Parallele und kommunizierende Prozesse

Durch den *Paralleloperator* *(merge)* $\|$ wird die parallele (besser: nebenläufige) Ausführung der beiden Prozesse dargestellt, die er als Argument hat. In den folgenden Regeln seien $v, w \in A$ und x, x', y, y' Prozessterme.

Folgen-Semantik

interleaving semantics

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

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

Folgen-Semantik

interleaving semantics

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

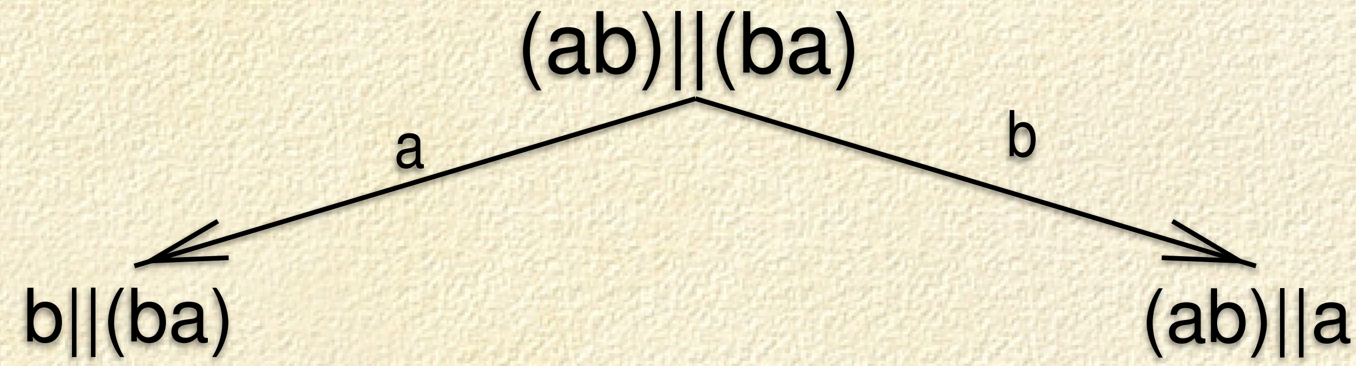
$$\frac{y \xrightarrow{v} \surd}{x \| y \xrightarrow{v} x} \qquad \frac{y \xrightarrow{v} y'}{x \| y \xrightarrow{v} x \| y'}$$

Beispiel

Der Prozessgraph von $(ab)\|(ba)$:

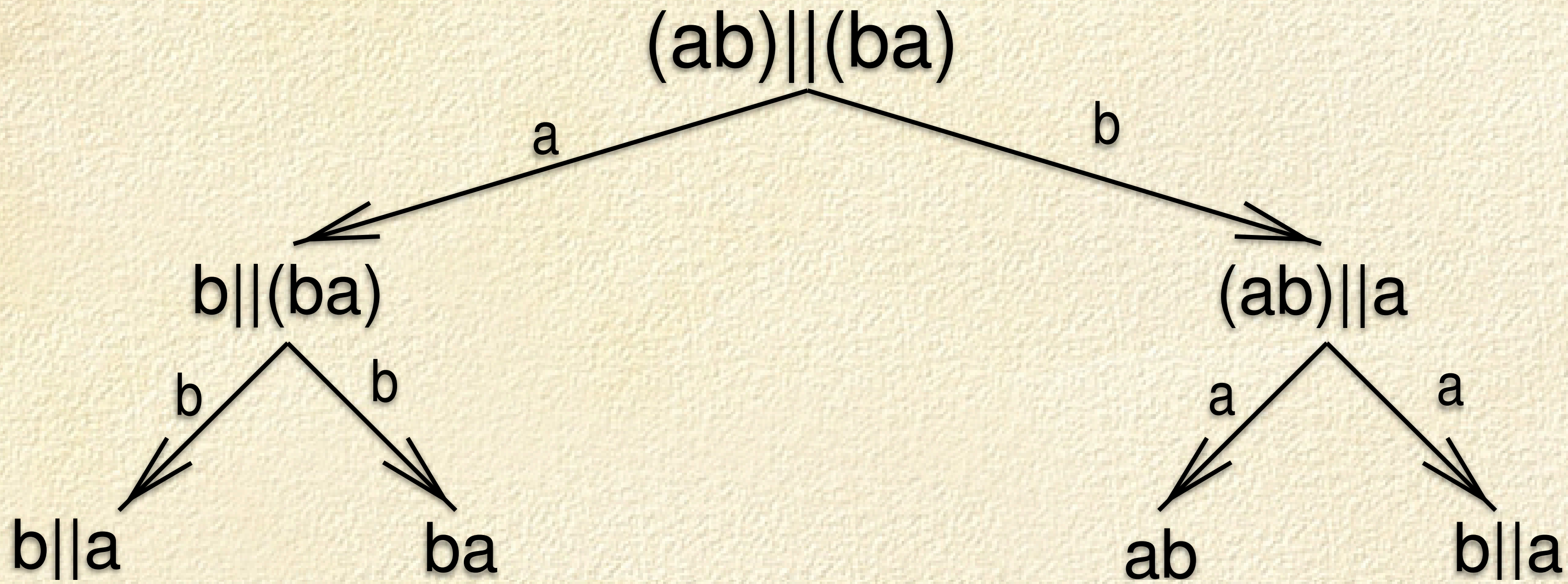
Beispiel

Der Prozessgraph von $(ab)\|(ba)$:



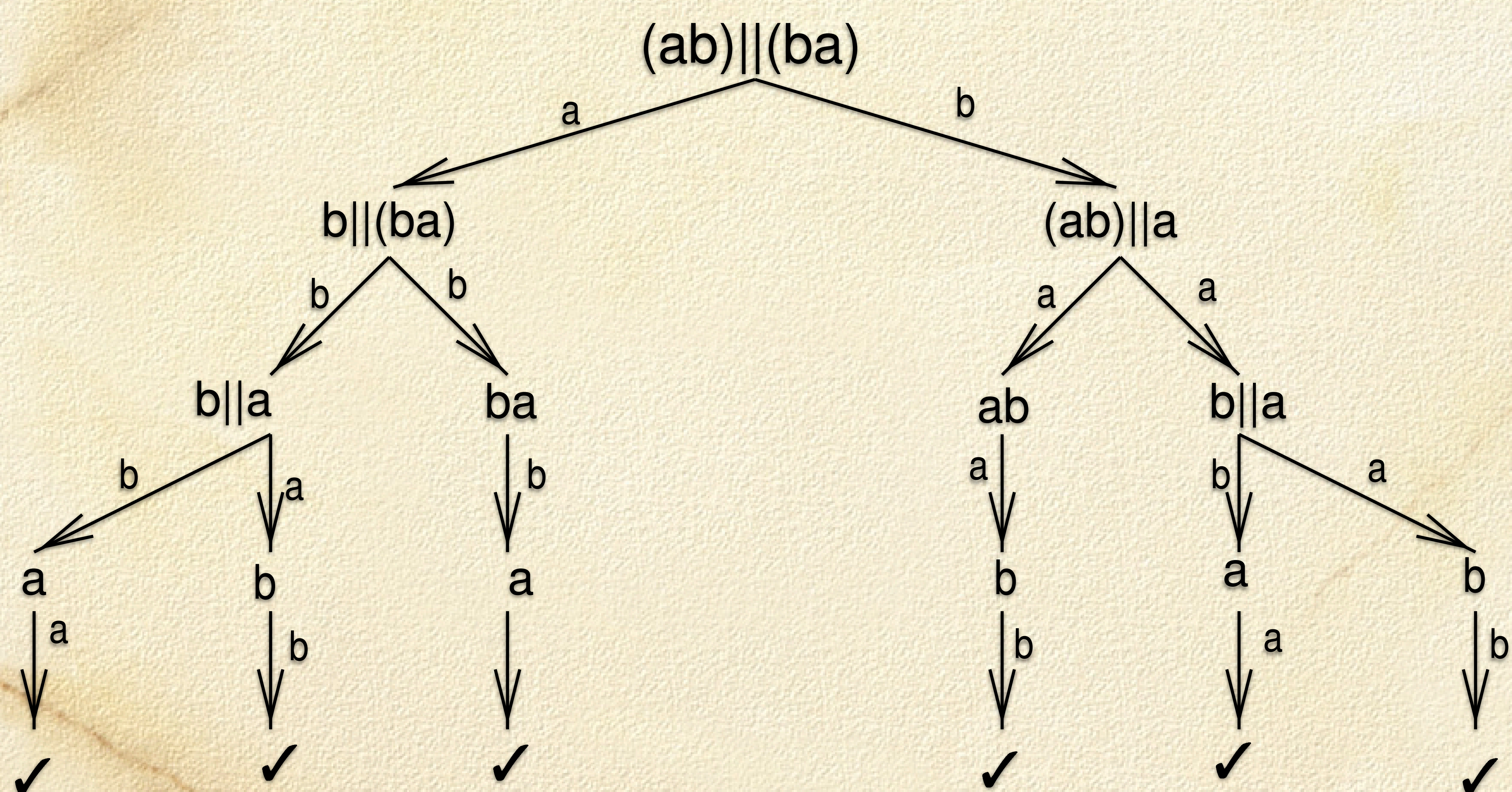
Beispiel

Der Prozessgraph von $(ab)\|(ba)$:



Beispiel

Der Prozessgraph von $(ab)\|(ba)$:



Zwei parallel ablaufende Prozesse kommunizieren mittels einer Kommunikationsfunktion.

Eine *Kommunikationsfunktion* $\gamma : A \times A \to A$ erzeugt für jedes Paar atomarer Aktionen a und b ihre Kommunikations-Aktion $\gamma(a, b)$.

Die Kommunikationsfunktion γ ist kommutativ und assoziativ:

$$\begin{aligned} \gamma(a, b) &\equiv \gamma(b, a) \\ \gamma(\gamma(a, b), c) &\equiv \gamma(a, \gamma(b, c)) \end{aligned}$$

Der Paralleloperator kann eine solche Kommunikation enthalten. Sie ist eine unteilbare Aktion beider Prozesse.

Der Paralleloperator kann eine solche Kommunikation enthalten. Sie ist eine unteilbare Aktion beider Prozesse.

Regeln für den Prozessgraphen:

$$\frac{x \xrightarrow{v} x' \qquad y \xrightarrow{w} y'}{x \| y \xrightarrow{\gamma(v,w)} x' \| y'}$$

Der Paralleloperator kann eine solche Kommunikation enthalten. Sie ist eine unteilbare Aktion beider Prozesse.

Regeln für den Prozessgraphen:

$$\frac{x \xrightarrow{v} \surd \quad y \xrightarrow{w} y'}{x \| y \xrightarrow{\gamma(v,w)} y'}$$

$$\frac{x \xrightarrow{v} x' \quad y \xrightarrow{w} y'}{x \| y \xrightarrow{\gamma(v,w)} x' \| y'}$$

Der Paralleloperator kann eine solche Kommunikation enthalten. Sie ist eine unteilbare Aktion beider Prozesse.

Regeln für den Prozessgraphen:

$$\frac{x \xrightarrow{v} \surd \qquad y \xrightarrow{w} \surd}{x \| y \xrightarrow{\gamma(v,w)} \surd}$$

$$\frac{x \xrightarrow{v} \surd \qquad y \xrightarrow{w} y'}{x \| y \xrightarrow{\gamma(v,w)} y'}$$

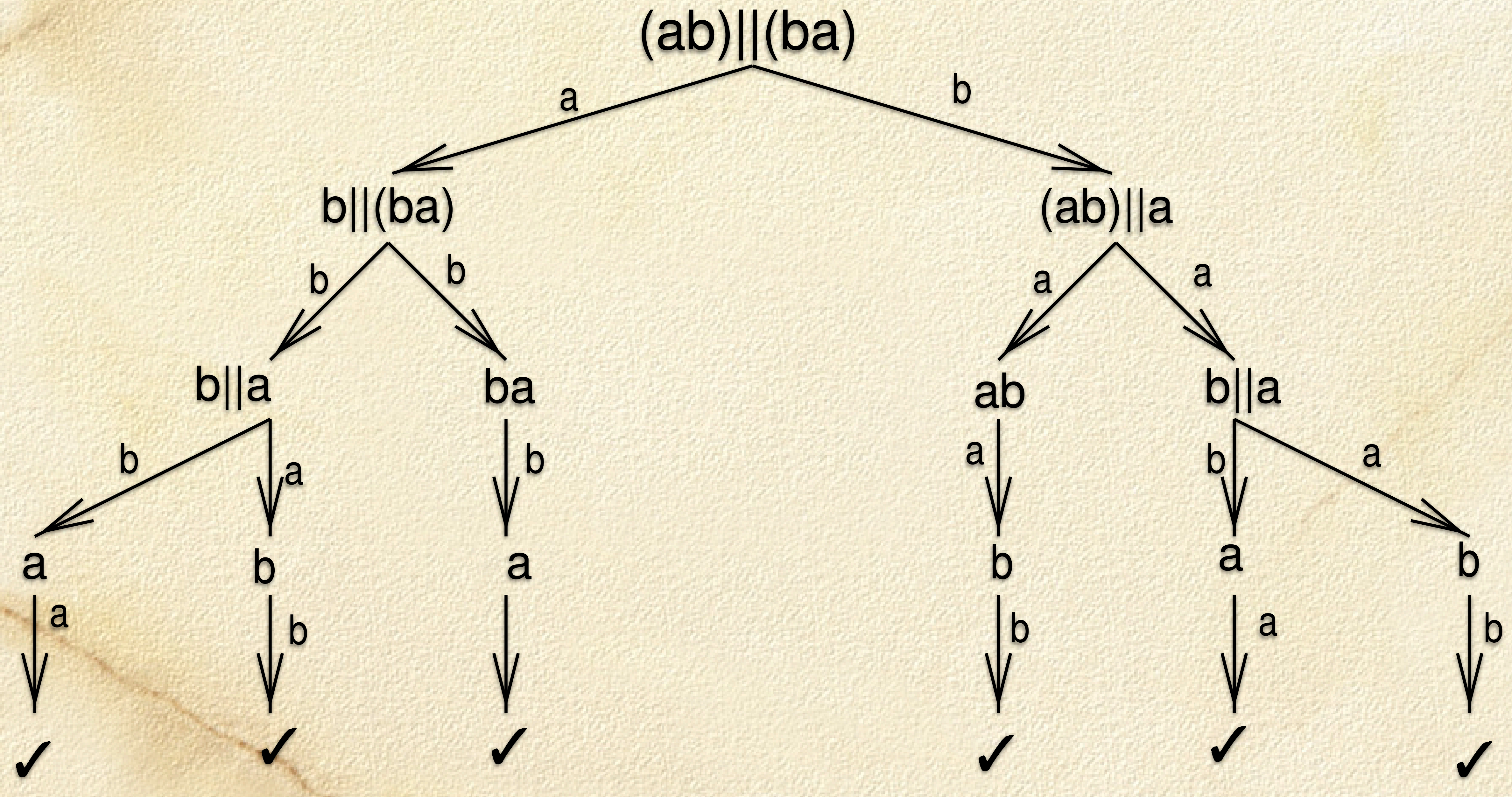
$$\frac{x \xrightarrow{v} x' \qquad y \xrightarrow{w} \surd}{x \| y \xrightarrow{\gamma(v,w)} x'}$$

$$\frac{x \xrightarrow{v} x' \qquad y \xrightarrow{w} y'}{x \| y \xrightarrow{\gamma(v,w)} x' \| y'}$$

Beispiel

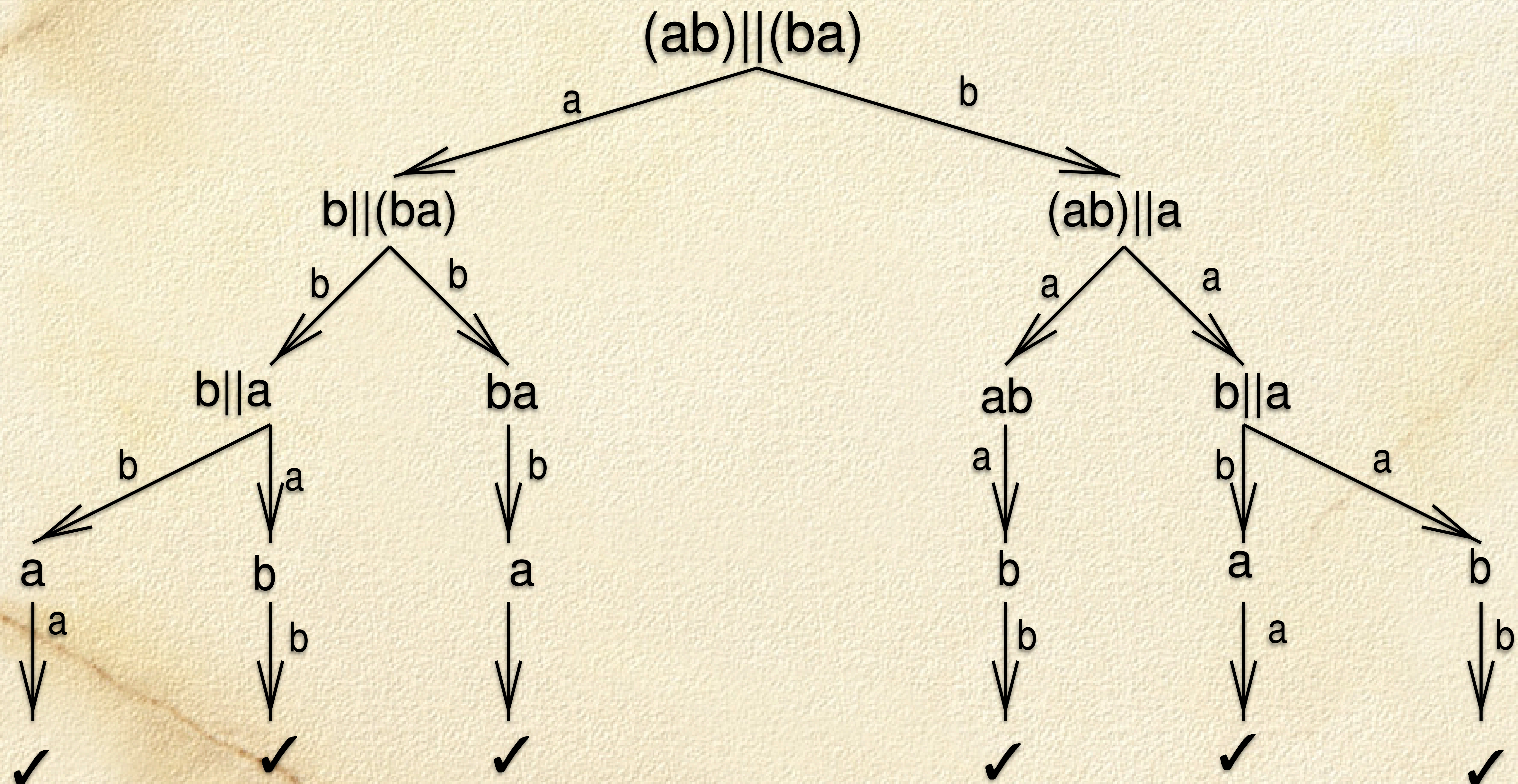
Der Prozessgraph von $(ab)\|(ba)$

für alle $x, y \in \{a, b\}$:



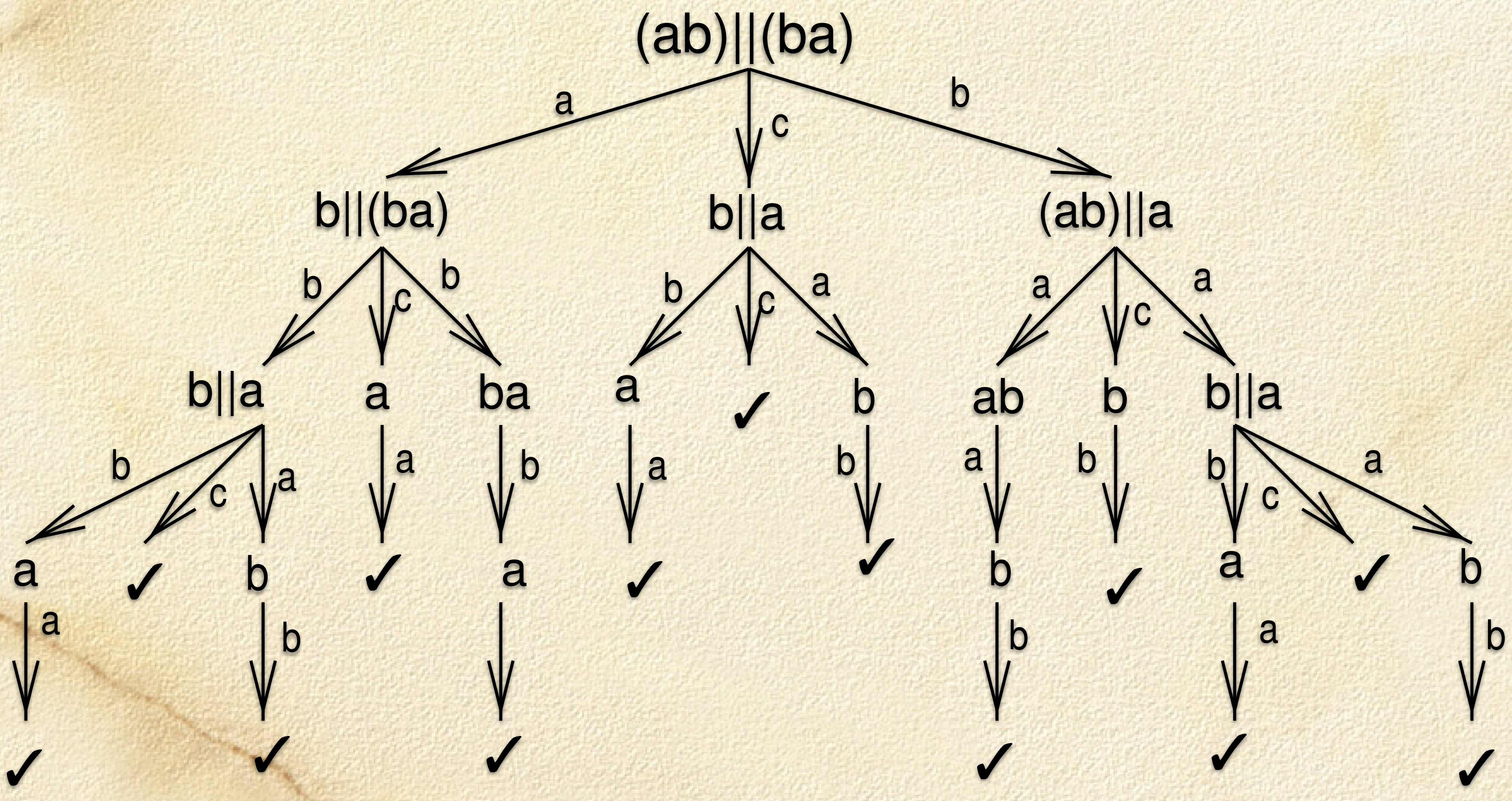
Beispiel

Der Prozessgraph von $(ab) \| (ba)$ mit $\gamma(x, y) = c$ für alle $x, y \in \{a, b\}$:



Beispiel

Der Prozessgraph von $(ab)\|(ba)$ mit $\gamma(x, y) = c$ für alle $x, y \in \{a, b\}$:



Links-Merge und Kommunikations-Merge

Um eine Axiomatisierung mit dem Paralleloperator zu erhalten, werden zwei Hilfsoperatoren benötigt:

Der *Links-Merge-Operator* *(left merge)* $\mathbin{⌊\!⌊}$ erlaubt die Ausführung der ersten Transition des ersten (linken) Argumentes:

$$\frac{x \xrightarrow{v} \surd}{x \mathbin{⌊\!⌊} y \xrightarrow{v} y} \qquad \frac{x \xrightarrow{v} x'}{x \mathbin{⌊\!⌊} y \xrightarrow{v} x' \| y}$$

Der *Kommunikations-Merge-Operator* *(communication merge)* $|$ stellt die Kommunikation der beiden ersten Transitionen der beiden Argumente dar:

Der *Kommunikations-Merge-Operator* *(communication merge)* $|$ stellt die Kommunikation der beiden ersten Transitionen der beiden Argumente dar:

$$\frac{x \xrightarrow{v} x' \quad y \xrightarrow{w} y'}{x|y \xrightarrow{\gamma(v,w)} x'\|y'}$$

Der *Kommunikations-Merge-Operator* *(communication merge)* $|$ stellt die Kommunikation der beiden ersten Transitionen der beiden Argumente dar:

$$\frac{x \xrightarrow{v} \surd \quad y \xrightarrow{w} y'}{x|y \xrightarrow{\gamma(v,w)} y'}$$

$$\frac{x \xrightarrow{v} x' \quad y \xrightarrow{w} y'}{x|y \xrightarrow{\gamma(v,w)} x' \| y'}$$

Der *Kommunikations-Merge-Operator* *(communication merge)* $|$ stellt die Kommunikation der beiden ersten Transitionen der beiden Argumente dar:

$$\frac{x \xrightarrow{v} \surd \quad y \xrightarrow{w} \surd}{x|y \xrightarrow{\gamma(v,w)} \surd} \qquad \frac{x \xrightarrow{v} \surd \quad y \xrightarrow{w} y'}{x|y \xrightarrow{\gamma(v,w)} y'}$$

$$\frac{x \xrightarrow{v} x' \quad y \xrightarrow{w} \surd}{x|y \xrightarrow{\gamma(v,w)} x'} \qquad \frac{x \xrightarrow{v} x' \quad y \xrightarrow{w} y'}{x|y \xrightarrow{\gamma(v,w)} x' \| y'}$$

Die Erweiterung der elementaren Prozessalgebra um die Operatoren *Paralleloperator (merge)* $\|$, *Links-Merge-Operator (left merge)* $\mathbin{\|\!\_}$ und *Kommunikations-Merge-Operator (communication merge)* $|$ heißt *PAP (Prozessalgebra mit Parallelismus)*.

Die Erweiterung der elementaren Prozessalgebra um die Operatoren *Paralleloperator (merge)* $\|$, *Links-Merge-Operator (left merge)* $\mathbin{\|\!\!\!\_}$ und *Kommunikations-Merge-Operator (communication merge)* $|$ heißt PAP (*Prozessalgebra mit Parallelismus*). In ihr sollen die neuen Paralleloperatoren stärker binden als $+$, d.h.: $a \mathbin{\|\!\!\!\_}\, b + a\|b$ steht für $(a \mathbin{\|\!\!\!\_}\, b) + (a\|b)$. Der Paralleloperator $\|$ kann durch $\mathbin{\|\!\!\!\_}$ und $|$ ausgedrückt werden: $s\|t \underline{\leftrightarrow} (s \mathbin{\|\!\!\!\_}\, t + t \mathbin{\|\!\!\!\_}\, s) + s|t$.

Anmerkung: PAP ist eine **konservative Erweiterung** von BPA.

Anmerkung: PAP ist eine **konservative Erweiterung** von BPA, d.h. die neuen Transitionsregeln verändern nicht die alten. Anders ausgedrückt bedeutet dies, dass der auf BPA eingeschränkte Prozessgraph unverändert bleibt.

Satz 5.14 *Die Äquivalenzrelation „Bisimulation" ist eine Kongruenzrelation in PAP, d.h.: wenn $s \underline{\leftrightarrow} s'$ und $t \underline{\leftrightarrow} t'$, dann*

- $s + t \underline{\leftrightarrow} s' + t'$,
- $s \cdot t \underline{\leftrightarrow} s' \cdot t'$,

Das galt schon in BPA.

Satz 5.14 *Die Äquivalenzrelation „Bisimulation“ ist eine Kongruenzrelation in PAP, d.h.: wenn $s \underline{\leftrightarrow} s'$ und $t \underline{\leftrightarrow} t'$, dann*

- $s + t \underline{\leftrightarrow} s' + t'$,
- $s \cdot t \underline{\leftrightarrow} s' \cdot t'$,
- $s \| t \underline{\leftrightarrow} s' \| t'$,
- $s \mathbin{⫡} t \underline{\leftrightarrow} s' \mathbin{⫡} t'$ *und*
- $s | t \underline{\leftrightarrow} s' | t'$.

Das galt schon in BPA.

Axiome des PAP-Kalküls:

Axiome A1, ..., A5 und

M1 $$x \| y = (x \mathbin{⫫} y + y \mathbin{⫫} x) + x | y$$

Axiome des PAP-Kalküls:

Axiome A1, ..., A5 und

M1 $$x \| y = (x \mathbin{⫫} y + y \mathbin{⫫} x) + x | y$$

LM4 $$(x + y) \mathbin{⫫} z = x \mathbin{⫫} z + y \mathbin{⫫} z$$

Axiome des PAP-Kalküls:

Axiome A1, . . . , A5 und

$$\text{M1} \qquad x \| y = (x \mathbin{⫡} y + y \mathbin{⫡} x) + x | y$$

$$\begin{aligned}
&\text{LM2} & v \mathbin{⫡} y &= v \cdot y \\
&\text{LM3} & (v \cdot x) \mathbin{⫡} y &= v \cdot (x \| y) \\
&\text{LM4} & (x + y) \mathbin{⫡} z &= x \mathbin{⫡} z + y \mathbin{⫡} z
\end{aligned}$$

$$\text{CM8} \qquad (v \cdot x)|(w \cdot y) = \gamma(v, w) \cdot (x \| y)$$

$$
\begin{array}{lrcl}
\text{CM5} & v|w & = & \gamma(v,w) \\
\text{CM6} & v|(w \cdot y) & = & \gamma(v,w) \cdot y \\
\text{CM7} & (v \cdot x)|w & = & \gamma(v,w) \cdot x \\
\text{CM8} & (v \cdot x)|(w \cdot y) & = & \gamma(v,w) \cdot (x \| y)
\end{array}
$$

$$
\begin{array}{lrcl}
\text{CM5} & v|w & = & \gamma(v,w) \\
\text{CM6} & v|(w \cdot y) & = & \gamma(v,w) \cdot y \\
\text{CM7} & (v \cdot x)|w & = & \gamma(v,w) \cdot x \\
\text{CM8} & (v \cdot x)|(w \cdot y) & = & \gamma(v,w) \cdot (x \| y) \\
\text{CM9} & (x+y)|z & = & x|z + y|z \\
\text{CM10} & x|(y+z) & = & x|y + x|z
\end{array}
$$

Satz 5.15 *Der PAP-Kalkül ist korrekt , d.h.:*
$s = t \;\Rightarrow\; s \underline{\leftrightarrow} t.$

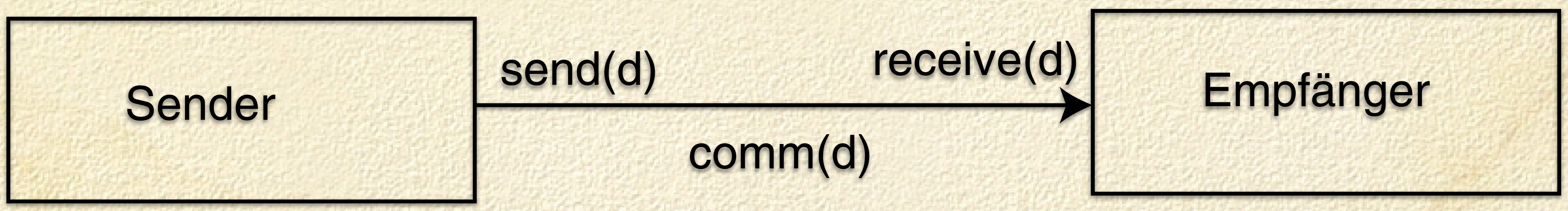
Beweis:
Beweisskizze: Da die Bisimulationsäquivalenz eine Kongruenz ist, genügt es für jedes Axiom $s = t$ die Relation $\sigma(s) \underline{\leftrightarrow} \sigma(t)$ für alle Substitutionen von den Variablen aus s und t in Processterme zu beweisen. $\square$

Satz 5.16 *Der PAP-Kalkül ist* *vollständig* ,
d.h.: $s \underline{\leftrightarrow} t \;\Rightarrow\; s = t$.

Beweis:

Beweisskizze: Dies kann man beweisen, indem die Axiome für PAP in ein (Term-) Ersetzungskalkül modulo + verwandelt. Jeder Prozessterm über PAP ist in Normalform reduzierbar. Wenn $s \underline{\leftrightarrow} t$ ist, wobei s und t Normalformen s' und t' haben, dann gilt $s' =_{\mathrm{AC}} t'$, also auch $s = s' =_{\mathrm{AC}} t' = t$. □

Abbruch (deadlock) und **Verdeckung** (encapsulation) dienen dazu, Teile einer Kommunikation (wie $send(d)$ und das zugehörige $receive(d)$) zu einer Operation (z.B. $comm(d)$) zu verschmelzen. Darüberhinaus können diese Operationen als Einzelaktionen unterbunden werden.



Der Operator *Abbruch* δ zeigt kein sichtbares Verhalten.
Es gibt daher auch dazu keine Transitionsregel.

Der Operator *Abbruch* δ zeigt kein sichtbares Verhalten. Es gibt daher auch dazu keine Transitionsregel.

"Abbrechen"

Der Operator Verdecken ∂_H, mit $H \subseteq A$, benennt alle Aktionen aus H, die bei ihm als Argument auftreten, in δ um:

$$\frac{x \xrightarrow{v} x' \quad (v \notin H)}{\partial_H(x) \xrightarrow{v} \partial_H(x')}$$

Der Operator *Abbruch* δ zeigt kein sichtbares Verhalten. Es gibt daher auch dazu keine Transitionsregel.

"Abbrechen"

Der Operator Verdecken ∂_H, mit $H \subseteq A$, benennt alle Aktionen aus H, die bei ihm als Argument auftreten, in δ um:

$$\frac{x \xrightarrow{v} \surd \quad (v \notin H)}{\partial_H(x) \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x' \quad (v \notin H)}{\partial_H(x) \xrightarrow{v} \partial_H(x')}$$

Wednesday, January 5, 2011

Der Operator *Abbruch* δ zeigt kein sichtbares Verhalten. Es gibt daher auch dazu keine Transitionsregel.

"Abbrechen"

Der Operator Verdecken ∂_H, mit $H \subseteq A$, benennt alle Aktionen aus H, die bei ihm als Argument auftreten, in δ um:

$$\frac{x \xrightarrow{v} \surd \quad (v \notin H)}{\partial_H(x) \xrightarrow{v} \surd} \qquad \frac{x \xrightarrow{v} x' \quad (v \notin H)}{\partial_H(x) \xrightarrow{v} \partial_H(x')}$$

Der Bildbereich der Kommunikationsfunktion γ wird um δ erweitert:

$$\gamma : A \times A \to A \cup \{\delta\}.$$

Der Verdeckungsoperator erzwingt Kommunikation. Beispielsweise kann $\partial_{\{a,b\}}(a \| b)$ nur als $\gamma(a,b)$ ausgeführt werden (falls $\gamma(a,b) \neq \delta$).

Der Verdeckungsoperator erzwingt Kommunikation. Beispielsweise kann $\partial_{\{a,b\}}(a \| b)$ nur als $\gamma(a, b)$ ausgeführt werden (falls $\gamma(a, b) \neq \delta$).

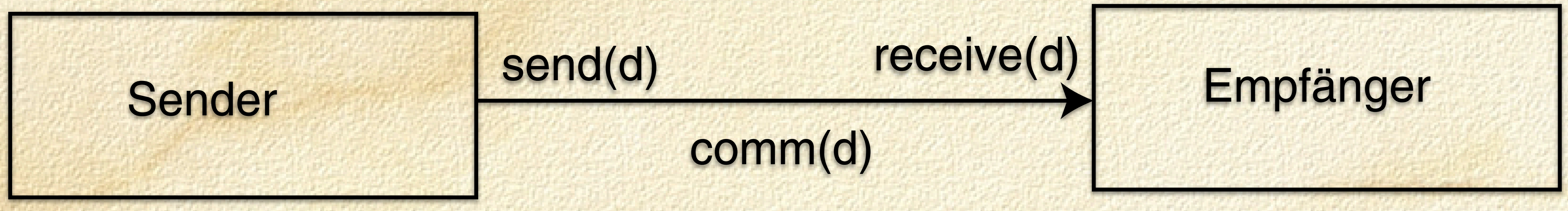
Definition 5.17 *Die Erweiterung des Kalküls PAP durch die nachstehenden Axiome für Abbruch und Verdeckung wird mit ACP bezeichnet (algebra of communicating processes).*

Axiome des Kalküls ACP : die Axiome von PAP und folgende für Abbruch und Verdeckung:

$$
\begin{array}{lrcl}
\text{A6} & x + \delta & = & x \\
\text{A7} & \delta \cdot x & = & \delta
\end{array}
$$

$$
\begin{array}{llcll}
\text{D1} & \partial_H(v) & = & v & (v \notin H) \\
\text{D2} & \partial_H(v) & = & \delta & (v \in H) \\
\text{D3} & \partial_H(\delta) & = & \delta & \\
\text{D4} & \partial_H(x+y) & = & \partial_H(x) + \partial_H(y) & \\
\text{D5} & \partial_H(x \cdot y) & = & \partial_H(x) \cdot \partial_H(y) &
\end{array}
$$

$$
\begin{array}{lrcl}
\text{LM11} & \delta \mathbin{⌊\!⌊} x & = & \delta \\
\text{CM12} & \delta | x & = & \delta \\
\text{CM13} & x | \delta & = & \delta
\end{array}
$$



Beispiel Der Prozessterm zum einführenden Beispiel zur Abbildung lautet:

$\partial_{\{send(0),send(1),read(0),read(1)\}}((send(0)+send(1))\|(read(0)+read(1)))$

mit $\gamma(send(d),read(d))=comm(d)$ für $d\in\{0,1\}$.

$$\partial_{\{send(0), send(1), read(0), read(1)\}}((send(0) + send(1)) \| (read(0) + read(1)))$$

mit $\gamma(send(d), read(d)) = comm(d)$ für $d \in \{0, 1\}$.

$$\partial_{\{send(0),send(1),read(0),read(1)\}}((\overset{a}{send}(0) + \overset{a'}{send}(1))\|(\overset{b}{read}(0) + \overset{b'}{read}(1)))$$

mit $\gamma(send(d), read(d)) = \underset{\substack{c \\ c'}}{comm}(d)$ für $d \in \{0, 1\}$.

$$\partial_{\{send(0),send(1),read(0),read(1)\}}((\overset{a}{send(0)} + \overset{a'}{send(1)})\|(\overset{b}{read(0)} + \overset{b'}{read(1)}))$$

mit $\gamma(send(d), read(d)) = \underset{c,\ c'}{comm(d)}$ für $d \in \{0, 1\}$.

Beispiel 4.20

Seinen $\gamma(a, b) \equiv c$ und $\gamma(a', b') \equiv c'$ zunächst die einzigen Kommunikationsaktionen zwischen Aktionen.

$(a + a')\|(b + b')$

$\overset{M1}{=}$

Beispiel 4.20

Seinen $\gamma(a,b) \equiv c$ und $\gamma(a',b') \equiv c'$ zunächst die einzigen Kommunikationsaktionen zwischen Aktionen.

$$(a + a') \| (b + b')$$

$$\stackrel{\text{M1}}{=}$$

$$(a + a') \llfloor (b + b') + (b + b') \llfloor (a + a')$$

$$+ (a + a') | (b + b')$$

$$\stackrel{\text{LM4, CM9,10}}{=}$$

$$(a + a') \mathbin{|\!\!\lfloor} (b + b') + (b + b') \mathbin{|\!\!\lfloor} (a + a')$$

$$+ (a + a') | (b + b')$$

$$\overset{\text{LM4, CM9,10}}{=}$$

$$(a+a') \mathbin{\|\!\!\lfloor} (b+b') + (b+b') \mathbin{\|\!\!\lfloor} (a+a')$$
$$+ (a+a') | (b+b')$$
$$\overset{\text{LM4, CM9,10}}{=}$$

$$a \mathbin{\|\!\!\lfloor} (b+b') + a' \mathbin{\|\!\!\lfloor} (b+b') + b \mathbin{\|\!\!\lfloor} (a+a') + b' \mathbin{\|\!\!\lfloor} (a+a')$$
$$+ a|b + a|b' + a'|b + a'|b'$$
$$\overset{\text{LM2, CM5}}{=}$$

$$a \cdot (b+b') + a' \cdot (b+b') + b \cdot (a+a') + b' \cdot (a+a')$$
$$+ c + \delta + \delta + c'$$
$$\overset{\text{A6}}{=}$$

$$a \cdot (b+b') + a' \cdot (b+b') + b \cdot (a+a') + b' \cdot (a+a')$$
$$+ c + c'$$

Beispiel 4.21 (Fortsetzung)

Sei nun $H = \{a, a', b, b'\}$.

$$\partial_H((a + a')\|(b + b'))$$

$$=$$

$$\partial_H(a \cdot (b + b') + a' \cdot (b + b') + b \cdot (a + a') + b' \cdot (a + a') + c + c')$$

$\overset{\text{D1,2,4,5}}{=}$

$\partial_H(a \cdot (b + b') + a' \cdot (b + b') + b \cdot (a + a')$

$+ b' \cdot (a + a') + c + c')$

H = {a,a',b,b'}

D1,$\underline{\underline{2}}$,4,5

$$\partial_H((a + a') \| (b + b'))$$

$\gamma(a, b) \equiv c$ und $\gamma(a', b') \equiv c'$

$$H = \{a, a', b, b'\}$$

$$\partial_H(a \cdot (b+b') + a' \cdot (b+b') + b \cdot (a+a')$$
$$+ b' \cdot (a+a') + c + c')$$

H = {a,a',b,b'}

$\overset{\text{D1,2,4,5}}{=}$

$$\delta \cdot \partial_H(b+b') + \delta \cdot \partial_H(b+b') + \delta \cdot \partial_H(a+a')$$
$$+ \delta \cdot \partial_H(a+a') + c + c'$$

$\overset{\text{A6,7}}{=}$

$$c + c'$$

∂_H erzwingt also die Kommunication zwischen a und b einerseits und zwischen a' und b' andererseits.

$$\partial_H((a+a') \| (b+b'))$$

$\gamma(a,b) \equiv c$ und $\gamma(a',b') \equiv c'$

$$H = \{a, a', b, b'\}$$

Theorem 5.19 *a) Bisimulation ist eine Kongruence für ACP: wenn* $s \underline{\leftrightarrow} s'$ *und* $t \underline{\leftrightarrow} t'$, *dann* $s + t \underline{\leftrightarrow} s' + t'$, $s \cdot t \underline{\leftrightarrow} s' \cdot t'$, $s \| t \underline{\leftrightarrow} s' \| t'$, $s \mathbin{⫡} t \underline{\leftrightarrow} s' \mathbin{⫡} t'$, $s | t \underline{\leftrightarrow} s' | t'$ *und* $\partial_H(s) \underline{\leftrightarrow} \partial_H(t)$.

b) Der Kalkül ACP ist korrekt:
$s = t \;\Rightarrow\; s \underline{\leftrightarrow} t.$

c) Der Kalkül ACP ist vollständig:
$s \underline{\leftrightarrow} t \;\Rightarrow\; s = t.$