Nebenläufige Prozesse und Reaktion

zu Kapitel 4 aus
Communicating and Mobile Systems: the Pi-Calculus, von Robin Milner

Arnaud Fietzke
Proseminar Theorie kommunizierender Systeme
Programming Systems Lab – Prof. Gert Smolka
Universität des Saarlandes

17. Oktober 2004

Inhaltsverzeichnis

1 Einleitung

Wir machen nun den Schritt von rein sequenziellen zu nebenläufigen Systemen. Im Wesentlichen entspricht die in diesem Kapitel eingeführte Teilsprache des π-Kalküls dem bereits 1980 in [3] eingeführten "Calculus of Communicating Systems" (CCS) ohne value-passing. Es handelt sich dabei um eine Prozess-Algebra, also eine algebraische Modellierung von Prozessen.
Ein Prozess wird hierbei in erster Linie als sein Verhalten, also eine Abfolge von atomaren Aktionen betrachtet. Zwei zentrale Begriffe sind dabei Beobachtung und Synchronisation. Beobachtung wird in Bezug auf einen externen Beobachter definiert, der bestimmte Abläufe im Prozess "sehen" kann, andere jedoch nicht.
Ziel wird es in den nachfolgenden Kapiteln sein, auf Basis der Beobachtung Äquivalenzbegriffe auf Prozessausdrücken zu definieren. Zwei Prozesse werden letztendlich als äquivalent betrachtet werden, wenn ein externer Beobachter sie nicht voneinander unterscheiden kann.
Natürlich entspricht der Beobachtung von außen auch eine Struktur im Inneren des Prozesses. Während ein Prozess sich für einen externen Beobachter wie eine "black box" verhält, aus deren Verhalten nicht ersichtlich ist, ob es sich um ein paralleles oder um ein rein sequentielles System handelt (siehe [4], Prop. 5.21), sollte eine Prozessalgebra natürlich die Beschreibung und Analyse der Struktur eines Prozesses ermöglichen. Interaktion zwischen nebenläufigen Komponenten wird beim vorliegenden Kalkül durch synchronisierte Transitionen modelliert.
Wir werden uns in diesem Kapitel auf die Beschreibung der strukturellen Eigenschaften von Prozessen konzentrieren; zum Verständnis der Semantik werden wir die Metapher der "Chemical Abstract Machine" von Berry und Boudol [1] verwenden.

2 Beobachtung und Reaktion

2.1 Von der Beobachtung zur Reaktion

Die Menge der Labels $\mathcal{L} := \mathcal{N} \cup \overline{\mathcal{N}}$, wurde bereits im letzten Kapitel eingeführt. Man konnte sie sich als "Knöpfe" auf einer "black box" vorstellen. Um eine Beobachtung von einer black box zu machen, musste man den entsprechenden Knopf drücken; die Beobachtung bestand darin, ob sich der Knopf drücken ließ oder nicht.

Gehen wir nun davon aus, dass ein Knopf nicht nur von einem Beobachter gedrückt werden kann, sondern dass die black box den Knopf auch selbst

auslösen kann. Stellen wir uns weiter vor, dass alle Knöpfe unterschiedliche Formen haben, die Formen von Knöpfen a und $\overline{a}$ jedoch jeweils ineinander passen, so können wir die Analogie erweitern und Paare $(a, \overline{a})$ als Kommunikationskanäle zwischen zwei black boxes ansehen, wobei die Aktion a der einen black box das Drücken des Knopfes $\overline{a}$ auf der anderen black box darstellt.

Ausgehend von endlichen Automaten wurde im letzten Kapitel ein Labelled Transition System (LTS) über Prozessausdrücken eingeführt. Die Labels entsprechen also gerade Zustandsänderungen (Transitionen) in Prozessen. Interaktion von zwei Prozessen über ein komplementäres Label-Paar bedeutet nun, dass beide Prozesse ihre jeweiligen Transitionen synchron ausführen; die beiden Aktionen verschmelzen zu einer neuen atomaren Aktion, die als **Reaktion** bezeichnet wird. Hier wird deutlich, wie Beobachter und System, bzw. zwei Systeme, die sich gegenseitig "beobachten", zu einem größeren System werden. Betrachten wir zum Beispiel das System $A|B$ mit

$$A := a.A' \quad B := b.B'$$
$$A' := \overline{b}.A' \quad B' := \overline{c}.B$$

so kann man folgendes Transitionsdiagramm zeichnen:

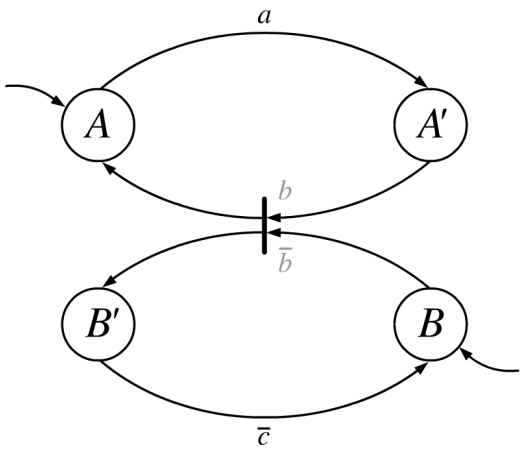

Abb. 1: Zwei nebenläufige Prozesse

Da an einem Interaktionspunkt immer zwei komplementäre Labels in Verbindung treten, kann ein Beobachter nicht mehr feststellen, welche Labels oder "Knöpfe" an einer Reaktion beteiligt waren. Er kann bestenfalls feststellen, dass eine *interne* Aktion stattgefunden hat. Interne Aktionen bzw.

Reaktionen werden deshalb durch ein spezielles Label dargestellt, welches kein Komplement besitzt:

$$Act := \mathcal{L} \cup \{\tau\}$$

2.2 Flowgraphs

Zur grafischen Veranschaulichung dieser Interaktionsbeziehungen führen wir **Flowgraphs** ein, sie stellen die Struktur eines Systems von Prozessen zu einem bestimmten Zeitpunkt dar. Hierbei werden jeweils Paare von komplementären Labels miteinander verbunden.

Diese Struktur kann sich im Laufe der Reaktionen verändern. Betrachten wir erneut das Sytem aus dem letzten Beispiel. Im Laufe der Reaktionen entstehen nacheinander folgende potentielle Interaktionspunkte:

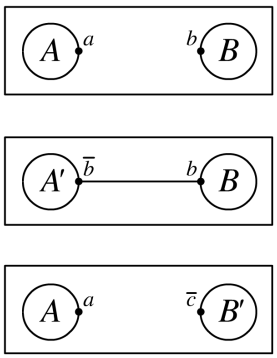

Abb. 2: Verschiedene Flowgrawphs im Laufe der Reaktion

3 Erweiterung der Prozess-Syntax

Flowgraphs ermöglichen eine anschauliche Darstellung der Struktur von parallelen Prozessen. Ziel ist jedoch weiterhin die Definition einer formalen Prozess-Algebra.

Wir erweitern daher die im letzten Kapitel eingeführte Syntax, um auch nebenläufige Prozesse beschreiben zu können. Wir führen hierzu zwei neue Operatoren ein, die **parallele Komposition** und die **Restriktion**:

Definition *Die Menge $\mathcal{P}$ der* nebenläufigen Prozessausdrücke *wird durch die folgende Syntax definiert:*

$$P ::= A\langle a_1, ..., a_n\rangle \mid \sum_{i\in I} \alpha_i.P_i \mid P_1|P_2 \mid \mathsf{new}\ a\ P$$

mit I endliche Indexmenge.

$P_1|P_2$ stellt gerade die nebenläufige Ausführung der Prozesse P_1 und P_2 dar. Die Restriktion $\mathsf{new}\ a\ P$ beschränkt den Skopus von a auf den Ausdruck P. Insbesondere bedeutet dies, dass der Name a durch $\mathsf{new}\ a$ **gebunden** wird. **Freie** Namen eines Ausdrucks P, geschrieben $\mathrm{fn}(P)$, sind gerade die Namen, die durch kein new gebunden werden.

Wir werden von nun an M und N als Platzhalter für Summen verwenden.

Es gelten darüber hinaus folgende Konventionen: Die Operationen $\alpha.$ und $\mathsf{new}\ a$ binden stärker als Summe und Komposition, zum Beispiel steht

$$\mathsf{new}\ a\ P|Q$$

für

$$(\mathsf{new}\ a)\ P|Q$$

und nicht etwa

$$\mathsf{new}\ a\ (P|Q)$$

Darüber hinaus werden mehrstellige Summen innerhalb von Restriktionen oder Kompositionen geklammert:

$$(a.P + b.Q)|c.R$$

Falls $I = \emptyset$, schreiben wir die leere Summe $\sum_{i\in I} \alpha_i.P_i$ als $\mathsf{0}$.

Wie im vorherigen Kapitel nehmen wir weiterhin an, dass jeder Prozessbezeichner A eine definierende Gleichung der Form

$$A(\vec{a}) := P_A$$

hat, wobei $\vec{a} = a_1..a_n$ alle freien Bezeichner von P_A enthält. $A\left\langle \vec{b} \right\rangle$ steht dann für $\left\{ \vec{b}/\vec{a} \right\} P_A$

Um eine Substitution korrekt durchzuführen, muss eventuell alpha-renaming angewendet werden, zum Beispiel gilt für $P = (\mathsf{new}\ b)a.b$:

$$\{b/a\}\ P = (\mathsf{new}\ b')b.b'$$

4 Semantik nebenläufiger Prozesse

4.1 Die Chemical Abstract Machine

Wir haben im ersten Abschnitt zur Veranschaulichung der Reaktionssemantik Prozesse als "black boxes" mit "Knöpfen" betrachtet. Diese Analogie stößt jedoch an ihre Grenzen, wenn wir die Dynamik von Prozessinteraktionen untersuchen wollen. Betrachtet man zum Beispiel einen Ausdruck der Form

$$a.A \mid \overline{a}.(B|C)$$

so müsste sich die zweite black box nach der Reaktion in zwei neue aufspalten – dies ist nicht wirklich intuitiv.

Um das Verhalten paralleler Prozesse zur verstehen, greifen wir daher auf das Modell der **"Chemical Abstract Machine"** (kurz Cham) aus [1] zurück. In ähnlicher Weise wie die Turingmaschine oder die "Categorical Abstract Machine" [2] als Modelle für sequentielle Berechnung dienen, lässt sich mit der Cham nebenläufige Berechnung auf intuitive Weise darstellen.

Die Grundidee der Cham ist die folgende: Prozessausdrücke entsprechen Molekülen in einer Lösung – durch parallele Komposition können immer neue Moleküle hinzugefügt werden. Die Lösung befindet sich in ständiger Bewegung, sodass jedes Molekül auf jedes andere treffen kann. Haben zwei Moleküle komplementäre Labels (man stelle sich $a.P_1$ und $\overline{a}.P_2$ als zwei entgegengesetzt geladene Ionen vor), so können sie miteinander reagieren und in neue Moleküle zerfallen, oder sich vollständig auflösen ($\mathsf{0}$).

Interessant ist bei diesem Modell, dass die Assoziativität und Kommutativität der parallelen Komposition auf natürliche Weise durch das ungeordnete "herumschwimmen" (sozusagen die Brownsche Molekularbewegung) der Moleküle in der Lösung modelliert wird.

Die formale Beschreibung der Cham hier aufzugreifen, würde den Rahmen dieses Papieres sprengen, deshalb verzichten wir auf Konzepte wie "Erhitzung", "Abkühlung" oder "Membranen", die in [1] eingeführt werden. Wir werden uns auf die Begriffe "Annäherung" und "Reaktion" beschränken. Moleküle können sich einander "annähern" und sie können reagieren.

Diesen beiden Begriffen entsprechen in der hier vorgestellten Teilsprache des π-Kalküls zwei Konzepte: die strukturelle Kongruenz und die Reaktions-Relation.

4.2 Strukturelle Kongruenz

Bei der strukturellen Kongruenz handelt es sich um eine rein syntaktische Äquivalenzrelation zwischen Prozessausdrücken. Sie wird es uns jedoch ermöglichen, potentielle "Reaktionspartner" auch tatsächlich reagieren zu lassen. Sie entspricht damit der "Annäherung" in der Cham. Warum dies der Fall ist, wird klar werden, wenn wir die formale Definition von Reaktion betrachen (siehe 4.3). Wir definieren hierzu zunächst eine allgemeine Prozess-Kongruenz. Um die Definitionen zu vereinfachen, führen wir zunächst den Begriff des **Prozess-Kontexts** ein:

Definition *Ein* Prozess-Kontext $\mathcal{C}$ *wird durch folgende Syntax definiert:*

$$\mathcal{C} ::= [\,] \;\mid\; \alpha.\mathcal{C} + M \;\mid\; \mathsf{new}\; a\; \mathcal{C} \;\mid\; \mathcal{C} \mid P \;\mid\; P \mid \mathcal{C}$$

wobei P ein Prozessausdruck ist. Informell ist ein Prozess-Kontext also ein Prozessausdruck mit einer Leerstelle. $\mathcal{C}\,[Q]$ beschreibt das Einsetzen von Prozessausdruck Q in Prozess-Kontext $\mathcal{C}$.
Insbesondere ist $\mathcal{C} = [\,]$ der Identitäts-Kontext, also $\mathcal{C}\,[Q] = Q$.

Eine **Kongruenz-Relation** $\cong$ über $\mathcal{P}$ lässt sich nun wie folgt definieren:

Definition *Sei $\cong$ eine Äquivalenzrelation (reflexiv, transitiv, symmetrisch) auf $\mathcal{P}$. Dann heisst* $\cong$ Prozess-Kongruenz, *wenn für alle Prozessausdrücke P, Q mit $P \cong Q$ gilt:*

$$\begin{aligned} \alpha.P + M &\cong \alpha.Q + M \\ \mathsf{new}\; a\; P &\cong \mathsf{new}\; a\; Q \\ P|R &\cong Q|R \\ R|P &\cong R|Q \end{aligned}$$

Es ist nun möglich, eine bestimmte Prozess-Kongruenz $\cong$ mit Hilfe eines Gleichungssystems $\mathcal{E}$ zu definieren:
$\cong$ ist die kleinste Prozess-Kongruenz, die folgende Bedingungen erfüllt:

- $\cong$ erfüllt $\mathcal{E}$
- $Q_1 \cong Q_n$ für jede Folge von Prozessausdrücken $Q_1 ... Q_n$ mit $Q_i = \mathcal{C}[P]$ und $Q_{i+1} = \mathcal{C}[P']$ für alle i, wobei $\mathcal{C}$ ein Prozess-Kontext und $P \cong P' \in \mathcal{E}$ oder $P' \cong P \in \mathcal{E}$

Intuitiv bedeutet dies, dass $Q \cong R$ genau dann, wenn Q durch wiederholte Anwendung einer Gleichung aus $\mathcal{E}$ auf beliebige Terme oder Subterme nach R transformiert werden kann.

Wir können nun die **strukturelle Kongruenz** auf Prozessausdrücken definieren:

Definition *Strukturelle Kongruenz $\equiv$ ist die Prozess-Kongruenz auf $\mathcal{P}$, die durch folgende Gleichungen bestimmt wird:*

1. Ändern gebundener Namen (alpha-renaming)
2. Umordnen der Terme in einer Summe
3. $P|\mathsf{0} \equiv P$, $P|Q \equiv Q|P$, $P|(Q|R) \equiv (P|Q)|R$
4. $\mathsf{new}\ a\ (P|Q) \equiv P|\mathsf{new}\ a\ Q \quad \textit{falls}\ a \notin \mathrm{fn}(P)$,
   $\mathsf{new}\ a\ \mathsf{0} \equiv \mathsf{0}$, $\mathsf{new}\ ab\ P \equiv \mathsf{new}\ ba\ P$
5. $A\left\langle \vec{b} \right\rangle \equiv \left\{ \vec{b}/\vec{a} \right\} P_A \quad \textit{falls}\ A(\vec{a}) := P_A$

Hier ist anzumerken, dass die zweite Regel zwar in [4] angegeben wird, im Grunde aber überflüssig ist, da Summen bereits als ungeordnet definiert wurden.

4.3 Reaktion

Wir kehren nun zur Chemical Abstract Machine zurück, und betrachten Reaktionen genauer. Wie schon oben erwähnt, finden Reaktionen dort statt, wo zwei komplementäre Labels in einer parallelen Komposition aufeinander treffen. Betrachtet man das aus der Komposition zweier Prozessausdrücke entstehende neue System, so zeigt sich eine Reaktion als τ-Transition.

Definition *Die Reaktions-Relation $\rightarrow$ über $\mathcal{P}$ enthält genau die Transitionen, die durch folgende Regeln inferiert werden können:*

$$\text{TAU}: \quad \tau.P + M \rightarrow P$$

$$\text{REACT}: \quad (a.P + M)|(\overline{a}.Q + N) \rightarrow P|Q$$

$$\text{PAR}: \quad \frac{P \rightarrow P'}{P|Q \rightarrow P'|Q}$$

$$\text{RES}: \quad \frac{P \rightarrow P'}{\mathsf{new}\ a\ P \rightarrow \mathsf{new}\ a\ P'}$$

$$\text{STRUCT}: \quad \frac{P \rightarrow P'}{Q \rightarrow Q'}\ falls\ P \equiv Q\ und\ P' \equiv Q'$$

Nun wird klar, warum die Strukturelle Kongruenz als "Annäherung" von Molekülen in der Cham betrachtet werden kann. Die STRUCT-Regel ermöglicht es nämlich, an jeder beliebigen Stelle in der Inferenz die strukturelle Kongruenz zu verwenden, um Prozessausdrücke in die Form der Inferenzregeln zu bringen und so Reaktionen ableiten zu können.

So ist jede Summe, die ein τ als Aktionspräfix enthält, kongruent zu $\tau.P + M$, und $\rightarrow P$ kann abgeleitet werden (Regel (2) der strukturellen Kongruenz).
Genauso kann jede Komposition von zwei Summen, die jeweils komplementäre Labels als Aktionspräfixe haben, durch alpha-renaming, Umordnung von Summen, Assoziativität und Kommutativität des "|"-Operators in die Form $(a.P + M)|(\overline{a}.Q + N)$ der REACT-Regel gebracht werden.
Für die PAR-Regel und die RES-Regel sind jeweils die Gleichungen aus (3) und (4) relevant.
Darüber hinaus ermöglicht die Gleichung $P|\mathsf{0} \cong P$, leere Summen (also "beendete" Prozesse bzw. "evaporierte" Moleküle) aus den Ausdrücken zu entfernen.

Die strukturelle Kongruenz entspricht also der "Annäherung" der Moleküle, da sie potenzielle Reaktionen für die Reaktions-Regeln erst "sichtbar" macht.

Es mag auf den ersten Blick seltsam erscheinen, dass die TAU-Regel explizit angegeben wird, da eine Reaktion per Definition zwischen komplementären Labels stattfindet, und dies bereits durch die REACT-Regel abgedeckt wird. In der Tat wird im nächsten Kapitel die Implikation

$$P \rightarrow P' \Rightarrow P \xrightarrow{\tau} \equiv P'$$

bewiesen werden.

Die Bedeutung des τ als Aktionspräfix wird jedoch deutlicher, wenn man ein Beispiel betrachtet. Wir geben eine Spezifikation für eine Lotterie wie folgt an:

$$Lotspec = \tau.b_1.Lotspec + ... + \tau.b_n.Lotspec$$

Die intendierte Bedeutung ist, dass die Lotterie eine interne Auswahl treffen kann, und somit indeterministisch einen Summanden auswählen kann. Danach kann der Beobachter den "ausgeworfenen" Ball $b_i (1 < i < n)$ "herausnehmen" (indem er $\overline{b_i}$ ausführt). Alternativ können wir dieselbe Spezifikation auch so schreiben:

$$Lotspec = a.\mathsf{0} \mid (\overline{a}.b_1.Lotspec + ... + \overline{a}.b_n.Lotspec)$$

Für den Beobachter würden sich die beiden Systeme identisch verhalten. Durch die explizite Angabe des τ im ersten Fall wird jedoch deutlich gemacht, dass eine interne Aktion stattfindet. Wodurch diese hervorgerufen wird, ist in diesem Fall irrelevant. Das τ dient also der Abstraktion von internen Aktionen.

Es sollte an dieser Stelle auch noch einmal daran erinnert werden, dass die Reaktions-Relation nur das *interne* Verhalten eines Prozesses beschreibt, und keine Aussagen über beobachtbare Aktionen macht. So kann es vorkommen, dass ein Prozess P keine Reaktionen mehr zulässt, also $P \not\rightarrow$; in diesem Fall wird P als **stabil** bezeichnet. P kann dann nur durch *externe* Interaktion wieder zu weiteren Reaktionen gebracht werden (wenn überhaupt).

4.4 Indeterminismus

Im Gegensatz zum λ-Kalkül, in dem für jeden Ausdruck eine eindeutige Auswertung angegeben werden kann (siehe Church-Rosser-Eigenschaft), ist CCS inhärent indeterministisch. Da es sich bei $\rightarrow$ um eine Relation handelt, kann es für einen Ausdruck mehrere mögliche Reaktionen geben. Zum einen geschieht das, wenn, wie in der Lotterie-Spezifikation, ein τ explizit als Aktions-Präfix angegeben wird. Der gleiche Effekt lässt sich aber auch wie in der alternativen Spezifikation erzielen; da die Summe exklusive Alternativen darstellt, fallen bei der Reaktion alle unbeteiligten Summanden weg. Ein Summand kann also sowohl durch externe Interaktion als auch durch interne ausgewählt werden.

Im Kontext der Cham kann man sich diese externe Auswahl als Vorhandensein mehrer Interaktionspunkte an der Moleküloberfläche vorstellen - findet an einem Punkt eine Reaktion mit einem anderen Molekül statt, so verschwinden die entsprechenden Teile des Moleküls. Die *interne* Auswahl

entspricht dann dem spontanen Zerfall eines Moleküls in ein bzw. mehrere kleinere.

Indeterminismus kann jedoch auch durch Interaktion mehrerer Moleküle stattfinden. Befindet sich zum Beispiel ein "positives" a-Ion mit mehreren "negativen" $\overline{a}$-Ionen in einer Lösung, so kann es nur mit einem von ihnen reagieren, da es danach verschwindet. Mit welchem es reagiert, ist jedoch nicht festgelegt. In CCS geschieht dies, indem mit einem Ausdruck der Form $a.Q$ mehrere Ausdrücke der Form $\overline{a}.P_i$ komponiert werden:

$$P := a.0 \mid \overline{a}.P_1 \mid ... \mid \overline{a}.P_n$$

Im Unterschied zur Auswahl von Summanden bleiben hier die anderen Komponenten erhalten, da sie ja gerade die unbeteiligten Moleküle darstellen:

$P \rightarrow P_1 \mid \overline{a}.P_2 \mid ... \mid \overline{a}.P_n$
...
$P \rightarrow \overline{a}.P_1 \mid ... \mid \overline{a}.P_{n-1} \mid P_n$

4.5 Standardform

Definition *Ein Prozessausdruck ist in* Standardform, *wenn er die Form* $\mathsf{new}\ \vec{a}\ (M_1|...|M_n)$ *hat, wobei jedes* M_i *eine nicht-leere Summe ist.*
Im Fall $n = 0$ *steht im Inneren der Klammer* 0*;*
falls $\vec{a}$ *leer ist, fällt* $\mathsf{new}\ \vec{a}$ *weg.*

Wir zeigen nun, dass *jeder Prozessausdruck strukturell kongruent zu einer Standardform* ist: Jede Restriktion $\mathsf{new}\ a$, die nicht innerhalb einer Summe steht, kann durch die Regel $P|\mathsf{new}\ a\ Q \equiv \mathsf{new}\ a\ (P|Q)$ (zusammen mit alpha-renaming) nach aussen gezogen werden. Zusammen mit Assoziativität und Kommutativität der Komposition lässt sich somit jeder Ausdruck in die Form $\mathsf{new}\ \vec{a}\ (M_1|...|M_n)$ bringen.
Leere Summen (0) können mit der Regel $P|0 \equiv P$ eliminiert werden.

Zudem kann man sicherstellen, dass die äußere Restriktion $\mathsf{new}\ \vec{a}$ genau die Namen enthält, die in mindestens einem Faktor M_i frei vorkommen: mit der 4. Regel der strukturellen Kongruenz lässt sich zeigen, dass $\mathsf{new}\ a\ P \equiv P$ falls a nicht frei in P vorkommt – daraus folgt die Behauptung.

Schaut man sich die Standardform genauer an, so stellt man fest, dass sie gerade die chemische Lösung der Cham beschreibt. Wie wir im letzten Abschnitt gesehen haben, stellt eine Summe gerade ein Molekül mit mehreren "Ports" dar. Die parallele Komposition entspricht dem Hinzufügen neuer

Moleküle in die Lösung. Das "Herausziehen" der Restriktionen nach außen und das damit einhergehende alpha-renaming stellen sicher, dass Namen, die vorher lokal deklariert waren, in der Restriktions-freien Komposition (der chemischen Lösung) nicht an Reaktionen teilnehmen:

$$(\mathsf{new}\ a)\ a.P_1 \mid (\mathsf{new}\ a)\ a.P_2 \mid a.P_3 \mid \overline{a}.P_4 \equiv \mathsf{new}\ xy\ (x.P_1 \mid y.P_2 \mid a.P_3 \mid \overline{a}.P_4)$$

4.6 Bemerkung

Im vorherigen Kapitel wurde das Labelled Transition System (LTS) über sequentiellen Prozessausdrücken eingeführt, im nachfolgenden Kapitel wird ein LTS für nebenläufige Prozessausdrücke (die in diesem Kapitel eingeführt wurden) definiert. Die Regeln, mit denen die Transitionen in diesem LTS inferiert werden, beschreiben gerade das Verhalten von nebenläufigen Prozessen bei Interaktion mit einem externen Beobachter. Da das LTS im Prinzip ein unendlicher Automat (ohne Start- und Endzustand) ist, gibt es zu jedem Zeitpunkt immer nur einen aktiven Zustand und in jedem Schritt wird höchstens eine Transition durchgeführt. Dies führt zur sogenannten Interleaving-Semantik, in der unabhängige Nebeneinanderausführung auf indeterministische sequentielle Ausführung reduziert wird:

$$a.b|c.d \ \sim\ a.b.c.d + a.c.b.d + c.d.a.b + c.a.b.d$$

Diese steht im Gegensatz zu anderen Semantiken, wie z.B. der von Petrinetzen. Befinden sich dort zwei Transitionen in einer "concurrency relation", so werden sie als von einander kausal unabhängig betrachtet und können in beliebiger Reihenfolge oder gleichzeitig ausgeführt werden. Diese Relation ist im hier vorgestellten Kalkül nicht vorhanden. Wenn zwei Prozesse komponiert werden, wird das resultierende System durch die Synchronisation ihrer Interaktionen bestimmt; ihre unabhängigen Aktionen können in beliebiger Reihenfolge auftreten, aber nicht simultan. Der Grund hierfür ist, dass wir vom externen Beobachter annehmen, dass er zu jedem Zeitpunkt nur eine Beobachtung machen kann. Diese Annahme ermöglicht die Einfachheit des Kalküls. Wie später gezeigt werden wird, ist diese Modellierung für unsere Zwecke ausreichend; für die schwache Äquivalenz ist es nicht nötig, alle möglichen Permutationen unabhängiger Aktionen zu betrachten. Damit $A \approx B$ gilt, genügt es zu zeigen, dass jede Aktion von A auch von B (eingeschlossen in eine beliebige Anzahl von τ-Transitionen) durchgeführt werden kann, und umgekehrt.
Die schwache Äquivalenz $\approx$ wird nun im 5. Kapitel von [4] definiert.

Literatur

[1] Gérard Berry and Gérard Boudol. The chemical abstract machine. In *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, Annual Symposium on Principles of Programming Languages, pages 81–94. ACM Press, 1989.

[2] Pierre-Louis Curien. *Categorical Combinators, Sequential Algorithms, and Functional Programming*. Research Notes in Theoretical Computer Science. Pitman, London, John Wiley & Sons, New York, Toronto, 1986.

[3] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.

[4] Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, 1999.