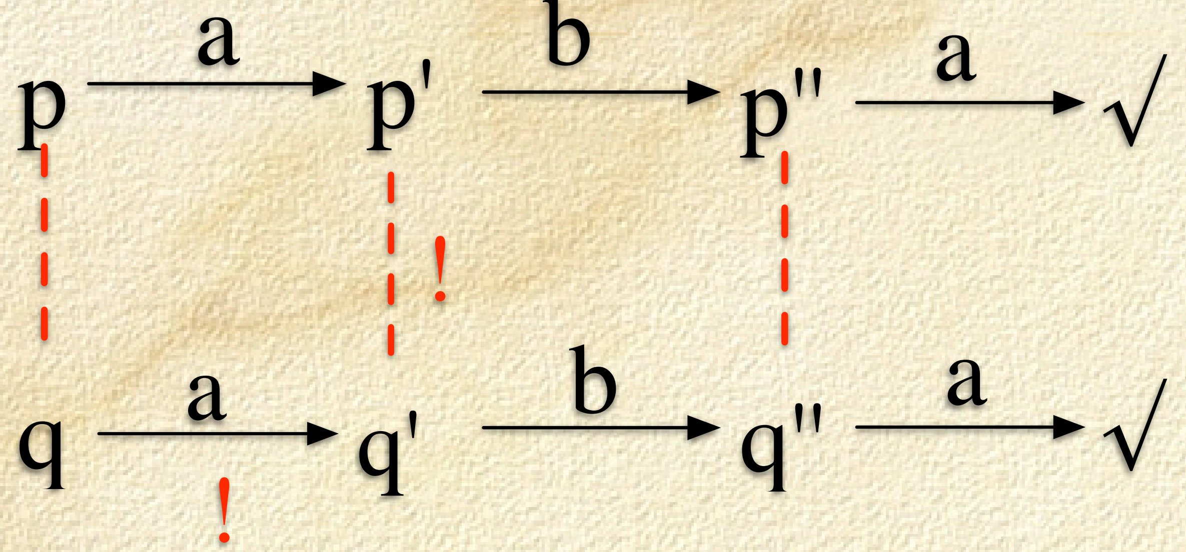
5.5

$$
\begin{tikzcd}
p \arrow[r, "a"] \arrow[d, dash, dashed] & p' \arrow[r, "b"] \arrow[d, dash, dashed, "!"] & p'' \arrow[r, "a"] \arrow[d, dash, dashed] & \surd \\
q \arrow[r, "a", "!"'] & q' \arrow[r, "b"] & q'' \arrow[r, "a"] & \surd
\end{tikzcd}
$$

$$
\begin{tikzcd}
p \arrow[r, "a"] \arrow[d, dash, dashed] & p' \arrow[r, "b"] \arrow[d, dash, dashed, "!"] & p'' \arrow[r, "a"] \arrow[d, dash, dashed] & \surd \\
q \arrow[r, "a", "!"'] & q' \arrow[r, "b"] & q'' \arrow[r, "a"] & \surd
\end{tikzcd}
$$

3. falls $p \,\mathcal{B}\, q$ und $p \xrightarrow{a} \surd$, dann $q \xrightarrow{a} \surd$
4. falls $p \,\mathcal{B}\, q$ und $q \xrightarrow{a} \surd$, dann $p \xrightarrow{a} \surd$

5.5

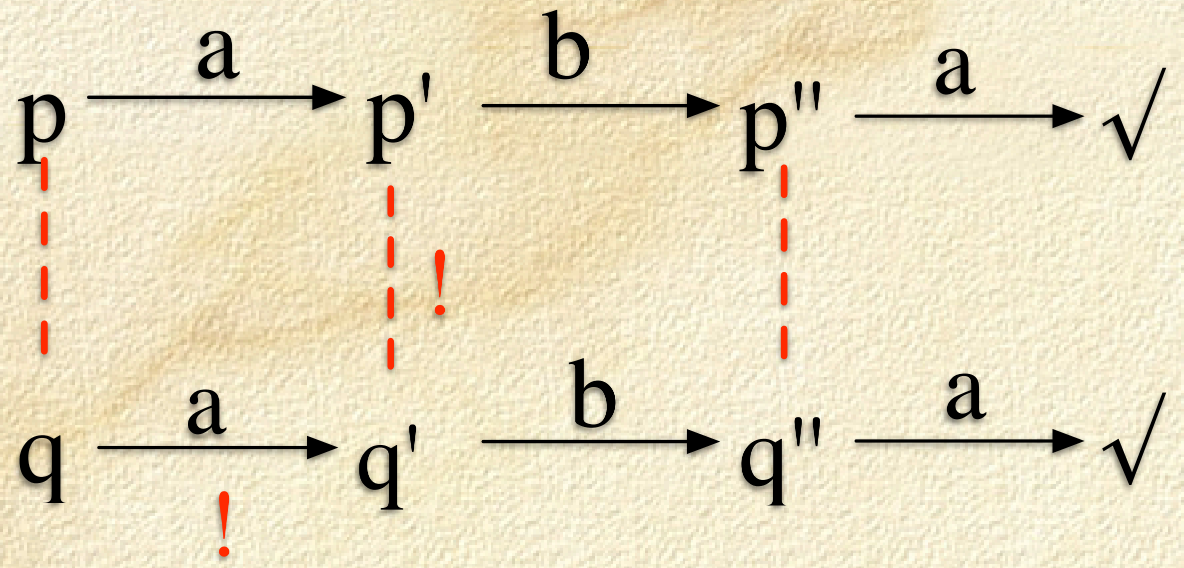


3. falls $p\,\mathcal{B}\,q$ und $p \xrightarrow{a} \surd$, dann $q \xrightarrow{a} \surd$

4. falls $p\,\mathcal{B}\,q$ und $q \xrightarrow{a} \surd$, dann $p \xrightarrow{a} \surd$

Zwei Prozesse p und q heißen *bisimilar* (Bezeichnung: $p \underline{\leftrightarrow} q$), falls es eine Bisimulation $\mathcal{B}$ mit $p\,\mathcal{B}\,q$ gibt.

5.5



3. falls $p\,\mathcal{B}\,q$ und $p \xrightarrow{a} \surd$, dann $q \xrightarrow{a} \surd$

4. falls $p\,\mathcal{B}\,q$ und $q \xrightarrow{a} \surd$, dann $p \xrightarrow{a} \surd$

Zwei Prozesse p und q heißen *bisimilar* (Bezeichnung: $p \underline{\leftrightarrow} q$), falls es eine Bisimulation $\mathcal{B}$ mit $p\,\mathcal{B}\,q$ gibt.

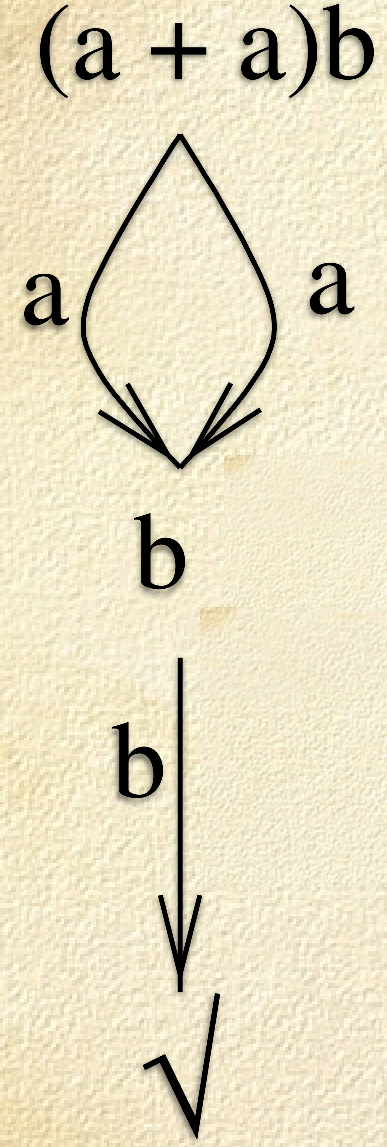
Lemma 5.5 *Die Bisimulation ist eine Äquivalenzrelation.*

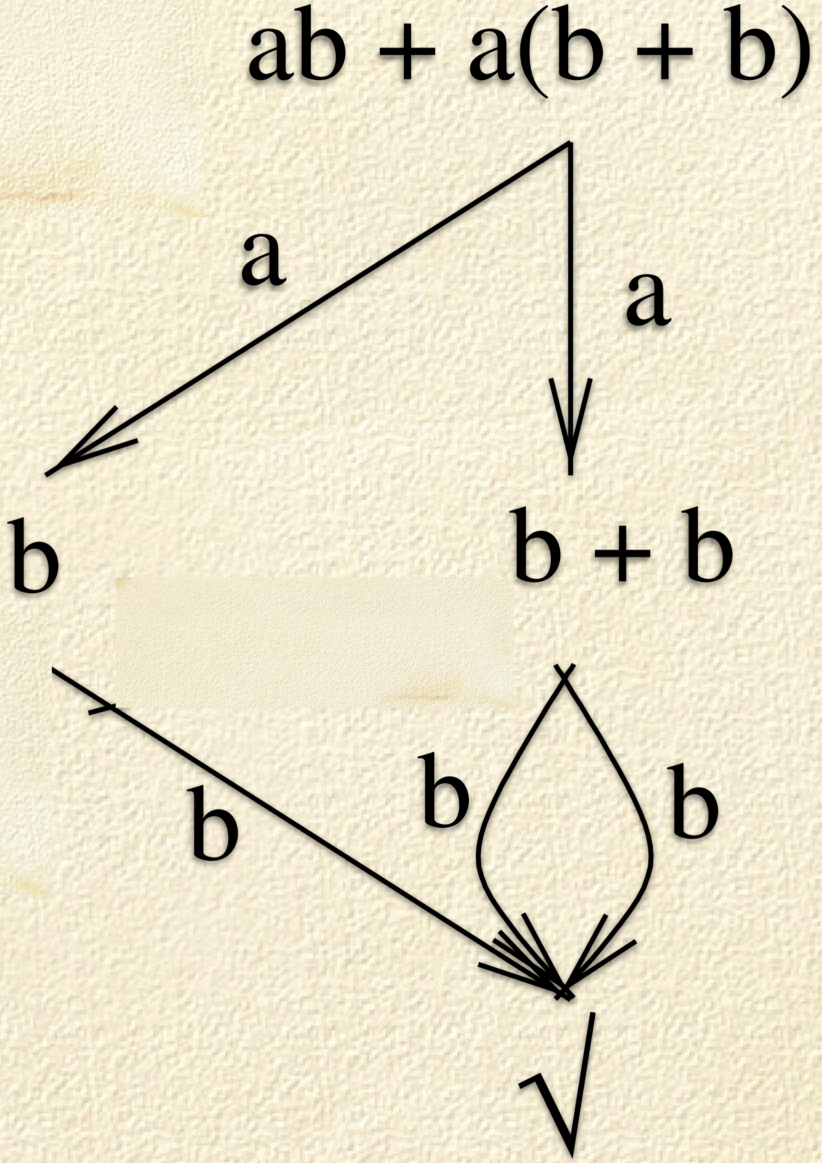
Beispiel: $(a+a)b \overset{?}{\underline{\leftrightarrow}} ab + a(b+b)$

Was ist zu tun?

Beispiel: $(a+a)b \ \overset{?}{\underline{\leftrightarrow}}\ ab + a(b+b)$

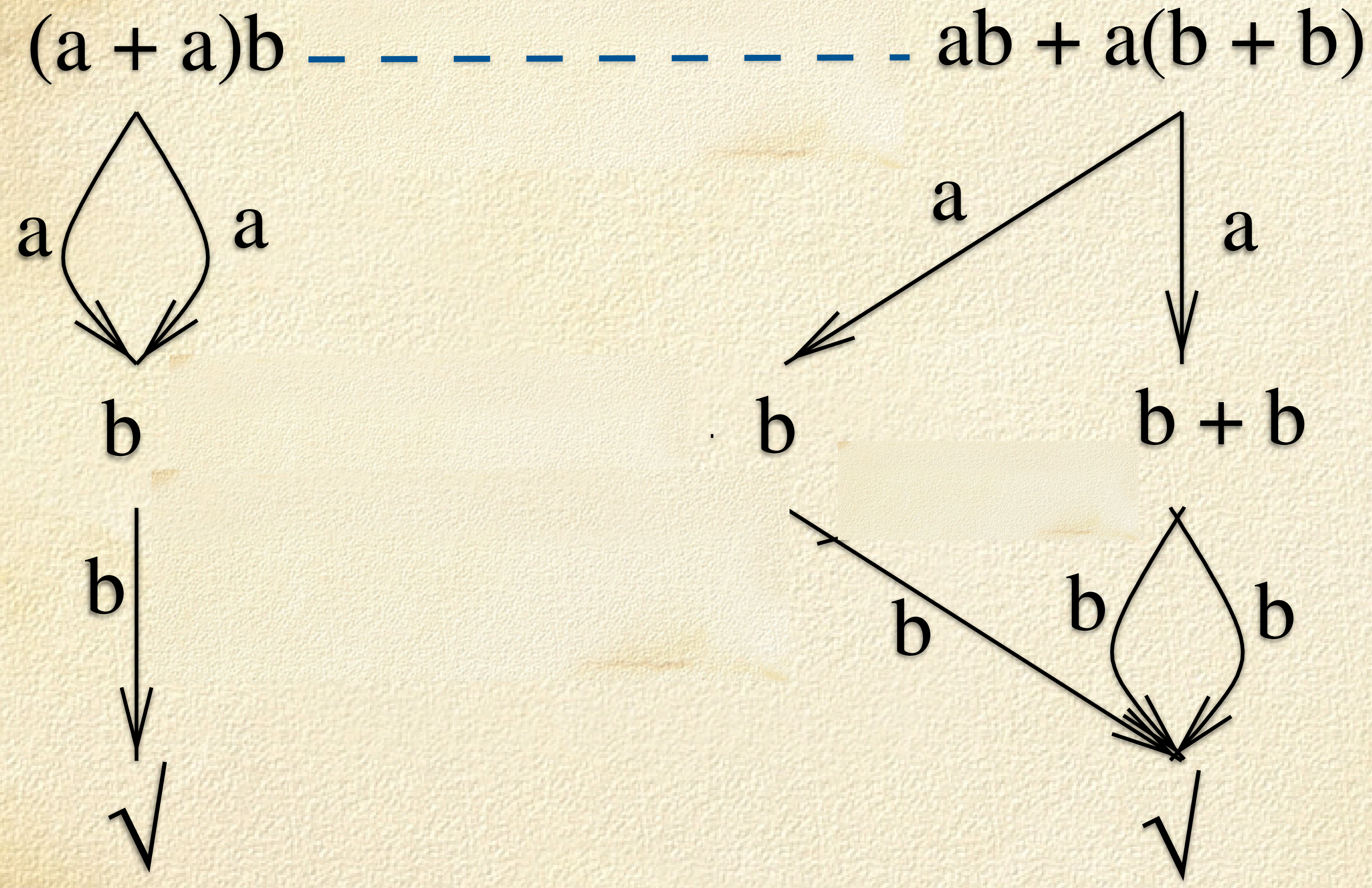
Was ist zu tun?





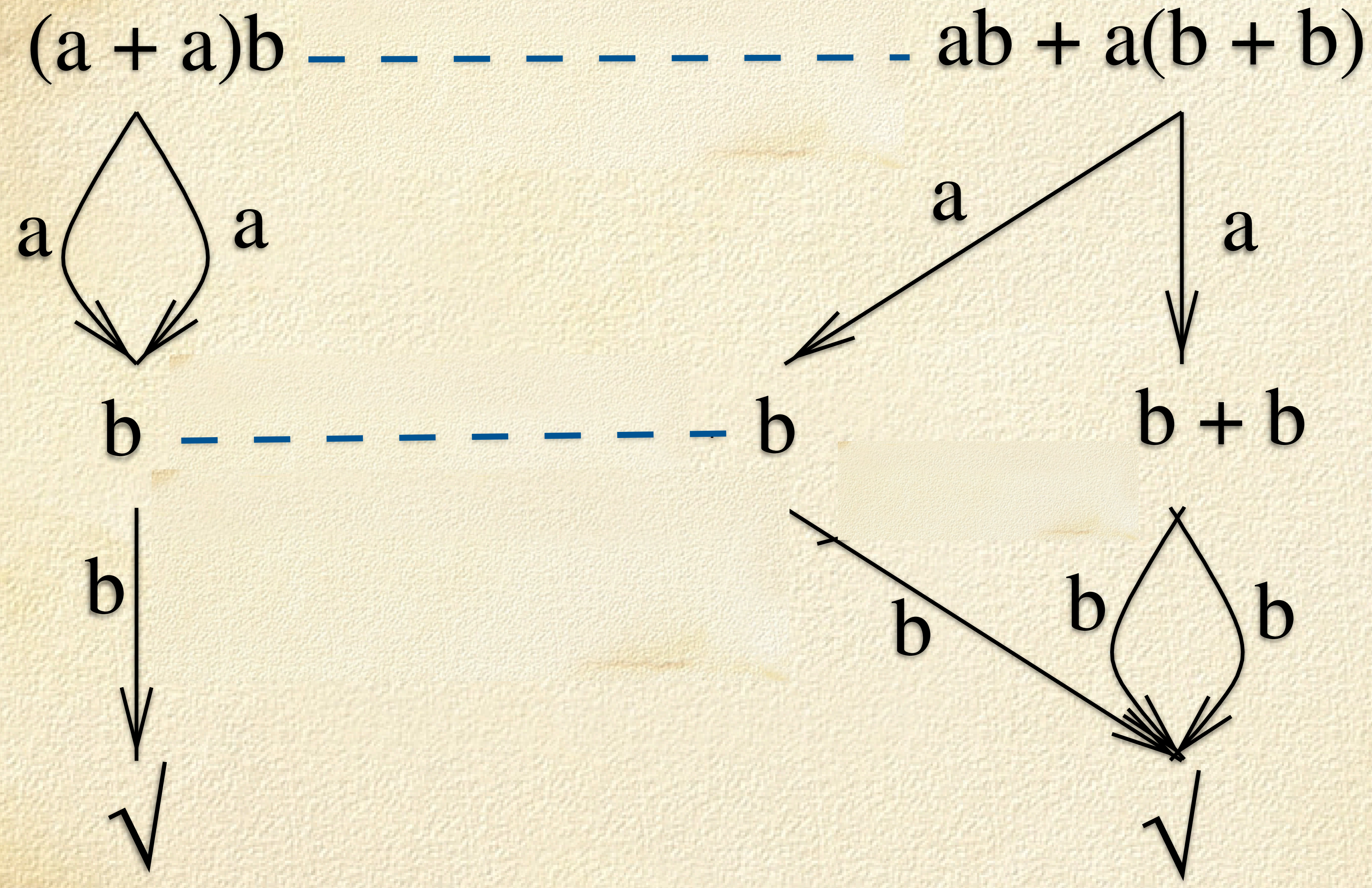
Beispiel: $(a+a)b \overset{?}{\underline{\leftrightarrow}} ab + a(b+b)$

Was ist zu tun?



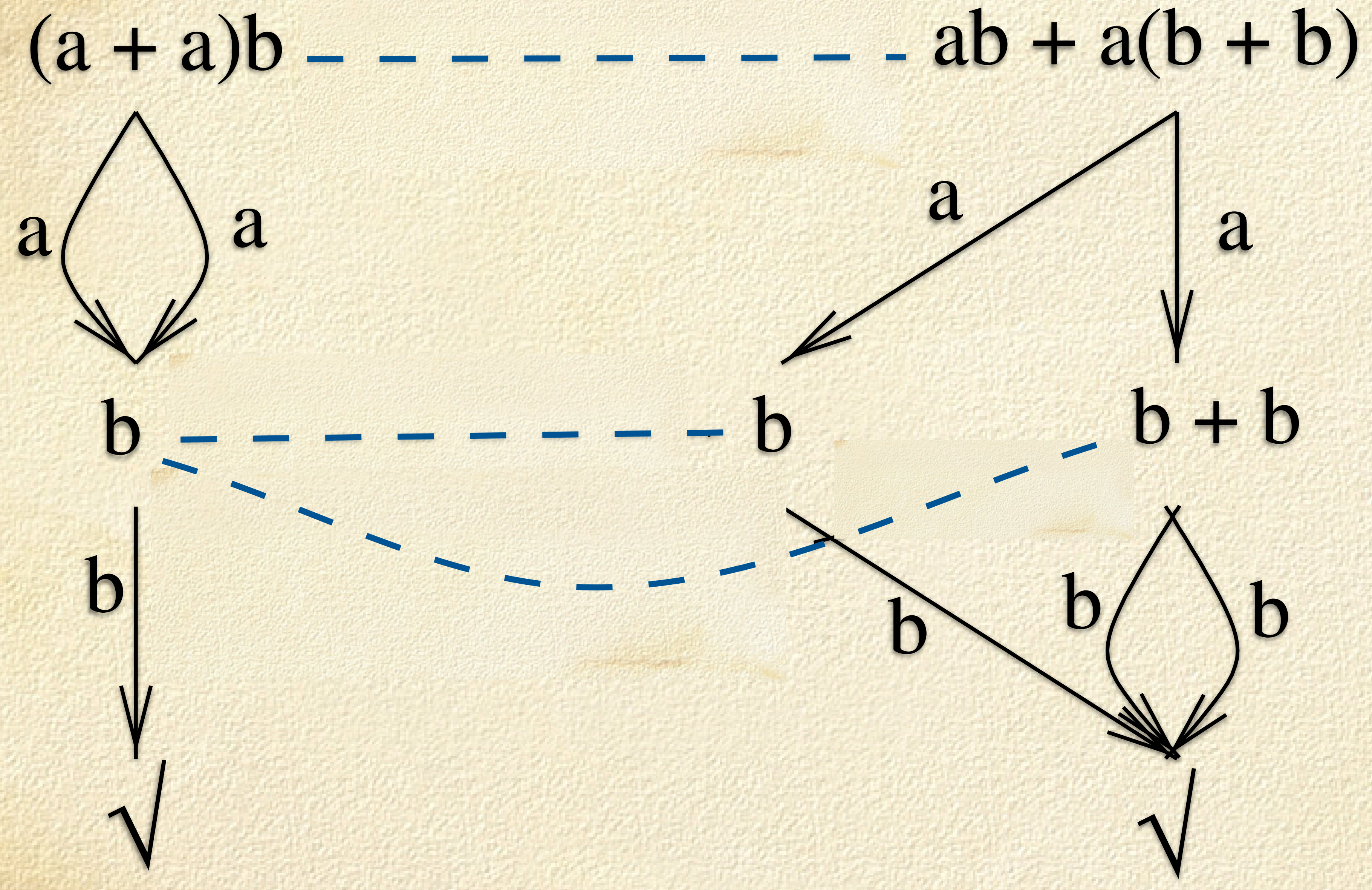
Beispiel: $(a + a)b \overset{?}{\underline{\leftrightarrow}} ab + a(b + b)$

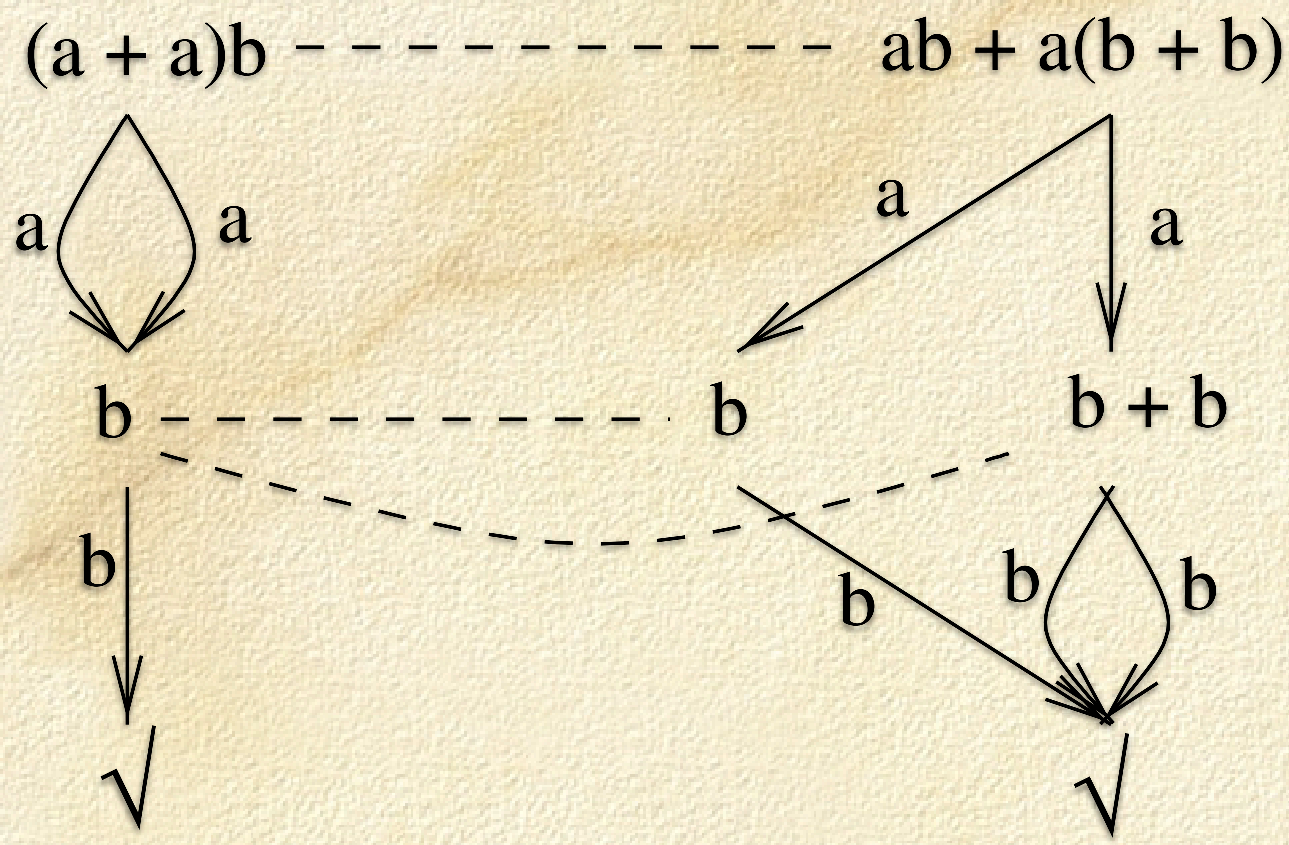
Was ist zu tun?



Beispiel: $(a+a)b \overset{?}{\underline{\leftrightarrow}} ab + a(b+b)$

Was ist zu tun?





Diese Bisimulation $\mathcal{B}$ wird definiert durch:

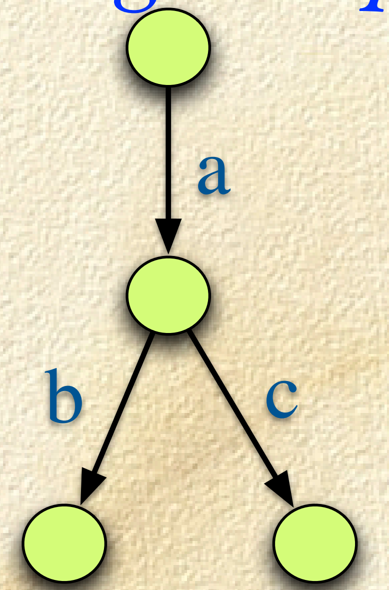
$(a + a)b \ \mathcal{B} \ ab + a(b + b)$

$b \ \mathcal{B} \ b$

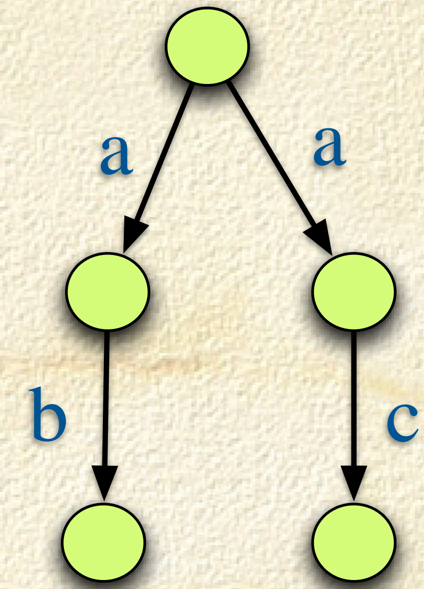
$b \ \mathcal{B} \ b + b.$

Wednesday, January 5, 2011

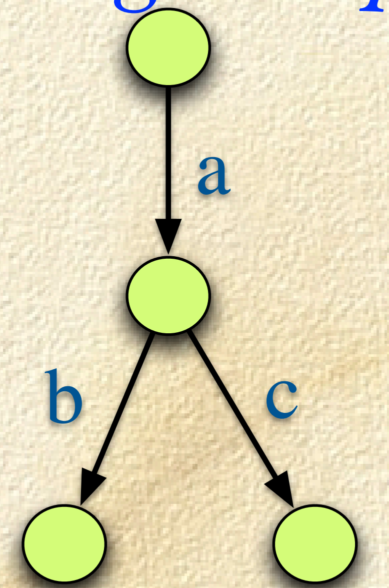
Gegenbeispiel:



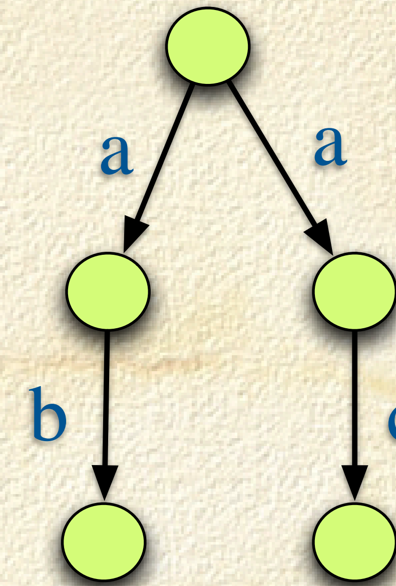
$$a \cdot (b + c) \stackrel{?}{=} a \cdot b + a \cdot c$$



Gegenbeispiel:

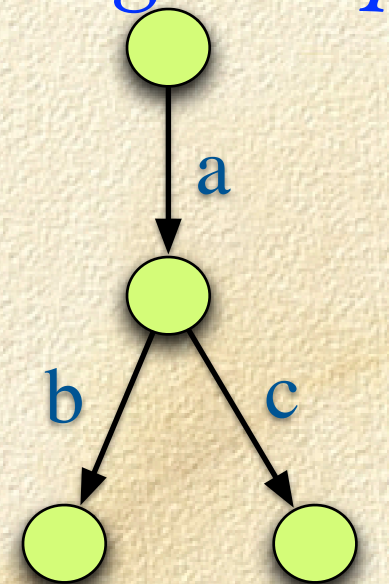


$$a \cdot (b + c) \overset{?}{=} a \cdot b + a \cdot c$$

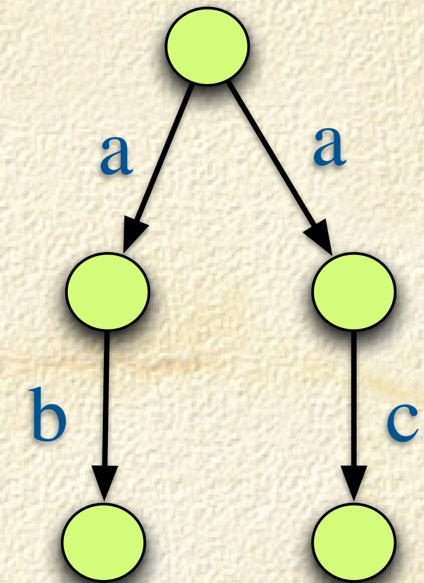


$$\begin{tikzcd} a \cdot (b + c) \arrow[d, "a"'] \\ b + c \arrow[d, bend right, "b"'] \arrow[d, bend left, "c"] \\ \surd \end{tikzcd}$$

Gegenbeispiel:

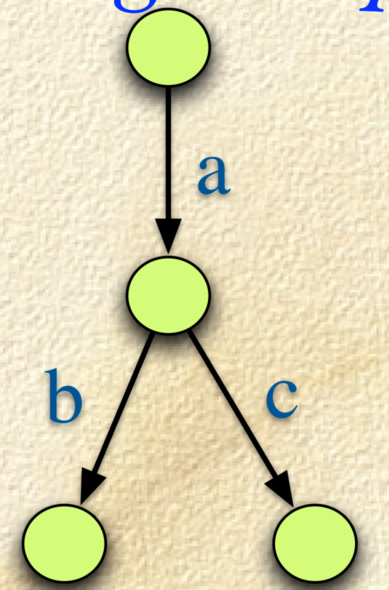


$$a \cdot (b + c) \overset{?}{=} a \cdot b + a \cdot c$$

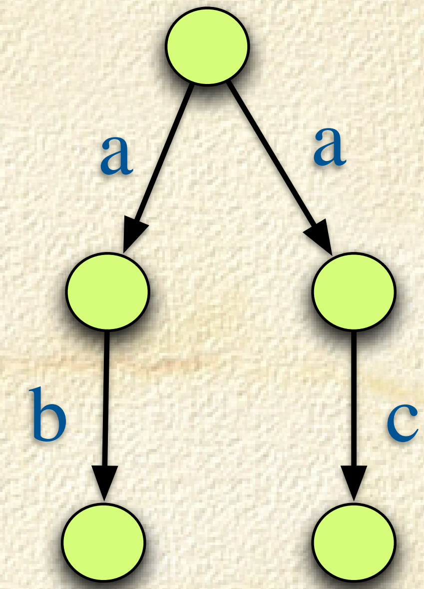


$$\begin{tikzcd} a \cdot (b + c) \arrow[d, "a"] \\ b + c \arrow[d, bend right, "b"'] \arrow[d, bend left, "c"] \\ \surd \end{tikzcd} \qquad \begin{tikzcd} & a \cdot b + a \cdot c \arrow[dl, "a"'] \arrow[dr, "a"] & \\ b \arrow[d, "b"'] & & c \arrow[d, "c"] \\ \surd & & \surd \end{tikzcd}$$

Gegenbeispiel:



$$a \cdot (b + c) \stackrel{?}{=} a \cdot b + a \cdot c$$

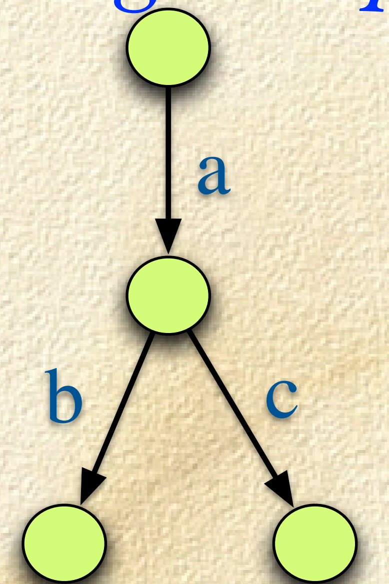


$$a \cdot (b + c) \;\text{- - -}\; a \cdot b + a \cdot c$$

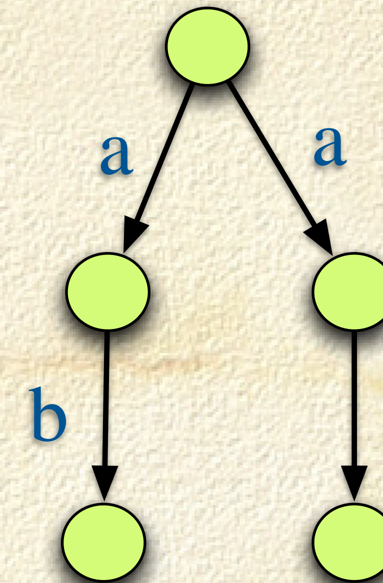
$$a \cdot (b + c) \xrightarrow{a} b + c, \quad b + c \xrightarrow{b} \surd, \quad b + c \xrightarrow{c} \surd$$

$$a \cdot b + a \cdot c \xrightarrow{a} b \xrightarrow{b} \surd, \quad a \cdot b + a \cdot c \xrightarrow{a} c \xrightarrow{c} \surd$$

Gegenbeispiel:



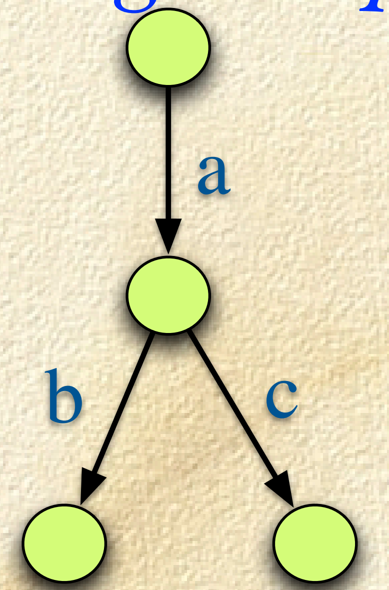
$$a \cdot (b + c) \stackrel{?}{=} a \cdot b + a \cdot c$$



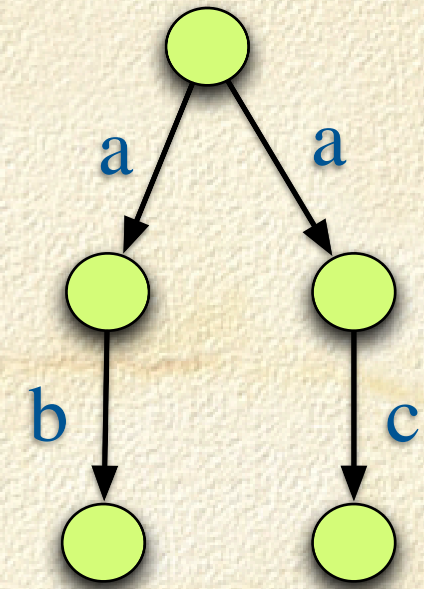
$$\begin{tikzcd}
a \cdot (b + c) \arrow[d, "a"] \arrow[r, dash, dashed] & a \cdot b + a \cdot c \arrow[d, "a"'] \arrow[dr, "a"] & \\
b + c \arrow[d, bend right, "b"'] \arrow[d, bend left, "c"] \arrow[r, dash, dashed] & b \arrow[d, "b"] & c \arrow[d, "c"] \\
\surd & \surd & \surd
\end{tikzcd}$$

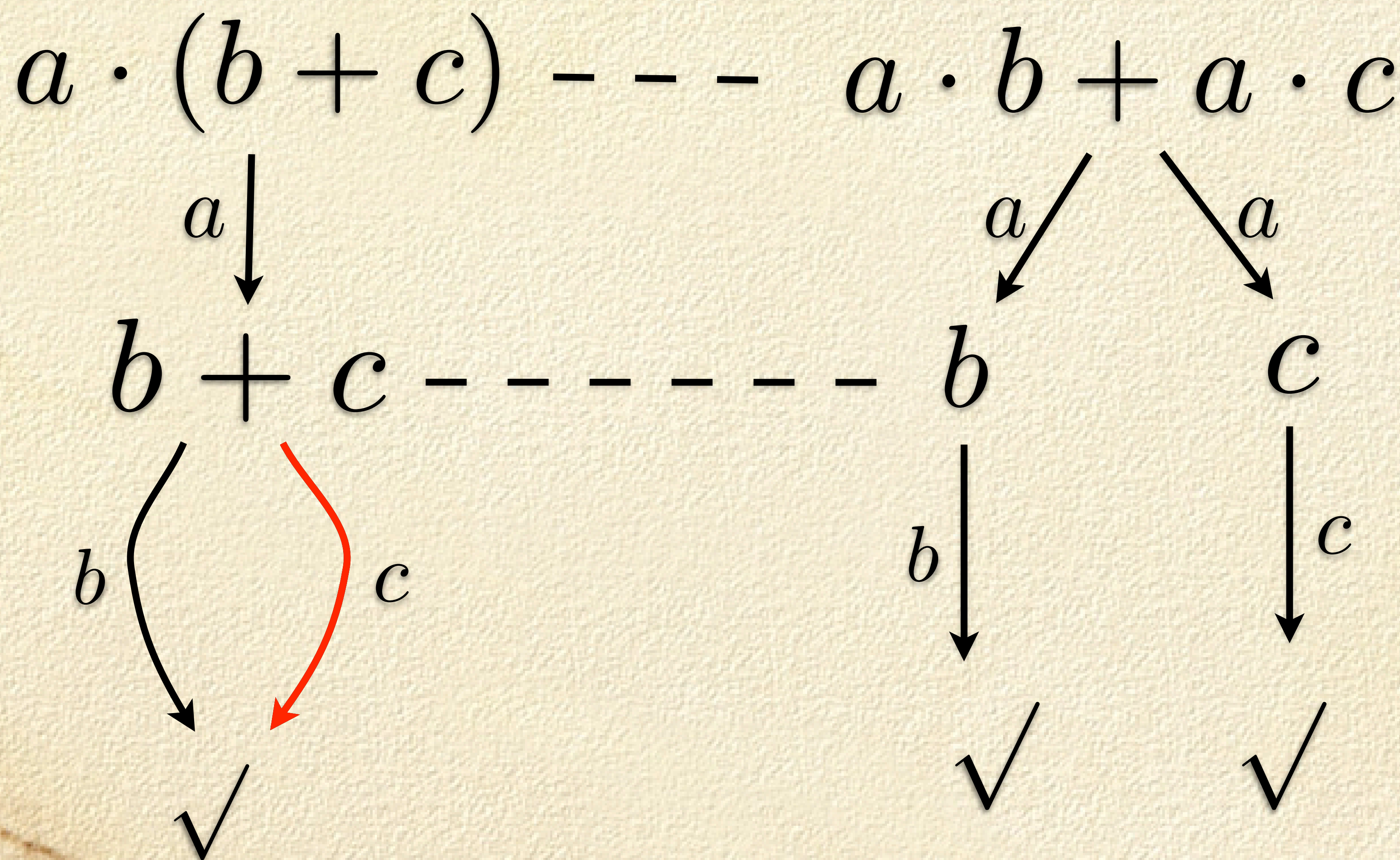
Wednesday, January 5, 2011

Gegenbeispiel:



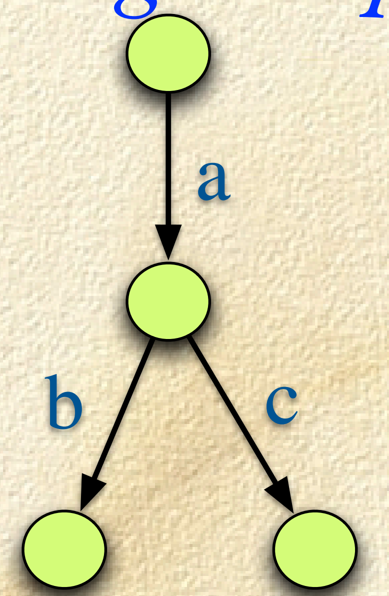
$$a \cdot (b + c) \stackrel{?}{=} a \cdot b + a \cdot c$$



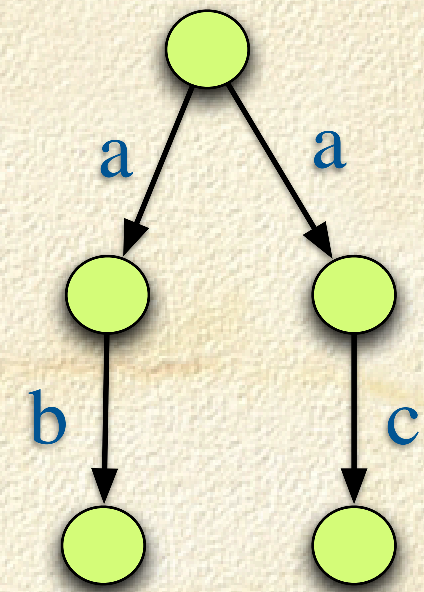


Wednesday, January 5, 2011

Gegenbeispiel:



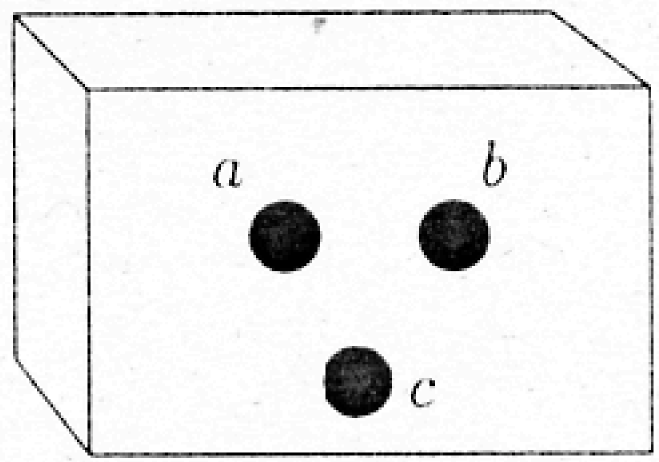
$$a \cdot (b + c) \stackrel{?}{=} a \cdot b + a \cdot c$$



$$\begin{tikzcd}[column sep=small]
a\cdot(b+c) \arrow[rr, dash, dashed] \arrow[d, "a"'] & & a\cdot b + a\cdot c \arrow[dl, "a"'] \arrow[dr, "a"] & \\
b+c \arrow[r, dash, dashed] \arrow[d, bend right, "b"'] \arrow[d, bend left, "c"] & b \arrow[d, "b"'] \arrow[dr, "?"] & & c \arrow[d, "c"] \\
\surd & \surd & & \surd
\end{tikzcd}$$

2.5 Black boxes, or reactive systems

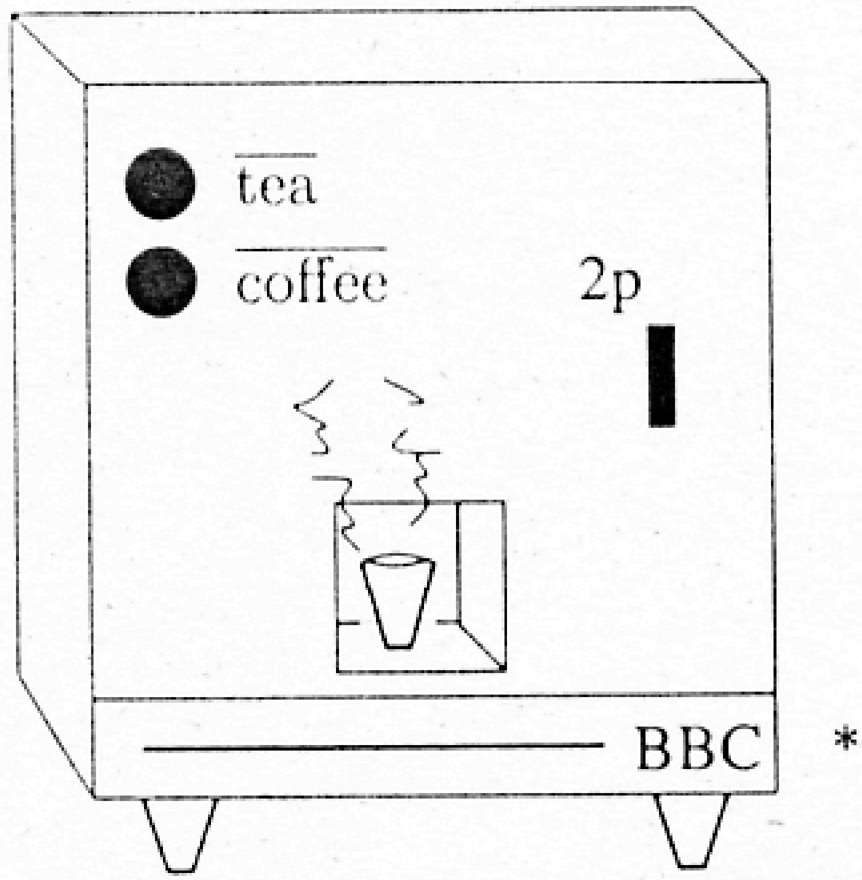
Let us now think of an automaton over $Act = \{a, b, c\}$ as a black box



with three buttons marked a, b and c. We interact with it by trying to press the buttons in some sequence. Sometimes the button goes down i.e. we succeed – and sometimes it doesn't; this is the only way we can tell the difference between black boxes with the same alphabet. In particular, we can't tell the difference between an accepting and a non-accepting *state*; we can't tell anything about states except indirectly through the behaviour of the buttons. (We shall often use the words 'button' and 'port', interchangeably, for the means of access to a process.)

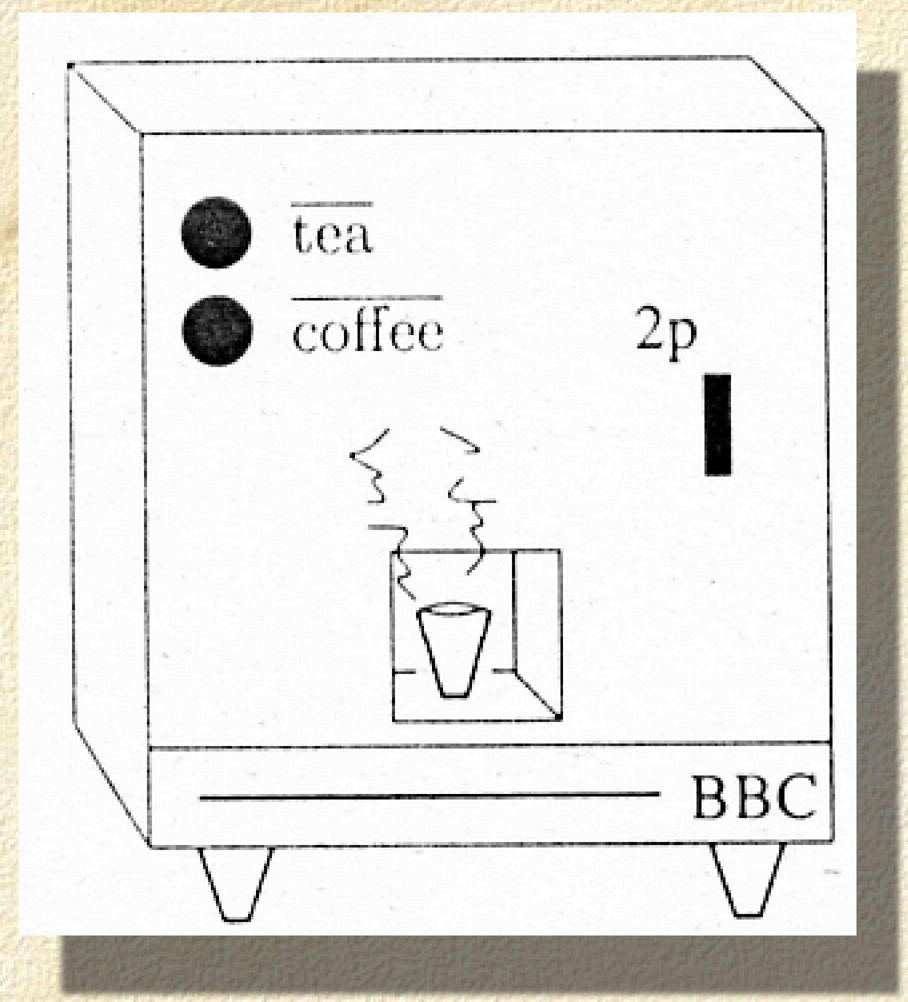
Wednesday, January 5, 2011

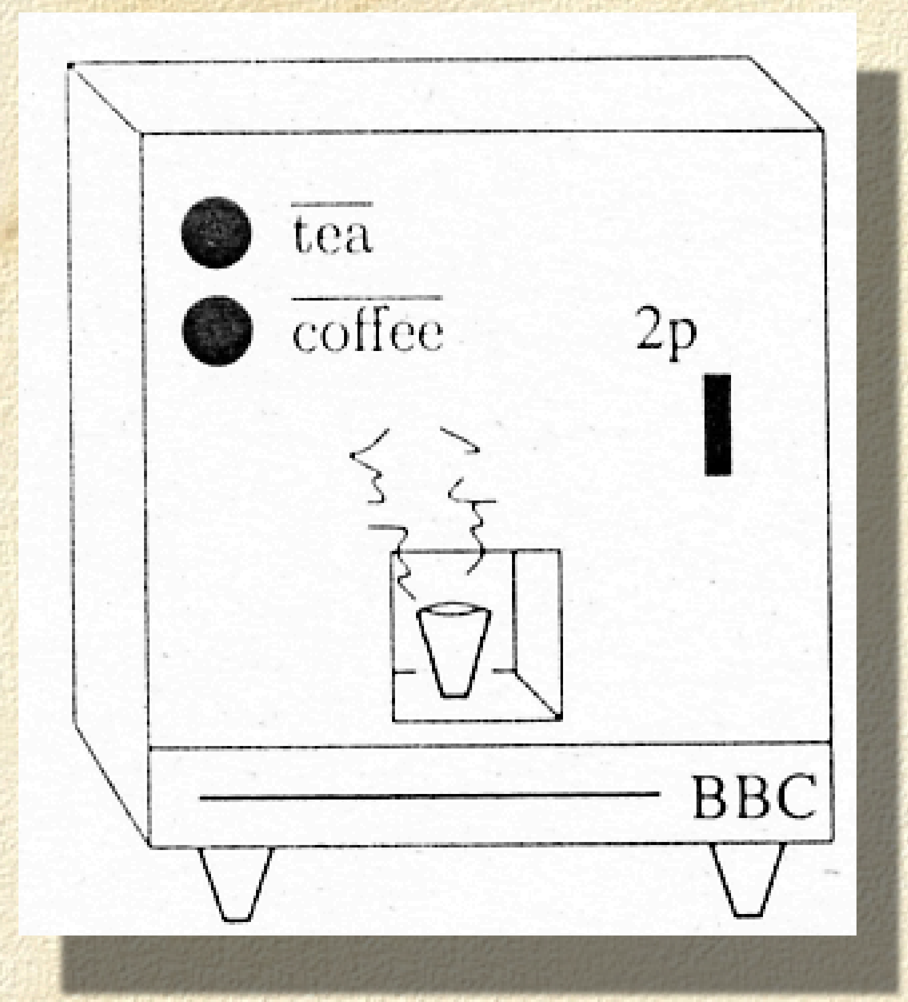
Example 2.8 Vending machine Here is a tea/coffee vending machine, a black box with a three-symbol alphabet $\{2\mathrm{p}, \overline{\mathrm{tea}}, \overline{\mathrm{coffee}}\}$. (For now ignore the overbars.)

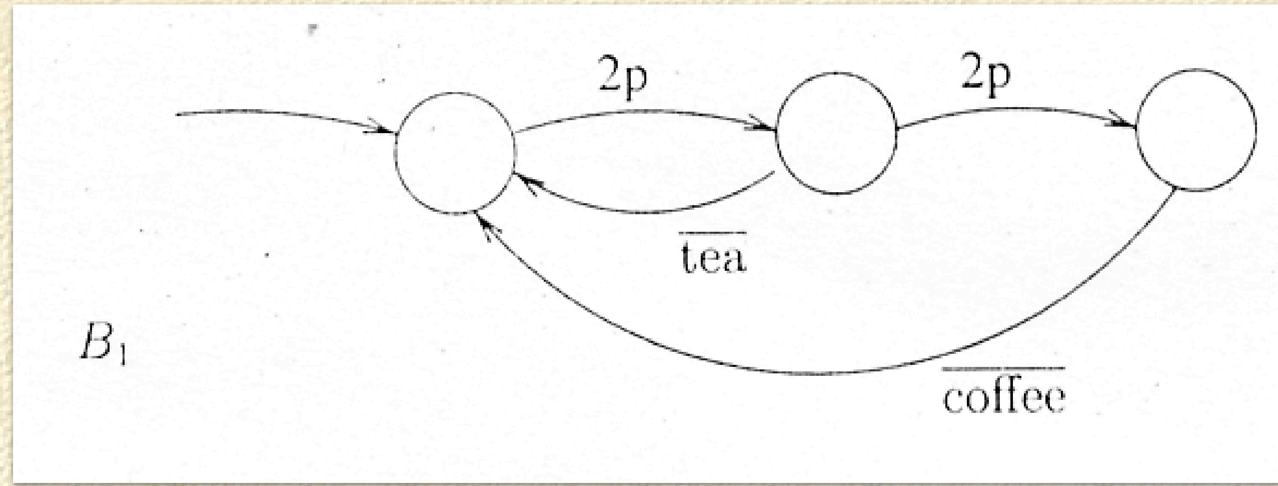


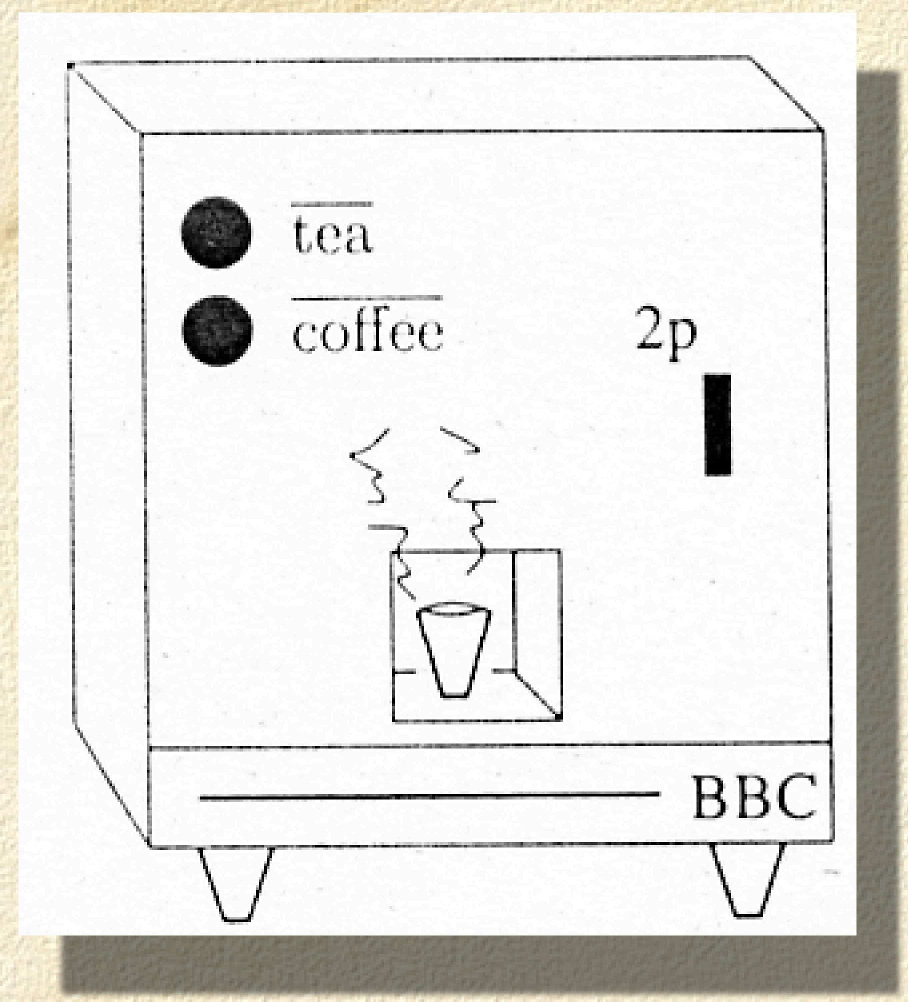
* Black Box Corporation

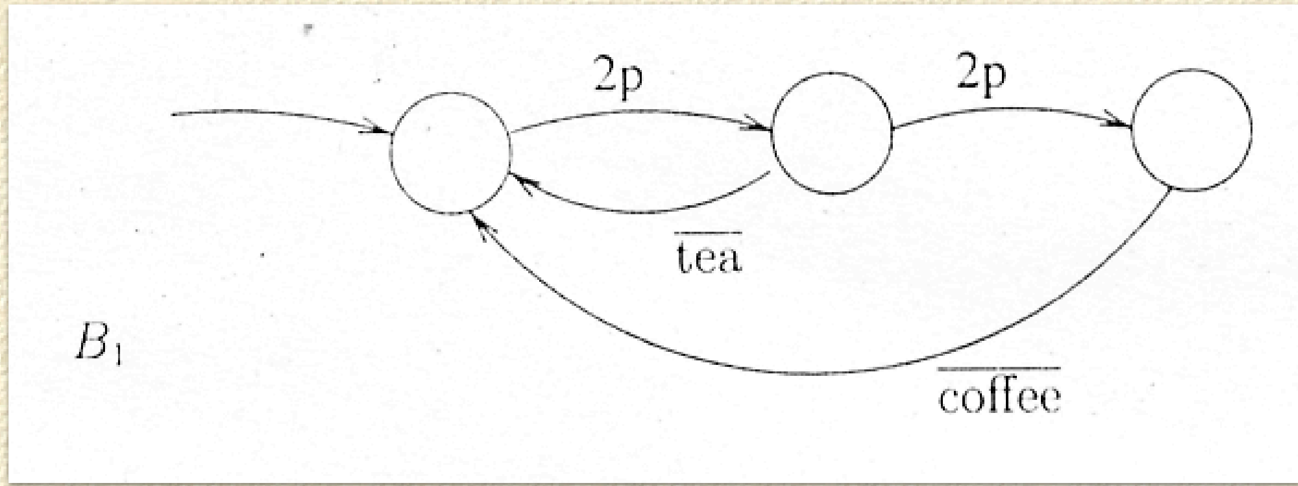
aus: R. Milner, *Communicating and Mobile Systems: the π-Calculus*, Cambridge University Press, 1999

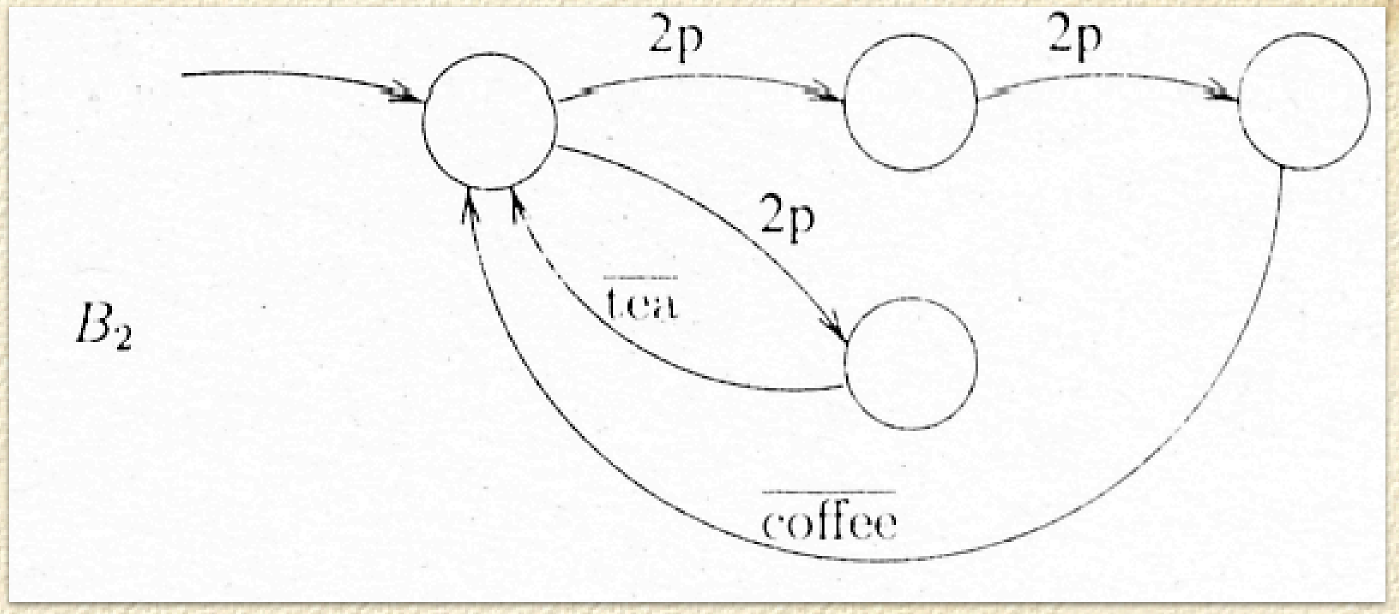

$\overline{\text{tea}}$
$\overline{\text{coffee}}$
2p
BBC


$\overline{\text{tea}}$
$\overline{\text{coffee}}$
2p
BBC


2p
2p
$\overline{\text{tea}}$
$\overline{\text{coffee}}$
B_1


$\overline{\text{tea}}$
$\overline{\text{coffee}}$
2p
BBC


2p
2p
$\overline{\text{tea}}$
B_1
$\overline{\text{coffee}}$


2p
2p
2p
$\overline{\text{tea}}$
B_2
$\overline{\text{coffee}}$

Lemma 5.1 *Die Bisimulation ist eine Äquivalenzrelation.*

$\rightarrow$ sogar $\rightarrow$ Kongruenzäquivalenz

Theorem 5.7 *Die Äquivalenz der Bisimulation ist eine Kongruenz bezgl.* $\{+, \cdot\}$ *auf BPA,*

Lemma 5.1 *Die Bisimulation ist eine Äquivalenzrelation.*

$\rightarrow$ sogar $\rightarrow$ Kongruenzäquivalenz

Theorem 5.7 *Die Äquivalenz der Bisimulation ist eine Kongruenz bezgl.* $\{+, \cdot\}$ *auf BPA,*
d.h.: falls $s \underline{\leftrightarrow} s'$ *und* $t \underline{\leftrightarrow} t'$, *dann auch*
$s + t \underline{\leftrightarrow} s' + t'$ *und* $s \cdot t \underline{\leftrightarrow} s' \cdot t'$.

Mit obenstehender Definition ist es $\boxed{\text{aufwendig}}$ zu überprüfen, ob zwei elementare Prozessterme bisimilar sind: es müssen zunächst die Prozessgraphen und dann zwischen ihnen die Bisimulationsrelation konstruiert werden. Es wird daher jetzt ein Kalkül für eine Gleichheitsrelation zwischen elementaren Prozesstermen eingeführt, der diese Aufgabe durch die Konstruktion von Ableitungen lösen soll.

Mit obenstehender Definition ist es aufwendig zu überprüfen, ob zwei elementare Prozessterme bisimilar sind: es müssen zunächst die Prozessgraphen und dann zwischen ihnen die Bisimulationsrelation konstruiert werden. Es wird daher jetzt ein Kalkül für eine Gleichheitsrelation zwischen elementaren Prozesstermen eingeführt, der diese Aufgabe durch die Konstruktion von Ableitungen lösen soll.

Mit obenstehender Definition ist es $\boxed{\text{aufwendig}}$ zu überprüfen, ob zwei elementare Prozessterme bisimilar sind: es müssen zunächst die $\boxed{\text{Prozessgraphen}}$ und dann zwischen ihnen die $\boxed{\text{Bisimulationsrelation}}$ konstruiert werden. Es wird daher jetzt ein Kalkül für eine Gleichheitsrelation zwischen elementaren Prozesstermen eingeführt, der diese Aufgabe durch die Konstruktion von Ableitungen lösen soll.

Mit obenstehender Definition ist es aufwendig zu überprüfen, ob zwei elementare Prozessterme bisimilar sind: es müssen zunächst die Prozessgraphen und dann zwischen ihnen die Bisimulationsrelation konstruiert werden. Es wird daher jetzt ein Kalkül für eine Gleichheitsrelation zwischen elementaren Prozesstermen eingeführt, der diese Aufgabe durch die Konstruktion von Ableitungen lösen soll.

Axiome des BPA-Kalküls:

$$
\begin{array}{lrcl}
\text{A1} & x + y & = & y + x \\
\text{A2} & (x + y) + z & = & x + (y + z) \\
\text{A3} & x + x & = & x
\end{array}
$$

Axiome des BPA-Kalküls:

$$
\begin{array}{lrcl}
\text{A1} & x + y & = & y + x \\
\text{A2} & (x + y) + z & = & x + (y + z) \\
\text{A3} & x + x & = & x \\
\text{A4} & (x + y) \cdot z & = & x \cdot z + y \cdot z \\
\text{A5} & (x \cdot y) \cdot z & = & x \cdot (y \cdot z)
\end{array}
$$

Schlussregeln des BPA-Kalküls:

- (SUBSTITUTION)

  Sei σ eine Substitution, die alle Variablen durch Terme aus BPA ersetzt. Ist dann $s = t$ ein Axiom, dann gilt $\sigma(s) = \sigma(t)$.

- (Äquivalenz)
  - $t = t$ für alle $t \in BPA$
  - falls $s = t$, dann $t = s$
  - falls $s = t$ und $t = u$, dann $s = u$

- (Äquivalenz)
  - $t = t$ für alle $t \in BPA$
  - falls $s = t$, dann $t = s$
  - falls $s = t$ und $t = u$, dann $s = u$
- (Kontext)

  Falls $s = s'$ und $t = t'$, dann $s + t = s' + t'$ und $s \cdot t = s' \cdot t'$.

Beispiel zu der Schlussregel 1
des BPA-Kalküls:

- (SUBSTITUTION)

  Sei σ eine Substitution, die alle Variablen durch Terme aus BPA ersetzt. Ist dann $s = t$ ein Axiom, dann gilt $\sigma(s) = \sigma(t)$.

$$\boxed{A4: \quad (x+y)z = xz + yz}$$

Substitution σ:

$$\boxed{A4: \quad (x+y)z = xz + yz}$$

Substitution σ: $\quad \begin{aligned} x &\mapsto ab \\ y &\mapsto a \\ z &\mapsto c + c \end{aligned}$

$$\boxed{A4: \quad (x+y)z = xz + yz}$$

$$\text{Substitution } \sigma: \quad \begin{array}{l} x \mapsto ab \\ y \mapsto a \\ z \mapsto c + c \end{array}$$

$$(ab + a)(c + c) = ab(c + c) + a(c + c)$$

$$\boxed{A4: \quad (x+y)z = xz + yz}$$ ⇨ Regel-„*Schema*“

$$\text{Substitution } \sigma: \quad \begin{aligned} x &\mapsto ab \\ y &\mapsto a \\ z &\mapsto c+c \end{aligned}$$

$$(ab+a)(c+c) = ab(c+c) + a(c+c)$$

Beispiel zu der Schlussregel 2 des BPA-Kalküls:

- (Äquivalenz)
  - $t = t$ für alle $t \in BPA$
  - falls $s = t$, dann $t = s$
  - falls $s = t$ und $t = u$, dann $s = u$

Transitivität

$$\frac{(x+y)+y = \quad x+(y+y) \qquad\qquad x+(y+y) \quad = x+y}{(x+y)+y = \qquad\qquad\qquad\qquad = x+y}$$

Beispiel zu der Schlussregel 3

des BPA-Kalküls:

- (Kontext) Falls $s = s'$ und $t = t'$, dann $s + t = s' + t'$ und $s \cdot t = s' \cdot t'$.

Beispiel zu der Schlussregel 3 des BPA-Kalküls:

- (KONTEXT) Falls $s = s'$ und $t = t'$, dann $s + t = s' + t'$ und $s \cdot t = s' \cdot t'$.

$$\frac{\begin{array}{ccc} x & = & x \\ y + y & = & y \end{array}}{}$$

Beispiel zu der Schlussregel 3

des BPA-Kalküls:

- (KONTEXT) Falls $s = s'$ und $t = t'$, dann $s + t = s' + t'$ und $s \cdot t = s' \cdot t'$.

$$\frac{\begin{array}{rcl} x & = & x \\ y + y & = & y \end{array}}{x + (y + y) \quad = \quad x + y}$$

Um die Relation *bisimilar* mittels des BPA-Kalküls zu berechen, muß natürlich bewiesen werden, dass zwei Terme genau dann bisimilar sind, wenn sie äquivalent sind.

Um die Relation *bisimilar* mittels des BPA-Kalküls zu berechen, muß natürlich bewiesen werden, dass zwei Terme genau dann bisimilar sind, wenn sie äquivalent sind. Traditionell wird diese Eigenschaft in zwei Teilen als *Korrektheit* und *Vollständigkeit* des BPA-Kalküls formuliert und bewiesen.

Korrektheit (soundness) Das BPA-Kalkül heißt *korrekt (sound)*, wenn für alle Terme $s, t \in BPA$ aus $s = t$ auch $s \underline{\leftrightarrow} t$ folgt.

Korrektheit **(soundness)** Das BPA-Kalkül heißt *korrekt (sound)*, wenn für alle Terme $s, t \in BPA$ aus $s = t$ auch $s \underline{\leftrightarrow} t$ folgt.

Vollständigkeit **(completeness)** Das BPA-Kalkül heißt *vollständig (complete)*, wenn für alle Terme $s, t \in BPA$ aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt.

Satz 5.8 *Der BPA-Kalkül ist korrekt.*

Beweis:

Es wird hier nur die Beweisstruktur dargestellt. Wie fast immer bei Beweisen für die Korrektheit eines Kalküls ist zu zeigen:

1.) Die Axiome sind korrekt.

2.) Die Regeln überführen korrekte Terme in korrekte Terme.

zu 1.): Erfolgt mit 2 a): Substitution

zu 2.):

a) Substitution: Es ist $\sigma(s) \underline{\leftrightarrow} \sigma(t)$ für jedes Axiom $s = t$ zu beweisen, wobei σ eine Substitution ist, die alle Variablen in s und t auf elementare Prozessterme abbildet. Dabei ist auf die Definition der Bisimulation zurückzugreifen. Dies wird hier nicht ausgeführt. Informell steht dahinter in Bezug auf die Axiome A1 ... A5 folgende Argumentation:

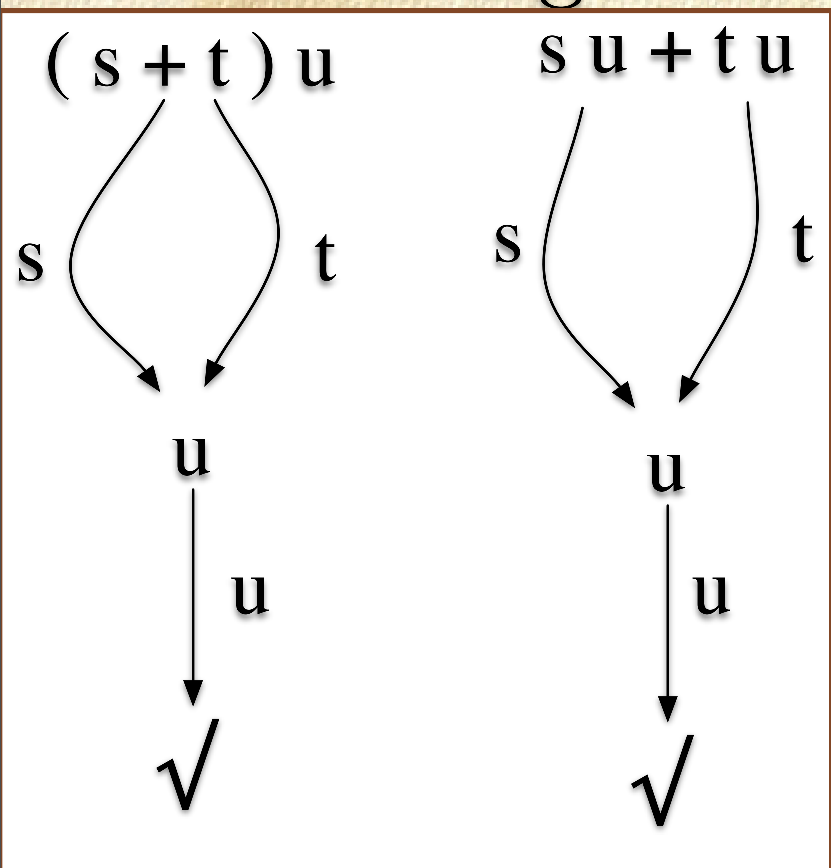
- A1: Die Terme $s + t$ und $t + s$ stellen beide eine Auswahl zwischen s und t dar.
- A2: Die Terme $(s + t) + u$ und $s + (t + u)$ stellen beide eine Auswahl zwischen s, t und u dar.
- A3: Eine Auswahl zwischen t und t ist eine Wahl für t.

- A4: Die Terme $(s + t) \cdot u$ und $s \cdot u + t \cdot u$ stellen beide eine Auswahl zwischen s und t dar, worauf u ausgeführt wird.
- A5: Die Terme $(s \cdot t) \cdot u$ und $s \cdot (t \cdot u)$ stellen beide die Aktion s dar, gefolgt von t und u.

b) Gilt, da die Bisimulations-Relation eine Äquivalenz ist.

c) Gilt, da die Bisimulations-Relation eine Kongruenz ist.

- A4: Die Terme $(s+t)\cdot u$ und $s\cdot u + t\cdot u$ stellen beide eine Auswahl zwischen s und t dar, worauf u ausgeführt wird.



erme $(s\cdot t)\cdot u$ und $s\cdot(t\cdot u)$ stellen beide s dar, gefolgt von t und u.

e Bisimulations-Relation eine Äquivalenz

e Bisimulations-Relation eine Kongruenz ist.

- A4: Die Terme $(s+t)\cdot u$ und $s\cdot u + t\cdot u$ stellen beide eine Auswahl zwischen s und t dar, worauf u ausgeführt wird.
- A5: Die Terme $(s\cdot t)\cdot u$ und $s\cdot(t\cdot u)$ stellen beide die Aktion s dar, gefolgt von t und u.

b) Gilt, da die Bisimulations-Relation eine Äquivalenz ist.

c) Gilt, da die Bisimulations-Relation eine Kongruenz ist.

Satz 5.10 *Der BPA-Kalkül ist vollständig.*

Beweis:

Zu beweisen ist also, dass aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{\mathrm{AC}}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{\mathrm{AC}} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

Satz 5.10 *Der BPA-Kalkül ist vollständig.*

Beweis:

Zu beweisen ist also, dass aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{\mathrm{AC}}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{\mathrm{AC}} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

$t =_{AC} s$ gdw nur die Gleichungen A1 und A2 werden benutzt

Satz 5.10 *I*

A1	$x + y$	$=$	$y + x$
A2	$(x+y)+z$	$=$	$x+(y+z)$

Beweis:

Zu beweisen ist also, dass aus $s \overset{*}{\underline{\leftrightarrow}} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{\mathrm{AC}}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{\mathrm{AC}} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

$t =_{AC} s$ gdw nur die Gleichungen A1 und A2 werden benutzt

Satz 5.10 *l*

A1	$x + y$	$=$	$y + x$
A2	$(x + y) + z$	$=$	$x + (y + z)$

Beweis:

Zu beweisen ist also, dass aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{AC}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{AC} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

$t =_{AC} s$ gdw nur die Gleichungen A1 und A2 werden benutzt

Repräsentanten haben die Form:

Satz 5.10 *[…]*

A1	$x + y$	$=$	$y + x$
A2	$(x + y) + z$	$=$	$x + (y + z)$

Beweis:

Zu beweisen ist also, dass aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{AC}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{AC} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

$t =_{AC} s$ gdw nur die Gleichungen A1 und A2 werden benutzt

Repräsentanten haben die Form:

$s_1 + s_2 + \ldots.\ s_n$

Satz 5.10 *[...]*

A1	$x + y$	$=$	$y + x$
A2	$(x + y) + z$	$=$	$x + (y + z)$

Beweis:

Zu beweisen ist also, dass aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{AC}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{AC} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

$t =_{AC} s$ gdw nur die Gleichungen A1 und A2 werden benutzt

Repräsentanten haben die Form:

$s_1 + s_2 + \ldots.\ s_n$

$s_i = a$ oder $s_i = a \cdot t$

Satz 5.10 *I*

A1	$x + y$	$=$	$y + x$
A2	$(x + y) + z$	$=$	$x + (y + z)$

Beweis:

Zu beweisen ist also, dass aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{AC}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{AC} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

$t =_{AC} s$ gdw nur die Gleichungen A1 und A2 werden benutzt

Repräsentanten haben die Form:

$s_1 + s_2 + \ldots.\ s_n$

$s_i = a$ oder $s_i = a \cdot t$

$a+bc =_{AC} bc+a$

Satz 5.10 *[partially covered]*

A1	$x + y$	$=$	$y + x$
A2	$(x + y) + z$	$=$	$x + (y + z)$

Beweis:

Zu beweisen ist also, dass aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{AC}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{AC} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

$t =_{AC} s$ gdw nur die Gleichungen A1 und A2 werden benutzt

Repräsentanten haben die Form:

$s_1 + s_2 + \ldots\ s_n$

$s_i = a$ oder $s_i = a \cdot t$

$(a+b)c \neq_{AC} ac+bc$

$a+bc =_{AC} bc+a$

Satz 5.10 *I*

$$
\begin{array}{lrcl}
\text{A1} & x+y & = & y+x \\
\text{A2} & (x+y)+z & = & x+(y+z)
\end{array}
$$

Beweis:

Zu beweisen ist also, dass aus $s \underline{\leftrightarrow} t$ auch $s = t$ folgt. Zunächst wird der Beweis für die einfachere Relation $=_{\mathrm{AC}}$ bewiesen, d.h. $s \underline{\leftrightarrow} t \Rightarrow s =_{\mathrm{AC}} t$. Daraus wird dann dann die Behauptung des Satzes abgeleitet.

$t =_{AC} s$ gdw nur die Gleichungen A1 und A2 werden benutzt

Repräsentanten haben die Form:

$s_1 + s_2 + \ldots\ s_n$

$s_i = a$ oder $s_i = a \cdot t$

$(a+b)c \neq_{AC} ac+bc$

$ab \neq_{AC} ba$

$a+bc =_{AC} bc+a$

Die übrigen Axiome werden in Ersetzungsregeln mit der Richtung „links nach rechts“ umgeformt:

$$
\begin{array}{lrcl}
\text{R1} & x + y & =_{\text{AC}} & y + x \\
\text{R2} & (x + y) + z & =_{\text{AC}} & x + (y + z) \\
\text{R3} & x + x & \to & x \\
\text{R4} & (x + y) \cdot z & \to & x \cdot z + y \cdot z \\
\text{R5} & (x \cdot y) \cdot z & \to & x \cdot (y \cdot z)
\end{array}
$$

Auf diese Weise erhalten wir ein Ersetzungskalkül, bei dem zwischen den Regelanwendungen Umformungen bzgl. der Relation $=_{\text{AC}}$ angewendet werden können

Die übrigen Axiome werden in Ersetzungsregeln mit der Richtung „links nach rechts“ umgeformt:

$$
\begin{array}{llrcl}
\color{blue}{\text{A1} =} & \text{R1} & x + y & =_{\text{AC}} & y + x \\
\color{blue}{\text{A2} =} & \text{R2} & (x + y) + z & =_{\text{AC}} & x + (y + z) \\
 & \text{R3} & x + x & \to & x \\
 & \text{R4} & (x + y) \cdot z & \to & x \cdot z + y \cdot z \\
 & \text{R5} & (x \cdot y) \cdot z & \to & x \cdot (y \cdot z)
\end{array}
$$

Auf diese Weise erhalten wir ein Ersetzungskalkül, bei dem zwischen den Regelanwendungen Umformungen bzgl. der Relation $=_{\text{AC}}$ angewendet werden können

Normalisierung ist nicht deterministisch.

Beispielsweise erlaubt der Term $t = ((a+a)+(b+b))c$ folgende Umformungen:

$$((a+a)+(b+b))c \xrightarrow{R3} (a+(b+b))c \xrightarrow{R3} (a+b)c \xrightarrow{R4} ac+bc$$

und

$$\begin{aligned}((a+a)+(b+b))c &\xrightarrow{R4} (a+a)c+(b+b)c\\ &\xrightarrow{R4} (ac+ac)+(b+b)c\\ &\xrightarrow{R4} (ac+ac)+(bc+bc)\\ &\xrightarrow{R3} ac+(bc+bc)\\ &\xrightarrow{R3} ac+bc\end{aligned}$$

Die Ersetzungsfolgen sind hier sogar unterschiedlich lang.

Man kann jedoch für diesen Ersetzungskalkül zeigen, dass die Normalform (trotz des Nichdeterminismus) eindeutig bestimmt ist, man also von *der* Normalform sprechen kann.

Dies folgt aus der Eigenschaft des Kalküls *konfluent* zu sein, was aber hier nicht bewiesen wird.

Wendet man die Regeln dieses Ersetzungskalküls immer wieder auf einen elementaren Prozessterm $s \in BPA$ an, so gelangt man nach einer endlichen Zahl von Schritten zu einem elementaren Prozessterm $t \in BPA$, der nicht weiter reduziert werden kann. Allgemein heißt ein solches t in Ersetzungskalkülen *Normalform.* Das Ersetzungskalkül selbst heißt *terminierend.* Dies wird üblicherweise mit einer *Gewichtsfuktion*

$$gew : BPA \mapsto \mathbb{N}$$

Wendet man die Regeln dieses Ersetzungskalküls immer wieder auf einen elementaren Prozessterm $s \in BPA$ an, so gelangt man nach einer endlichen Zahl von Schritten zu einem elementaren Prozessterm $t \in BPA$, der nicht weiter reduziert werden kann. Allgemein heißt ein solches t in Ersetzungskalkülen *Normalform.* Das Ersetzungskalkül selbst heißt *terminierend.* Dies wird üblicherweise mit einer *Gewichtsfuktion*

$$gew : BPA \mapsto \mathbb{N}$$

bewiesen, die im vorliegenden Fall folgendermaßen definiert werden kann ($v \in A, s, t \in BPA$).

$$\begin{aligned} gew(v) &:= 2 \\ gew(s+t) &:= gew(s) + gew(t) \\ gew(s \cdot t) &:= gew(s)^2 \cdot gew(t) \end{aligned}$$

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Beispiel:

$$gew(v) := 2$$

$$gew(s+t) := gew(s) + gew(t)$$

$$gew(s \cdot t) := gew(s)^2 \cdot gew(t)$$

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Beispiel:

$$
\begin{aligned}
gew(v) &:= 2 \\
gew(s+t) &:= gew(s) + gew(t) \\
gew(s \cdot t) &:= gew(s)^2 \cdot gew(t)
\end{aligned}
$$

$$
\begin{aligned}
x + x &\rightarrow x \\
(x+y) \cdot z &\rightarrow x \cdot z + y \cdot z \\
(x \cdot y) \cdot z &\rightarrow x \cdot (y \cdot z)
\end{aligned}
$$

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Beispiel:

$$gew(v) := 2$$

$$gew(s+t) := gew(s) + gew(t)$$

$$gew(s \cdot t) := gew(s)^2 \cdot gew(t)$$

$$\begin{array}{rcl} 2+2 \quad x + x & \rightarrow & x \\ (x+y) \cdot z & \rightarrow & x \cdot z + y \cdot z \\ (x \cdot y) \cdot z & \rightarrow & x \cdot (y \cdot z) \end{array}$$

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Beispiel:

$$
\begin{aligned}
gew(v) &:= 2 \\
gew(s+t) &:= gew(s) + gew(t) \\
gew(s \cdot t) &:= gew(s)^2 \cdot gew(t)
\end{aligned}
$$

$$
\begin{array}{lrcl}
2+2 & x + x & \rightarrow & x \quad 2 \\
 & (x+y) \cdot z & \rightarrow & x \cdot z + y \cdot z \\
 & (x \cdot y) \cdot z & \rightarrow & x \cdot (y \cdot z)
\end{array}
$$

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Beispiel:

$$gew(v) := 2$$

$$gew(s+t) := gew(s) + gew(t)$$

$$gew(s \cdot t) := gew(s)^2 \cdot gew(t)$$

$$
\begin{array}{rrcll}
2+2 & x + x & \longrightarrow & x & 2 \\
(2+2)^2 \cdot 2 = 32 & (x+y) \cdot z & \longrightarrow & x \cdot z + y \cdot z & \\
 & (x \cdot y) \cdot z & \longrightarrow & x \cdot (y \cdot z) &
\end{array}
$$

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Beispiel:

$$gew(v) := 2$$

$$gew(s+t) := gew(s) + gew(t)$$

$$gew(s \cdot t) := gew(s)^2 \cdot gew(t)$$

$$
\begin{array}{rrcll}
2+2 & x + x & \rightarrow & x & 2 \\
(2+2)^2 \cdot 2 = 32 & (x+y) \cdot z & \rightarrow & x \cdot z + y \cdot z & 2^2 \cdot 2 + 2^2 \cdot 2 = 16 \\
 & (x \cdot y) \cdot z & \rightarrow & x \cdot (y \cdot z) &
\end{array}
$$

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Beispiel:

$$gew(v) := 2$$

$$gew(s+t) := gew(s) + gew(t)$$

$$gew(s \cdot t) := gew(s)^2 \cdot gew(t)$$

$2+2$	$x + x$	$\rightarrow$	x	2
$(2+2)^2 \cdot 2 = 32$	$(x+y) \cdot z$	$\rightarrow$	$x \cdot z + y \cdot z$	$2^2 \cdot 2 + 2^2 \cdot 2 = 16$
$(2^2 \cdot 2)^2 \cdot 2 = 128$	$(x \cdot y) \cdot z$	$\rightarrow$	$x \cdot (y \cdot z)$	

Da $gew(s_1) > gew(s_2) > \ldots > gew(s_q) > \ldots$ für jede Ableitung $s_1, s_2, \ldots, s_q, \ldots$ gilt, kann es keine unendlichen Ableitungen geben.

Beispiel:

$$gew(v) := 2$$

$$gew(s+t) := gew(s) + gew(t)$$

$$gew(s \cdot t) := gew(s)^2 \cdot gew(t)$$

$$\begin{array}{rccl l}
2+2 & x + x & \rightarrow & x & 2 \\
(2+2)^2 \cdot 2 = 32 & (x+y) \cdot z & \rightarrow & x \cdot z + y \cdot z & 2^2 \cdot 2 + 2^2 \cdot 2 = 16 \\
(2^2 \cdot 2)^2 \cdot 2 = 128 & (x \cdot y) \cdot z & \rightarrow & x \cdot (y \cdot z) & 2^2 \cdot (2^2 \cdot 2) = 32
\end{array}$$

Terme in Normalform haben die Struktur $t_1 + \ldots + t_k$, wobei jedes t_i eine atomare Aktion $a \in A$ ist oder die Form $a \cdot s$ ($a \in A$, s in Normalform) hat. Durch Induktion über ihre Länge beweist man für Normalformen n und n' :

Lemma: $n \underline{\leftrightarrow} n' \Rightarrow n =_{\mathrm{AC}} n'$

Terme in Normalform haben die Struktur $t_1 + \ldots + t_k$, wobei jedes t_i eine atomare Aktion $a \in A$ ist oder die Form $a \cdot s$ ($a \in A$, s in Normalform) hat. Durch Induktion über ihre Länge beweist man für Normalformen n und n' :

Lemma: $n \underline{\leftrightarrow} n' \Rightarrow n =_{\mathrm{AC}} n'$

$$\boxed{a(cd) + b(cd) + a \quad \underline{\leftrightarrow} \quad a + b(cd) + a(cd)}$$

Terme in Normalform haben die Struktur $t_1 + \ldots + t_k$, wobei jedes t_i eine atomare Aktion $a \in A$ ist oder die Form $a \cdot s$ ($a \in A$, s in Normalform) hat. Durch Induktion über ihre Länge beweist man für Normalformen n und n' :

Lemma: $n \underline{\leftrightarrow} n' \Rightarrow n =_{\mathrm{AC}} n'$

$$\boxed{a(cd) + b(cd) + a \quad \underline{\leftrightarrow} \quad a + b(cd) + a(cd)}$$

Beweis:

Terme in Normalform haben die Struktur $t_1 + \ldots + t_k$, wobei jedes t_i eine atomare Aktion $a \in A$ ist oder die Form $a \cdot s$ ($a \in A$, s in Normalform) hat. Durch Induktion über ihre Länge beweist man für Normalformen n und n' :

Lemma: $n \underline{\leftrightarrow} n' \Rightarrow n =_{\mathrm{AC}} n'$

$$\boxed{a(cd) + b(cd) + a \quad \underline{\leftrightarrow} \quad a + b(cd) + a(cd)}$$

Beweis:

- Falls n einen Summanden der Form a enthält, dann gilt $n \xrightarrow{a} \surd$ und wegen $n \underline{\leftrightarrow} n'$ auch $n' \xrightarrow{a} \surd$. Also ist a auch in n' als Summand enthalten.

- Falls n einen Summanden der Form $a \cdot s$ enthält, dann gilt $n \xrightarrow{a} s$ und wegen $n \underline{\leftrightarrow} n'$ auch $n' \xrightarrow{a} t$ mit $s \underline{\leftrightarrow} t$. Also ist $a \cdot t$ in n' als Summand enthalten. Da s und t in Normalform, aber kleiner als n und n' sind, folgt durch Induktion $s =_{\mathrm{AC}} t$.

Da somit n und n' dieselben Summanden haben, gilt $n =_{\mathrm{AC}} n'$.

$$\boxed{a(cd) + b(cd) + a \quad \underline{\leftrightarrow} \quad a + b(cd) + a(cd)}$$

- Falls n einen Summanden der Form $a \cdot s$ enthält, dann gilt $n \xrightarrow{a} s$ und wegen $n \,\underline{\leftrightarrow}\, n'$ auch $n' \xrightarrow{a} t$ mit $s \,\underline{\leftrightarrow}\, t$. Also ist $a \cdot t$ in n' als Summand enthalten. Da s und t in Normalform, aber kleiner als n und n' sind, folgt durch Induktion $s =_{\text{AC}} t$.

Da somit n und n' dieselben Summanden haben, gilt $n =_{\text{AC}} n'$.

$$\boxed{a(cd) + b(cd) + a \quad \underline{\leftrightarrow} \quad a + b(cd) + a(cd)}$$

$$b(cd) \xrightarrow{a} cd$$

- Falls n einen Summanden der Form $a \cdot s$ enthält, dann gilt $n \xrightarrow{a} s$ und wegen $n \underline{\leftrightarrow} n'$ auch $n' \xrightarrow{a} t$ mit $s \underline{\leftrightarrow} t$. Also ist $a \cdot t$ in n' als Summand enthalten. Da s und t in Normalform, aber kleiner als n und n' sind, folgt durch Induktion $s =_{\mathrm{AC}} t$.

Da somit n und n' dieselben Summanden haben, gilt $n =_{\mathrm{AC}} n'$.

$$\begin{tikzcd}
a(cd) + b(cd) + a \arrow[d, "a"] \arrow[r, phantom, "\underline{\leftrightarrow}"] & a + b(cd) + a(cd) \arrow[d, "a"] \\
cd \arrow[r, phantom, "\underline{\leftrightarrow}"] & cd
\end{tikzcd}$$

Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt:
$s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme: $s \ \underline{\longleftrightarrow}\ t$

Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:

$$\begin{tikzcd} s \arrow[r, leftrightarrow] \arrow[d, dashed, red] & t \\ n & \end{tikzcd}$$

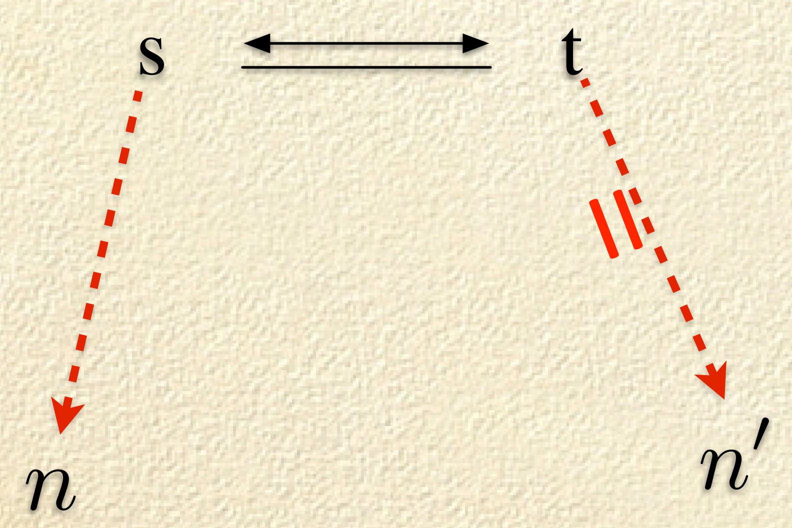
Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:

$$\begin{tikzcd} s \arrow[r, leftrightarrow] \arrow[d, dashed] & t \arrow[d, dashed] \\ n & n' \end{tikzcd}$$

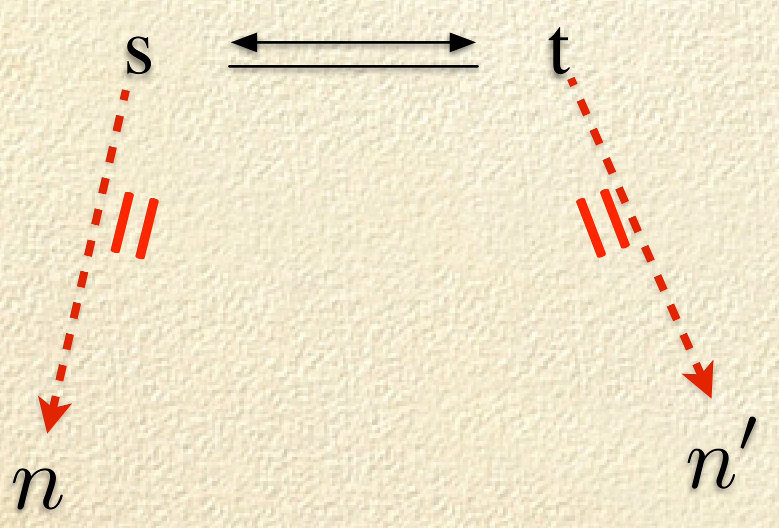
Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:



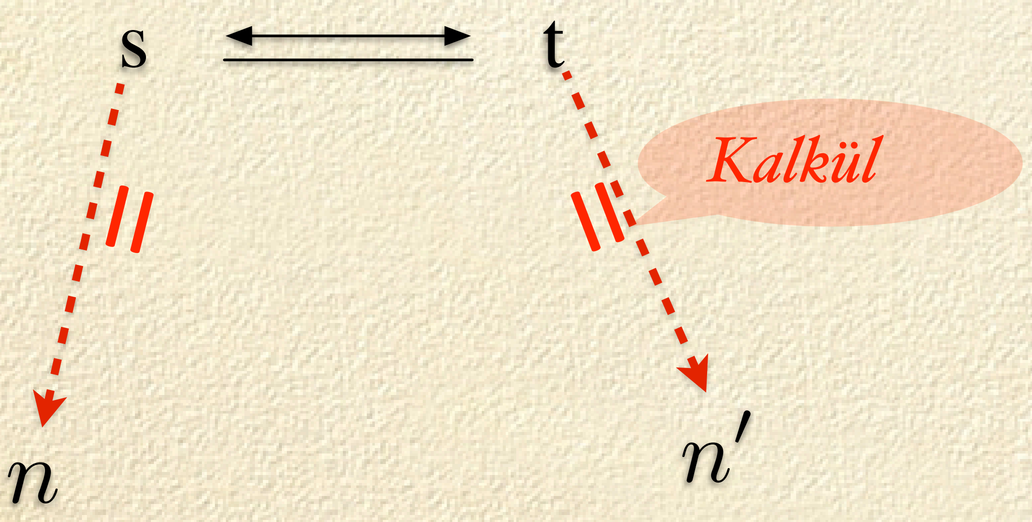
Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:



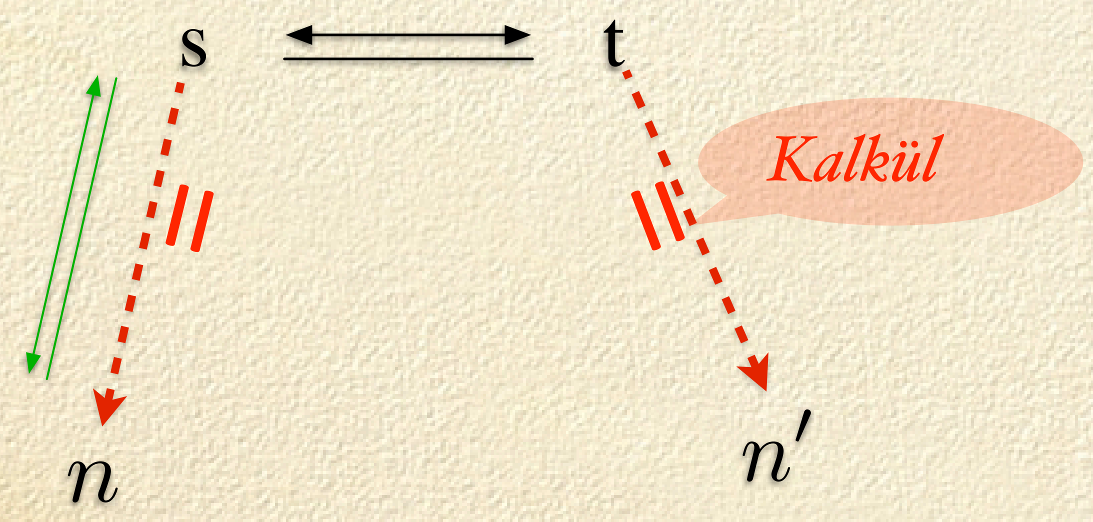
Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:



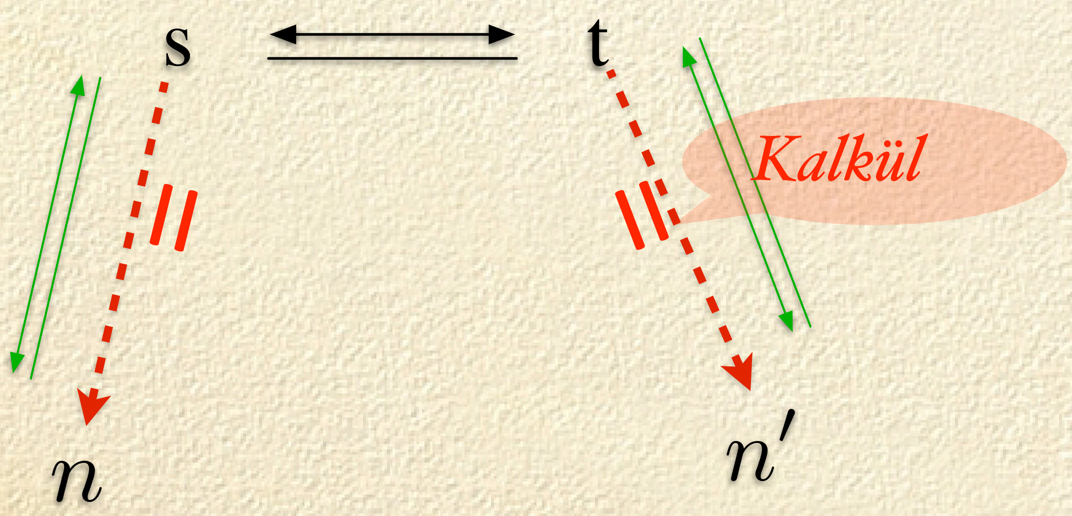
Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:

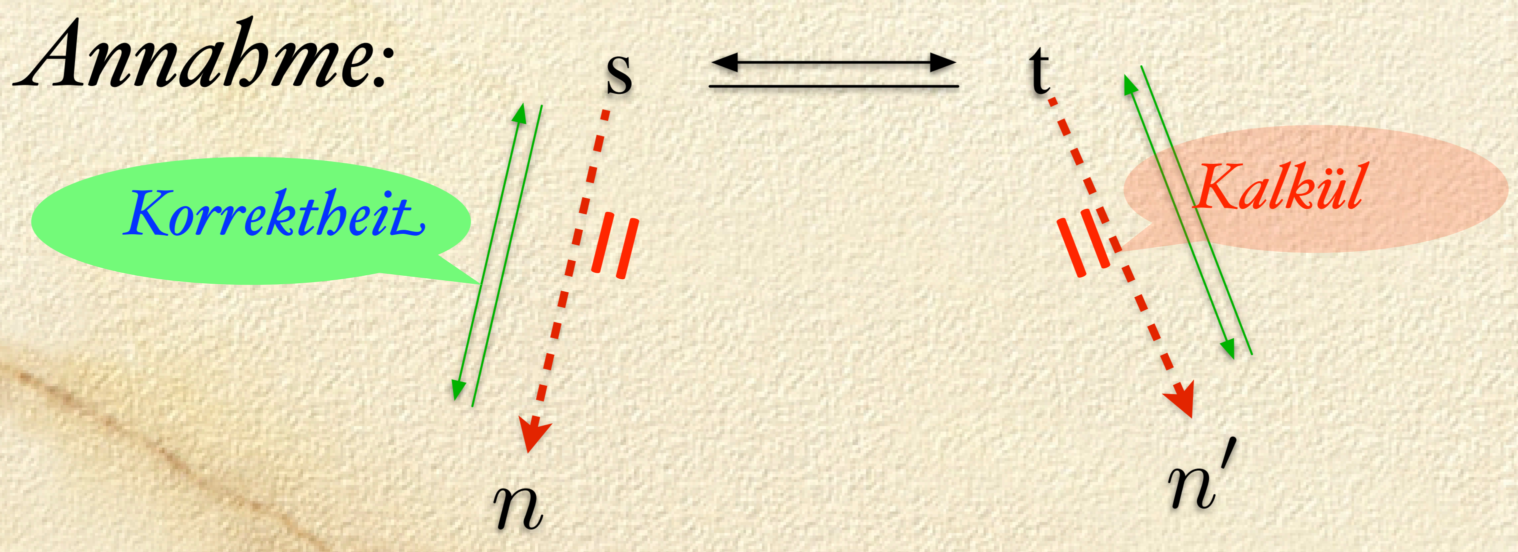


Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:

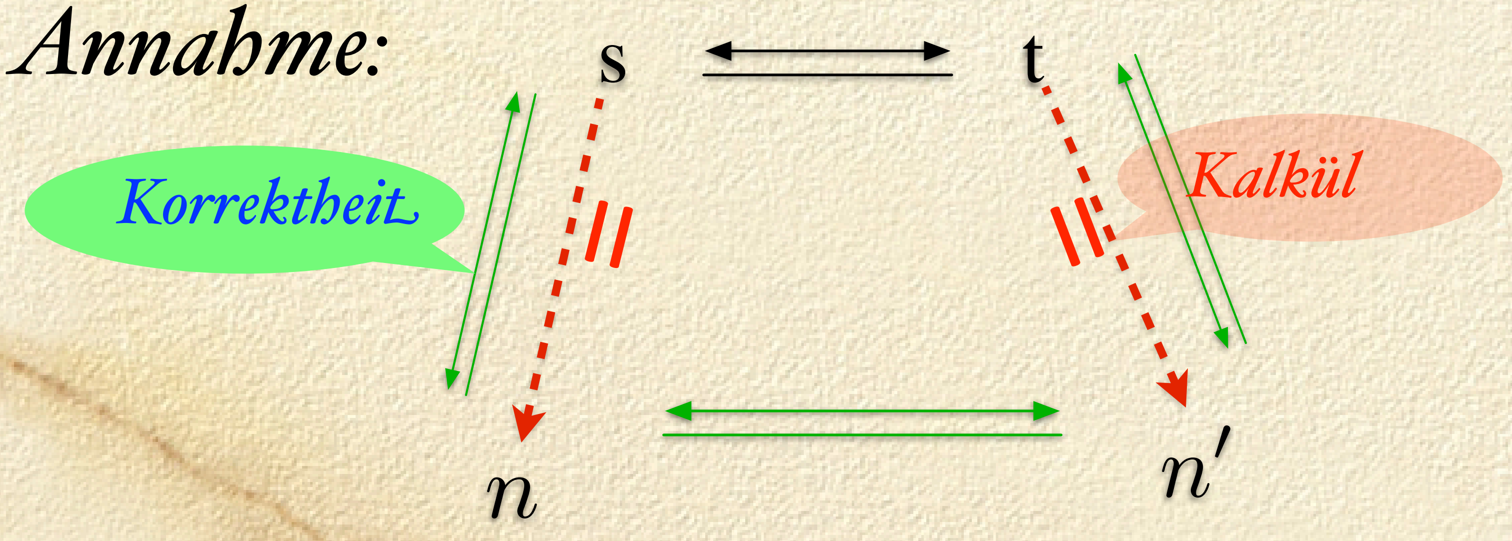


Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt:
$s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.



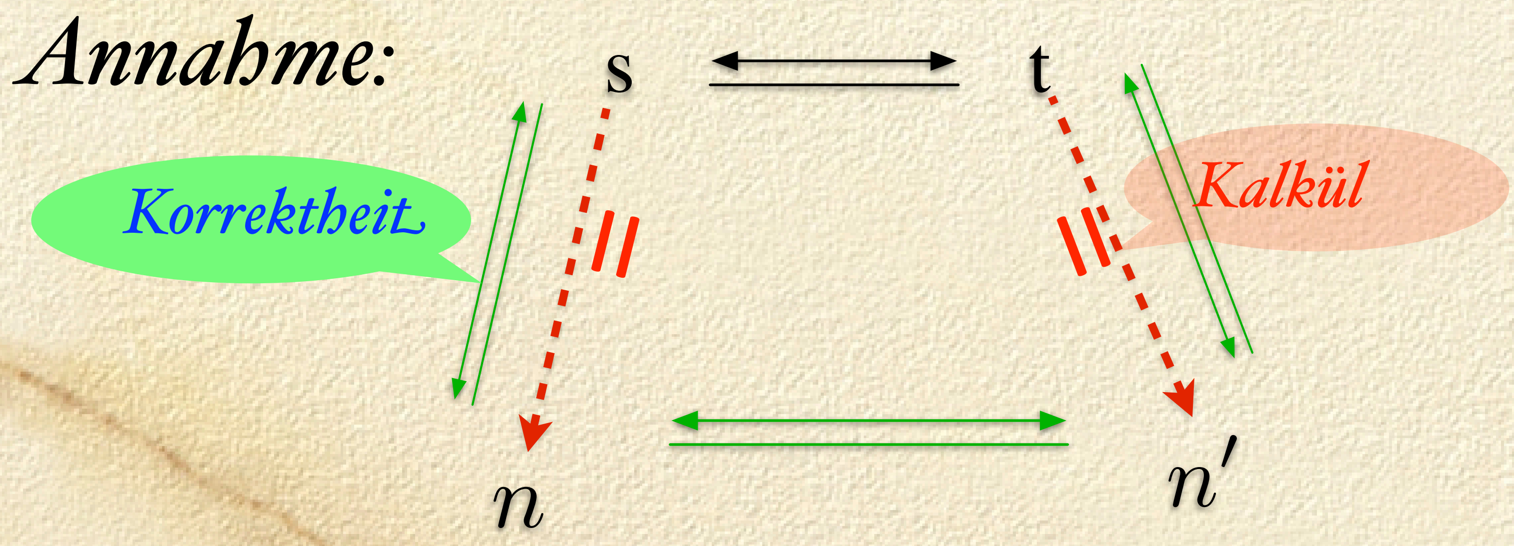
Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:



Wednesday, January 5, 2011

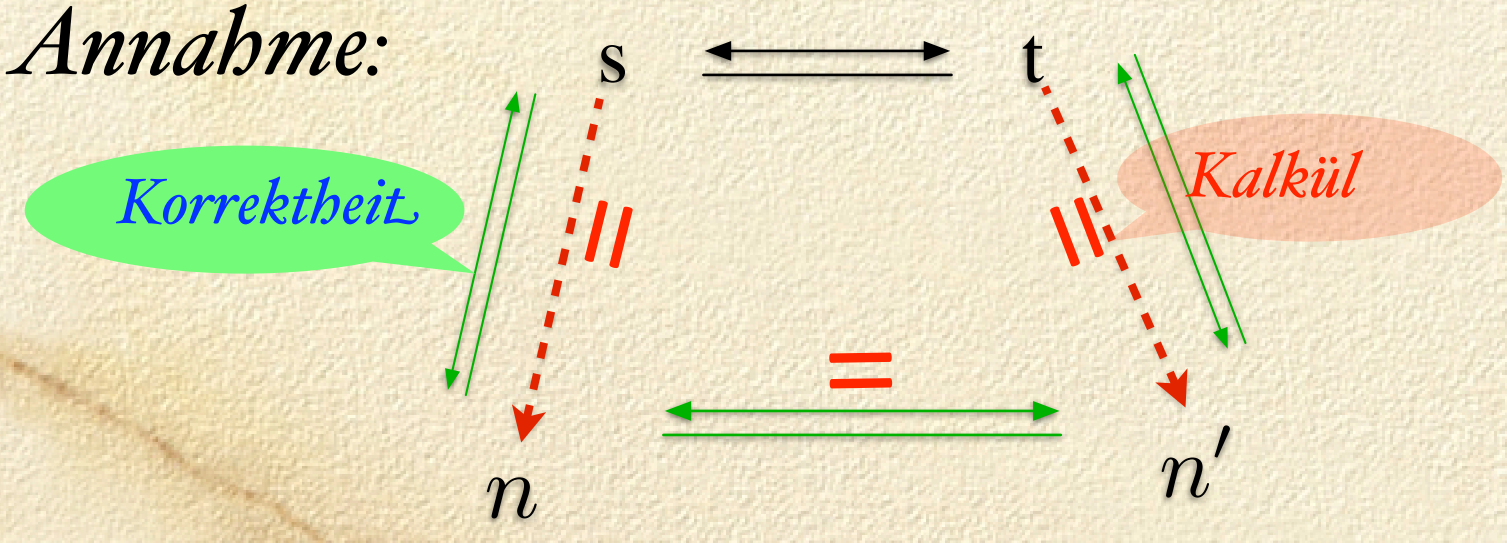
Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.



Lemma: $n \underline{\leftrightarrow} n' \Rightarrow n =_{\mathrm{AC}} n'$

Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt:
$s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

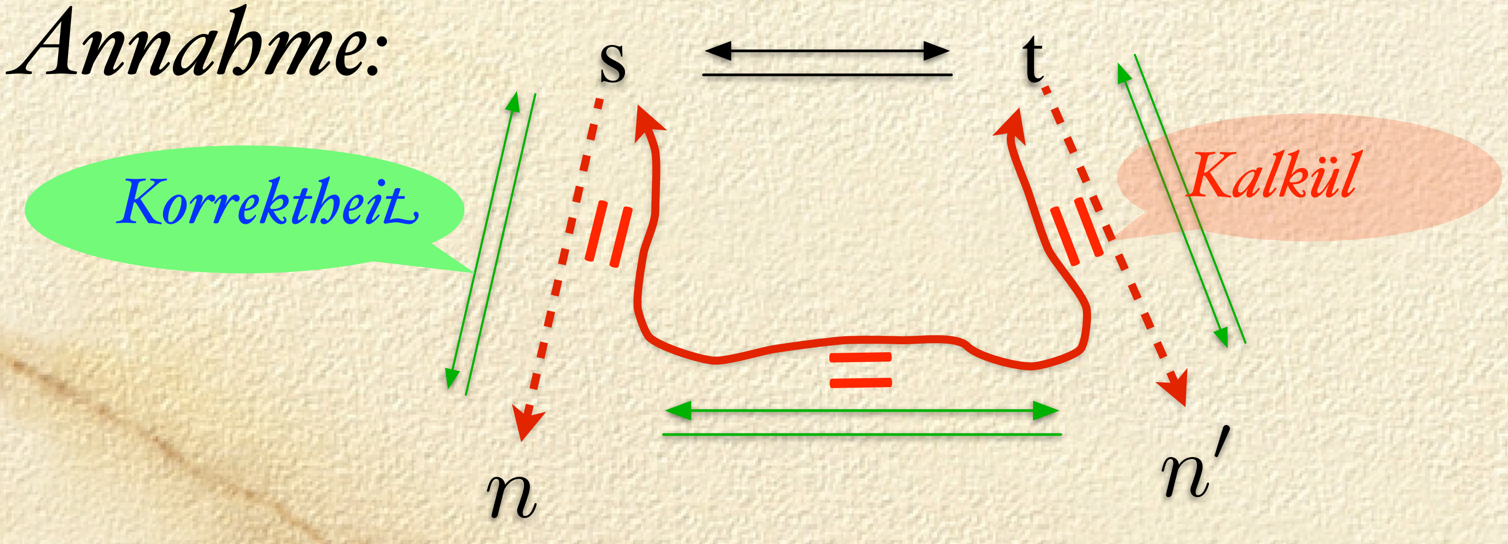
Annahme:



Lemma: $n \underline{\leftrightarrow} n' \Rightarrow n =_{\mathrm{AC}} n'$

Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt: $s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

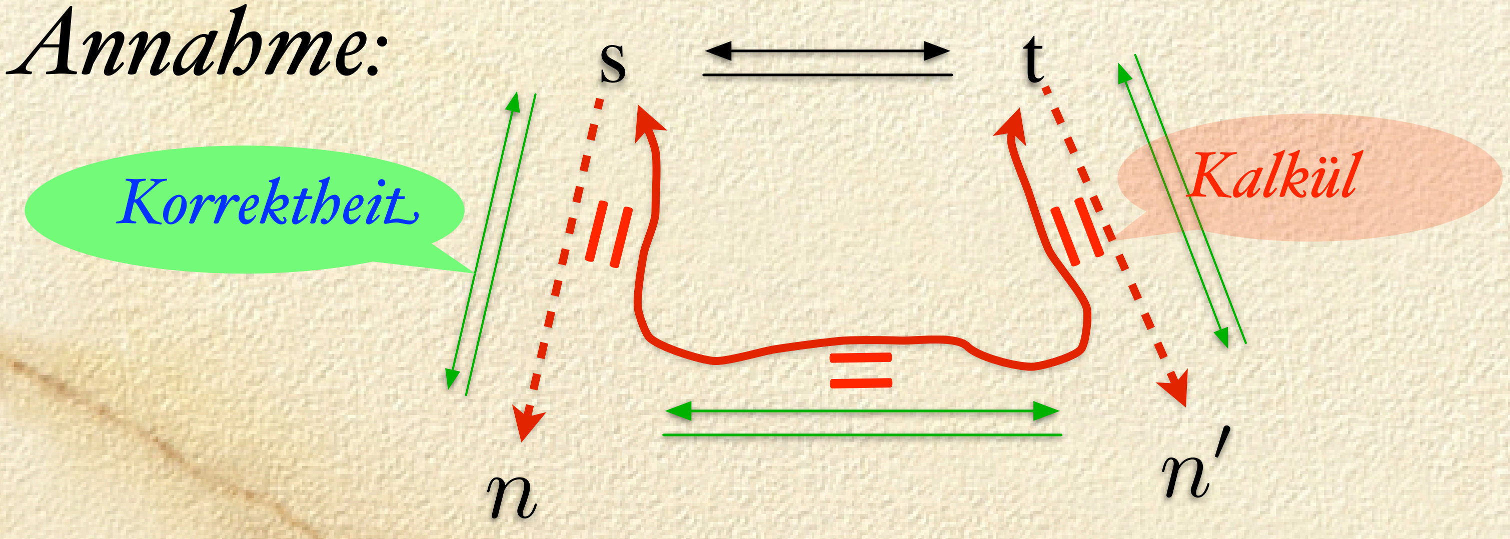
Annahme:



Lemma: $n \underline{\leftrightarrow} n' \Rightarrow n =_{\mathrm{AC}} n'$

Um nun den Beweis von $\boxed{s \underline{\leftrightarrow} t \Rightarrow s = t}$ zu führen, sei $s \underline{\leftrightarrow} t$ angenommen. s und t können durch das Ersetzungssystem zu Normalformen n und n' reduziert werden. Dies könnte auch durch den Kalkül geschehen, d.h. es gilt: $s = n$ und $t = n'$. Aus der Korrektheit des Kalküls folgt $s \underline{\leftrightarrow} n$ und $t \underline{\leftrightarrow} n'$, also insgesamt $n \underline{\leftrightarrow} n'$. Für solche Normalformen wurde aber oben gezeigt: $n =_{\mathrm{AC}} n'$. Damit ergibt sich insgesamt:
$s = n =_{\mathrm{AC}} n' = t$, d.h. $s = t$.

Annahme:



Lemma: $n \underline{\leftrightarrow} n' \Rightarrow n =_{\mathrm{AC}} n'$

also: s = t

Anmerkung: Aus dem Beweis ergibt sich ein **Verfahren** , mit dem man durch einen Kalkül $s \underline{\leftrightarrow} t$ bzw. $s = t$ entscheiden kann. (siehe das folgende Beispiel).

Algorithmus 4.1 (Entscheiden von $s \underline{\leftrightarrow} t$ bzw. $s = t$)

Input - Zwei Prozessterme s und t.
Output - TRUE falls $s = t$; FALSE falls die Eigenschaft $s = t$ nicht erfüllt ist.

1. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf s an.
2. Nenne das Ergebnis n (n ist ein Prozessterm in Normalform).
3. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf t an.
4. Nenne das Ergebnis n' (n' ist ein Prozessterm in Normalform).
5. Falls $n =_{AC} n'$, gebe TRUE aus, sonst FALSE.
   (Dieser Schritt kann mit den Regeln R1 und R2 durchgeführt werden (wobei auf Termination oder durch andere Verfahren der Textverarbeitung.)

$$
\begin{array}{lrcl}
\text{R1} & x + y & =_{\text{AC}} & y + x \\
\text{R2} & (x + y) + z & =_{\text{AC}} & x + (y + z) \\
\text{R3} & x + x & \rightarrow & x \\
\text{R4} & (x + y) \cdot z & \rightarrow & x \cdot z + y \cdot z \\
\text{R5} & (x \cdot y) \cdot z & \rightarrow & x \cdot (y \cdot z)
\end{array}
$$

Algorithmus 4.1 (Entscheiden von $s \underline{\leftrightarrow} t$ bzw. $s = t$)

Input - Zwei Prozessterme s und t.
Output - TRUE falls $s = t$; FALSE falls die Eigenschaft $s = t$ nicht erfüllt ist.

1. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf s an.
2. Nenne das Ergebnis n (n ist ein Prozessterm in Normalform).
3. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf t an.
4. Nenne das Ergebnis n' (n' ist ein Prozessterm in Normalform).
5. Falls $n =_{AC} n'$, gebe TRUE aus, sonst FALSE.
   (Dieser Schritt kann mit den Regeln R1 und R2 durchgeführt werden (wobei auf Termination oder durch andere Verfahren der Textverarbeitung.)

$$\begin{array}{lrcl} \text{R1} & x+y & =_{\text{AC}} & y+x \\ \text{R2} & (x+y)+z & =_{\text{AC}} & x+(y+z) \\ \text{R3} & x+x & \to & x \\ \text{R4} & (x+y)\cdot z & \to & x\cdot z + y\cdot z \\ \text{R5} & (x\cdot y)\cdot z & \to & x\cdot(y\cdot z) \end{array}$$

Algorithmus 4.1 (Entscheiden von $s \underline{\leftrightarrow} t$ bzw. $s = t$)

Input - Zwei Prozessterme s und t.
Output - TRUE falls $s = t$; FALSE falls die Eigenschaft $s = t$ nicht erfüllt ist.

1. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf s an.
2. Nenne das Ergebnis n (n ist ein Prozessterm in Normalform).
3. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf t an.
4. Nenne das Ergebnis n' (n' ist ein Prozessterm in Normalform).
5. Falls $n =_{AC} n'$, gebe TRUE aus, sonst FALSE.
   (Dieser Schritt kann mit den Regeln R1 und R2 durchgeführt werden (wobei auf Termination oder durch andere Verfahren der Textverarbeitung.)

$$
\begin{array}{lrcl}
\text{R1} & x + y & =_{\text{AC}} & y + x \\
\text{R2} & (x + y) + z & =_{\text{AC}} & x + (y + z) \\
\text{R3} & x + x & \to & x \\
\text{R4} & (x + y) \cdot z & \to & x \cdot z + y \cdot z \\
\text{R5} & (x \cdot y) \cdot z & \to & x \cdot (y \cdot z)
\end{array}
$$

Algorithmus 4.1 (Entscheiden von $s \underline{\leftrightarrow} t$ bzw. $s = t$)

Input - Zwei Prozessterme s und t.
Output - TRUE falls $s = t$; FALSE falls die Eigenschaft $s = t$ nicht erfüllt ist.

1. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf s an.
2. Nenne das Ergebnis n (n ist ein Prozessterm in Normalform).
3. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf t an.
4. Nenne das Ergebnis n' (n' ist ein Prozessterm in Normalform).
5. Falls $n =_{AC} n'$, gebe TRUE aus, sonst FALSE.
   (Dieser Schritt kann mit den Regeln R1 und R2 durchgeführt werden (wobei auf Termination oder durch andere Verfahren der Textverarbeitung.)

R1	$x + y$	$=_{\mathrm{AC}}$	$y + x$
R2	$(x + y) + z$	$=_{\mathrm{AC}}$	$x + (y + z)$
R3	$x + x$	$\rightarrow$	x
R4	$(x + y) \cdot z$	$\rightarrow$	$x \cdot z + y \cdot z$
R5	$(x \cdot y) \cdot z$	$\rightarrow$	$x \cdot (y \cdot z)$

Algorithmus 4.1 (Entscheiden von $s \underline{\leftrightarrow} t$ bzw. $s = t$)

Input - Zwei Prozessterme s und t.
Output - TRUE falls $s = t$; FALSE falls die Eigenschaft $s = t$ nicht erfüllt ist.

1. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf s an.
2. Nenne das Ergebnis n (n ist ein Prozessterm in Normalform).
3. Wende die Regeln R3, R4 und R5 von Seite 139 solange wie möglich auf t an.
4. Nenne das Ergebnis n' (n' ist ein Prozessterm in Normalform).
5. Falls $n =_{AC} n'$, gebe TRUE aus, sonst FALSE.
   (Dieser Schritt kann mit den Regeln R1 und R2 durchgeführt werden (wobei auf Termination oder durch andere Verfahren der Textverarbeitung.)

Beispiel

Zu entscheiden ist:

$$(a + a)(cd) + (bc)(d + d) = ((b + a)(c + c))d$$

Beispiel

Zu entscheiden ist:

$$(a+a)(cd)+(bc)(d+d) = ((b+a)(c+c))d$$

linke Seite $\quad (\underline{a+a})(cd)+(bc)(d+d)$

$\xrightarrow{A3} \quad a(cd)+(bc)(\underline{d+d})$

Beispiel

Zu entscheiden ist:

$$(a+a)(cd)+(bc)(d+d) = ((b+a)(c+c))d$$

linke Seite $\quad (\underline{a+a})(cd)+(bc)(d+d)$

$\xrightarrow{A3} \quad a(cd)+(bc)(\underline{d+d})$

Beispiel

Zu entscheiden ist:

$$(a+a)(cd)+(bc)(d+d) = ((b+a)(c+c))d$$

linke Seite

$$\begin{aligned} & (\underline{a+a})(cd)+(bc)(d+d) \\ \xrightarrow{A3}\ & a(cd)+(bc)(\underline{d+d}) \\ \xrightarrow{A3}\ & a(cd)+\underline{(bc)d} \\ \xrightarrow{A5}\ & a(cd)+b(cd) \end{aligned}$$

rechte
Seite:

rechte Seite:

$$
\begin{aligned}
& ((b+a)(\underline{c+c}))d \\
\xrightarrow{\text{A3}}\ & \underline{((b+a)c)d} \\
\xrightarrow{\text{A5}}\ & \underline{(b+a)(cd)} \\
\xrightarrow{\text{A4}}\ & b(cd)+a(cd)
\end{aligned}
$$

rechte Seite:

$$((b+a)(\underline{c+c}))d$$
$$\xrightarrow{A3} \underline{((b+a)c)d}$$
$$\xrightarrow{A5} \underline{(b+a)(cd)}$$
$$\xrightarrow{A4} b(cd)+a(cd)$$

$$(\underline{a+a})(cd)+(bc)(d+d)$$
$$\xrightarrow{A3} a(cd)+(bc)(\underline{d+d})$$
$$\xrightarrow{A3} a(cd)+\underline{(bc)d}$$
$$\xrightarrow{A5} a(cd)+b(cd)$$

linke Seite

rechte Seite:

$$((b+a)(\underline{c+c}))d$$
$$\xrightarrow{A3} \underline{((b+a)c)d}$$
$$\xrightarrow{A5} \underline{(b+a)(cd)}$$
$$\xrightarrow{A4} b(cd)+a(cd)$$

$$(\underline{a+a})(cd)+(bc)(d+d)$$
$$\xrightarrow{A3} a(cd)+(bc)(\underline{d+d})$$
$$\xrightarrow{A3} a(cd)+\underline{(bc)d}$$
$$\xrightarrow{A5} a(cd)+b(cd)$$

linke Seite

Die beiden berechneten Normalformen sind äquivalent (modulo $=_{\mathrm{AC}}$) und daher auch die Ausgangsterme.