

## 9.4 Rekursion

Bislang wurden nur Prozesse endlicher Länge spezifiziert. Unendliche Prozesse, wie z.B. der Prozess  $ababab\dots$  mit dem Transitionssystem von Abb. 9.4 werden durch Rekursion definiert.

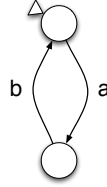


Abbildung 9.4: Transitionssystem für  $abab\dots$

Konkret geschieht dies z.B. durch das folgende *rekursive Gleichungssystem*:

$$\begin{aligned} X &\stackrel{!}{=} aY \\ Y &\stackrel{!}{=} bX \end{aligned}$$

Dabei sind  $X$  und  $Y$  *Rekursionsvariablen*, die zwei Zustände des Prozesses repräsentieren.

**Definition 9.29** Eine rekursive Spezifikation für die Variablen  $X_1, \dots, X_n$  ist ein System von Gleichungen der Form:

$$\begin{aligned} X_1 &\stackrel{!}{=} t_1(X_1, \dots, X_n) \\ &\vdots \\ X_n &\stackrel{!}{=} t_n(X_1, \dots, X_n) \end{aligned}$$

Hierbei sind die  $t_i(X_1, \dots, X_n)$  ACP-Terme ACP, welche die Variablen  $X_1, \dots, X_n$  enthalten dürfen.

Notationshinweis: Wir verwenden das Symbol  $\stackrel{!}{=}$  in den Gleichungen  $X_n \stackrel{!}{=} t_n(X_1, \dots, X_n)$ , um zu zeigen, dass es sich hierbei um etwas anderes handelt als um die Aussagen  $s = t$  im Kalkül.

Wenn durch den Kontext eine Verwechslung ausgeschlossen ist, notieren wir aber  $\stackrel{!}{=}$  auch als  $=$ .

**Definition 9.30** Prozesse  $p_1, \dots, p_n$  werden als eine Lösung einer rekursiven Spezifikation

$$\{X_i \stackrel{!}{=} t_i(X_1, \dots, X_n) \mid i \in \{1, \dots, n\}\}$$

(modulo Bisimulations-Äquivalenz) bezeichnet, falls  $p_i \Leftrightarrow t_i(p_1, \dots, p_n)$  für  $i \in \{1, \dots, n\}$  gilt.

Mit den Prozessen  $p_1, \dots, p_n$  sind hier ihre Prozessgraphen gemeint. Damit dann der Ausdruck  $t_i(p_1, \dots, p_n)$  Sinn ergibt, verstehen wir die Operationen  $+$  und  $\cdot$  auch als Operationen auf Prozessgraphen.

Wir wollen, dass Lösungen eindeutig bezüglich Bisimulations-Äquivalenz sind, d.h. falls  $p_1, \dots, p_n$  und  $q_1, \dots, q_n$  zwei solche Lösungen sind, dann erwarten wir  $p_i \Leftrightarrow q_i$  für alle  $i \in \{1, \dots, n\}$ .

Dies muss aber nicht für jede Spezifikation der Fall sein.

### Beispiel 9.31

- a)  $X \stackrel{!}{=} X$  hat alle Prozesse als Lösung.
- b) Alle Prozesse  $p$  mit  $p \xrightarrow{a} \surd$  sind Lösungen von der rekursiven Spezifikation  $\{X \stackrel{!}{=} a + X\}$ .  
Beispiel  $p \equiv a + cd$ : Die linke Seite ist nach Einsetzen ( $X \equiv p \equiv a + cd$ ) bisimilar zur rechten Seite nach Einsetzen ( $a + X \equiv a + a + X$ ).
- c) Alle nichtterminierenden Prozesse sind Lösungen von der rekursiven Spezifikation  $\{X \stackrel{!}{=} Xa\}$ .
- d)  $\{X \stackrel{!}{=} aY, Y \stackrel{!}{=} bX\}$  hat als einzige Lösung den Prozess  

$$X \xrightarrow{a} Y \xrightarrow{b} X \xrightarrow{a} Y \xrightarrow{b} \dots \text{ für } X \text{ und } Y \xrightarrow{b} X \xrightarrow{a} Y \xrightarrow{b} X \xrightarrow{a} \dots \text{ für } Y.$$
- e)  $\{X \stackrel{!}{=} Y, Y \stackrel{!}{=} aX\}$  hat als einzige Lösung  $X \xrightarrow{a} X \xrightarrow{a} X \xrightarrow{a} X \xrightarrow{a} \dots$  für  $X$  und  $Y$ .
- f)  $\{X \stackrel{!}{=} (a + b) \parallel X\}$  hat als einzige Lösung:

$$X \xrightarrow[a]{a} X \xrightarrow[a]{a} X \xrightarrow[a]{a} X \xrightarrow[a]{a} \dots$$

„Einzig“ bzw. „zwei verschiedene“ ist hier jeweils modulo Bisimulation zu verstehen.

## Geschützte rekursive Spezifikation

Wir definieren jetzt einen Spezialfall rekursiver Spezifikation, um Existenz und Eindeutigkeit zu garantieren.

**Definition 9.32** *Eine rekursive Spezifikation*

$$\{X_i \stackrel{!}{=} t_i(X_1, \dots, X_n) \mid i \in \{1, \dots, n\}\}$$

heißt geschützt (guarded), falls die rechten Seiten ihrer Gleichungen mit Hilfe der ACP-Kalküls in die folgende Form überführt werden können:

$$a_1 \cdot s_1(X_1, \dots, X_n) + \dots + a_k \cdot s_k(X_1, \dots, X_n) + b_1 + \dots + b_\ell$$

Außerdem dürfen bei dieser Umformung Variable einer Rekursionsgleichung durch die entsprechende rechte Seite ersetzt werden.

**Beispiel 9.33** Sei  $\gamma(a, b) \equiv c$  und  $\gamma(b, b) \equiv c$ .

$\{X \stackrel{!}{=} Y \parallel Z, Y \stackrel{!}{=} Z + a, Z \stackrel{!}{=} bZ\}$  ist geschützt, denn:

- $Z \stackrel{!}{=} bZ$  hat schon die gewünschte Form.
- $Y \stackrel{!}{=} Z + a$  wird transformiert, indem  $Z$  durch  $bZ$  ersetzt wird.
- $X \stackrel{!}{=} Y \parallel Z$  wird transformiert, indem  $Y$  durch  $bZ + a$  und  $Z$  durch  $bZ$  ersetzt wird und der so erhaltene Term  $(bZ + a) \parallel bZ$  mittels der Axiome transformiert wird:

$$\begin{aligned}
 & (bZ + a) \parallel bZ \\
 = & (bZ + a) \parallel bZ + bZ \parallel (bZ + a) + (bZ + a) \parallel bZ \\
 = & bZ \parallel bZ + a \parallel bZ + bZ \parallel (bZ + a) + bZ \parallel bZ + a \parallel bZ \\
 = & b(Z \parallel bZ) + abZ + b(Z \parallel (bZ + a)) + c(Z \parallel Z) + cZ
 \end{aligned}$$

**Satz 9.34** Eine rekursive Spezifikation besitzt genau dann eine eindeutige Lösung (modulo Bisimulation), wenn sie geschützt ist.

Zum Beispiel sind einige der rekursiven Spezifikationen von Beispiel 9.31 nicht geschützt. Wegen der Existenz und der Eindeutigkeit können wir von der Lösung sprechen.

**Definition 9.35** Für eine geschützte rekursive Spezifikation  $E$  über den Variablen  $X_1, \dots, X_n$  bezeichnet

$$\langle X_i | E \rangle$$

die Lösung der Variablen  $X_i, i = 1, \dots, n$ .

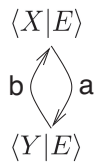
### Transitionsregeln für Rekursion

$$\begin{array}{c}
 \frac{t_i(\langle X_1 | E \rangle, \dots, \langle X_n | E \rangle) \xrightarrow{v} \surd}{\langle X_i | E \rangle \xrightarrow{v} \surd} \\
 \\
 \frac{t_i(\langle X_1 | E \rangle, \dots, \langle X_n | E \rangle) \xrightarrow{v} y}{\langle X_i | E \rangle \xrightarrow{v} y}
 \end{array}$$

d.h.:  $\langle X_i | E \rangle$  übernimmt das Verhalten von  $t_i(\langle X_1 | E \rangle, \dots, \langle X_n | E \rangle)$ :

### Beispiel 9.36

Sei  $E$  die geschützte rekursive Spezifikation  $\{X = aY, Y = bX\}$ . Der Prozessgraph von  $\langle X | E \rangle$  ist:



Wir leiten her:  $\langle X|E \rangle \xrightarrow{a} \langle Y|E \rangle$ :

$$\begin{array}{c}
 \frac{a \xrightarrow{a} \surd}{a \cdot \langle Y|E \rangle \xrightarrow{a} \langle Y|E \rangle} \quad \frac{\overline{v \xrightarrow{v} \surd}}{x \cdot y \xrightarrow{v} y} \quad v := a, \quad x := a, \quad y := \langle Y|E \rangle \\
 \hline
 \langle X|E \rangle \xrightarrow{a} \langle Y|E \rangle \quad \frac{a \cdot \langle Y|E \rangle \xrightarrow{v} y}{\langle X|E \rangle \xrightarrow{v} y} \quad v := a, \quad y := \langle Y|E \rangle
 \end{array}$$

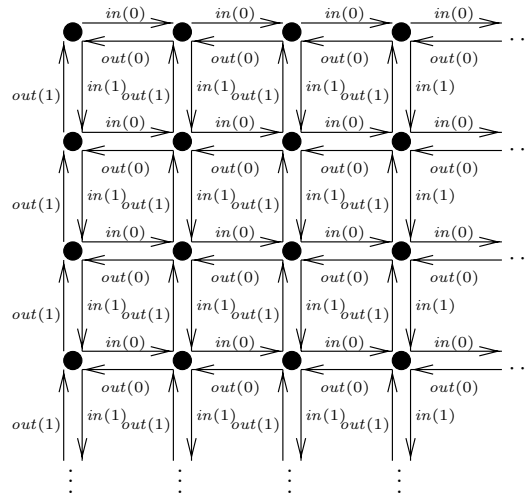
**Aufgabe 9.37** Leiten Sie für die folgenden drei Prozess-Terme den Prozessgraphen ab:

- a)  $\langle X|X = ab \rangle$
- b)  $\langle X|X = aXb \rangle$
- c)  $\langle X|X = aXb + c \rangle$

**Anmerkung:** ACP mit geschützter Rekursion ist eine konservative Erweiterung von ACP und Bisimulation ist eine Kongruenzrelation.

**Beispiel 9.38** Multimenge (bag) über  $\{0, 1\}$ .

Die Elemente 0 and 1 werden durch  $in(0)$  und  $in(1)$  in die Multimenge eingefügt und können durch  $out(0)$  und  $out(1)$  in beliebiger Reihenfolge entfernt werden.



Eine rekursive Spezifikation ist:

$$X = in(0)(X \parallel out(0)) + in(1)(X \parallel out(1))$$

**Aufgabe 9.39** Geben Sie eine Bisimulations-Relation für Beispiel 9.38 an. Dabei ist das gezeichnete Transitionssystem mit dem Zustand links oben als Anfangszustand zu verstehen. Die Bisimulation soll zwischen diesem Transitionssystem und dem Prozessgraphen einer Lösung  $\langle X|E \rangle$  der rekursiven Spezifikation konstruiert werden.

**Definition 9.40** Sei  $E$  eine rekursive Spezifikation  $E$  der Form:

$$\{X_i = t_i(X_1, \dots, X_n) \mid i \in \{1, \dots, n\}\}$$

Folgende Axiome gelten für ACP mit Rekursion:

- **Rekursives Definitionsprinzip:**

$$\langle X_i|E \rangle = t_i(\langle X_1|E \rangle, \dots, \langle X_n|E \rangle) \quad (i \in \{1, \dots, n\}) \quad (\text{RDP})$$

- **Rekursives Spezifikationsprinzip:**

$$\begin{aligned} &\text{Falls } y_i = t_i(y_1, \dots, y_n) \text{ für } i \in \{1, \dots, n\}, \\ &\text{dann } y_i = \langle X_i|E \rangle \quad (i \in \{1, \dots, n\}) \end{aligned} \quad (\text{RSP})$$

**Beispiel 9.41**

$$\begin{aligned} \langle Z \mid Z=aZ \rangle &\stackrel{\text{RDP}}{=} a\langle Z \mid Z=aZ \rangle \\ &\stackrel{\text{RDP}}{=} a(a\langle Z \mid Z=aZ \rangle) \\ &\stackrel{\text{A5}}{=} (aa)\langle Z \mid Z=aZ \rangle \end{aligned}$$

Also gilt mit RSP (denn wir haben eine Gleichung mit  $Z$  der Form  $x = aax$ ):

$$\langle Z \mid Z=aZ \rangle = \langle X \mid X=(aa)X \rangle$$

Weiter gilt:

$$\begin{aligned} \langle Z \mid Z=aZ \rangle &\stackrel{\text{RDP}}{=} a\langle Z \mid Z=aZ \rangle \\ &\stackrel{\text{RDP}}{=} a(a\langle Z \mid Z=aZ \rangle) \\ &\stackrel{\text{RDP}}{=} a(a(a\langle Z \mid Z=aZ \rangle)) \\ &\stackrel{\text{A5}}{=} ((aa)a)\langle Z \mid Z=aZ \rangle \end{aligned}$$

Also gilt mit RSP (denn wir haben eine Gleichung mit  $Z$  der Form  $y = aaay$ ):

$$\langle Z \mid Z=aZ \rangle = \langle Y \mid Y=((aa)a)Y \rangle$$

Also:

$$\begin{aligned} \langle X \mid X=(aa)X \rangle &= \langle Z \mid Z=aZ \rangle \\ &= \langle Y \mid Y=((aa)a)Y \rangle \end{aligned}$$

**Beispiel 9.42** Wir wollen für  $\gamma(a, b) \equiv c$  die folgende Gleichung beweisen:

$$\partial_{\{a,b\}}(\langle X \mid X=aX \rangle \parallel \langle Y \mid Y=bY \rangle) = \langle Z \mid Z=cZ \rangle$$

Ansatz: Es gilt:

$$\begin{aligned} & \langle X \mid X=aX \rangle \parallel \langle Y \mid Y=bY \rangle \\ = & \langle X \mid X=aX \rangle \sqcup \langle Y \mid Y=bY \rangle \\ + & \langle Y \mid Y=bY \rangle \sqcup \langle X \mid X=aX \rangle \\ + & \langle X \mid X=aX \rangle \mid \langle Y \mid Y=bY \rangle \\ = & a(\langle X \mid X=aX \rangle \parallel \langle Y \mid Y=bY \rangle) \\ + & b(\langle Y \mid Y=bY \rangle \parallel \langle X \mid X=aX \rangle) \\ + & c(\langle X \mid X=aX \rangle \parallel \langle Y \mid Y=bY \rangle) \end{aligned}$$

Aus

$$\begin{aligned} & \partial_{\{a,b\}}(\langle X \mid X=aX \rangle \parallel \langle Y \mid Y=bY \rangle) \\ = & c \cdot \partial_{\{a,b\}}(\langle X \mid X=aX \rangle \parallel \langle Y \mid Y=bY \rangle) \end{aligned}$$

folgt mit RSP das Ergebnis.

**Theorem 9.43** *Der Kalkül ACP mit geschützter Rekursion ist korrekt bezüglich Bisimulation:*

$$s = t \Rightarrow s \leftrightarrow t$$

RSP ist **nicht** korrekt für ungeschützte Rekursion. Beispielsweise folgt mit RSP aus  $t = t$ , dass  $t$  das Muster der Spezifikation  $X = X$  besitzt, d.h. dass  $t$  nach RSP eine Lösung ist:

$$t = \langle X \mid X=X \rangle$$

Da diese Gleichung für alle Prozess-Terme  $t$  gilt, erhalten wir also  $t_1 = \langle X \mid X=X \rangle$  und  $t_2 = \langle X \mid X=X \rangle$  und hieraus folgt direkt  $t_1 = t_2$  und das für beliebige Terme  $t_1, t_2$ . Mit anderen Worten: Aus RSP würde für ungeschützte Rekursion folgen, dass alle Terme gleich sind, es also nur einen einzigen Term gibt.

Bemerkung: Der Kalkül ACP mit geschützter Rekursion ist jedoch *nicht* vollständig. Wir werden daher die Form der Rekursion weiter einschränken.

**Definition 9.44** *Eine rekursive Spezifikation ist linear, wenn ihre rekursiven Gleichungen die folgende Form haben:*

$$X = a_1 X_1 + \cdots + a_k X_k + b_1 + \cdots + b_l \quad a_i, b_j \in A$$

Die linearen rekursiven Spezifikationen sind vollständig axiomatisierbar.

**Theorem 9.45 (Vollständigkeit)** *Das Kalkül ACP mit den Axiomen RDP und RSP ist eine vollständige Axiomatisierung für Bisimulation.*

**Umformungen in rekursiven Spezifikationen**

Sei  $E, E'$  geschützte rekursive Spezifikationen, wobei  $E'$  aus  $E$  entsteht, indem die rechten Seiten der Gleichungen folgendermaßen modifiziert werden:

- Die Axiome für ACP mit geschützter Rekursion dürfen benutzt werden.
- Rekursionsvariablen dürfen durch die rechte Seite ihrer Rekursionsgleichung ersetzt werden.

Dann kann  $\langle X|E \rangle = \langle X|E' \rangle$  im Kalkül ACP mit geschützter Rekursion für alle Rekursionsvariablen abgeleitet werden.

Zum Beispiel kann

$$\langle X \mid X=aX+aX \rangle = \langle X \mid X=(aa)X \rangle$$

abgeleitet werden, indem

- erst A3 angewandt wird:  $(x + x = x)$ ,
- dann  $X$  durch die rechte Seite  $aX$  ersetzt wird,
- und dann zuletzt A5 angewandt wird:  $((xy)z = x(yz))$ .

$$X=aX+aX \rightsquigarrow X=aX \rightsquigarrow X=a(aX) \rightsquigarrow X=(aa)X$$

## 9.5 Abstraktion

Wird ein spezifiziertes System implementiert, so führt es i.A. mehr Aktionen aus, als in der Spezifikation vorgegeben sind. Diese Aktionen sind zwar für den internen Ablauf wichtig, für den Benutzer oder Auftraggeber jedoch nicht relevant. Um dies zu beschreiben, werden *stille* (silent), *spontane* oder *interne* Aktionen eingeführt. Ihre Bezeichnung ist meist  $\tau$  (Tau).

Der stille Systemschritt  $\tau$  kann ohne Vorbedingung ausgeführt werden und terminiert dann erfolgreich:

$$\overline{\tau \xrightarrow{\tau} \surd}$$

Der Kalkül ACP über  $A'$  mit  $\tau$ -Transition und Abstraktionsoperator wird mit  $ACP_\tau$  bezeichnet.

Die Transitionsregeln werden nun so erweitert, dass sie  $\tau$  enthalten, d.h. die neue Menge von Aktionen ist  $A' := A \cup \{\tau\}$  mit einer neuen Kommunikationsfunktion  $\gamma : A' \times A' \rightarrow A \cup \{\delta\}$  derart, dass das Bild immer  $\delta$  ist, falls im Argument ein  $\tau$  steht, d.h. es findet keine Kommunikation mit der internen Aktion statt.

Der *Abstraktions-Operator*  $\tau_I$ , mit  $I \subseteq A$ , benennt alle Aktionen aus  $I$ , die er als Argument führt, in  $\tau$  um:

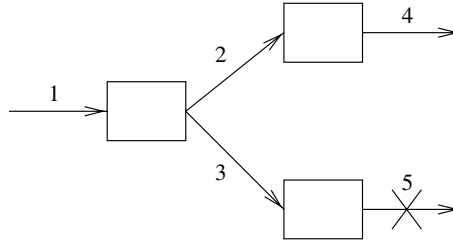
$$\begin{array}{cc} \frac{x \xrightarrow{v} \surd \ (v \notin I)}{\tau_I(x) \xrightarrow{v} \surd} & \frac{x \xrightarrow{v} x' \ (v \notin I)}{\tau_I(x) \xrightarrow{v} \tau_I(x')} \\ \frac{x \xrightarrow{v} \surd \ (v \in I)}{\tau_I(x) \xrightarrow{\tau} \surd} & \frac{x \xrightarrow{v} x' \ (v \in I)}{\tau_I(x) \xrightarrow{\tau} \tau_I(x')} \end{array}$$

Wenn in einer Folge  $abc$  von Aktionen  $b$  intern ist, dann ist  $a\tau c$  (für die externe Spezifikation) äquivalent zu  $ac$ , d.h. stille Aktionen sind (extern) wirkungslos.

Wie das folgende Beispiel zeigt, kann nicht generell still gleich wirkungslos gesetzt werden.

### Beispiel 9.46

Daten  $d$  werden über Kanal 1 empfangen ( $r_1(d)$ ) und über Kanal 2 oder 3 weitergeleitet ( $c_2(d), c_3(d)$ ). Im ersten Fall wird  $d$  über Kanal 4 weitergeleitet ( $s_4(d)$ ), im zweiten Fall ist dies jedoch nicht möglich, da der Kanal 5 blockiert ist, d.h.  $s_5(d)$  ist blockiert.





Dieses Verhalten wird durch folgenden Prozessausdruck beschrieben:

$$\begin{aligned} & \partial_{\{s_5(d)\}}(r_1(d) \cdot (c_2(d) \cdot s_4(d) + c_3(d) \cdot s_5(d))) \\ &= r_1(d) \cdot (c_2(d) \cdot s_4(d) + c_3(d) \cdot \delta) \end{aligned}$$

(Die Umformung erfolgt mit den Regeln D1,D2,D4,D5 von Seite 211.)

Spezifiziert man  $c_2(d)$  und  $c_3(d)$  als interne Aktionen, so erhält man

$$r_1(d) \cdot (\tau \cdot s_4(d) + \tau \cdot \delta).$$

Werden beide stille Aktionen  $\tau$  gestrichen, so erhält man mit  $r_1(d) \cdot (s_4(d) + \delta)$  einen Prozess ohne Verklemmung. Er wird daher als nicht (Verhaltens-)äquivalent betrachtet.

**Beispiel 9.47** Weitere nicht-äquivalente Prozessausdrücke:

- $a + \tau\delta$  und  $a$
- $\partial_{\{b\}}(a + \tau b)$  und  $\partial_{\{b\}}(a + b)$
- $a + \tau b$  und  $a + b$

Wenn still nicht generell mit wirkungslos gleichgesetzt werden kann, dann stellt sich die Frage: Welche  $\tau$ -Transitionen sind wirkungslos?

Antwort: Die  $\tau$ -Transitionen, deren Streichung nicht das Verhalten ändert.

Zum Beispiel:  $a + \tau(a + b)$  und  $a + b$ , denn nach der Ausführung von  $\tau$  ist  $a$  immer noch ausführbar. Dies wird durch die folgende Definition der *Verzweigungs-Bisimulation* (branching bisimulation) formalisiert. (Hierbei bezeichne  $\equiv$  die syntaktische Gleichheit.)

**Definition 9.48** Eine Verzweigungs-Bisimulation ist eine binäre Relation  $\mathcal{B}$  auf Prozessen, für die mit  $a \in A' = A \cup \{\tau\}$  gilt:

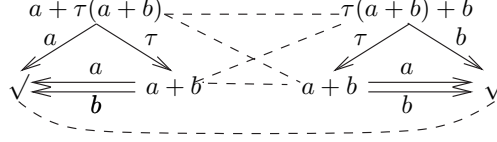
1. Wenn  $p \mathcal{B} q$  und  $p \xrightarrow{a} p'$ , dann
  - entweder  $a \equiv \tau$  und  $p' \mathcal{B} q$
  - oder  $q \xrightarrow{\tau} \dots \xrightarrow{\tau} q_0$  mit  $p \mathcal{B} q_0$  und  $q_0 \xrightarrow{a} q'$  mit  $p' \mathcal{B} q'$
2. Wenn  $p \mathcal{B} q$  und  $q \xrightarrow{a} q'$ , dann
  - entweder  $a \equiv \tau$  und  $p \mathcal{B} q'$
  - oder  $p \xrightarrow{\tau} \dots \xrightarrow{\tau} p_0$  mit  $p_0 \mathcal{B} q$  und  $p_0 \xrightarrow{a} p'$  mit  $p' \mathcal{B} q'$
3. Wenn  $p \mathcal{B} q$  und  $p$  ist terminiert, dann  $q \xrightarrow{\tau} \dots \xrightarrow{\tau} q_0$ , so dass  $p \mathcal{B} q_0$  und  $q_0$  ist terminiert.
4. Wenn  $p \mathcal{B} q$  und  $q$  ist terminiert, dann  $p \xrightarrow{\tau} \dots \xrightarrow{\tau} p_0$ , so dass  $p_0 \mathcal{B} q$  und  $p_0$  ist terminiert.

Die  $\tau$ -Folgen haben eine endliche Länge  $n \geq 0$ .

Zwei Prozesse  $p$  und  $q$  heißen verzweigungs-bisimilar (branching bisimilar), in Zeichen  $p \underline{\leftrightarrow}_b q$ , wenn es eine Verzweigungs-Bisimulations-Relation  $\mathcal{B}$  gibt mit  $p \mathcal{B} q$ .

**Beispiel 9.49**

$$a + \tau(a + b) \xrightarrow{b} \tau(a + b) + b$$



Die Relation  $\mathcal{B}$  ist definiert durch:

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

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

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

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

$$\surd \mathcal{B} \surd$$

**Aufgabe 9.50**

1. Geben Sie eine Verzweigungs-Bisimulations-Relation an, die zeigt dass  $\tau(\tau(a + b) + b) + a$  und  $a + b$  verzweigungsbisimilar sind.
2. Begründen Sie, warum  $\tau a + \tau b$  und  $a + b$  nicht verzweigungsbisimilar sind.

Die Verzweigungs-Bisimulations-Relation ist zwar eine Äquivalenzrelation, aber keine Kongruenz bezüglich BPA. Beispielsweise sind  $\tau a$  und  $a$  verzweigungsbisimilar, aber nicht  $\tau a + b$  und  $a + b$ .

Ansatz: Einige Terme werden nicht mehr als äquivalent angesehen, indem wir die Bedingungen verschärfen.

Milner [Mil89] hat gezeigt, dass man dieses Problem dadurch lösen kann, dass eine Initialisierungsbedingung gefordert wird: Initiale  $\tau$ -Transitionen werden nie eliminiert.

Dann wird in folgenden Fällen wird keine Äquivalenz erzeugt:

- $a + \tau\delta$  und  $a$
- $\partial_{\{b\}}(a + \tau b)$  und  $\partial_{\{b\}}(a + b)$
- $a + \tau b$  und  $a + b$
- $b$  und  $\tau b$

**Definition 9.51** Eine initiale Verzweigungs-Bisimulation (rooted branching bisimulation) ist eine binäre Relation  $\mathcal{B}$  auf Prozessen, für die mit  $a \in A' = A \cup \{\tau\}$  gilt:

1. Wenn  $p \mathcal{B} q$  und  $p \xrightarrow{a} p'$ , dann  $q \xrightarrow{a} q'$  mit  $p' \xrightarrow{b} q'$ .
2. Wenn  $p \mathcal{B} q$  und  $q \xrightarrow{a} q'$ , dann  $p \xrightarrow{a} p'$  mit  $p' \xrightarrow{b} q'$ .
3. Wenn  $p \mathcal{B} q$  und  $p$  terminiert, dann terminiert auch  $q$ .
4. Wenn  $p \mathcal{B} q$  und  $q$  terminiert, dann terminiert auch  $p$ .

Zwei Prozesse  $p$  und  $q$  heißen initial verzweigungsbisimilar (rooted branching bisimilar), in Zeichen  $p \xrightarrow{rb} q$ , wenn es eine initiale Verzweigungs-Bisimulations-Relation  $\mathcal{B}$  gibt mit  $p \mathcal{B} q$ .

Initiale Verzweigungs-Bisimulation ist (wie Verzweigungs-Bisimulation auch) eine Äquivalenzrelation. Das folgende Lemma zeigt, dass Initiale Verzweigungs-Bisimulation feiner als Verzweigungs-Bisimulation ist.

**Lemma 9.52** *Zwischen den Äquivalenzen gelten die Beziehungen:*

$$\Leftrightarrow \subseteq \Leftrightarrow_{rb} \subseteq \Leftrightarrow_b$$

*Falls  $\tau$  in den Termen nicht vorkommt (wie z.B. in ACP), dann gilt sogar:*

$$\Leftrightarrow = \Leftrightarrow_{rb} = \Leftrightarrow_b$$

**Aufgabe 9.53** Bestimmen Sie für jedes der folgenden Paare von Prozess-Termen, ob es bisimilar, initial verzweigungs-bisimilar oder verzweigungs-bisimilar ist bzw. nicht ist - möglichst mit stichhaltiger Begründung:

- a)  $(a + b)(c + d)$  und  $ac + ad + bc + bd$ ,
- b)  $(a + b)(c + d)$  und  $(b + a)(d + c) + a(c + d)$ ,
- c)  $\tau(b + a) + \tau(a + b)$  und  $a + b$ ,
- d)  $c(\tau(b + a) + \tau(a + b))$  und  $c(a + b)$  sowie
- e)  $a(\tau b + c)$  und  $a(b + \tau c)$ .

### 9.5.1 Geschützte lineare Rekursion

Betrachten wir lineare Rekursion im Zusammenhang mit  $\tau$ -Aktionen.

Alle Prozess-Terme  $\tau s$  stellen eine Lösung für die Spezifikation  $X = \tau X$  dar, da  $\tau s \Leftrightarrow_{rb} \tau \tau s$ .

Die  $\tau$ -Aktion reicht also nicht aus, um eine Eindeutigkeit zu garantieren. Wir müssen also den Begriff der geschützten Rekursion anpassen.

**Definition 9.54** 1. Eine rekursive Spezifikation ist linear, wenn ihre rekursiven Gleichungen die folgende Form haben:

$$X = a_1 X_1 + \dots + a_k X_k + b_1 + \dots + b_\ell \quad (a_i, b_j \in A \cup \{\tau\})$$

2. Eine lineare rekursive Spezifikation  $E$  ist geschützt, wenn es keine unendliche Folge von  $\tau$ -Transitionen der folgenden Form gibt:

$$\langle X|E \rangle \xrightarrow{\tau} \langle X'|E \rangle \xrightarrow{\tau} \langle X''|E \rangle \xrightarrow{\tau} \dots$$

Die geschützten linearen rekursiven Spezifikationen sind genau diejenigen rekursiven Spezifikationen, die eine eindeutige Lösung modulo initialer Verzweigungs-Bisimulation haben. Wieder geben wir eine Axiomatisierung des erweiterten Kalküls  $ACP_\tau$  an.

$ACP_\tau$  bezeichne jetzt den Kalkül ACP mit  $\tau$ -Aktion, Abstraktionsoperator, geschützter linearer Rekursion und den folgenden Axiomen:

**Axiome für Abstraktion** (mit  $v \in A' = A \cup \{\tau\}, I \subseteq A$ )

$$\begin{array}{ll}
 \text{B1} & v \cdot \tau = v \\
 \text{B2} & v \cdot (\tau \cdot (x + y) + x) = v \cdot (x + y) \\
 \\ 
 \text{TI1} & \tau_I(v) = v \ (v \notin I) \\
 \text{TI2} & \tau_I(v) = \tau \ (v \in I) \\
 \text{TI3} & \tau_I(\delta) = \delta \\
 \text{TI4} & \tau_I(x + y) = \tau_I(x) + \tau_I(y) \\
 \text{TI5} & \tau_I(x \cdot y) = \tau_I(x) \cdot \tau_I(y)
 \end{array}$$

**Satz 9.55** *Initiale Verzweigungs-Bisimulation ist eine Kongruenzrelation für  $ACP_\tau$  und geschützter linearer Rekursion.*

**Theorem 9.56** *Der Kalkül  $ACP_\tau$  mit Abstraktion und geschützter linearer Rekursion ist korrekt und vollständig in Bezug auf initiale Verzweigungs-bisimulation.*

**Aufgabe 9.57** Leiten sie  $\tau_{\{b\}}(\langle X | X = aY, Y = bX \rangle) = \langle Z | Z = aZ \rangle$  in  $ACP_\tau$  ab.

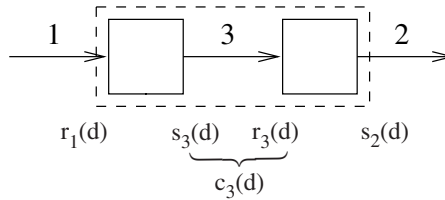
**Beispiel 9.58** (Tandempuffer)

Konvention: Um Ausdrücke zu vereinfachen, wird in den folgenden Beispielrechnungen eine Lösung  $\langle Q | Q = r_1 s_1 Q \rangle$  einer rekursiven Spezifikation abkürzend als  $Q$  geschrieben. Betrachtet werden zwei in Serie angeordnete Puffer der Kapazität 1 die mit Kanälen 1, 2 und 3 verbunden sind.  $r_i(d)$  bzw.  $s_i(d)$  bezeichne das Schreiben bzw. Lesen vom Kanal  $i$ .  $\Delta$  ist eine endliche Menge von Daten. Das Verhalten ist:

$$\begin{aligned}
 Q_1 &= \sum_{d \in \Delta} r_1 s_3 Q_1 \\
 Q_2 &= \sum_{d \in \Delta} r_3 s_2 Q_2
 \end{aligned}$$

wobei  $d \in \Delta$  weggelassen wird, d.h. wir tun so, als ob  $\Delta$  nur ein Element enthalten würde:

$$\begin{aligned}
 Q_1 &= r_1 s_3 Q_1 \\
 Q_2 &= r_3 s_2 Q_2
 \end{aligned}$$



Die Puffer  $Q_1$  und  $Q_2$  der Kapazität 1 arbeiten parallel und sind durch die Aktion  $\gamma(s_3, r_3) = c_3$  synchronisiert:

$$\tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_2 \parallel Q_1))$$

Wir beweisen, dass das gemeinsame Verhalten dasjenige eines Puffers der Kapazität 2 ist:

$$\begin{aligned} X &= r_1 Y \\ Y &= r_1 Z + s_2 X \\ Z &= s_2 Y \end{aligned}$$

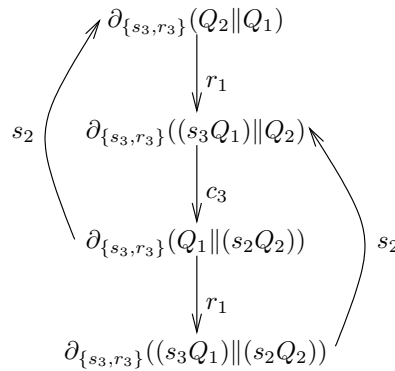
Dazu überlegen wir, was die Lösung dieses rekursiven Gleichungssystems  $E$  ist. Als erstes berechnen wir:

$$\begin{aligned} & Q_2 \parallel Q_1 \\ \stackrel{\text{M1}}{=} & Q_2 \sqcup Q_1 + Q_1 \sqcup Q_2 + Q_2 | Q_1 \\ \stackrel{\text{RDP}}{=} & (r_3 s_2 Q_2) \sqcup Q_1 + (r_1 s_3 Q_1) \sqcup Q_2 \\ & + (r_3 s_2 Q_2) | (r_1 s_3 Q_1) \\ \stackrel{\text{LM3,CM8}}{=} & r_3 \cdot ((s_2 Q_2) \parallel Q_1) + r_1 \cdot ((s_3 Q_1) \parallel Q_2) \\ & + \delta \cdot ((s_2 Q_2) \parallel (s_3 Q_1)) \\ \stackrel{\text{A7,A6}}{=} & r_3 \cdot ((s_2 Q_2) \parallel Q_1) + r_1 \cdot ((s_3 Q_1) \parallel Q_2) \\ \\ & \partial_{\{s_3, r_3\}}(Q_2 \parallel Q_1) \\ = & \partial_{\{s_3, r_3\}}(r_3 \cdot ((s_2 Q_2) \parallel Q_1) + r_1 \cdot ((s_3 Q_1) \parallel Q_2)) \\ = & \partial_{\{s_3, r_3\}}(r_3 \cdot ((s_2 Q_2) \parallel Q_1)) \\ & + \partial_{\{s_3, r_3\}}(r_1 \cdot ((s_3 Q_1) \parallel Q_2)) \\ = & \partial_{\{s_3, r_3\}}(r_3) \cdot \partial_{\{s_3, r_3\}}((s_2 Q_2) \parallel Q_1) \\ & + \partial_{\{s_3, r_3\}}(r_1) \cdot \partial_{\{s_3, r_3\}}((s_3 Q_1) \parallel Q_2) \\ = & \delta \cdot \partial_{\{s_3, r_3\}}((s_2 Q_2) \parallel Q_1) \\ & + r_1 \cdot \partial_{\{s_3, r_3\}}((s_3 Q_1) \parallel Q_2) \\ = & r_1 \cdot \partial_{\{s_3, r_3\}}((s_3 Q_1) \parallel Q_2) \end{aligned}$$

Weiter rechnen wir:

$$\begin{aligned} \partial_{\{s_3, r_3\}}((s_3 Q_1) \parallel Q_2) &= c_3 \cdot \partial_{\{s_3, r_3\}}(Q_1 \parallel (s_2 Q_2)) \\ \partial_{\{s_3, r_3\}}(Q_1 \parallel (s_2 Q_2)) &= r_1 \cdot \partial_{\{s_3, r_3\}}((s_3 Q_1) \parallel (s_2 Q_2)) \\ &\quad + s_2 \cdot \partial_{\{s_3, r_3\}}(Q_2 \parallel Q_1) \\ \partial_{\{s_3, r_3\}}((s_3 Q_1) \parallel (s_2 Q_2)) &= s_2 \cdot \partial_{\{s_3, r_3\}}((s_3 Q_1) \parallel Q_2) \end{aligned}$$

Wir fassen die gerechneten Transitionsübergänge zusammen:



Die Axiome für die  $\tau$ -Aktion und Abstraktion ergeben:

$$\begin{aligned}
 & \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_2 \| Q_1)) \\
 = & \tau_{\{c_3\}}(r_1 \cdot \partial_{\{s_3, r_3\}}((s_3 Q_1) \| Q_2)) \\
 = & r_1 \cdot \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}((s_3 Q_1) \| Q_2)) \\
 = & r_1 \cdot \tau_{\{c_3\}}(c_3 \cdot \partial_{\{s_3, r_3\}}(Q_1 \| (s_2 Q_2))) \\
 = & r_1 \cdot \tau \cdot \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_1 \| (s_2 Q_2))) \\
 = & r_1 \cdot \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_1 \| (s_2 Q_2)))
 \end{aligned}$$

Weiter rechnen wir:

$$\begin{aligned}
 \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_1 \| (s_2 Q_2))) &= \\
 & r_1 \cdot \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}((s_3 Q_1) \| (s_2 Q_2))) \\
 & + s_2 \cdot \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_2 \| Q_1)) \\
 \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}((s_3 Q_1) \| (s_2 Q_2))) &= \\
 & s_2 \cdot \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_1 \| (s_2 Q_2)))
 \end{aligned}$$

Damit erhalten wir als Lösung für die lineare rekursive Spezifikation  $E$  für einen Puffer der Kapazität 2:

$$\begin{aligned}
 X &:= \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_2 \| Q_1)) \\
 Y &:= \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}(Q_1 \| (s_2 Q_2))) \\
 Z &:= \tau_{\{c_3\}}(\partial_{\{s_3, r_3\}}((s_3 Q_1) \| (s_2 Q_2)))
 \end{aligned}$$

### 9.5.2 Die Fairness-Regel

Es ist möglich, durch Abstraktion  $\tau$ -Schleifen zu konstruieren: So führt  $\tau_{\{a\}}(\langle X \mid X = aX \rangle)$  unendlich lange die  $\tau$ -Aktion aus.

**Definition 9.59** Sei  $E$  eine geschützte lineare rekursive Spezifikation,  $C$  eine Teilmenge ihrer Variablen und  $I \subset A$  eine Menge von Aktionen.

- $C$  heißt (Fairness-)Gruppe (cluster) für  $I$ , falls für je zwei Rekursionsvariablen  $X$  und  $Y$  in  $C$   $\langle X \mid E \rangle \xrightarrow{b_1} \dots \xrightarrow{b_m} \langle Y \mid E \rangle$  und  $\langle Y \mid E \rangle \xrightarrow{c_1} \dots \xrightarrow{c_n} \langle X \mid E \rangle$  für Aktionen  $b_1, \dots, b_m, c_1, \dots, c_n \in I \cup \{\tau\}$  gilt.
- $a$  und  $aX$  heißen Ausgang (exit) der Gruppe  $C$  falls:
  1.  $a$  oder  $aX$  ein Summand auf der rechten Seite der Rekursionsgleichung für eine Rekursionsvariable in  $C$  ist, und
  2. im Fall  $aX$  zusätzlich  $a \notin I \cup \{\tau\}$  oder  $X \notin C$  gilt.

**Die Fairness-Regel CFAR:** Falls  $X$  in einer Gruppe  $C$  für  $I$  mit Ausgängen  $\{v_1 Y_1, \dots, v_m Y_m, w_1, \dots, w_n\}$  ist, dann gilt

$$\tau \cdot \tau_I(\langle X \mid E \rangle) = \tau \cdot \tau_I(v_1 \langle Y_1 \mid E \rangle + \dots + v_m \langle Y_m \mid E \rangle + w_1 + \dots + w_n)$$