REIHE INFORMATIK
35/95

Parallelität in Basic LOTOS

H. T. Do
Universität Mannheim
Fakultät für Mathematik und Informatik
Lehrstuhl für Praktische Informatik I
Seminargebäude A5
D-68131 Mannheim

Parallelität in Basic LOTOS

Universität Mannheim
Fakultät für Mathematik und Informatik
Lehrstuhl für Praktische Informatik I
Seminargebäude A5
E-mail: do@pi1.informatik.uni-mannheim.de

H. T. DO

Zusammenfassung

In diesem Bericht werden Parallelitätsbegriffe auf der Halbordnungssemantik nach [Lan92] für Basic LOTOS eingeführt und diskutiert. Darunter versteht man die Unterscheidung zweier äquivalenter Prozesse $B1$ und $B2$, wenn $B1$ (bzw. $B2$) im Laufe seines Ablaufs mehr Aktionen parallel zur Ausführung bringen kann als $B2$ (bzw. $B1$). Ferner wird ein Ausblick auf einen Ansatz zur Parallelisierung von parallelen Prozessen gegeben.

Inhaltsverzeichnis

1 Einführung

Durch die rasante Entwicklung in den letzen Jahrzehnten im Bereich verteilter Systeme ist deren korrekte Entwurf heutzutage ohne formale Spezifikationssprachen kaum denkbar. Formale Sprachen wie LOTOS, ESTELLE und SDL [Hog89], denen eine eindeutige Semantik zugeordnet wird, sind natürlicher Sprachen gegenüber eindeutig interpretierbar. Die potentiellen Mißverständnisse und Fehler, die sich oft aus unpräzisen informellen Beschreibungen ergeben, treten nicht mehr auf. Dies ist deshalb von Bedeutung, da Entwickler eine gemeinsame Sprache sprechen müssen, wenn verteilte Systeme von verschiedenen Gruppen entwickelt werden.

Formale Sprachen besitzen im Hinblick auf Korrektheit und innere Konsistenz von Systemen eine angenehme Eigenschaft, nämlich, daß sie eine mathematische Theorie anbieten, die die Verifikation von verteilten Systemen erlaubt. Beispiele dazu findet man auf dem Gebiet der Spezifikation von Kommuniktionsprotokollen, wo oft verifiziert wird, daß eine Protokollspezifikation eine Dienstspezifikation erfüllt. In der Praxis scheitert jedoch oft eine solche Verifkation an der Komplexität eines verteilten Systems.

Um dieses Problem zu umgehen, haben sich in den letzten Jahren zwei Lösungsansätze entwickelt. Beim ersten Ansatz wird anstatt Verifikation das Testen von verteilten Systemen durchgeführt. Somit wird natürlich deren Korrektheit nicht vollständig gezeigt, sondern von dem Teil des Systems, der den vom Benutzer gestellten Anforderungen genügt [Bri88]. Der zweite Ansatz basiert auf semantikerhaltenden Transformationen. Dabei wird die Ausgangsspezifikation eines Systems in eine verfeinerte Spezifikation transformiert, die im Vergleich zu der Ausgangsspezifikation nach wie vor funktional korrekt ist, jedoch mehr Informationen über die innere Struktur des Systems enthalten kann. Beispielsweise findet man in [Lan92] und [BL95] die Transformationsmethode „splitting processes“, bei der ein Prozeß in zwei Teilprozesse zerlegt wird, die miteinander synchron bzw. asynchron kommunizieren. Diese Methode wird in der Praxis oft angewendet, wenn es darum geht, aus einer vorgebenenen Dienstspezifikation systematisch eine Protokollspezifikation zu bestimmen, die aus genau zwei Protokollinstanzen besteht. Sollte die Protokollspezifikation aus mehr als zwei Protokollinstanzen bestehen, die über einen zuverlässigen, d.h. fehlerfreien, Kanal kommunizieren, so könnte diese mit der Methode von [KBK89] bestimmt werden.

In [PHQ+92] wurde eine andere Transformationsmethode, „inverse expansion“ genannt, entwickelt, die in Abhängigkeit von zwei vorgebenenen Aktionenmengen die Rückführung des sogenannten Expansionstheorems bestimmt. Verwandt mit „inverse expansion“ ist die Transformation von [Jan85], womit ein sequentielles System in Abhängigkeit von mehr als zwei vorgegebenen Aktionenmengen in ein paralleles System transformiert wird. Darüberhinaus gibt es weitere Arten von Transformationen, die hier nicht mehr weiter aufgezählt werden können. Eine Sammlung von Transformationsmethoden findet man in [Bol92].

Die bis jetzt bekannten Transformationen berücksichtigen den Grad der Parallelität nicht. D.h. die Anzahl der parallel (gleichzeitig) ausführbaren Aktionen eines transformierten Prozesses wird nicht beachtet. Daher ist es interessant, eine Transformationsmethode zu finden, die diesen Aspekt berücksichtigt. Steht einmal eine solche Methode zur Verfügung, so besteht die berechtigte Hoffnung, mit ihrer Hilfe die Ausführungsgeschwindigkeit eines Prozesses zu steigern.

Um dieses Ziel zu verfolgen, werden in diesem Bericht vier Parallelitätsbegriffe auf der

Halbordnungssemantik, die Langerak für Basic LOTOS definiert hat [Lan92], eingeführt und diskutiert. Drei davon basieren auf dem Parallelitätsbegriff von [Ace91], der für eine CCS-artige Sprache ohne Synchronisation eingeführt wurde. Beim vierten Begriff handelt es sich um den Paralellitätsgrad eines Prozesses, d.h. die maximale Anzahl der Aktionen, die im Laufe des Prozeßablaufes parallel ausgeführt werden können.

Der Bericht gliedert sich wie folgt:

Im Abschnitt 2 werden die Syntax, die operationelle Semantik sowie die Halbordnungssemantik von Basic LOTOS aus [Lan92] wiedergegeben. Im Abschnitt 3 werden Parallelitätsbegriffe eingeführt. Ferner wird ein Ausblick auf einen Ansatz zur Parallelisierung von parallelen Prozessen gegeben. Im Abschnitt 4 werden die Ergebnisse zusammengefaßt.

2 Syntax und Semantiken von Basic LOTOS

Dieser Abschnitt gibt als Grundlage die formale Definition der Syntax, der operationellen Semantik und der Halbordnungssemantik (auch True-Concurrency-Semantik genannt) von Basic LOTOS aus [Lan92] wieder. Dabei werden auch Begriffe, für die Langerak in seiner Arbeit nur eine informelle Beschreibung gegeben hat, formal definiert. Die meisten Notationen lehnen sich nach wie vor an [Lan92]. Die prinzipiellen Unterschiede und die Intentionen der beiden Semantiken werden kurz erläutert. Für Einzelheiten wird der Leser auf [Lan92, Rei87] verwiesen.

2.1 Syntax

Wie der Name bereits darauf hindeutet, ist Basic LOTOS eine Teilsprache von (Full) LOTOS (= Basic LOTOS + abstrakte Datentypen). In Basic LOTOS können Datentypen, die z.B. oft in der Spezifikation von Kommunikationsprotokollen vorkommen, nicht beschrieben werden. Dennoch werden die meisten, theoretischen Untersuchungen nach wie vor auf dieser Teilsprache durchgeführt. Der Grund ist: 1) Basic LOTOS basiert auf dem Konzept der Prozeßkalküle CCS [Mil89] und CSP [Hoa85] und ist daher theoretisch leichter zugänglich. 2) Prozesse dargestellt in Full LOTOS lassen sich unter Einschränkung, daß die Datentypen endlich sind, mit Basic LOTOS semantisch äquivalent darstellen [Bol92]. Damit lassen sich viele Eigenschaften von Prozessen leicht verifizieren, die sich mit Full LOTOS nicht oder nur schwer behandeln lassen.

Leser, die sich für Full LOTOS interessieren, werden auf [ISO89] verwiesen.

Definition 2.1 (Basic LOTOS) *Sei $\mathcal{G}$ eine Menge von Aktionsnamen, $\mathcal{P}$ eine Menge von Prozeßnamen(-variablen), $g \in \mathcal{G} \cup \{\mathbf{i}\}$ mit $\mathbf{i} \notin \mathcal{G}$ als die interne Aktion, $g_i, a_i, b_i \in \mathcal{G}$ mit $1 \leq i \leq n$ und $P \in \mathcal{P}$. Dann ist Basic LOTOS, bezeichnet als $\mathcal{BL}$, durch folgende Grammatik*

$$
\begin{array}{rcl}
B & ::= & \mathbf{stop} \;\mid\; \mathbf{exit} \;\mid\; g;B \;\mid\; B\,[]\,B \;\mid\; B \gg B \;\mid\; B\,[>\,B \;\mid\; \\
 & & \mathbf{hide}\ g_1,\ldots,g_n\ \mathbf{in}\ B \;\mid\; B[b_1/a_1,\ldots,b_n/a_n] \;\mid\; \\
 & & B\,|[g_1,\ldots,g_n]|\,B \;\mid\; P
\end{array}
$$

definiert. □

Informell beschrieben haben die Operatoren folgende Bedeutung:

- *inaction*: **stop**

  Mit *stop* wird ein Prozeß beschrieben, der nichts tut und somit nach außen kein Verhalten zeigt. *stop* dient in der Praxis der Darstellung eines Verklemmungszustandes.

- *successfull termination*: **exit**

  exit repräsentiert einen Prozeß, der erfolgreich terminiert ist. *exit* unterscheidet sich von *stop* dadurch, daß es sich hier nicht um einen Verklemmungszustand, sondern um einen erfolgreich terminierten Zustand handelt.

- *action prefix*: $g; B$

  Der Prozeß $g; B$ beschreibt einen Prozeß, der zunächst die Aktion g ausführt und sich anschließend wie B verhält. Dabei ist $g \in \mathcal{G}$ eine beobachtbare (d.h. nach außen sichtbare) Aktion und $g = \mathbf{i}$ eine nicht beobachtbare (d.h. interne, nach außen unsichtbare) Aktion.

- *choice*: $B_1 \,[]\, B_2$

  Der Prozeß $B_1 \,[]\, B_2$ verhält sich entweder wie der Teilprozeß B_1 oder wie der Teilprozeß B_2. Die Entscheidung fällt zugunsten desjenigen Teilprozesses, der zuerst eine Aktion ausführt. Ist diese Aktion von B_1 (bzw. B_2), so wird B_2 (bzw. B_1) aus dem weiteren Prozeßgeschehen entfernt.

  In der Praxis wird der Auswahloperator [] oft nur dann verwendet, wenn der Prozeß der Umgebung eine Auswahl von beobachtbaren Aktionen anbietet, die von der Umgebung zur Ausführung gebracht werden. Die Entscheidung, welcher Prozeß als erster eine beobachtbare Aktion ausführen darf, wird durch die Interaktion (Synchronisation) mit der Umgebung getroffen.

- *enabling*: $B_1 \gg B_2$

  Der Prozeß $B_1 \gg B_2$ beschreibt die Hintereinanderschaltung von zwei Prozessen. Terminiert der erste Prozeß B_1 erfolgreich, d.h. begibt B_1 sich nicht in einen Verklemmungszustand, so wird der zweite Prozeß B_2 aktiviert.

- *disabling*: $B_1 \,[>\, B_2$

  $B_1 \,[>\, B_2$ definiert einen Prozeß, bei dem der Prozeß B_2 zu jedem Zeitpunkt die Ausführung von B_1 unterbrechen kann. Diese Unterbrechung kann sogar vor der Ausführung von B_1 stattfinden. Wenn allerdings B_1 erfolgreich terminiert ist, kann B_2 nicht mehr ausgeführt werden.

- *hiding*: **hide** $g_1, \ldots, g_n$ **in** B

  Der **hiding**-Operator verdeckt Aktionen $g_1, \ldots, g_n$ in B. Diese Aktionen werden in die interne Aktion **i** umbenannt. Diese umbenannten Aktionen sind somit nach außen unsichtbar.

- *renaming*: $B[b_1/a_1, \ldots, b_n/a_n]$

  Jede Aktion b_i in B mit $i = 1, \ldots, n$ wird in die Aktion a_i umbenannt.

- *parallel composition*: $B_1 \,|[g_1, \ldots, g_n]|\, B_2$

  Hierbei handelt es sich um eine Parallelschaltung von zwei Prozessen B_1 und B_2. D.h. B_1 und B_2 können ihre Aktionen bis auf $g_1, \ldots, g_n$ parallel unabhängig voneinander ausführen. Bezüglich $g_1, \ldots, g_n$ müssen sie sich jedoch synchronisieren.

- *process instantion*: P

  P als Prozeßname steht für die Definition eines Prozesses, d.h. $P := B$ mit $B \in \mathcal{BL}$. Damit ist die Möglichkeit gegeben, einen rekursiven Prozeß zu definieren, z.B.: P := g; P.

Bevor wir zu weiteren Abschnitten und Kapiteln übergehen, werden an dieser Stelle eine kürzere Schreibweise für $|[g_1, \ldots, g_n]|$ eingeführt sowie eine Einschränkung auf $\mathcal{BL}$ getroffen. Diese Einschränkung beruht darauf, daß Langerak eine Halbordnungssemantik für $\mathcal{BL}$ definiert hat, bei der er voraussetzt, daß in B mit $P := B$ keine Prozeßvariablen außer P vorkommen. Wir halten dies daher in der folgenden Bemerkung fest.

Bemerkung 2.1

- Für $B_1 \,|[]|\, B_2$ bzw. $B_1 \,|[g_1, \ldots, g_n]|\, B_2$ mit $\{g_1, \ldots, g_n\} = \mathcal{G}$ schreiben wir auch kurz $B_1 \,|||\, B_2$ bzw. $B_1 \,||\, B_2$.

- Sei $B \in \mathcal{BL}$ und $P := B$, dann dürfen in B keine Prozeßvariablen außer P vorkommen. Z.B. $P := a; P \,[]\, b; Q$ ist nicht erlaubt. □

2.2 Operationelle Semantik

Mit operationeller Semantik wird das beobachtbare Verhalten eines Systems beschrieben, wobei unter „beobachtbares Verhalten" die zeitliche Reihenfolge des Eintretens von beobachtbaren Aktionen zu verstehen ist. Dabei werden die Aktionen als instantan betrachtet, d.h. die Ausführungszeit einer Aktion a, also die Zeit zwischen dem Zeitpunkt, zu dem a ausführbar ist, und dem Zeitpunkt, zu dem a ausgeführt wird, ist Null [BB87].

Die Beschreibung eines beobachtbaren Verhaltens erfolgt in Form eines beschrifteten Transitionssytems.

Definition 2.2 (Transitionssystem) *Sei L eine Menge. Dann heißt*

$$T = (Q, \Leftrightarrow\!\!\rightarrow, q_0)$$

ein (beschriftetes) Transitionssystem, wenn gilt:

- *Q ist eine Menge (von Zuständen).*
- *$\Leftrightarrow\!\!\rightarrow \subseteq Q \times L \times Q$ (Übergangsrelation)*
- *$q_0 \in Q$ (Anfangszustand)* □

Bemerkung 2.2 *Für $(p, e, q) \in \Leftrightarrow\!\!\rightarrow$ wird auch kurz $p \overset{e}{\Leftrightarrow\!\!\rightarrow} q$ geschrieben.* □

Definition 2.3 (operationelle Semantik) *Sei $B \in \mathcal{BL}$, $Act := \mathcal{G} \cup \{\mathbf{i}, \delta\}$ mit $\delta \notin \mathcal{G} \cup \{\mathbf{i}\}$. Dann ist die operationelle Semantik von B ein Transitionssystem*

$$\mathcal{OS}(B) := (\mathcal{BL}, \Leftrightarrow\!\!\rightarrow, B),$$

wobei $\Leftrightarrow\!\!\rightarrow \subseteq \mathcal{BL} \times Act \times \mathcal{BL}$ die kleinste Relation ist, die den folgenden Regeln genügt:

1. $\mathbf{exit} \overset{\delta}{\Leftrightarrow\!\!\rightarrow} \mathbf{stop}$
2. $g; B \overset{g}{\Leftrightarrow\!\!\rightarrow} B$
3. $\dfrac{B_1 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 \,[]\, B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1'}$
4. $\dfrac{B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_2'}{B_1 \,[]\, B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_2'}$
5. $\dfrac{B_1 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \neq \delta}{B_1 \gg B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1' \gg B_2}$
6. $\dfrac{B_1 \overset{\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 \gg B_2 \overset{\mathbf{i}}{\Leftrightarrow\!\!\rightarrow} B_2}$
7. $\dfrac{B_1 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \neq \delta}{B_1 \,[>\, B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1' \,[>\, B_2}$
8. $\dfrac{B_1 \overset{\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 \,[>\, B_2 \overset{\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}$
9. $\dfrac{B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_2'}{B_1 \,[>\, B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_2'}$
10. $\dfrac{B \overset{a}{\Leftrightarrow\!\!\rightarrow} B' \wedge a \in \{g_1, \ldots, g_n\}}{\mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B \overset{\mathbf{i}}{\Leftrightarrow\!\!\rightarrow} \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B'}$
11. $\dfrac{B \overset{a}{\Leftrightarrow\!\!\rightarrow} B' \wedge a \notin \{g_1, \ldots, g_n\}}{\mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B \overset{a}{\Leftrightarrow\!\!\rightarrow} \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B'}$
12. $\dfrac{B \overset{a}{\Leftrightarrow\!\!\rightarrow} B' \wedge \exists i \in \{1, \ldots, n\} : a = a_i}{B[b_1/a_1, \ldots, b_n/a_n] \overset{b_i}{\Leftrightarrow\!\!\rightarrow} B'[b_1/a_1, \ldots, b_n/a_n]}$
13. $\dfrac{B \overset{a}{\Leftrightarrow\!\!\rightarrow} B' \wedge a \notin \{a_1, \ldots, a_n\}}{B[b_1/a_1, \ldots, b_n/a_n] \overset{a}{\Leftrightarrow\!\!\rightarrow} B'[b_1/a_1, \ldots, b_n/a_n]}$
14. $\dfrac{B_1 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \notin \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1' \,|[g_1, \ldots, g_n]|\, B_2}$
15. $\dfrac{B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_2' \wedge a \notin \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1 \,|[g_1, \ldots, g_n]|\, B_2'}$
16. $\dfrac{B_1 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_2' \wedge a \in \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} B_1' \,|[g_1, \ldots, g_n]|\, B_2'}$
17. $\dfrac{P := B \wedge B \overset{a}{\Leftrightarrow\!\!\rightarrow} B'}{P \overset{a}{\Leftrightarrow\!\!\rightarrow} B'}$

□

Dadurch, daß jedem Term $B \in \mathcal{BL}$ ein Transitionssystem zugeordnet wird, ist es möglich, einen Äquivalenzbegriff auf Prozessen einzuführen, der unserer Intuition entspricht. Dieser beruht nämlich auf der Äquivalenz von Transitionssystemen und besagt, daß zwei Prozesse nur dann äquivalent sind (im folgenden als bisimulationsäquivalent bezeichnet), wenn sie das gleiche beobachtbare Verhalten aufweisen.

Definition 2.4 (Äquivalenz auf Transitionssystemen)

Sei $T_i = (Q_i, \rightarrow, q_i)$ mit $i = 1, 2$ ein Transitionssystem. T_1 und T_2 sind bisimulationsäquivalent ($T_1 \sim_s T_2$), wenn es eine Relation $R \subseteq Q_1 \times Q_2$ mit $(q_1, q_2) \in R$ gibt und für alle $(p, q) \in R$ gilt:

1. *Gilt $p \stackrel{a}{\Leftrightarrow} p'$, dann $\exists q' \in Q_2 : q \stackrel{a}{\Leftrightarrow} q'$ und $(p', q') \in R$.*
2. *Gilt $q \stackrel{a}{\Leftrightarrow} q'$, dann $\exists p' \in Q_2 : p \stackrel{a}{\Leftrightarrow} p'$ und $(p', q') \in R$.*

Eine Relation R, die die Bedingung 1. und 2. erfüllt, heißt eine Bisimulation. □

Definition 2.5 (Bisimulationsäquivalenz)

Sei $B_i \in \mathcal{BL}$ mit $i = 1, 2$. B_1 und B_2 sind bisimulationsäquivalent ($B_1 \sim B_2$), wenn $\mathcal{OS}(B_1) \sim_s \mathcal{OS}(B_2)$ gilt. □

Beispiel 2.1

1. $a; b; \mathbf{stop} + b; a; \mathbf{stop} \sim a; \mathbf{stop} \,|||\, b; \mathbf{stop}$
2. $a; (b; c; \mathbf{stop} + b; d; \mathbf{stop}) \not\sim a; b; c; \mathbf{stop} + a; b; d; \mathbf{stop}$
3. $a; \mathbf{stop} \sim P$ *mit* $P := a \,[]\, P$ □

Lemma 2.1 *Sei $B, C \in \mathcal{BL}$ mit $B \sim C$. Gilt $B \stackrel{a}{\Leftrightarrow} B'$ mit $a \in Act$, dann gilt $\exists C' : C \stackrel{a}{\Leftrightarrow} C'$ und $B' \sim C'$.*

Beweis: Der Beweis ist einfach und wird deshalb nicht ausgeführt. □

Satz 2.1 *$\sim$ ist auf $\mathcal{BL}$ eine Kongruenzrelation, d.h. sei $B, B', C, C' \in \mathcal{BL}$ mit $B \sim C$ und $B' \sim C'$ sowie $op \in \{[], \gg, [>, |[g_1, \ldots, g_n]|\}$, dann gilt*

- $B \; op \; B' \sim C \; op \; C'$
- **hide** $g_1, \ldots, g_n$ **in** $B \sim$ **hide** $g_1, \ldots, g_n$ **in** C
- $B[b_1/a_1, \ldots, b_n/a_n] \sim C[b_1/a_1, \ldots, b_n/a_n]$

Beweis: Wir zeigen nur für den Fall $op = \gg$. Die Beweise für die restlichen Fälle verlaufen analog. Sei

$$R = \{(B1 \gg B2, C1 \gg C2) \,|\, B1 \sim C1 \wedge B2 \sim C2\} \cup \{(B, C) \,|\, B \sim C\},$$

dann ist R eine Bisimulation, denn es gilt für $(B1 \gg B2, C1 \gg C2) \in R$:

1. Gilt $B1 \gg B2 \stackrel{a}{\Leftrightarrow} B1' \gg B2$ mit $a \neq \delta$, dann gilt nach Lemma 2.1 $\exists C1' : C1 \gg C2 \stackrel{a}{\Leftrightarrow} C1' \gg C2$ und $B1' \sim C1'$. Somit gilt $(B1' \gg B2, C1' \gg C2) \in R$.

   Gilt $B1 \gg B2 \stackrel{\mathbf{i}}{\Leftrightarrow} B2$, so gilt $B1 \stackrel{\delta}{\Leftrightarrow} B1'$ und nach Lemma 2.1 $\exists C1' : C1 \stackrel{\delta}{\Leftrightarrow} C1'$, was $C1 \gg C2 \stackrel{\mathbf{i}}{\Leftrightarrow} C2$ und $(B2, C2) \in R$ bedeutet.

2. Symmetrisch zu 1.

Für $(B, C) \in R$ mit $B \sim C$ brauchen wir nicht zu betrachten, da die Bedingung a) und b) aus Lemma 2.1 folgt. Da $(B \gg B', C \gg C') \in R$ gilt, gilt $B \gg B' \sim C \gg C'$. □

Die Kongruenzeigenschaft von $\sim$ besagt anschaulich, daß jeder Teilprozeß eines komplexen Prozesses B durch einen äquivalenten Teilprozeß ersetzt werden kann, ohne daß sich dabei das Verhalten von B ändert. $\sim$ kann daher wie eine Gleichheitsrelation betrachtet werden. Diese Eigenschaft wird, wie wir später noch sehen werden, für einen wichtigen Beweis verwendet.

Aus obiger Definition 2.3 ist die Zustandsmenge von $\mathcal{OS}(B)$ die Sprache $\mathcal{BL}$. In dieser Menge gibt es Zustände, die von B nicht erreichbar sind. Da aber die nicht erreichbaren Zustände das gesamte Verhalten eines Systems nicht beeinflussen, genügt es, sich nur auf die erreichbaren Zustände einzuschränken. Daß dies tatsächlich gilt, zeigt der nachfolgende Satz. $Er(B)$ in der folgenden Definition gibt anschaulich die Menge der erreichbaren Zustände von B wieder.

Definition 2.6 ($Er(B)$, $Er(\mathcal{OS}(B))$) *Sei $B \in \mathcal{BL}$. Die Menge $Er(B)$ ist die kleinste Menge, für die folgendes gilt:*

- $B \in Er(B)$
- *Gilt $B \in Er(B)$ und $B \overset{a}{\Leftrightarrow} C$ mit $a \in Act$, dann gilt $C \in Er(B)$.*

Sei $\mathcal{OS}(B) = (\mathcal{BL}, \Leftrightarrow, B)$, dann ist $Er(\mathcal{OS}(B)) := (Er(B), \Leftrightarrow', B)$, wobei

$$\Leftrightarrow' = \Leftrightarrow \cap (Er(B) \times Er(B))$$

□

Satz 2.2 *Sei $B \in \mathcal{BL}$, dann gilt $\mathcal{OS}(B) \sim_s Er(\mathcal{OS}(B))$.*

Beweis: Der Beweis ist einfach und wird deshalb nicht ausgeführt. □

Am Ende dieses Abschnittes führen wir noch einen weiteren Begriff ein, der später oft benötigt wird.

Definition 2.7 ($Act(B)$) *Sei $B \in \mathcal{BL}$. $Act(B)$ ist induktiv wie folgt definiert:*

- $Act(\mathbf{stop}) := \emptyset$
- $Act(\mathbf{exit}) := \emptyset$
- $Act(g; B) := \begin{cases} Act(B) & falls \quad g = \mathbf{i} \\ \{g\} \cup Act(B) & falls \quad sonst \end{cases}$
- $Act(B_1 \,[]\, B_2) := Act(B_1) \cup Act(B_2)$
- $Act(B_1 \gg B_2) := Act(B_1) \cup Act(B_2)$
- $Act(B_1 \,[>\, B_2) := Act(B_1) \cup Act(B_2)$
- $Act(\mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B) := Act(B) \setminus \{g_1, \ldots, g_n\}$

- $Act(B[b_1/a_1, \ldots, b_n/a_n]) := \{h(g) \mid g \in Act(B)\}$, *wobei* $h(g) := \begin{cases} b_i & \textit{falls} \quad g = a_i \\ g & \textit{falls} \quad \textit{sonst} \end{cases}$
- $Act(B_1 \,|[g_1, \ldots, g_n]|\, B_2) := Act(B_1) \cup Act(B_2)$
- $Act(P) := Act(B)$, *falls* $P := B$. □

$Act(B)$ gibt also nichts anderes als die Menge aller beobachtbaren Aktionen von B an. Auch Aktionen, die nicht ausführbar sind, sind in $Act(B)$ enthalten. Z.B. sind a und b in $Act(B)$ mit $B = a; \mathbf{stop} \,|[a,b]|\, b; \mathbf{stop}$, aber sie sind nicht ausführbar, da es keine Übergänge mit a und b gibt.

2.3 Halbordnungssemantik

Nach [Lan92] und [Rei87] berücksichtigt die operationelle Semantik folgende Aspekte:

- Mit operationeller Semantik wird nur das beobachtbare Verhalten eines Systems beschrieben, wobei dessen konkrete Struktur völlig abstrahiert wird. Es wird nicht unterschieden, ob es sich um ein sequentielles oder ein verteiltes System handelt. Lediglich auf das beobachtbare Systemverhalten kommt es an. Kausale Unabhängigkeit von Aktionen wird daher nicht richtig und soll nicht erfaßt werden. Die kausale Ordnungsrelation, die angibt, daß eine Aktion a notwendigerweise vor einer Aktion b stattfinden muß, damit b ausgeführt werden kann, wird in der operationellen Semantik zu einer totalen Halbordnung erweitert. Dies bedeutet, daß jede Aktion mindestens in einer Ordnung mit einer anderen Aktion stehen muß, auch wenn die Aktionen kausal unabhängig voneinander sind. Diese totale Halbordnung gibt aber genau die temporale Ordnung wieder, nach der die Ereignisse zeitlich auftreten [Rei87]. Das Beispiel

  $$a; \mathit{stop} \,|||\, b; \mathit{stop} \sim a; b; \mathit{stop} \,[]\, b; a; \mathit{stop}$$

  zeigt, daß wegen der Äquivalenz der beiden Prozesse deren Struktur nicht beachtet wird. Auf der linken Seite haben wir einen Prozeß mit einem Parallelkonstrukt, dagegen auf der rechten Seite einen Prozeß mit einer alternativen Auswahl. In der Tat kann man das Verhalten der beiden Prozesse voneinander nicht unterscheiden, wenn man sie als Black-Boxes betrachtet.

- Aufgrund der konkreten Strukturabstraktion eines Systems wird nur ein globaler Zustand definiert. Man will nämlich ein (verteiltes) System als ein Ganzes und nicht als ein System mit verteilten Resourcen betrachten.

- Aus den obengenannten Gesichtspunkten ist die operationelle Semantik in der Anfangsphase des Spezifikationsentwurfs ein sehr geeignetes, semantisches Model [Lan92]. Sie gibt nämlich die Eigenschaften wieder, die man in dieser Phase tatsächlich erwartet. Diese sind wie oben erläutert die Abstraktion einer konkreten Systemstruktur und die der Ausführungszeit einer Aktion. Diese Eigenschaften sind aber für das Einhalten des Hierarchievorgehens in der Entwurfsphase eine notwendige Voraussetzung. Jedes Hinzunehmen von weiteren Informationen erschwert nur den Verlauf des Entwurfs.

Da mit operationeller Semantik das beobachtbare Verhalten eines Systems beschrieben wird, kann die Theorie des Testens basierend auf dieser Semantik einfach entwickelt werden. Beim Testen werden nämlich bestimmte, beobachtbare Eigenschaften des Systems auf die Richtigkeit geprüft und nicht die Eigenschaften, die von der Struktur des Systems, d.h. der kausalen Abhängigkeit der Aktionen, abhängig sind [Bri88]. In den späteren Phasen des Spezifikationsentwurfs (nicht in der Realisierungsphase, z.B. mit Sprache C) bringt die operationelle Semantik jedoch einige Probleme mit sich. Dies hat folgende Gründe:

- Die Annahme, daß eine Aktion instantan ist, ist nicht mehr zulässig, da die Ausführung einer Aktion eine bestimmte Dauer hat. Vernachlässigt man nämlich die Dauer einer Aktion, so läßt sich das System nicht effizient implementieren. Die Leistungsbewertung eines Systems (z.B. Durchsatz, Wartezeit, usw.) kann nämlich vor der Realisierungsphase nicht durchgeführt werden (siehe z.B. [RB91], [GHR93], [HR94]).

- Für eine effziente Implementierung einer formalen Spezifikation darf auch die konkrete Struktur eines Systems nicht mehr vernachlässigt werden. Die Effizienz einer Implementierung hängt entscheidend davon ab, ob das System verteilt oder nicht verteilt (sequentiell) ist. Betrachten wir nochmals das obige Beipiel, wobei wir folgendes annehmen:

  1. Der Prozeß $B_1 = a; b; \mathbf{stop} \,[]\, b; a; \mathbf{stop}$ kann nur von einem einzigen Prozessor ausgeführt werden.
  2. Der Prozeß $B_2 = a; \mathbf{stop} \,|||\, b; \mathbf{stop}$ läuft auf einem Rechnersystem mit zwei Prozessoren P_1 und P_2 ab, wobei P_1 den Prozeß $a; \mathbf{stop}$ ausführt und P_2 den Prozeß $b; \mathbf{stop}$.

  so benötigen wir bei 1. $D(a) + D(b)$ Zeiteinheiten für die Ausführung von B_1, wenn $D(a)$ und $D(b)$ die Dauer der entsprechenden Aktion angeben. Bei 2. wird dagegen der Prozeß B_2 innerhalb von $Max(D(a), D(b))$ Zeiteinheiten abgeschlossen ausgeführt.

  Gibt es eine Semantik, die die Systemstrukturen unterscheidet, so besteht die Möglichkeit, zu untersuchen, welches System besser für eine effiziente Implementierung geeignet ist.

Darüber hinaus hat [Rei87] in seiner Arbeit gezeigt, daß viele Eigeschaften, die von der internen Struktur bedingt sind, nicht beobachtbar sind. Z.B. läßt sich der Nichtdeterminismus der alternativen Auswahl [] nicht beobachten.

Aus den obengenannten Gründen hat Langerak [Lan92] für Basic LOTOS eine Halbordnungssemantik definiert, die die Strukturen von Systemen unterscheidet, also die kausalen Abhängigkeiten der Aktionen berücksichtigt. Langerak hat als semantisches Model seine selbstdefinierte, sogenannte „bundle event structure“ verwendet. Diese ist eine abgewandelte Form von „primes event structure“, „flow event structure“ und „stable event structure“, die Winskel in [Win89] eingeführt hat.

Definition 2.8 (e.b.e.s) *Ein 4er-Tupel* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *mit*

- E *als eine Menge von Ereignissen*

- $\rightsquigarrow \subseteq E \times E$ *(Konfliktrelation)*
- $\mapsto \subseteq 2^E \times E$ *(Bündel-Menge)*
- $l : E \to Act$

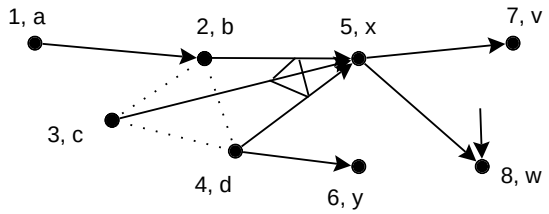
heißt eine extended bundle event structure (kurz e.b.e.s), wenn gilt:

1. $X \mapsto e \Longrightarrow \forall e_1, e_2 \in X : (e_1 \neq e_2 \Longrightarrow e_1 \rightsquigarrow e_2)$
2. $\rightsquigarrow$ *ist irreflexiv, d.h.* $\forall e, e' \in E : (e \rightsquigarrow e' \Longrightarrow e \neq e')$.

Die Menge aller e.b.e.s. bezeichnen wir als $\mathcal{ES}$ □

Graphisch wird eine e.b.e.s. wie folgt dargestellt: Jedes Ereignis $e \in E$ wird durch einen Punkt repräsentiert. In der Nähe jedes Punktes wird neben dem Ereignisnamen e auch noch $l(e)$ eingetragen. Jedes Bündel $X \mapsto e$ wird durch die gerichteten Kanten zwischen Elementen $e' \in X$ und e wiedergegeben, die mit Linien verbunden werden. $e \rightsquigarrow e'$ wird durch einen punktierten Pfeil $e \cdots\cdots \triangleright e'$ dargestellt. Gilt sogar $e' \rightsquigarrow e$, so wird nur eine punktierte Linie hingezeichnet.

Beispiel 2.2 Aus [Lan92] übernommen.



In dieser Struktur sind die Zahlen die Ereignisse und die Buchstaben die Aktionen. Bündel sind z.B. $\{1\} \mapsto 2$, $\{2,3,4\} \mapsto 5$ und $\emptyset \mapsto 8$. □

Oft beschränken wir uns nur auf die Isomorphieklassen von e.b.e.s. Anschaulich sind zwei e.b.e.s. isomorph zueinander, wenn sie sich nur durch die Ereignisnamen unterscheiden.

Definition 2.9 (Isomorphismus, $\cong$) *Sei* $\mathcal{E}_i$ *e.b.e.s. mit* $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ *und* $i = 1, 2$. $h : E_1 \to E_2$ *heißt ein Isomorphismus, wenn gilt:*

- h *ist bijektiv.*
- $e \rightsquigarrow_1 e' \Longleftrightarrow h(e) \rightsquigarrow_2 h(e')$
- $X \mapsto_1 e \Longleftrightarrow h(X) \mapsto_2 h(e)$, *wobei* $X \neq \emptyset$, *und* $\emptyset \mapsto_1 e \Longleftrightarrow \emptyset \mapsto_2 h(e)$
- $l_1(e) = l_2(h(e))$

Wir schreiben $\mathcal{E}_1 \cong \mathcal{E}_2$, *wenn es einen Isomorphismus zwischen* E_1 *und* E_2 *gibt.* □

Definition 2.10 (Ausführungsfolge) *Sei* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *eine e.b.e.s. Eine endliche Folge von verschiedenen Ereignissen* $e_1, \ldots, e_n$ *mit* $e_1, \ldots, e_n \in E$ *heißt eine Ausführungsfolge* (engl.: *proving sequence*) *von* $\mathcal{E}$, *wenn folgendes gilt:*

1. $\forall e_i, e_j : e_i \rightsquigarrow e_j \Longrightarrow i < j$
2. $X \mapsto e_i \Longrightarrow \{e_1, \ldots, e_{i-1}\} \cap X \neq \emptyset$

Wir bezeichnen die Menge aller Ausführungsfolgen von $\mathcal{E}$ als $PS(\mathcal{E})$. □

Aus dem Beispiel 2.2 ist also $1, 2, 5, 7$ in dieser Reihenfolge eine Ausführungsfolge. Dagegen ist $1, 2, 5, 8$ in jeder beliebigen Permutation keine Ausführungsfolge, da $\emptyset \mapsto 8$ gilt.

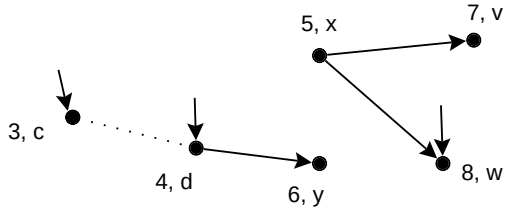
Definition 2.11 (Konfiguration) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine e.b.e.s. Eine Menge $K \subseteq E$ heißt eine Konfiguration von $\mathcal{E}$, wenn es eine Ausführungsfolge $e_1, \ldots, e_n$ gibt, so daß $K = \{e_1, , \ldots, e_n\}$ gilt.* □

Definition 2.12 ($\mathcal{E}[K]$) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine e.b.e.s. und K eine Konfiguration von $\mathcal{E}$. Dann ist $\mathcal{E}[K] := (E', \rightsquigarrow', \mapsto', l')$, wobei*

- $E' = E \setminus K$
- $\rightsquigarrow' = \rightsquigarrow \cap (E' \times E')$
- $\mapsto' = (\mapsto \setminus \{(X, e) \mid X \mapsto e \wedge X \cap K \neq \emptyset\}) \cup \{(\emptyset, e) \mid \exists e' \in K : e \rightsquigarrow e'\}$
- $l' := l|_{E'}$ □

Stellt man sich die Ereignisstruktur $\mathcal{E}$ als ein Zustand vor, so ist $\mathcal{E}[K]$ der Nachfolgezustand von $\mathcal{E}$, wenn K ausgeführt wird. Man beachte, daß $\mathcal{E}[K]$ eindeutig ist.

Beispiel 2.3 Sei $\mathcal{E}$ die Ereignisstruktur aus dem Beispiel 2.2, dann ist $\mathcal{E}[\{1, 2\}] =$



□

Definition 2.13 ($\prec_K$) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine e.b.e.s. und K eine Konfiguration von $\mathcal{E}$. Eine Relation $\prec_K \subseteq K \times K$ heißt eine Präzedenz-Relation, wenn gilt:*

$$e \prec_K e' \Leftrightarrow (\exists X \subseteq E : (e \in X \wedge X \mapsto e')) \vee e \rightsquigarrow e'$$

Die reflexive und transitive Hülle von $\prec_K$ wird als $\leq_K$ bezeichnet, d.h. $\leq_K := \prec_K^*$. □

Anschaulich gibt $\prec_K$ genau die kausale Abhängigkeit der Ereignisse in K wieder, nach der diese Ereignisse auftreten. Stehen z.B. zwei Ereignisse e und e' nicht in der Relation $\prec_K$, dann bedeutet dies, daß e und e' kausal unabhängig sind. e und e' können daher parallel ausgeführt werden. Stellt man sich also eine Konfiguration K als eine Berechnung vor, so ist die Ausführungsgeschwindigkeit dieser Berechnung abhängig davon, wie groß $\prec_K$ ist. Ist z.B. $\prec_K$ eine Identitätsrelation, also die kleinste Relation, die $\prec_K$ annehmen kann, so kann K in der kürzesten Zeit ausgeführt werden. Alle Elemente in K sind nämlich kausal unabhängig und können daher parallel ausgeführt werden.

Lemma 2.2 *$\leq_K$ ist eine Halbordnung auf K.*

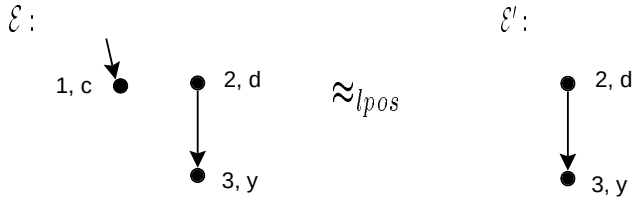
Beweis: siehe [Lan92]. □

Definition 2.14 (lposet) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine e.b.e.s. und K eine Konfiguration von $\mathcal{E}$. Die lposet (labelled partially ordered set) von $\mathcal{E}$ bezüglich K, bezeichnet als $\overline{K}$, ist definiert als $\overline{K} := (K, \leq_K, l|_K)$. $Lpos(\mathcal{E})$ bezeichnet die Menge aller lposets von $\mathcal{E}$.* □

Definition 2.15 ($\overline{K}$-Transition) *Sei $\mathcal{E}$ eine e.b.e.s und $\overline{K}$ eine lposet von $\mathcal{E}$. $\mathcal{E}$ hat eine $\overline{K}$-Transition, bezeichnet als $\mathcal{E} \stackrel{\overline{K}}{\Leftrightarrow} \mathcal{E}'$, wenn $\mathcal{E}' = \mathcal{E}[K]$ gilt.* □

Definition 2.16 (lposet-Äquivalenz) *Zwei e.b.e.s. $\mathcal{E}$ und $\mathcal{E}'$ sind lposet-äquivalent ($\mathcal{E} \approx_{lpos} \mathcal{E}'$), wenn $Lpos(\mathcal{E}) = Lpos(\mathcal{E}')$ gilt.* □

Der Leser würde vielleicht an dieser Stelle die Frage stellen, ob zwei e.b.e.s. $\mathcal{E}$ und $\mathcal{E}'$ identisch sind, wenn sie lposet-äquivalent sind. Daß dies nicht gilt, zeigt das folgende Beispiel:



Theorem 2.1 *Sei $\mathcal{E}$ und $\mathcal{E}'$ e.b.e.s. Dann gilt $\mathcal{E} \approx_{lpos} \mathcal{E}'$ genau dann, wenn $PS(\mathcal{E}) = PS(\mathcal{E})$ gilt.*

Beweis: [Lan92] □

Definition 2.17 (Präfix einer lposet) *Eine lposet $\overline{K}$ heißt Präfix einer lposet $\overline{H}$, wenn gilt:*

- $K \subseteq H$
- $\leq_K = \leq_H \cap (H \times K)$
- $l_K = l_H|_K$

□

Theorem 2.2 *Sei $\mathcal{E}$ eine e.b.e.s, K eine Konfiguration von $\mathcal{E}$, $H \subseteq E$ und $K \cap H = \emptyset$. H ist eine Konfiguration von $\mathcal{E}[K]$ genau dann, wenn $K \cup H$ eine Konfiguration von $\mathcal{E}$ und $\overline{K}$ Präfix von $\overline{K \cup H}$ gilt.*

Beweis: [Lan92] □

Nachdem die notwendigen Definitionen und Theoreme über e.b.e.s. zitiert wurden, wird nun die Halbordnungssemantik für $\mathcal{BL}$, die jedem Term $B \in \mathcal{BL}$ eine e.b.e.s. zuordnet, definiert. Die Definition erfolgt in folgenden Schritten: Zunächst werden entprechend den syntaktischen Operatoren auf $\mathcal{BL}$ die semantischen Operatoren auf $\mathcal{ES}$ definiert. Anschließend wird eine stetige Funktion $\mathcal{F}_B$ auf $\mathcal{ES}$ konstruiert, die die Semantik eines rekursiven Prozesses festlegt. Zum Schluß wird die Halbordnungssemantik von $\mathcal{BL}$ mit Hilfe der semantischen Operatoren und der Funktion $\mathcal{F}_B$ denotational definiert.

Definition 2.18 (Notationen) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine e.b.e.s. Dann*

- $init(\mathcal{E}) := \{e \in E \mid \neg(\exists X \subseteq E : X \mapsto e)\}$
- $exit(\mathcal{E}) := \{e \in E \mid l(e) = \delta\}$ □

Definition 2.19 (Operatoren auf e.b.e.s) *Sei $\mathcal{E}_1 = (E_1, \rightsquigarrow_1, \mapsto_1, l_1)$ und $\mathcal{E}_2 = (E_2, \rightsquigarrow_2, \mapsto_2, l_2)$ mit $E_1 \cap E_2 = \emptyset$. Sei weiterhin E_U eine universale Ereignismenge. Dann*

- $\overline{\mathbf{stop}} := (\emptyset, \emptyset, \emptyset, \emptyset)$
- $\overline{\mathbf{exit}}_e := (\{e\}, \emptyset, \emptyset, (e, \delta))$ *für* $e \in E_U$
- *Für* $g \in \mathcal{G} \cup \{\mathbf{i}\}$ *und* $e \in E_U$ *mit* $e \notin E_1$ *ist* $\overline{g_e;}\ \mathcal{E}_1 := (E, \rightsquigarrow_1, \mapsto, l)$, *wobei*
  1. $E = E_1 \cup \{e\}$
  2. $\mapsto = \mapsto_1 \cup (\{\{e\}\} \times init(\mathcal{E}_1))$
  3. $l = l_1 \cup \{(e, g)\}$
- $\mathcal{E}_1\ \overline{[]}\ \mathcal{E}_2 := (E, \rightsquigarrow, \mapsto, l)$, *wobei*
  1. $E = E_1 \cup E_2$
  2. $\rightsquigarrow = \rightsquigarrow_1 \cup \rightsquigarrow_2 \cup (init(\mathcal{E}_1) \times init(\mathcal{E}_2)) \cup (init(\mathcal{E}_2) \times init(\mathcal{E}_1))$
  3. $\mapsto = \mapsto_1 \cup \mapsto_2$
  4. $l = l_1 \cup l_2$
- $\mathcal{E}_1\ \overline{\gg}\ \mathcal{E}_2 := (E, \rightsquigarrow, \mapsto, l)$, *wobei*
  1. $E = E_1 \cup E_2$
  2. $\rightsquigarrow = \rightsquigarrow_1 \cup \rightsquigarrow_2 \cup \{(e, e') \mid e, e' \in exit(\mathcal{E}_1), e \neq e'\}$
  3. $\mapsto = \mapsto_1 \cup \mapsto_2 \cup (\{exit(\mathcal{E}_1)\} \times init(\mathcal{E}_2))$
  4. $l = ((l_1 \cup l_2) \setminus (exit(\mathcal{E}_1) \times \{\delta\})) \cup (exit(\mathcal{E}_1) \times \{\mathbf{i}\})$
- $\mathcal{E}_1\ \overline{[>}\ \mathcal{E}_2 := (E, \rightsquigarrow, \mapsto, l)$, *wobei*
  1. $E = E_1 \cup E_2$
  2. $\rightsquigarrow = \rightsquigarrow_1 \cup \rightsquigarrow_2 \cup (E_1 \times init(\mathcal{E}_2)) \cup (init(\mathcal{E}_2) \times exit(\mathcal{E}_1))$
  3. $\mapsto = \mapsto_1 \cup \mapsto_2$
  4. $l = l_1 \cup l_2$
- *Für* $g_i \in \mathcal{G}$ *mit* $i \in \{1, \ldots, n\}$ *ist* $\overline{\mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}}\ \mathcal{E}_1 := (E_1, \rightsquigarrow_1, \mapsto_1, l)$, *wobei*

$$l(e) = \begin{cases} \mathbf{i} & \textit{falls} \quad l_1(e) = g_i \textit{ mit } i \in \{1, \ldots, n\} \\ l_1(e) & \textit{falls} \quad \textit{sonst} \end{cases}$$

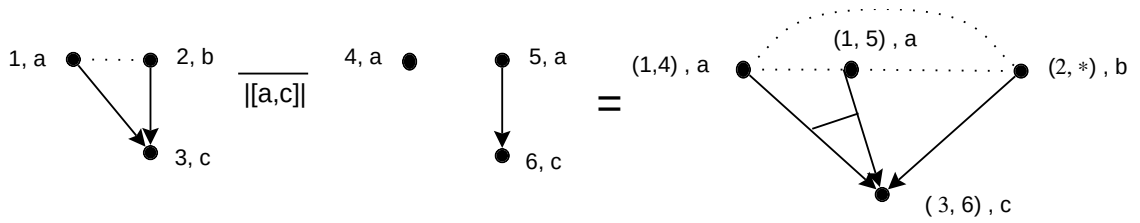
- *Für* $b_i, a_i \in \mathcal{G}$ *mit* $i \in \{1, \ldots, n\}$ *ist* $\mathcal{E}_1\overline{[b_1/a_1, \ldots, b_n/a_n]} := (E_1, \rightsquigarrow_1, \mapsto_1, l)$, *wobei*

$$l(e) = \begin{cases} b_i & \textit{falls} \quad l_1(e) = a_i \textit{ mit } i \in \{1, \ldots, n\} \\ l_1(e) & \textit{falls} \quad \textit{sonst} \end{cases}$$

- $\mathcal{E}_1 \overline{|[g_1,\ldots,g_n]|} \mathcal{E}_2 := (E, \rightsquigarrow, \mapsto, l)$, *wobei*

  1. $E = (E_1^f \times \{*\}) \cup (\{*\} \times E_2^f) \cup \{(e_1,e_2) \in E_1^s \times E_2^s \,|\, l(e_1) = l(e_2)\}$
     mit $E_i^s = \{e \in E_i \,|\, l(e) \in \{g_1,\ldots,g_n,\delta\}\}$ *für* $i = 1,2$ *und* $E_i^f = E_i \setminus E_i^s$ *für* $i = 1,2$.
  2. $(e_1,e_2) \rightsquigarrow (e_1',e_2')$ *genau dann, wenn*
     $(e_1 \rightsquigarrow_1 e_1') \vee (e_2 \rightsquigarrow_2 e_2') \vee (e_1 = e_1' \neq * \wedge e_2 \neq e_2') \vee (e_2 = e_2' \neq * \wedge e_1 \neq e_1')$
  3. $X \mapsto (e_1,e_2)$ *genau dann, wenn*
     $\exists X_1 \subseteq E_1 : (X_1 \mapsto_1 e_1 \wedge X = \{(e_i,e_j) \in E \,|\, e_i \in X_1\}) \vee$
     $\exists X_2 \subseteq E_2 : (X_2 \mapsto_2 e_2 \wedge X = \{(e_i,e_j) \in E \,|\, e_i \in X_2\})$
  4. $l(e_1,e_2) = \begin{cases} l_2(e_2) & \textit{falls} \quad e_1 = * \\ l_1(e_1) & \textit{falls} \quad \textit{sonst} \end{cases}$ □

Beispiel 2.4



□

Bis jetzt wird jedem syntaktischen Operator aus $\mathcal{BL}$ ein entsprechender, semantischer Operator zugeordnet. Damit kann eine Semantik für B denotational, d.h. entsprechend dem Termaufbau von B, definiert werden, wenn B keine Prozeßvariablen enthält. Die Semantik einer Prozeßdefinition $P := B'$ mit $B' \in \mathcal{BL}$ muß daher noch festgelegt werden.

Wie in Definition von denotationaler Semantik üblich ist, wird die Semantik von P mit Hilfe einer stetigen Funktion auf einer total halbgeordneteten Menge (cpo) definiert. Diesem Ansatz folgend hat Langerak eine stetige Funktion $\mathcal{F}_B$ auf $\mathcal{ES}$ konstruiert, die in Abhängigkeit von B festgelegt ist. B ist hier ein Prozeß, der zusätzlich die Information über die Ereignisnamen enthält.

Im Vergleich zu [Lan92] werden wir im folgenden für $\mathcal{F}_B$ an Stelle einer informellen Beschreibung eine formale Definition angeben. Dazu wird zunächst ein Prozeß $B \in \mathcal{BL}$ in einen Prozeß B' transformiert, der sich dadurch ergibt, daß jedes Vorkommen einer Aktion a, einer Prozeßvariable P und des Terms **exit** in B eindeutig mit einem Ereignisnamen markiert wird.

Definition 2.20 ($\mathcal{BL}^{ev}$) *Sei $\mathcal{A}$ eine abzählbar unendliche Menge von Ereignisnamen, wobei für $e \in \mathcal{A}$ gilt:*

- $e \neq (e')$, *wobei e' ein beliebiges Element ist.*
- $e \neq e'\sigma$, *wobei $e' \in \mathcal{A}$ und $\sigma \neq \epsilon$ (ϵ = leeres Wort) ein beliebiges Element ist.*

Sei ferner $g \in \mathcal{G} \cup \{\mathbf{i}\}$, $g_i, a_i, b_i \in \mathcal{G}$ mit $1 \leq i \leq n$, $e \in \mathcal{A}$ und $P \in \mathcal{P}$. Dann ist $\mathcal{BL}^{ev}$ durch folgende Grammatik

$$B ::= \mathbf{stop} \mid \mathbf{exit}_e \mid g_e; B \mid B \,[]\, B \mid B \gg B \mid B \,[>\, B \mid \mathbf{hide}\ g_1,\ldots,g_n\ \mathbf{in}\ B \mid B[b_1/a_1,\ldots,b_n/a_n] \mid B \,|[g_1,\ldots,g_n]|\, B \mid P_e$$

definiert. Dabei kommt jedes $e \in \mathcal{A}$ in B mit $B \in \mathcal{BL}^{ev}$ genau einmal vor, d.h. e ist in B eindeutig. □

Definition 2.21 ($e(\mathcal{E})$) *Sei Y eine Menge von Ereignissen, $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ und $e \in E_U$, dann*

- $eY := \{ee' \mid e' \in Y\}$
- $e(\mathcal{E}) := (eE, \rightsquigarrow', \mapsto', l')$, *wobei*
  1. $ee_1 \rightsquigarrow' ee_2 \iff e_1 \rightsquigarrow e_2$
  2. $eX \mapsto' ee' \iff X \mapsto e'$
  3. $l'(ee') = l(e)$

□

Definition 2.22 ($\mathcal{F}_B(\mathcal{E})$) *Sei $B \in \mathcal{BL}^{ev}$. Dann ist $\mathcal{F}_B : \mathcal{ES} \to \mathcal{ES}$ induktiv wie folgt definiert:*

- *Sei $B =$* **stop**, *dann ist $\mathcal{F}_B(\mathcal{E}) := \overline{\textbf{stop}}$.*
- *Sei $B =$* $\textbf{exit}_e$, *dann ist $\mathcal{F}_B(\mathcal{E}) := \overline{\textbf{exit}_e}$.*
- *Sei $B = g_e; C$, dann ist $\mathcal{F}_B(\mathcal{E}) := \overline{g_e;}\mathcal{F}_C(\mathcal{E})$.*
- *Sei $B = B1 \,[]\, B2$, dann ist $\mathcal{F}_B(\mathcal{E}) := \mathcal{F}_{B1}(\mathcal{E}) \;\overline{[]}\; \mathcal{F}_{B2}(\mathcal{E})$.*
- *Sei $B = B1 \gg B2$, dann ist $\mathcal{F}_B(\mathcal{E}) := \mathcal{F}_{B1}(\mathcal{E}) \;\overline{\gg}\; \mathcal{F}_{B2}(\mathcal{E})$.*
- *Sei $B = B1 \,[> B2$, dann ist $\mathcal{F}_B(\mathcal{E}) := \mathcal{F}_{B1}(\mathcal{E}) \;\overline{[>}\; \mathcal{F}_{B2}(\mathcal{E})$.*
- *Sei $B =$* **hide** $g_1, \ldots, g_n$ **in** B_1, *dann ist $\mathcal{F}_B(\mathcal{E}) := \overline{\textbf{hide } g_1, \ldots, g_n \textbf{ in}}\; \mathcal{F}_{B1}(\mathcal{E})$.*
- *Sei $B = B1[b_1/a_1, \ldots, b_n/a_n]$, dann ist $\mathcal{F}_B(\mathcal{E}) := \mathcal{F}_{B1}(\mathcal{E})\overline{[b_1/a_1, \ldots, b_n/a_n]}$.*
- *Sei $B = B1 \,|[g_1, \ldots, g_n]|\, B2$, dann ist $\mathcal{F}_B(\mathcal{E}) := \mathcal{F}_{B1}(\mathcal{E}) \;\overline{|[g_1, \ldots, g_n]|}\; \mathcal{F}_{B2}(\mathcal{E})$.*
- *Sei $B = P_e$, dann ist $\mathcal{F}_B(\mathcal{E}) := e(\mathcal{E})$.*

□

Man beachte, daß die obige Definition 2.22 eine korrekte Definition ist, obwohl wir nicht voraussgesetzt haben, daß e nicht in $\mathcal{F}_C(\mathcal{E})$ liegen darf, und daß die Ereignismengen von $\mathcal{F}_{B1}(\mathcal{E})$ und $\mathcal{F}_{B2}(\mathcal{E})$ disjunkt sein müssen. Diese Aussage bestätigt das folgende Lemma.

Lemma 2.3 *Sei $B \in \mathcal{BL}^{ev}$, $\mathcal{E}$ eine e.b.e.s., $\mathcal{F}_{B1}(\mathcal{E}) = (E_1, \rightsquigarrow_1, \mapsto_1, l_1)$ und $\mathcal{F}_{B2}(\mathcal{E}) = (E_2, \rightsquigarrow_2, \mapsto_2, l_2)$. Dann gilt*

1. $B = g_e; B1 \Longrightarrow e \notin E_1$
2. *Für $B = B1$ op $B2$ mit op $\in \{[], \gg, [>, |[g_1, \ldots, g_n]|\}$ gilt $E_1 \cap E_2 = \emptyset$.*

Beweis:

1. Wir zeigen zunächst mit Hilfe der strukturellen Induktion über den Termaufbau, daß gilt: Sei $B \in \mathcal{BL}^{ev}$ und $\mathcal{F}_B(\mathcal{E}) = (E, \rightsquigarrow, \mapsto, l)$. Gilt $e \in E$ und $e \in \mathcal{A}$, dann gibt es einen Teilterm B' in B mit $B' = a_e; B''$ oder $B' = \mathbf{exit}_e$.

   Für $B = P_e$ und $B = B_1 \,|[g_1, \ldots, g_n]|\, B_2$ brauchen wir nicht zu zeigen, da es kein $e \in E$ mit $e \in \mathcal{A}$ gibt. Der Rest gilt offenbar.

   Wäre nun $e \in E_1$, dann gäbe es einen Teilterm B_1' in B_1 mit $B_1' = a_e; B_1''$ oder $B_1' = \mathbf{exit}_e$. Daraus würde folgen, daß e nicht eindeutig ist. Widerspruch.

2. Angenommen sei $E_1 \cap E_2 \neq \emptyset$, dann gilt zunächst $\exists e : (e \in E_1 \wedge e \in E_2)$. Da e nur von folgender Form

   (a) $e \in \mathcal{A}$

   (b) $e = (\xi, \phi)$

   (c) $\exists e' \in \mathcal{A} : e = e'\sigma$ mit $\sigma \neq \epsilon$

   sein kann (Induktionsbeweis über den Termaufbau), ist $e = (\xi, \phi)$, denn: (a) Wäre $e \in \mathcal{A}$, so hätten wir den gleichen Widerspruch wie bei 1. (b) Wäre $e = e'\sigma$, so gäbe es eine Prozeßvariable P in $B1$ und eine Prozeßvariable Q in $B2$, die mit e markiert sind (Induktionsbeweis über den Termaufbau). Widerspruch.

   Ist $e = (\xi, \phi)$, so gibt es einen Teilterm $TB1 = T1 \,|[g_1, \ldots, g_n]|\, T2$ in $B1$ und einen Teilterm $TB2 = T3 \,|[g_1, \ldots, g_n]|\, T4$ in $B2$ (Induktionsbeweis über den Termaufbau), so daß gilt:

   - ξ liegt sowohl in $\mathcal{F}_{T1}$ als auch in $\mathcal{F}_{T3}$, falls $\xi \neq *$.
   - ϕ liegt sowohl in $\mathcal{F}_{T2}$ als auch in $\mathcal{F}_{T4}$, falls $\phi \neq *$.

   Da $(\xi \neq * \vee \phi \neq *)$ und $\xi, \phi \notin \mathcal{A}$ sowie $\xi \neq \xi'\sigma'$ (bzw. $\phi \neq \phi'\sigma''$) mit $\xi' \in \mathcal{A}$ (bzw. $\phi' \in \mathcal{A}$) und $\sigma' \neq \epsilon$ (bzw. $\sigma'' \neq \epsilon$) gilt (denn sonst hätten wir den gleichen Widerspruch wie oben), gilt $\xi = (\xi_1^1, \xi_1^2)$ oder $\phi = (\xi_1^1, \xi_1^2)$.

   Da ein Term endlich lang ist und daher die Anzahl der parallelen Kompositionen $|[g_1, \ldots, g_n]|$ in einem Term endlich ist, gibt es ein $n \in I\!N$, so daß gilt:

   - $\psi_1 = e = (\xi, \phi)$
   - $\psi_2 = (\psi_2^1, \psi_2^2) \Longrightarrow \xi = \psi_2$ oder $\phi = \psi_2$
   - $\psi_3 = (\psi_3^1, \psi_3^2) \Longrightarrow \psi_2^1 = \psi_3$ oder $\psi_2^2 = \psi_3$
   - $\ldots$
   - $\psi_n = (\psi_n^1, \psi_n^2) \Longrightarrow \psi_{n-1}^1 = \psi_n$ oder $\psi_{n-1}^2 = \psi_n$
   - $\psi_n^1 \neq (\ldots, \ldots)$ und $\psi_n^2 \neq (\ldots, \ldots)$

   Da $\psi_n^1, \psi_n^2 \notin \mathcal{A}$ und $\psi_n^1 \neq \psi' w$ sowie $\psi_n^2 \neq \psi'' w'$ mit $\psi', \psi'' \in \mathcal{A}$ und $w \neq \epsilon$, $w' \neq \epsilon$ gilt, so liegt ein Widerspruch vor. □

Langerak hat auf der Menge $\mathcal{ES}$ eine Halbordnung definiert und gezeigt, daß $\mathcal{F}_B$ bezüglich dieser Halbordnung eine stetige Funktion ist. Dieses Resultat wird in folgendem zitiert.

Definition 2.23 ($\trianglelefteq$) *Sei $\mathcal{E}_i$ mit $i = 1, 2$, dann ist $\mathcal{E}_1 \trianglelefteq \mathcal{E}_2$, wenn gilt:*

1. $E_1 \subseteq E_2$
2. $\rightsquigarrow_1 \subseteq \rightsquigarrow_2$ *und* $((e \rightsquigarrow_2 e' \wedge e, e' \in E_1) \Longrightarrow e \rightsquigarrow_1 e')$
3. $X \mapsto_1 e \Longrightarrow \exists Y \subseteq E_2 : (Y \mapsto_2 e \wedge X = Y \cap E_1)$ *und*
   $(X \mapsto_2 e \wedge e \in E_1) \Longrightarrow (X \cap E_1) \mapsto_1 e$
4. $l_1 = l_2|_{E_1}$ □

Theorem 2.3 $(\mathcal{ES}, \trianglelefteq, \perp)$ *ist eine c.p.o., wobei* $\perp := (\emptyset, \emptyset, \emptyset)$.

Beweis: siehe [Lan92] □

Definition 2.24 *Eine Funktion* $F : V \to V$ *ist stetig auf einer c.p.o.* $(V, \sqsubseteq, \perp_v)$, *wenn für jede aufsteigende* $\sqsubseteq$*-Kette* $< x_i \mid i \in I\!N_0 >$ *gilt:*

$$F(\bigsqcup_{i \in I\!N_0} x_i) = \bigsqcup_{i \in I\!N_0} F(x_i)$$

□

Theorem 2.4 *Sei* F *stetig auf einer c.p.o.* $(V, \sqsubseteq, \perp_v)$, *dann ist* $\bigsqcup_{i \in I\!N_0} F^i(\perp_v)$ *der kleinste Fixpunkt von* F.

Beweis: siehe [Mü87] □

Satz 2.3 *Sei* $B \in \mathcal{BL}^{ev}$, *wobei in* B *höchstens eine Prozeßvariable vorkommen darf, dann ist* $\mathcal{F}_B$ *stetig auf* $(\mathcal{ES}, \trianglelefteq, \perp)$.

Beweis: siehe [Lan92] □

Damit ist erreicht, daß $\mathcal{F}_B$ einen kleinsten Fixpunkt besitzt, welcher, wie im folgenden zu sehen wird, der Semantik einer Prozeßvariable P mit $P := B$ entspricht.

Definition 2.25 *Sei* $B \in \mathcal{BL}$ *und* $\mathcal{A}$ *die Menge aus Definition 2.20. Dann heißt* C *ein markierter Term von* B, *wenn gilt*

- $B = \mathbf{stop} \Longrightarrow C = \mathbf{stop}$
- $B = \mathbf{exit} \Longrightarrow C = \mathbf{exit}_e$ *mit* $e \in \mathcal{A}$
- *Ist* $B = g; B1$, *so ist* $C = g_e; C1$ *mit* $e \in \mathcal{A}$
- $B = B1 \,[]\, B2 \Longrightarrow C = C1 \,[]\, C2$
- $B = B1 \gg B2 \Longrightarrow C = C1 \gg C2$
- $B = B1 \,[>\, B2 \Longrightarrow C = C1 \,[>\, C2$
- $B = \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B1 \Longrightarrow C = \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ C1$
- $B = B1[b_1/a_1, \ldots, b_n/a_n] \Longrightarrow C = C1[b_1/a_1, \ldots, b_n/a_n]$
- $B = B1 \,|[g_1, \ldots, g_n]|\, B2 \Longrightarrow C = C1 \,|[g_1, \ldots, g_n]|\, C2$
- $B = P \Longrightarrow C = P_e \quad mit \quad e \in \mathcal{A}$

Dabei sind $C1$ und $C2$ markierte Terme von $B1$ und $B2$. Ist sogar jedes $e \in \mathcal{A}$ in C eindeutig, so heißt C ein eindeutig markierter Term von B. □

Definition 2.26 ($\mathcal{M}(B)$) *Sei $B \in \mathcal{BL}$, dann bezeichnet $\mathcal{M}(B)$ einen eindeutig markierten Term von B.* □

Ein eindeutig markierter Term $\mathcal{M}(B)$ ist also ein Element von $\mathcal{BL}^{ev}$. Damit sind wir in der Lage, die Halbordnungssemantik für $\mathcal{BL}$ aus [Lan92] wiederzugeben.

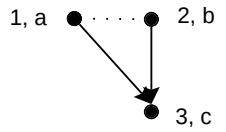
Definition 2.27 (Halbordnungssemantik) *Sei $B \in \mathcal{BL}$, $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ mit $i = 1, 2$, $E_1 \cap E_2 = \emptyset$ und $\mathcal{E}_i \in [\![B_i]\!]$. Dann ist $[\![.]\!] : \mathcal{BL} \to \mathcal{ES}_{/\cong}$ induktiv wie folgt definiert:*

- *Sei $B = \mathbf{stop}$, dann $[\![B]\!] := [\ \overline{\mathbf{stop}}\ ]_{/\cong}$.*
- *Sei $B = \mathbf{exit}$, dann $[\![B]\!] := [\overline{\mathbf{exit}}_e]_{/\cong}$ für $e \in E_U$.*
- *Sei $B = g; B_1$, dann $[\![B]\!] := [\overline{g_e;}\ \mathcal{E}_1]_{/\cong}$ mit $e \in E_U$ und $e \notin E_1$.*
- *Sei $B = B_1 \,[]\, B_2$, dann $[\![B]\!] := [\mathcal{E}_1\ \overline{[]}\ \mathcal{E}_2]_{/\cong}$.*
- *Sei $B = B_1 \gg B_2$, dann $[\![B]\!] := [\mathcal{E}_1\ \overline{\gg}\ \mathcal{E}_2]_{/\cong}$.*
- *Sei $B = B_1 \,[> B_2$, dann $[\![B]\!] := [\mathcal{E}_1\ \overline{[>}\ \mathcal{E}_2]_{/\cong}$.*
- *Sei $B = \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B_1$, dann $[\![B]\!] := [\overline{\mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}}\ \mathcal{E}_1]_{/\cong}$.*
- *Sei $B = B_1[b_1/a_1, \ldots, b_n/a_n]$, dann $[\![B]\!] := [\mathcal{E}_1\ \overline{[b_1/a_1, \ldots, b_n/a_n]}\ ]_{/\cong}$.*
- *Sei $B = B_1 \,|[g_1, \ldots, g_n]|\, B_2$, dann $[\![B]\!] := [\mathcal{E}_1 \overline{|[g_1, \ldots, g_n]|}\ \mathcal{E}_2]_{/\cong}$.*
- *Sei $B = P$ und $P := B'$, dann $[\![B]\!] := [\bigsqcup_i \mathcal{F}^i_{\mathcal{M}(B')}(\bot)]_{/\cong}$.* □

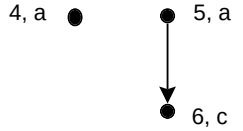
$[\![B]\!]$ ordnet also jedem Term $B \in \mathcal{BL}$ eine Isomorphieklasse zu. Es ist jedoch anzumerken, daß ein Repräsentant dieser Klasse gemeint ist, wenn über $[\![B]\!]$ gesprochen wird. Wir schreiben daher an Stelle $\mathcal{E} \in [\![B]\!]$ auch $[\![B]\!] = \mathcal{E}$ oder $\mathcal{E} = [\![B]\!]$.

Beispiel 2.5

- Sei $B1 = (a; \mathbf{stop} \,[]\, b; c; \mathbf{stop})\ |[a, c]|\ a; c; \mathbf{stop}$, dann $[\![B]\!] =$



- Sei $B2 = a; \mathbf{stop} \,|||\, a; c; \mathbf{stop}$, dann $[\![B1]\!] =$



- Sei $B = B1\ |[a, c]|\ B2$, dann hat $[\![B]\!]$ die Ereignisstruktur wie im Beispiel 2.4 angegeben. □

2.4 Konsistenz der Interleaving- und Halbordnungssemantik

In diesem Abschnitt wird das Konsistenzresultat von [Lan92] zitiert, das die Grundbasis für die noch kommenden Lemmata und Sätze darstellt. Langerak hat die Konsistenz der Semantiken dadurch gezeigt, daß er zunächst für $\mathcal{BL}$ eine operationelle Semantik definiert hat, die zusätzlich noch die Ereignisnamen enthält. Er zeigt dann, daß die Menge der Ereignisspuren, die sich aus dieser operationellen Semantik ergibt, die Menge der Ausführungsfolgen ist, die man aus der Halbordnungssemantik gewinnt. Dazu wird $\mathcal{BL}$ zu einer Sprache $\mathcal{BL}^{ev}_{er}$ erweitert, deren Sprachelemente mit Ereignisnamen gekennzeichnet sind. Diese Ereignisnamen müssen im Gegensatz zu $\mathcal{BL}^{ev}$ nicht unbedingt eindeutig sein.

Definition 2.28 ($\mathcal{BL}^{ev}_{er}$) *Sei $\mathcal{A}$, $\mathcal{G}$ und $\mathcal{P}$ die Menge aus Definition 2.20. Sei ferner $g \in \mathcal{G} \cup \{\mathbf{i}\}$, $g_i, a_i, b_i \in \mathcal{G}$ mit $1 \leq i \leq n$, $e \in \mathcal{A}$ und $P \in \mathcal{P}$. Dann ist $\mathcal{BL}^{ev}_{er}$ durch folgende Grammatik*

$$\begin{aligned} B \;::=\; & \mathbf{stop} \;\mid\; \mathbf{exit}_e \;\mid\; g_e; B \;\mid\; B \,[]\, B \;\mid\; B \gg B \;\mid\; B \,[>\, B \;\mid \\ & \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B \;\mid\; B[b_1/a_1, \ldots, b_n/a_n] \;\mid \\ & B \,|[g_1, \ldots, g_n]|\, B \;\mid\; P_e \;\mid\; e(B) \end{aligned}$$

definiert. □

Wie aus obiger Definition 2.28 zu erkennen ist, ist $\mathcal{BL}^{ev}$ eine Teilsprache von $\mathcal{BL}^{ev}_{er}$. Es wird nämlich nicht verlangt, daß $e \in \mathcal{A}$ in B mit $B \in \mathcal{BL}^{ev}_{er}$ eindeutig ist. Außerdem ist jeder markierter Term ein Sprachelelement aus der Sprache $\mathcal{BL}^{ev}_{er}$.

Man beachte, daß in [Lan92] $\mathcal{BL}^{ev}$ und $\mathcal{BL}^{ev}_{er}$ nicht explizit definiert werden. Wir führen hier eine formale Defintion dieser Sprachen deshalb ein, da diese für einen formalen Beweis in Abschnitt 3 benötigt wird.

Definition 2.29 (Ev) *Sei $\mathcal{A}$ die Menge aus Definition 2.20. Dann ist Ev die kleinste Menge, für die folgendes gilt:*

1. $\mathcal{A} \subseteq Ev$
2. $e \in Ev \Longrightarrow (e, *) \in Ev \wedge (*, e) \in Ev$
3. $e, e' \in Ev \Longrightarrow (e, e') \in Ev$
4. $e \in Ev \wedge e' \in \mathcal{A} \Longrightarrow e'e \in Ev$ □

Definition 2.30 (operationelle Semantik mit events) *Sei $B \in \mathcal{BL}^{ev}_{er}$. Dann ist die operationelle Semantik von B ein Transitionssystem $ETS(B) := (\mathcal{BL}^{ev}_{er}, \Leftrightarrow\!\!\rightarrow, B)$, wobei*

$$\Leftrightarrow\!\!\rightarrow \subseteq \mathcal{BL}^{ev}_{er} \times (Ev \times Act) \times \mathcal{BL}^{ev}_{er}$$

die kleinste Relation ist, die den folgenden Regeln genügt:

1. $\mathbf{exit}_e \overset{e,\delta}{\Leftrightarrow\!\!\rightarrow} \mathbf{stop}$
2. $g_e; B \overset{e,g}{\Leftrightarrow\!\!\rightarrow} B$

3. $$\frac{B_1 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 \,[]\, B_2 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_1'}$$

4. $$\frac{B_2 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_2'}{B_1 \,[]\, B_2 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_2'}$$

5. $$\frac{B_1 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \neq \delta}{B_1 \gg B_2 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_1' \gg B_2}$$

6. $$\frac{B_1 \overset{e,\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 \gg B_2 \overset{e,\mathbf{i}}{\Leftrightarrow\!\!\rightarrow} B_2}$$

7. $$\frac{B_1 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \neq \delta}{B_1 [> B_2 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_1' [> B_2}$$

8. $$\frac{B_1 \overset{e,\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 [> B_2 \overset{e,\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}$$

9. $$\frac{B_2 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_2'}{B_1 [> B_2 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_2'}$$

10. $$\frac{B \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B' \wedge a \in \{g_1, \ldots, g_n\}}{\textbf{hide } g_1, \ldots, g_n \textbf{ in } B \overset{e,\mathbf{i}}{\Leftrightarrow\!\!\rightarrow} \textbf{hide } g_1, \ldots, g_n \textbf{ in } B'}$$

11. $$\frac{B \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B' \wedge a \notin \{g_1, \ldots, g_n\}}{\textbf{hide } g_1, \ldots, g_n \textbf{ in } B \overset{e,a}{\Leftrightarrow\!\!\rightarrow} \textbf{hide } g_1, \ldots, g_n \textbf{ in } B'}$$

12. $$\frac{B \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B' \wedge \exists i \in \{1, \ldots, n\} : a = a_i}{B[b_1/a_1, \ldots, b_n/a_n] \overset{e,b_i}{\Leftrightarrow\!\!\rightarrow} B'[b_1/a_1, \ldots, b_n/a_n]}$$

13. $$\frac{B \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B' \wedge a \notin \{a_1, \ldots, a_n\}}{B[b_1/a_1, \ldots, b_n/a_n] \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B'[b_1/a_1, \ldots, b_n/a_n]}$$

14. $$\frac{B_1 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \notin \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{(e,*),a}{\Leftrightarrow\!\!\rightarrow} B_1' \,|[g_1, \ldots, g_n]|\, B_2}$$

15. $$\frac{B_2 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_2' \wedge a \notin \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{(*,e),a}{\Leftrightarrow\!\!\rightarrow} B_1 \,|[g_1, \ldots, g_n]|\, B_2'}$$

16. $$\frac{B_1 \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge B_2 \overset{e',a}{\Leftrightarrow\!\!\rightarrow} B_2' \wedge a \in \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{(e,e'),a}{\Leftrightarrow\!\!\rightarrow} B_1' \,|[g_1, \ldots, g_n]|\, B_2'}$$

17. $$\frac{P := B \wedge B \overset{e',a}{\Leftrightarrow\!\!\rightarrow} B'}{P_e \overset{ee',a}{\Leftrightarrow\!\!\rightarrow} e(B')}$$

18. $$\frac{B \overset{e',a}{\Leftrightarrow\!\!\rightarrow} B'}{e(B) \overset{ee',a}{\Leftrightarrow\!\!\rightarrow} e(B')}$$

□

Analog zum Lemma 2.3 gilt auch hier für $ETS(B)$ mit $B \in \mathcal{BL}^{ev}$ die Tatsache, daß jeder Ereignisname in $ETS(B)$ eindeutig ist. Wir wollen dies in dem nachfolgenden Lemma festhalten, wofür wir jedoch aufgrund der Analogie zum Lemma 2.3 keinen Beweis angeben.

Definition 2.31 ($Er(B)$, $Er(ETS(B))$) *Sei $B \in \mathcal{BL}^{ev}_{er}$. Die Menge $Er(B)$ ist die kleinste Menge, für die folgendes gilt:*

- $B \in Er(B)$

- *Gilt* $B \in Er(B)$ *und* $B \overset{e,a}{\Leftrightarrow\!\!\!\rightarrow} C$, *dann gilt* $C \in Er(B)$.

Sei $ETS(B) = (\mathcal{BL}, \Leftrightarrow\!\!\!\rightarrow, B)$, *dann ist* $Er(ETS(B)) := (Er(B), \Leftrightarrow\!\!\!\rightarrow', B)$, *wobei*

$$\Leftrightarrow\!\!\!\rightarrow' = \Leftrightarrow\!\!\!\rightarrow \cap (Er(B) \times Er(B))$$

□

Definition 2.32 *Sei* $B \in \mathcal{BL}^{ev}_{er}$ *und* $T = Er(ETS(B)) = (Er(B), \Leftrightarrow\!\!\!\rightarrow, B)$. *Dann ist* $Ev(T) := \{e \mid \exists B' \in Er(B) : B' \overset{e,a}{\Leftrightarrow\!\!\!\rightarrow} B''\}$.

Lemma 2.4 *Sei* $B \in \mathcal{BL}^{ev}$. *Dann gilt:*

- *Sei* $B = g_e; C$, *dann gilt* $e \notin Ev(T)$ *mit* $T = Er(ETS(C))$
- *Sei* $B = B1$ *op* $B2$ *mit op* $\in \{[], \gg, [>, |[g_1, \ldots, g_n]|\}$, *dann gilt* $Ev(T1) \cap Ev(T2) = \emptyset$ *mit* $T1 = Er(ETS(B1))$ *und* $T2 = Er(ETS(B2))$.

Beweis: Analog zum Beweis des Lemmas 2.3. □

Auch die Tatsache, daß $ETS(B)$ deterministisch ist, d.h. jeder Übergang hat einen eindeutigen Folgezustand, wird in dem nachfolgenden Lemma festgehalten.

Lemma 2.5 *Sei* $B \in \mathcal{BL}^{ev}$. *Gilt* $B \overset{e,a}{\Leftrightarrow\!\!\!\rightarrow} B'$ *und* $B \overset{e,a}{\Leftrightarrow\!\!\!\rightarrow} B''$, *so gilt* $B' = B''$.

Beweis: Induktionsbeweis über die Tiefe der Ableitung (Der Begriff „Ableitung“ ist im Sinne der Logik zu verstehen. Bei der operationellen Semantik befassen wir uns im Grunde genommen mit einem Regelssystem, bei dem die Regeln 1 und 2 als Axiome zu verstehen sind.). Der Beweis läßt sich mit Hilfe des Lemmas 2.4 einfach ausführen und wird dehalb nicht angegeben. □

Definition 2.33 *Sei* $B \in \mathcal{BL}^{ev}$, *dann heißt* $\sigma = e_1 \ldots e_n$ *mit*

$$B \overset{e_1,a_1}{\Leftrightarrow\!\!\!\!\!\!\rightarrow} B_1 \ldots B_{n-1} \overset{e_n,a_n}{\Leftrightarrow\!\!\!\!\!\!\rightarrow} B_n$$

eine Ereignisspur von B. *Die Menge aller Ereignisspuren von* B *bezeichnen wir als* $ETr(B)$. □

Wenn wir nun eine Halbordnungsemantik für einen Prozeß $B \in \mathcal{BL}^{ev}$ festlegen, bei der die Ereignisnamen genau die sind, die in B vorkommen, dann gilt $ETr(B) = PS([\![B]\!])$. Das ist das Konsistenzresultat, das Langerak in seiner Arbeit erzielt hat.

Definition 2.34 (Halbordnungssemantik für $\mathcal{BL}^{ev}$) *Sei* $B \in \mathcal{BL}^{ev}$. *Dann ist* $[\![.]\!] : \mathcal{BL}^{ev} \to \mathcal{ES}$ *induktiv wie folgt definiert:*

- *Sei* $B = \mathbf{stop}$, *dann* $[\![B]\!] := \overline{\mathbf{stop}}$.
- *Sei* $B = \mathbf{exit}_e$, *dann* $[\![B]\!] := \overline{\mathbf{exit}}_e$.
- *Sei* $B = g_e; B_1$, *dann* $[\![B]\!] := \overline{g_e;}\ [\![B_1]\!]$.
- *Sei* $B = B_1 \,[]\, B_2$, *dann* $[\![B]\!] := [\![B_1]\!]\ \overline{[]}\ [\![B_2]\!]$.

- *Sei* $B = B_1 \gg B_2$, *dann* $[\![B]\!] := [\![B_1]\!] \overline{\gg} [\![B_2]\!]$.
- *Sei* $B = B_1 [> B_2$, *dann* $[\![B]\!] := [\![B_1]\!] \overline{[>} [\![B_2]\!]$.
- *Sei* $B = \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B'$, *dann* $[\![B]\!] := \overline{\mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}}\ [\![B']\!]$.
- *Sei* $B = B'[b_1/a_1, \ldots, b_n/a_n]$, *dann* $[\![B]\!] := [\![B']\!]\ \overline{[b_1/a_1, \ldots, b_n/a_n]}$.
- *Sei* $B = B_1 \,|[g_1, \ldots, g_n]|\, B_2$, *dann* $[\![B]\!] := [\![B_1]\!]\ \overline{|[g_1, \ldots, g_n]|}\ [\![B_2]\!]$.
- *Sei* $B = P_e$ *und* $P := B'$, *dann* $[\![B]\!] := e(\bigsqcup_i \mathcal{F}^i_{B'}(\bot))$. □

In der obigen Definition 2.34 haben wir nicht vorausgesetzt, daß $E_1 \cap E_2 = \emptyset$ bzw. $e \notin E_1$ mit $[\![B_i]\!] = (E_1, \rightsquigarrow_1, \mapsto_1, l_1)$ und $i = 1, 2$ gilt. Dies folgt direkt aus dem Lemma 2.3.

Es läßt sich leicht zeigen, daß $[\![\mathcal{M}(B)]\!] \in [\![B]\!]$ für $B \in \mathcal{BL}$ gilt, wenn wir $[\![\mathcal{M}(P)]\!] \in [\![P]\!]$ voraussetzen, wobei P eine Prozeßvariable ist. Diese Voraussetzung ist natürlich immer erfüllt, wenn die Funktion $\mathcal{F}_B$ bei beiden Semantiken auf einen gemeinsamen Prozeß B angewendet wird.

Lemma 2.6 *Sei* $B \in \mathcal{BL}$, *dann gilt* $[\![\mathcal{M}(B)]\!] \in [\![B]\!]$.

Beweis: Induktionsbeweis über den Termaufbau. Der Beweis ist einfach und wird deshalb nicht ausgeführt. □

Theorem 2.5 *Sei* $B, B1, B2 \in \mathcal{BL}^{ev}$ *mit* $[\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$, $PS([\![B1]\!]) = ETr(B1)$ *und* $PS([\![B2]\!]) = ETr(B2)$. *Dann gilt:*

1. *Sei* $B = \mathbf{stop}$, *dann* $PS([\![B]\!]) = ETr(B)$.
2. *Sei* $B = \mathbf{exit}_e$, *dann* $PS([\![B]\!]) = ETr(B)$.
3. *Sei* $B = g_e; B_1$, *dann* $PS([\![B]\!]) = ETr(B)$.
4. *Sei* $B = B_1 \,[]\, B_2$, *dann* $PS([\![B]\!]) = ETr(B)$.
5. *Sei* $B = B_1 \gg B_2$, *dann* $PS([\![B]\!]) = ETr(B)$.
6. *Sei* $B = B_1 [> B_2$, *dann* $PS([\![B]\!]) = ETr(B)$.
7. *Sei* $B = \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B1$, *dann* $PS([\![B]\!]) = ETr(B)$.
8. *Sei* $B = B1[b_1/a_1, \ldots, b_n/a_n]$, *dann* $PS([\![B]\!]) = ETr(B)$.
9. *Sei* $B = B_1 \,|[g_1, \ldots, g_n]|\, B_2$, *dann* $PS([\![B]\!]) = ETr(B)$.

Ferner gilt für B *aus 1., 2., . . . , und 9.:*

$$\forall e_1 \ldots e_n \in ETr(B) : B \overset{e_1, l(e_1)}{\leftrightarrows\!\!\!\rightarrow} B_1 \ldots B_{n-1} \overset{e_n, l(e_n)}{\leftrightarrows\!\!\!\rightarrow} B_n.$$

Beweis: siehe [Lan92] □

Theorem 2.6 *Sei $B \in \mathcal{BL}^{ev}$, $P := B$ und $[\![P]\!] = (E, \rightsquigarrow, \mapsto, l)$, dann gilt $PS([\![P]\!]) = ETr(P)$ und für $e_1 \ldots e_n \in ETr(P)$:*

$$P \overset{e_1, l(e_1)}{\Longleftrightarrow} B_1 \ldots B_{n-1} \overset{e_n, l(e_n)}{\Longleftrightarrow} B_n.$$

Dabei ist P als P_ϵ zu verstehen.

Beweis: siehe [Lan92] □

Korollar 2.1 *Sei $B \in \mathcal{BL}^{ev}$, $P := B$ und $[\![P_e]\!] = (E, \rightsquigarrow, \mapsto, l)$, dann gilt $PS([\![P_e]\!]) = ETr(P_e)$ und für $e_1 \ldots e_n \in ETr(P_e)$:*

$$P_e \overset{e_1, l(e_1)}{\Longleftrightarrow} B_1 \ldots B_{n-1} \overset{e_n, l(e_n)}{\Longleftrightarrow} B_n.$$

□

Korollar 2.2 *Sei $B \in \mathcal{BL}^{ev}$ und $[\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$, dann gilt $PS([\![B]\!]) = ETr(B)$ und für $e_1 \ldots e_n \in ETr(B)$:*

$$B \overset{e_1, l(e_1)}{\Longleftrightarrow} B_1 \ldots B_{n-1} \overset{e_n, l(e_n)}{\Longleftrightarrow} B_n.$$

□

Dieses Korollar stellt, wie später zu sehen wird, die Grundlage für die in diesem Bericht erzielten Resultate dar.

3 Parallelitätsbegriffe für Basic LOTOS

In diesem Abschnitt werden Parallelitätsbegriffe für Basic LOTOS eingeführt und miteinander verglichen. Im ersten Teil werden die notwendigen Lemmata und Sätze zitiert und bewiesen, auf denen sich die Definition der Parallelitätsbegriffe im zweiten Teil aufbaut. Im dritten Teil wird eine Teilsprache von $\mathcal{BL}$ ohne Rekursion betrachtet, für die der maximale Parallitätsgrad sich abschätzen läßt. Im vierten Teil wird ein Ausblick auf Parallelisierung auf dieser Teilsprache gegeben.

3.1 Grundlegende Sätze und Lemmata

Im Abschnitt 2 wurde eine operationelle Semantik für $\mathcal{BL}$ und $\mathcal{BL}^{ev}$ angegeben, die jedem Term aus dieser Sprache ein Transitionssystem zuordnet. Diese Transitionssysteme unterscheiden sich im wesentlichem dadurch, daß $ETS(B')$ für $B' \in \mathcal{BL}^{ev}$ keine Zyklen enthält, während $\mathcal{OS}(B)$ für $B \in \mathcal{BL}$ Zyklen enthalten kann.

Beispiel 3.1 Sei $P := a; P$ in $\mathcal{BL}$ und $\mathcal{M}(P) = P_3$, wobei $P := a_1; P_2$ in $\mathcal{BL}^{ev}$ sei, dann haben $\mathcal{OS}(P)$ und $ETS(\mathcal{M}(P))$ die Strukturen wie in Abbildung 1 dargestellt.

Entfernt man die Ereignisnamen an den Übergängen von $ETS(\mathcal{M}(B))$, so läßt sich zeigen, daß $ETS(\mathcal{M}(B)) \sim_s \mathcal{OS}(B)$ gilt.

Definition 3.1 *Sei $B \in \mathcal{BL}^{ev}_{er}$ und $ETS(B)) = (\mathcal{BL}^{ev}_{er}, \rightarrow, B)$, dann ist $l(B) := (\mathcal{BL}^{ev}_{er}, \rightarrow_l, q_0)$, wobei gilt:*

$$B_1 \overset{a}{\Longrightarrow}_l B_2 \iff B_1 \overset{\epsilon, a}{\Longrightarrow} B_2$$

□

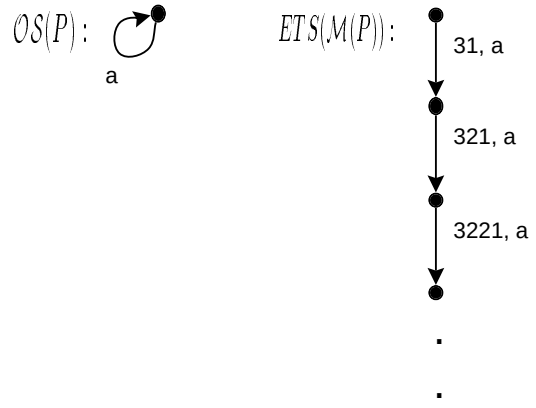


Abbildung 1: Unterschied zwischen $\mathcal{OS}(P)$ und $ETS(\mathcal{M}(P))$

Wie für $\mathcal{BL}$ führen wir für $\mathcal{BL}_{er}^{ev}$ einen identischen Bisimulationsäquivalenzbegriff ein.

Definition 3.2 (Bisimulationsäquivalenz)

Sei $B_i \in \mathcal{BL}_{er}^{ev}$ mit $i = 1,2$. B_1 und B_2 sind bisimulationsäquivalent ($B_1 \sim B_2$), wenn $l(B_1) \sim_s l(B_2)$ gilt. □

Satz 3.1 *$\sim$ ist auf $\mathcal{BL}_{er}^{ev}$ eine Kongruenzrelation.*

Beweis: Analog zum Beweis für $\sim$ auf $\mathcal{BL}$ (siehe Satz 2.1). □

Lemma 3.1 *Sei $\mathcal{A}$ die Menge aus Definition 2.20, $e \in \mathcal{A}$ und $B \in \mathcal{BL}_{er}^{ev}$, dann gilt $e(B) \sim B$.*

Beweis: Offenbar ist $R = \{(e(B), B) \mid B \in \mathcal{BL}_{er}^{ev}\}$ eine Bisimulation. □

Definition 3.3 (Bisimulation zu $\sim_s$) *Eine Relation $R \subseteq \mathcal{BL} \times \mathcal{BL}_{er}^{ev}$ heißt eine Bisimulation zu $\sim_s$, wenn für alle $(B, C) \in R$ gilt:*

1. *Gilt $B \overset{a}{\Leftrightarrow}_l B'$, dann gilt $\exists C' \in \mathcal{BL}_{er}^{ev} : C \overset{a}{\Leftrightarrow}_l C'$ und $B' \sim R \sim C'$.*
2. *Gilt $C \overset{a}{\Leftrightarrow}_l C'$, dann gilt $\exists B' \in \mathcal{BL}^{ev} : B \overset{a}{\Leftrightarrow}_l B'$ und $B' \sim R \sim C'$.*

Dabei bedeutet die Notation $B' \sim R \sim C'$, daß es ein B'' und C'' gibt, so daß $B' \sim B''$, $C'' \sim C'$ und $(B'', C'') \in R$ gilt. □

Lemma 3.2 *Sei R eine Bisimulation zu $\sim_s$, dann gilt:*

1. *$\sim R \sim$ ist eine Bisimulation.*
2. *$R \subseteq \sim R \sim$*

Beweis: siehe [Mil89]. □

Wir verwenden das Lemma 3.2, um den nachfolgenden, wichtigen Satz zu beweisen.

Satz 3.2 *Sei $B \in \mathcal{BL}$, dann gilt $\mathcal{OS}(B) \sim_s l(\mathcal{M}(B))$.*

Beweis: (Der folgende Beweis erfolgt analog zu einem Beweis, den Milner in seinem Buch [Mil89] vorgeführt hat.) Wir zeigen mit Hilfe der Induktion über die Tiefe der Ableitung (der Begriff „Ableitung“ ist im Sinne der Logik zu verstehen, denn wir haben in Definition 2.3 und 2.30 im Grunde genommen mit einem Regelsystem zu tun.), daß

$$R = \{(G, H) \mid \quad G \in \mathcal{BL} \text{ und } H \text{ ein markiertier Term von } G\}$$

eine Bisimulation zu $\sim_s$ ist. Ist R eine Bisimulation zu $\sim_s$, so gilt $\mathcal{OS}(B) \sim_s l(\mathcal{M}(B))$, da $(B, \mathcal{M}(B)) \in R$ gilt.

- $G = \mathbf{stop}$

  trivial.

- $G = \mathbf{exit}$

  Es gilt $G = \mathbf{exit} \overset{\delta}{\Leftrightarrow} \mathbf{stop} \Leftrightarrow H = \mathbf{exit}_e \overset{\delta}{\Leftrightarrow}_l \mathbf{stop}$ und $(\mathbf{stop}, \mathbf{stop}) \in R$.

- $G = a; G'$

  Dann gilt $H = a_e; H'$, wobei H' ein markierter Term von B ist. Wir haben dann

  $$\begin{aligned} & G = a; G' \overset{a}{\Leftrightarrow} G' \\ \Leftrightarrow\ & H = a_e; H' \overset{a}{\Leftrightarrow}_l H' \end{aligned}$$

  und $(G', H') \in R$.

- $G = G_1 \,[]\, G_2$

  Zunächst gilt $H = H_1 \,[]\, H_2$ mit H_i als ein markierter Term von G_i und $i = 1, 2$. Damit gilt dann

  $$\begin{aligned} & G = G_1 \,[]\, G_2 \overset{a}{\Leftrightarrow} C' \\ \Rightarrow\ & \exists i \in \{1, 2\} : G_i \overset{a}{\Leftrightarrow} C' \\ \Rightarrow\ & H_i \overset{a}{\Leftrightarrow}_l D'' \sim D' \text{ mit } (C', D') \in R \\ & \quad \text{(nach Induktionsannahme)} \\ \Rightarrow\ & H \overset{a}{\Leftrightarrow}_l D'' \sim D' \text{ mit } (C', D') \in R \end{aligned}$$

  Die Umkehrung gilt analog.

- $G = G_1 \gg G_2$

  Analog zum obigen Beweis gilt auch hier $H = H_1 \gg H_2$ mit H_i als ein markierter Term von G_i und $i = 1, 2$. Dann gilt

  1. $$\begin{aligned} & G \overset{a}{\Leftrightarrow} G_1' \gg G_2 \\ \Rightarrow\ & G_1 \overset{a}{\Leftrightarrow} G_1' \text{ mit } a \neq \delta \\ \Rightarrow\ & H_1 \overset{a}{\Leftrightarrow}_l D'' \sim D' \text{ mit } (G_1', D') \in R \text{ (nach Induktionsannahme)} \\ \Rightarrow\ & H \overset{a}{\Leftrightarrow}_l D'' \gg H_2 \end{aligned}$$

     Da nach Satz 3.1 $\sim$ eine Kongruenzrelation in $\mathcal{BL}^{ev}_{er}$ ist, gilt

     $$D'' \gg H_2 \sim H_1' \gg H_2,$$

wobei $H_1' = D'$ ein markierter Term von G_1' ist. Setzen wir also $G' = G_1' \gg G_2$ und $H' = H_1' \gg H_2$, so ist H' ein markierter Term von G' und daher $(G', H') \in R$.

Die Umkehrung gilt analog.

2.

$$\begin{aligned}
&G \overset{\mathbf{i}}{\Leftrightarrow\!\!\rightarrow} G_2 \\
&\Rightarrow G_1 \overset{\delta}{\Leftrightarrow\!\!\rightarrow} G_1' \\
&\Rightarrow H_1 \overset{\delta}{\Leftrightarrow\!\!\rightarrow}_l D'' \sim D' \text{ mit } (G_1', D') \in R \text{ (nach Induktionsannahme)} \\
&\Rightarrow H \overset{\delta}{\Leftrightarrow\!\!\rightarrow}_l H_2 \text{ mit } (G_2, H_2) \in R
\end{aligned}$$

Die Umkehrung gilt analog.

- $G = G_1 [> G_2$

  Analog zum Beweis für $\gg$.

- $G = \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ G'$ und $G = G'[b_1/a_1, \ldots, b_n/a_n]$

  trivial.

- $G = G_1 \,|[g_1, \ldots, g_n]|\, G_2$

  Dann gilt $H = H_1 \,|[g_1, \ldots, g_n]|\, H_2$ mit H_i als ein markierter Term von G_i und $i = 1, 2$. Es gilt

  1.

  $$\begin{aligned}
  &G \overset{a}{\Leftrightarrow\!\!\rightarrow} G_1' \,|[g_1, \ldots, g_n]|\, G_2' \text{ mit } a \in \{g_1, \ldots, g_n\} \\
  &\Rightarrow G_1 \overset{a}{\Leftrightarrow\!\!\rightarrow} G_1' \text{ und } G_2 \overset{a}{\Leftrightarrow\!\!\rightarrow} G_2' \\
  &\Rightarrow H_1 \overset{a}{\Leftrightarrow\!\!\rightarrow}_l D_1'' \sim D_1' \text{ mit } (G_1', D_1') \in R \text{ und} \\
  &\quad H_2 \overset{a}{\Leftrightarrow\!\!\rightarrow}_l D_2'' \sim D_2' \text{ mit } (G_2', D_2') \in R \text{ (nach Induktionsannahme)} \\
  &\Rightarrow H \overset{a}{\Leftrightarrow\!\!\rightarrow}_l D_1'' \,|[g_1, \ldots, g_n]|\, D_2''
  \end{aligned}$$

  Da nach Satz 3.1 $\sim$ eine Kongruenzrelation in $\mathcal{BL}_{er}^{ev}$ ist, gilt

  $$D_1'' \,|[g_1, \ldots, g_n]|\, D_2'' \sim H_1' \,|[g_1, \ldots, g_n]|\, H_2',$$

  wobei $H_1' = D_1'$ und $H_2' = D_2'$ markierte Terme von G_1' und G_1' sind. Setzen wir also $G' = G_1' \,|[g_1, \ldots, g_n]|\, G_2'$ und $H' = H_1' \,|[g_1, \ldots, g_n]|\, H_2'$, so ist H' ein markierter Term von G' und daher $(G', H') \in R$.

  Die Umkehrung gilt analog.

  2. $G \overset{a}{\Leftrightarrow\!\!\rightarrow} G_1' \,|[g_1, \ldots, g_n]|\, G_2$ mit $a \notin \{g_1, \ldots, g_n\}$

     Analog zu 1.

  3. $G \overset{a}{\Leftrightarrow\!\!\rightarrow} G_1 \,|[g_1, \ldots, g_n]|\, G_2'$ mit $a \notin \{g_1, \ldots, g_n\}$

     Analog zu 1.

- $G = P$ mit $P := B$

Sei $H = P_e$ und B' ein markierter Term von B.

$$\begin{aligned}
&G = P \overset{a}{\Leftrightarrow\!\!\!\rightarrow} C' \\
&\Rightarrow B \overset{a}{\Leftrightarrow\!\!\!\rightarrow} C' \\
&\Rightarrow B' \overset{a}{\Leftrightarrow\!\!\!\rightarrow}_l D'' \sim D' \text{ mit } (C', D') \in R \text{ (nach Induktionsannahme)} \\
&\Rightarrow P_e \overset{a}{\Leftrightarrow\!\!\!\rightarrow}_l e(D'') \sim D' \text{ mit } (C', D') \in R, \\
&\quad \text{da nach Lemma 3.1 } e(D'') \sim D'' \text{ gilt.}
\end{aligned}$$

Die Umkehrung gilt analog. □

Die im folgenden zitierten, einfachen Lemmata hängen wenig eng miteinander zusammen, sind jedoch für den nächsten Abschnitt von Bedeutung.

Lemma 3.3 *Sei $B \in \mathcal{BL}$, $\mathcal{E} = [\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$ und $e \in E$. Dann gilt $\mathcal{E} \overset{\overline{\{e\}}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}'$ genau dann, wenn $e \in init(\mathcal{E})$ gilt.*

Beweis: Der Beweis ist einfach und wird deshalb nicht ausgeführt. □

Lemma 3.4 *Sei $\mathcal{E}$ eine e.b.e.s. $\mathcal{E} \overset{\overline{K}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}'$ mit $e_1, \ldots, e_n$ als Ausführungsfolge von K gilt genau dann, wenn*

$$\mathcal{E} \overset{\overline{\{e_1\}}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}[\{e_1\}] \overset{\overline{K'}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}'$$

mit $K' = \{e_2, \ldots, e_n\}$ und $e_2 \ldots e_n$ als Ausführungsfolge von K' gilt.

Beweis:

- „$\Rightarrow$“
  Da $\{e_1\} \cap \{e_2, \ldots, e_n\} = \emptyset$ und

  $$\begin{aligned}
  \leq_K \cap (K \times \{e_1\}) &= \{(e, e_1) \mid e \in K \wedge e \leq_K e_1\} \\
  &= \{(e_1, e_1)\}
  \end{aligned}$$

  (da $e \leq_K e_1$ mit $e \neq e_1$ die Aussage $\exists e_i \in K \setminus \{e_1\} : (e_i \leq_K e_1 \wedge i \geq 1)$ impliziert und somit ein Widerspruch zur Definition 2.10 vorliegt.) gilt, ist $\overline{\{e_1\}}$ ein Präfix von $\overline{K}$. Somit gilt nach Theorem 2.2

  $$\mathcal{E} \overset{\overline{\{e_1\}}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}[\{e_1\}] \overset{\overline{K'}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}'$$

  mit $K' = \{e_2, \ldots, e_n\}$. Zu zeigen bleibt es noch, daß $e_2 \ldots e_n$ eine Ausführungsfolge von K' ist.

  1. Da aus $e_i \rightsquigarrow e_j$ in $\mathcal{E}[\{e_1\}]$ auch $e_i \rightsquigarrow e_j$ in $\mathcal{E}$ für $e_i, e_j \in K'$ folgt, gilt $i < j$.
  2. Wir unterscheiden zwei Fälle:
     (a) Gilt $X \mapsto e_i$ in $\mathcal{E}[\{e_1\}]$ mit $e_i \in K'$ und $X \neq \emptyset$, so gilt nach Definition 2.12 auch $X \mapsto e_i$ in $\mathcal{E}$ und somit $X \cap \{e_2, \ldots, e_{i-1}\} \neq \emptyset$, da $X \cap \{e_1\} = \emptyset$ und $X \cap \{e_1, \ldots, e_{i-1}\} \neq \emptyset$ gilt.
     (b) Gilt $\emptyset \mapsto e_i$ in $\mathcal{E}[\{e_1\}]$, so gilt wegen $e_1, \ldots, e_n \in PS(\mathcal{E})$ und nach Definition 2.12 $e_i \rightsquigarrow e_1$ in $\mathcal{E}$. Dies ist ein Widerspruch, da $i > 1$ gilt.

Aus 1. und 2. ist also $e_2 \dots e_n$ eine Ausführungsfolge von K'.

- „$\Leftarrow$“

  Nach Theorem 2.2 gilt $\mathcal{E} \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}'$. Wir zeigen nun, daß $e_1 e_2 \dots e_n \in PS(\mathcal{E})$ gilt.

  1. Zunächst gilt nach Definition 2.12 $\forall e_i, e_j \in K' : (e_i \rightsquigarrow e_j \text{ in } \mathcal{E}[\{e_1\}] \iff e_i \rightsquigarrow e_j \text{ in } \mathcal{E})$. Da $e_i \rightsquigarrow e_1$ in $\mathcal{E}$ mit $e_i \in K'$ die Relation $\emptyset \mapsto e_i$ in $\mathcal{E}[\{e_1\}]$ und somit $e_2 \dots e_n \notin PS(\mathcal{E}'_1)$ impliziert, folgt $\forall e_i, e_j \in K : (e_i \rightsquigarrow e_j \Longrightarrow i < j)$.
  2. Da $\neg(\exists X : X \mapsto e_1)$ gilt, bleibt es, nur die Elemente $e_2, \dots, e_n$ auf die Bedingung 2 der Definition 2.10 zu prüfen. Gilt $X \mapsto e_i$ in $\mathcal{E}$ mit $e_i \in K'$, so gilt entweder $X \cap \{e_1\} = \emptyset$ oder $X \cap \{e_1\} \neq \emptyset$. Im ersten Fall gilt $X \mapsto e_i$ in $\mathcal{E}[\{e_1\}]$ und wegen $X \cap \{e_2, \dots, e_{i-1}\} \neq \emptyset$ auch $X \cap \{e_1, e_2, \dots, e_{i-1}\} \neq \emptyset$. Im zweiten Fall folgt sofort $X \cap \{e_1, e_2, \dots, e_{i-1}\} \neq \emptyset$. □

Korollar 3.1 *Sei $\mathcal{E}$ eine e.b.e.s. und $e_1, \dots, e_n$ eine Ausführungsfolge von K. $\mathcal{E} \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}'$ gilt genau dann, wenn gilt:*

$$\mathcal{E} \overset{\overline{\{e_1\}}}{\Leftrightarrow} \mathcal{E}_1 \dots \mathcal{E}_{n-1} \overset{\overline{\{e_n\}}}{\Leftrightarrow} \mathcal{E}_n = \mathcal{E}'$$

Beweis: Wendet man das Lemma 3.4 n-mal an, so erhält man das Gewünschte. □

3.2 Parallelitätsvergleich

Die im folgenden eingeführten Parallelitätsbegriffe basieren auf einer Grundidee von Aceto, die in [Ace91] vorgestellt wurde. Aceto hat einen Parallelitätsbegriff für eine CCS-artige Sprache ohne Synchronisation eingeführt, der auf sogenannten Pomset-Transitionssystemen definiert ist. Damit können zwei äquivalente Prozesse $B1$ und $B2$ unterschieden werden, wenn $B1$ (bzw. $B2$) im Laufe seiner Existenz mehr Aktionen parallel zur Ausführung bringen kann als $B2$ (bzw. $B1$).

Wir wollen nun den Parallelitätsbegriff von [Ace91] zunächst sinngemäß auf e.b.e.s übertragen und definieren anschließend daraus zwei neue Begriffe, die in bestimmten Fällen für die praktischen Anforderungen besser geeignet sind. Doch zuvor sind noch einige grundlegende Definitionen und Sätze nötig.

Definition 3.4 ($\equiv$) *Sei $\overline{K_i} = (K_i, \leq_i, l_i)$ mit $i = 1, 2$ eine lposet. Gibt es eine Funktion $h : K_1 \to K_2$, wobei*

- *h ist bijektiv*
- *$\forall e \in K_1 : l_1(e) = l_2(h(e))$*

gilt, dann schreiben wir $\overline{K_1} \equiv \overline{K_2}$. □

$\equiv$ gibt anschaulich an, daß es sich um äquivalente Berechnungen handelt, wenn man sich eine Konfiguration als eine Berechnung vorstellt. Wie diese Berechnung durchgeführt wird, d.h. in welcher Reihenfolge die Aktionen ausgeführt werden, ist nicht von Relevanz.

Definition 3.5 ($Er(\mathcal{E})$) *Sei $\mathcal{E}$ eine e.b.e.s. $Er(\mathcal{E})$ ist die kleinste Menge, für die gilt:*

- $\mathcal{E} \in Er(\mathcal{E})$
- *Gilt* $\mathcal{E}_1 \in Er(\mathcal{E})$ *und* $\mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_2$, *dann gilt* $\mathcal{E}_2 \in Er(\mathcal{E})$. □

$Er(\mathcal{E})$ ist also die Menge der e.b.e.s, die von $\mathcal{E}$ aus erreichbar sind.

Definition 3.6 ($\approx_K$, $\approx_E$)

- *Eine Relation* $R \subseteq \mathcal{ES} \times \mathcal{ES}$ *heißt eine K-Bisimulation, wenn für alle* $(\mathcal{E}_1, \mathcal{E}_2) \in R$ *gilt:*
  1. $\mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_1' \Longrightarrow \exists \overline{K'} : (\overline{K} \equiv \overline{K'} \wedge \mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow} \mathcal{E}_2' \wedge (\mathcal{E}_1', \mathcal{E}_2') \in R)$
  2. $\mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow} \mathcal{E}_2' \Longrightarrow \exists \overline{K} : (\overline{K} \equiv \overline{K'} \wedge \mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_1' \wedge (\mathcal{E}_1', \mathcal{E}_2') \in R)$
- $\mathcal{E}_1$ *und* $\mathcal{E}_2$ *sind K-bisimulationsäquivalent* ($\mathcal{E}_1 \approx_K \mathcal{E}_2$), *wenn es eine K-Bisimulation* $R \subseteq Er(\mathcal{E}_1) \times Er(\mathcal{E}_2)$ *gibt, so daß* $(\mathcal{E}_1, \mathcal{E}_2) \in R$ *gilt.*
- *Ist* $R \subseteq \mathcal{ES} \times \mathcal{ES}$ *eine K-Bisimulation und* K *eine einelementige Menge, dann heißt* R *eine E-Bisimulation.*
- $\mathcal{E}_1$ *und* $\mathcal{E}_2$ *sind E-bisimulationsäquivalent* ($\mathcal{E}_1 \approx_E \mathcal{E}_2$), *wenn es eine E-Bisimulation* $R \subseteq Er(\mathcal{E}_1) \times Er(\mathcal{E}_2)$ *gibt, so daß* $(\mathcal{E}_1, \mathcal{E}_2) \in R$ *gilt.* □

Satz 3.3 *Seien* $\mathcal{E}$ *und* $\mathcal{E}'$ *e.b.e.s. Dann gilt* $\mathcal{E} \approx_K \mathcal{E}'$ *genau dann, wenn* $\mathcal{E} \approx_E \mathcal{E}'$ *gilt.*

Beweis:

- „$\Rightarrow$“

  Ist R die zugehörige K-Bisimulation, dann ist R offenbar auch eine E-Bisimulation.

- „$\Leftarrow$“

  Sei R die E-Bisimulation mit $(\mathcal{E}, \mathcal{E}') \in R$. Wir zeigen nun, daß R auch eine K-Bisimulation ist. Sei also $(\mathcal{E}_1, \mathcal{E}_2) \in R$.

  1. Gilt $\mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_1'$, dann gibt es eine Ausführungsfolge $e_1, \ldots, e_n$ mit $K = \{e_1, \ldots, e_n\}$, so daß nach Korollar 3.1 folgendes gilt:

     $$\mathcal{E}_1 \overset{\overline{\{e_1\}}}{\Leftrightarrow} \mathcal{E}_1^1 \ldots \mathcal{E}_1^{n-1} \overset{\overline{\{e_n\}}}{\Leftrightarrow} \mathcal{E}_1^n = \mathcal{E}_1'$$

     Da $\mathcal{E}_1 \approx_E \mathcal{E}_2$ gilt, gibt es eine Folge von $e_1', \ldots, e_n'$, so daß

     $$\mathcal{E}_2 \overset{\overline{\{e_1'\}}}{\Leftrightarrow} \mathcal{E}_2^1 \ldots \mathcal{E}_2^{n-1} \overset{\overline{\{e_n'\}}}{\Leftrightarrow} \mathcal{E}_2^n = \mathcal{E}_2'$$

     und $(\mathcal{E}_1^i, \mathcal{E}_2^i) \in R \wedge l_1(e_i) = l_2(e_i')$ mit $i = 1, \ldots, n$ gilt. Sei also $\overline{H}$ mit $H = \{e_1', \ldots, e_n'\}$, dann gilt $\overline{K} \equiv \overline{H}$ und nach Korollar 3.1 $\mathcal{E}_2 \overset{\overline{H}}{\Leftrightarrow} \mathcal{E}_2'$. Also gibt es ein $\overline{K'}$ mit $\overline{K} \equiv \overline{K'}$, so daß $\mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow} \mathcal{E}_2'$ und $(\mathcal{E}_1', \mathcal{E}_2') \in R$ gilt. Daß $\mathcal{E}_1' \in Er(\mathcal{E})$ und $\mathcal{E}_2' \in Er(\mathcal{E}')$ gilt, ist offensichtlich.

  2. Für $\mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow} \mathcal{E}_2'$ gilt es analog zu zeigen.

Aus 1. und 2. ist R eine K-Bisimulation. $\square$

Wir ordnen nun jeder e.b.e.s. ein Transitionssystem zu.

Definition 3.7 ($\mathcal{T}(\mathcal{E})$) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine e.b.e.s., dann ist*

$$\mathcal{T}(\mathcal{E}) := (Er(\mathcal{E}), \Leftrightarrow\!\!\rightarrow, \mathcal{E})$$

mit $\Leftrightarrow\!\!\rightarrow \subseteq Er(\mathcal{E}) \times Act \times Er(\mathcal{E})$ und $\Leftrightarrow\!\!\rightarrow = \{(\mathcal{E}', l(e), \mathcal{E}'') \mid \mathcal{E}', \mathcal{E}'' \in Er(\mathcal{E}) \wedge \exists e : \mathcal{E}' \overset{\overline{\{e\}}}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}''\}$.

$\square$

Lemma 3.5 *Sei $\mathcal{E}, \mathcal{E}'$ e.b.e.s., dann gilt $\mathcal{E} \approx_E \mathcal{E}' \Longleftrightarrow \mathcal{T}(\mathcal{E}) \sim_s \mathcal{T}(\mathcal{E}')$.*

Beweis: Offenbar ist $R \subseteq Er(\mathcal{E}) \times Er(\mathcal{E}')$ mit $(\mathcal{E}, \mathcal{E}') \in R$ eine E-Bisimulation genau dann, wenn R eine Bisimulation ist. $\square$

Es läßt sich zeigen, daß $\mathcal{T}(\mathcal{E}) \sim_s \mathcal{T}(\mathcal{E}')$ gilt, wenn $\mathcal{E}$ und $\mathcal{E}'$ isomorph sind.

Lemma 3.6 *Sei $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ eine e.b.e.s. mit $i = 1, 2$ und $\mathcal{E}_1 \cong \mathcal{E}_2$. Dann gilt $\mathcal{T}(\mathcal{E}) \sim_s \mathcal{T}(\mathcal{E}')$.*

Beweis: Sei $h : E_1 \to E_2$ ein Isomorphismus zwischen $\mathcal{E}_1$ und $\mathcal{E}_2$. Damit läßt sich folgendes leicht zeigen:

$$\text{Gilt } \mathcal{E}_1 \overset{\overline{\{e\}}}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_1', \text{ dann gilt } \mathcal{E}_2 \overset{\overline{\{h(e)\}}}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_2' \text{ und } \mathcal{E}_1' \cong \mathcal{E}_2'.$$

Nach Lemma 3.3 ist $e \in init(\mathcal{E}_1)$. Wegen Isomorphie gilt dann $h(e) \in init(\mathcal{E}_2)$ und somit $\mathcal{E}_2 \overset{\overline{\{h(e)\}}}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_2'$. Sei nun $f : E_1 \setminus \{e\} \to E_2 \setminus \{h(e)\}$ mit $f(e') := h(e')$, dann folgt sofort, daß f ein Isomorphismus zwischen $\mathcal{E}_1'$ und $\mathcal{E}_2'$ ist. Sei $R = \{(\mathcal{E}, \mathcal{E}') \mid \mathcal{E} \in Er(\mathcal{E}_1) \wedge \mathcal{E}' \in Er(\mathcal{E}_2) \wedge \mathcal{E} \cong \mathcal{E}'\}$, dann ist R offenbar eine Bisimulation. $\square$

Unser Ziel ist es jetzt, die Bisimulationsäquivalenz, die auf der operationellen Semantik definiert ist, mittels der Halbordnungssemantik auszudrücken. Dazu brauchen wir das nachfolgende Lemma.

Lemma 3.7 *Sei $B \in \mathcal{BL}^{ev}$ und $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ mit $ETr(B) = PS(\mathcal{E})$. Sei ferner*

$$B \overset{e_1, l(e_1)}{\Leftrightarrow\!\!\!\!\rightarrow} B_1 \ldots B_{n-1} \overset{e_n, l(e_n)}{\Leftrightarrow\!\!\!\!\rightarrow} B_n$$

für $e_1 \cdots e_n \in ETr(B)$. Dann gilt:

1. $B \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B' \Longleftrightarrow \mathcal{E} \overset{\overline{\{e\}}}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}'$ *mit $l(e) = a$*
2. *Sei $B \overset{e,a}{\Leftrightarrow\!\!\rightarrow} B'$ und $\mathcal{E} \overset{\overline{\{e\}}}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}'$, so gilt $ETr(B') = PS(\mathcal{E}')$ und*

$$B' \overset{\psi_1, l(\psi_1)}{\Leftrightarrow\!\!\!\!\rightarrow} B_1' \ldots B_{n-1} \overset{\psi_n, l(\psi_n)}{\Leftrightarrow\!\!\!\!\rightarrow} B_n'$$

für $\psi_1 \cdots \psi_n \in ETr(B')$.

Beweis: Zu 1.: trivial. Zu 2.: Sei $e_2 \dots e_n \in ETr(B')$, dann gilt $e_1 e_2 \dots e_n \in ETr(B)$ mit $e_1 = e$ und somit auch $e_1 e_2 \dots e_n \in PS(\mathcal{E})$. Nach Lemma 3.4 gilt $e_2 \dots e_n \in PS(\mathcal{E}')$. Gilt umgekehrt $e_2 \dots e_n \in PS(\mathcal{E}')$, so gilt nach Lemma 3.4 $e_1 e_2 \dots e_n \in PS(\mathcal{E})$. Also ist $e_1 e_2 \dots e_n \in ETr(B)$. Da nach Lemma 2.5 B' eindeutig ist, folgt sofort $e_2 \dots e_n \in ETr(B')$. Die Aussage

$$B' \stackrel{\psi_1, l(\psi_1)}{\Longrightarrow} B'_1 \dots B_{n-1} \stackrel{\psi_n, l(\psi_n)}{\Longrightarrow} B'_n$$

für $\psi_1 \cdots \psi_n \in ETr(B')$ gilt offenbar. □

Lemma 3.8 *Sei $B \in \mathcal{BL}$ und $\mathcal{E} = [\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$, dann gilt $\mathcal{OS}(B) \sim_s \mathcal{T}(\mathcal{E})$.*

Beweis: Da $\mathcal{OS}(B) \sim_s l(\mathcal{M}(B))$ und nach Lemma 3.6 $\mathcal{T}([\![\mathcal{M}(B)]\!]) \sim_s \mathcal{T}(\mathcal{E})$ gilt, zeigen wir wegen Transitivität von $\sim_s$ nur die Gültigkeit von

$$l(\mathcal{M}(B)) \sim_s \mathcal{T}([\![\mathcal{M}(B)]\!]).$$

Sei $R \subseteq \mathcal{BL}^{ev} \times \mathcal{ES}$ mit

$$\begin{aligned} R \;=\; \{(B', \mathcal{E}') \mid\; & B' \in \mathcal{BL}^{ev} \wedge ETr(B') = PS(\mathcal{E}') \wedge \\ & \forall e_1 \cdots e_n \in Etr(B') : B' \stackrel{e_1, l'(e_1)}{\Longrightarrow} B'_1 \dots B_{n-1} \stackrel{e_n, l'(e_n)}{\Longrightarrow} B'_n\}. \end{aligned}$$

Dabei ist $\mathcal{E}' = (E', \rightsquigarrow', \mapsto', l')$. Dann ist R eine Bisimulation, denn es gilt für $(B', \mathcal{E}') \in R$:

1. Gilt $B' \stackrel{a}{\Longrightarrow}_l B''$, so gilt $\exists e : B' \stackrel{a,e}{\Longrightarrow} B''$. Daraus folgt wegen $ETr(B') = PS(\mathcal{E}')$ auch $\mathcal{E}' \stackrel{\overline{\{e\}}}{\Longrightarrow} \mathcal{E}''$ und somit $\mathcal{E}' \stackrel{a}{\Longrightarrow} \mathcal{E}''$. Nach Lemma 3.7 gilt $(B'', \mathcal{E}'') \in R$.
2. Analog zu 1.

Da aus Korollar 2.2 $(\mathcal{M}(B), [\![\mathcal{M}(B)]\!]) \in R$ folgt, gilt $l(\mathcal{M}(B)) \sim_s \mathcal{T}([\![\mathcal{M}(B)]\!])$. □

Satz 3.4 *Sei $B_1, B_2 \in \mathcal{BL}$ mit $\mathcal{E}_1 = [\![B_1]\!]$ und $\mathcal{E}_2 = [\![B_2]\!]$. Dann gilt $B_1 \sim B_2$ genau dann, wenn $\mathcal{E}_1 \approx_K \mathcal{E}_2$ gilt.*

Beweis:

$$\begin{aligned} B_1 \sim B_2 &\iff \mathcal{OS}(B_1) \sim_s \mathcal{OS}(B_2) \\ &\iff \mathcal{T}(\mathcal{E}_1) \sim_s \mathcal{T}(\mathcal{E}_2) \text{ nach Lemma 3.8} \\ &\iff \mathcal{E}_1 \approx_E \mathcal{E}_2 \text{ nach Lemma 3.5} \\ &\iff \mathcal{E}_1 \approx_K \mathcal{E}_2 \text{ nach Satz 3.3} \end{aligned}$$

□

Mit Hilfe des Satzes 3.4 läßt sich der Parallelitätsbegriff nach [Ace91] leicht auf e.b.e.s. übertragen.

Definition 3.8 (Projektion, $\lhd$) *Sei $\overline{K_i} = (K_i, \leq_i, l_i)$ mit $i = 1, 2$ eine lposet. Eine Funktion $h : K_2 \to K_1$ heißt eine Projektion von K_2 auf K_1, wenn gilt:*

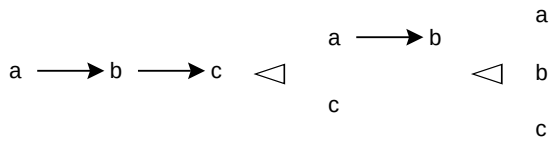
- *h ist bijektiv*
- $\forall e, e' \in K_2 : (e \leq_2 e' \Longrightarrow h(e) \leq_1 h(e'))$

- $\forall e \in K_2 : l_2(e) = l_1(h(e))$

Gibt es eine Projektion von K_2 auf K_1, so schreiben wir $\overline{K_1} \lhd \overline{K_2}$. □

Der Begriff „Projektion“ stammt von [Ace91] und besagt anschaulich, daß eine Konfiguration K mit $\overline{K} \lhd \overline{K'}$ mehr kausale Abhängigkeiten als K' enthält. Es gibt also in K' mehr Aktionen als in K, die kausal unabhängig voneinander ausgeführt werden.

Beispiel 3.2 Aus [Ace91] übernommen.



□

Der Parallelitätsbegriff nach [Ace91] lautet wie folgt:

Definition 3.9 ($\sqsubseteq_A$) *Sei $B, B' \in \mathcal{BL}$ mit $\mathcal{E} = [\![B]\!]$, $\mathcal{E}' = [\![B']\!]$. Dann ist B A-parallel zu B' ($B \sqsubseteq_A B'$), wenn es eine Relation $R \subseteq Er(\mathcal{E}) \times Er(\mathcal{E}')$ mit $(\mathcal{E}, \mathcal{E}') \in R$ gibt und für alle $(\mathcal{E}_1, \mathcal{E}_2) \in R$ gilt:*

1. $\mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_1' \Longrightarrow \exists \overline{K'} : (\overline{K} \lhd \overline{K'} \wedge \mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow} \mathcal{E}_2' \wedge (\mathcal{E}_1', \mathcal{E}_2') \in R)$
2. $\mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow} \mathcal{E}_2' \Longrightarrow \exists \overline{K} : (\overline{K} \lhd \overline{K'} \wedge \mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_1' \wedge (\mathcal{E}_1', \mathcal{E}_2') \in R)$ □

$\sqsubseteq_A$ hat folgende anschauliche Bedeutung: Steht B in der Relation $\sqsubseteq_A$ zu B', dann: 1) Zu jeder Berechnung in B gibt es eine äquivalente Berechnung in B', in der mehr Aktionen parallel ausgeführt werden. 2) Zu jeder Berechnung in B' gibt es eine äquivalente Berechnung in B, in der weniger Aktionen parallel ausgeführt werden.

Beispiel 3.3 In diesem Beispiel und auch in den nachfolgenden Beispielen werden wir den Prozeß **stop** nicht explizit angeben. Wir schreiben z.B. statt $a; \mathbf{stop}$ kurz a.

1. $a; b \,[]\, b; a \sqsubseteq_A a \,|||\, b$
2. $a; b \,[]\, (a \,|||\, b) \sqsubseteq_A a \,|||\, b$, aber $a \,|||\, b \not\sqsubseteq_A a; b \,[]\, (a \,|||\, b)$
3. $a \,|||\, a; a \sqsubseteq_A (a \,|||\, a; a) \,[]\, (a \,|||\, a \,|||\, a) \sqsubseteq_A a \,|||\, a \,|||\, a$,
   aber $a \,|||\, a \,|||\, a \not\sqsubseteq_A (a \,|||\, a; a) \,[]\, (a \,|||\, a \,|||\, a)$ □

Korollar 3.2 *Sei $B, C \in \mathcal{BL}$. Gilt $B \sqsubseteq_A C$, dann gilt $B \sim C$.* □

Mit $\sqsubseteq_A$ lassen sich also Prozesse vergleichen, die sich bezüglich ihrer Ausführung unterscheiden. $\sqsubseteq_A$ weist jedoch in mancher Hinsicht eine Schwäche auf. Betrachten wir z.B. die Prozesse $B1 = a; a; a \,[]\, (a \,|||\, a \,|||\, a)$ und $B2 = (a \,|||\, a; a) \,[]\, (a \,|||\, a \,|||\, a)$, dann gilt nach Aceto $B1 \sqsubseteq_A B2$. Nehmen wir nun (wie in [RB91]) folgendes an:

1. Für $B1$ ist die Wahrscheinlichkeit, daß der rechte Teilprozeß von $B1$ ausgewählt wird, gleich eins.

2. Für $B2$ ist dagegen die Wahrscheinlichkeit, daß der rechte Teilprozeß von $B2$ ausgewählt wird, gleich Null.

Dann kann offenbar $B1$ nicht langsamer bezüglich der Ausführungsgeschwindigkeit (und somit bezüglich der Parallelität nicht schlechter) als $B2$ betrachtet werden. Im Gegenteil wird $B1$ immer vor $B2$ abgeschlossen ausgeführt sein.

Um diesen Fall auch noch zu berücksichtigen, müssen die Bedingungen für $\sqsubseteq_A$ anders formuliert werden. Wir kommen daher zu einem neuen (strengeren) Parallelitätsbegriff.

Definition 3.10 (Notationen) *Sei $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ mit $i = 1, 2$.*

1. *Sei $\sigma = e_1 \cdots e_n$ eine Ausführungsfolge, dann $\overline{\sigma} := \overline{\{e_1, \ldots, e_n\}}$.*

2. *Sei $\sigma = e_1 \cdots e_n$ eine Ausführungsfolge einer e.b.e.s. $\mathcal{E}$. Dann schreiben wir an Stelle*
$$\mathcal{E} \overset{\overline{\{e_1\}}}{\Leftrightarrow} \mathcal{E}_1 \cdots \mathcal{E}_{n-1} \overset{\overline{\{e_n\}}}{\Leftrightarrow} \mathcal{E}_n = \mathcal{E}'$$
auch kurz $\mathcal{E} \overset{\sigma}{\rightarrow} \mathcal{E}'$.

3. *$Abl(\mathcal{E}_1, \mathcal{E}_2)$ ist die kleinste Menge, für die folgendes gilt:*
Sei $\sigma = \xi_1 \cdots \xi_n$ eine Ausführungsfolge von $\mathcal{E}_1$ und $\mu = \psi_1 \cdots \psi_n$ eine Ausführungsfolge von $\mathcal{E}_2$. Sei ferner
$$\mathcal{E}_1 \overset{\overline{\{\xi_1\}}}{\Leftrightarrow} \mathcal{E}_1^1 \cdots \mathcal{E}_1^{n-1} \overset{\overline{\{\xi_n\}}}{\Leftrightarrow} \mathcal{E}_1^n = \mathcal{E}_1'$$
und
$$\mathcal{E}_2 \overset{\overline{\{\psi_1\}}}{\Leftrightarrow} \mathcal{E}_2^1 \cdots \mathcal{E}_2^{n-1} \overset{\overline{\{\psi_n\}}}{\Leftrightarrow} \mathcal{E}_2^n = \mathcal{E}_2'.$$
Gilt $l_1(\xi_i) = l_2(\psi_i)$ und $\mathcal{T}(\mathcal{E}_1^i) \sim_s \mathcal{T}(\mathcal{E}_2^i)$ für $i = 1, \cdots, n$, dann $(\mathcal{E}_1 \overset{\sigma}{\rightarrow} \mathcal{E}_1', \mathcal{E}_2 \overset{\mu}{\rightarrow} \mathcal{E}_2') \in Abl(\mathcal{E}_1, \mathcal{E}_2)$. □

Man beachte, daß wir auch $(\mathcal{E}_1 \overset{\epsilon}{\rightarrow} \mathcal{E}_1, \mathcal{E}_2 \overset{\epsilon}{\rightarrow} \mathcal{E}_2) \in Abl(\mathcal{E}_1, \mathcal{E}_2)$ zulassen. Außerdem soll $\overline{\epsilon} \lhd \overline{\epsilon}$ gelten.

Definition 3.11 ($\sqsubseteq_P$) *Sei $B, C \in \mathcal{BL}$ mit $\mathcal{E}_B = [\![B]\!]$ und $\mathcal{E}_C = [\![C]\!]$. Dann ist B P-parallel zu C ($B \sqsubseteq_P C$), wenn folgendes gilt:*

- $\mathcal{T}(\mathcal{E}_B) \sim_s \mathcal{T}(\mathcal{E}_C)$
- *Sei $(\mathcal{E}_B \overset{\sigma}{\rightarrow} \mathcal{E}_B', \mathcal{E}_C \overset{\sigma'}{\rightarrow} \mathcal{E}_C') \in Abl(\mathcal{E}_B, \mathcal{E}_C)$ und μ eine Ausführungsfolge. Dann gilt:*
$$\mathcal{E}_B' \overset{\mu}{\rightarrow} \mathcal{E}_B'' \Longrightarrow \forall \mu' : ((\mathcal{E}_B' \overset{\mu}{\rightarrow} \mathcal{E}_B'', \mathcal{E}_C' \overset{\mu'}{\rightarrow} \mathcal{E}_C'') \in Abl(\mathcal{E}_B', \mathcal{E}_C') \Longrightarrow \overline{\mu} \lhd \overline{\mu'})$$

□

$\sqsubseteq_P$ hat folgende anschauliche Bedeutung: Steht B in der Relation $\sqsubseteq_P$ zu C und erreichen B und C über eine äquivalente Berechnung einen äquivalenten Zustand Z, so gibt es zu jeder Berechnung im Zustand Z in C keine äquivalente Berechnung im Zustand Z in B, in der mehr Aktionen parallel ausgeführt werden. Unter einer äquivalenten Berechnung soll hier eine Ausführungsfolge verstanden werden, die die Bedingung der Definition 3.10(3) erfüllt.

Wenn wir also davon ausgehen, daß einer Aktion bei beiden Prozessen eine gleiche Ausführungsdauer zugeordnet wird, dann können wir annehmen, daß $B2$ seine Aktionen immer in einer kürzeren Zeitdauer ausführt als $B1$.

Beispiel 3.4

1. $a;b \,[]\, b;a \sqsubseteq_P a \,|||\, b$
2. $(a;(b \,[]\, c)) \,[]\, (a;b \,[]\, b;a) \sqsubseteq_P (a;(b \,[]\, c)) \,[]\, (a \,|||\, b)$
3. $a;a;a \,[]\, (a \,|||\, a \,|||\, a) \not\sqsubseteq_P (a \,|||\, a;a) \,[]\, (a \,|||\, a \,|||\, a)$ □

Satz 3.5 *Sei $B, C \in \mathcal{BL}$, dann gilt $B \sqsubseteq_P C \Longrightarrow B \sqsubseteq_A C$.*

Beweis: Sei $\mathcal{E}_B = [\![B]\!] = (E_B, \rightsquigarrow_B, \mapsto_B, l_B)$, $\mathcal{E}_C = [\![C]\!] = (E_C, \rightsquigarrow_C, \mapsto_C, l_C)$ und

$$S = \{(\mathcal{E}_1, \mathcal{E}_2) \,|(\mathcal{E}_B \stackrel{\sigma}{\rightarrow} \mathcal{E}_1, \mathcal{E}_C \stackrel{\sigma'}{\rightarrow} \mathcal{E}_2) \in Abl(\mathcal{E}_B, \mathcal{E}_C)\}.$$

Dann gilt zunächst $S \subseteq Er(\mathcal{E}_B) \times Er(\mathcal{E}_C)$ und $(\mathcal{E}_B, \mathcal{E}_C) \in S$. Zu zeigen ist also, daß die Bedingungen 1. und 2. der Definition 3.9 gelten. Sei $(\mathcal{E}_1, \mathcal{E}_2) \in S$.

1. Sei $\mathcal{E}_1 \stackrel{\overline{K1}}{\Leftrightarrow} \mathcal{E}_1'$ und $\xi_1, \cdots, \xi_n$ eine Ausführungsfolge für K, also

   $$\mathcal{E}_1 \stackrel{\overline{\{\xi_1\}}}{\Leftrightarrow} \mathcal{E}_1^1 \cdots \mathcal{E}_1^{n-1} \stackrel{\overline{\{\xi_n\}}}{\Leftrightarrow} \mathcal{E}_1^n = \mathcal{E}_1'.$$

   Dann gibt es wegen $\mathcal{T}(\mathcal{E}_1) \sim_s \mathcal{T}(\mathcal{E}_2)$ eine Ausführungsfolge $\psi_1, \cdots, \psi_n$ und eine E-Bisimulation R, so daß

   $$\mathcal{E}_2 \stackrel{\overline{\{\psi_1\}}}{\Leftrightarrow} \mathcal{E}_2^1 \cdots \mathcal{E}_2^{n-1} \stackrel{\overline{\{\psi_n\}}}{\Leftrightarrow} \mathcal{E}_2^n = \mathcal{E}_2',$$

   $l_B(\xi_i) = l_C(\psi_i)$ und $(\mathcal{E}_1^i, \mathcal{E}_2^i) \in R$ (und somit $\mathcal{T}(\mathcal{E}_1^i) \sim_s \mathcal{T}(\mathcal{E}_2^i)$) für $i = 1, \cdots, n$ gilt. Setzen wir $\mu = \xi_1 \cdots \xi_n$ und $\mu' = \psi_1 \cdots \psi_n$, dann gilt offenbar

   $$(\mathcal{E}_B \stackrel{\sigma\mu}{\Leftrightarrow} \mathcal{E}_1', \mathcal{E}_C \stackrel{\sigma'\mu'}{\Leftrightarrow} \mathcal{E}_2') \in Abl(\mathcal{E}_B, \mathcal{E}_C)$$

   und somit $(\mathcal{E}_1', \mathcal{E}_2') \in S$. Daß $\overline{K1} \lhd\!\!\rhd \overline{\mu'}$ gilt, folgt aus der Voraussetzung $B \sqsubseteq_P C$.

2. Analog zu 1. □

Wie das Beispiel 3.4(3.) zeigt, gilt die Umkehrung nicht.

Es ist anzumerken, daß $\sqsubseteq_P$ und $\sqsubseteq_A$ keine Halbordnungen sind, da die Antisymmetrie verletzt ist. Beispiel: Sei $B1 = (a \,[]\, b) \,|[b]|\, (b \,|||\, c)$ und $B2 = (a \,[]\, b) \,|||\, c$, dann $B1 \sqsubseteq_P B2$ und $B2 \sqsubseteq_P B1$.

Außerdem ist $\sqsubseteq_P$ nicht reflexiv, da $a \,|||\, a;a \not\sqsubseteq_P a \,|||\, a;a$ gilt. Es gibt sogar Prozesse, die bezüglich $\sqsubseteq_A$ und $\sqsubseteq_P$ nicht vergleichbar sind.

Beispiel 3.5 Sei $B1 = ((a;d) \,|||\, c) \,[]\, d;a;c$ und $B2 = ((a;c) \,|||\, d) \,[]\, c;a;d$, dann gilt $B1 \not\sqsubseteq_A B2$ (und somit $B1 \not\sqsubseteq_P B2$) und $B2 \not\sqsubseteq_A B1$ (und somit $B2 \not\sqsubseteq_P B1$), obwohl $B1 \sim B2$ gilt. □

$\sqsubseteq_A$ und $\sqsubseteq_P$ sind in manchen Fällen viel zu streng, da sie Prozesse nicht vergleichen, die, intuitiv gesehen, vergleichbar sind. Betrachten wir beispielsweise die Prozesse $B1 = ((a;d) \,|||\, c) \,[]\, d;a;c$ und $B2 = ((a;d) \,|||\, c) \,[]\, (d \,|||\, a;c)$, dann sind $B1$ und $B2$ bezüglich $\sqsubseteq_A$ und somit bezüglich $\sqsubseteq_P$ nicht vergleichbar, obwohl in $B2$ die Aktionen d und a (bzw. d und c) im linken Teilprozeß kausal unabhängig sind.

Aus diesem Grund führen wir nun einen abgeschwächten Parallelitätsbegriff ein.

Definition 3.12 ($\sqsubseteq_W$) *Sei $B, B' \in \mathcal{BL}$ mit $\mathcal{E} = [\![B]\!]$, $\mathcal{E}' = [\![B']\!]$. Dann ist B W-parallel zu B' ($B \sqsubseteq_W B'$), wenn es eine Relation $R \subseteq Er(\mathcal{E}) \times Er(\mathcal{E}')$ mit $(\mathcal{E}, \mathcal{E}') \in R$ gibt und für alle $(\mathcal{E}_1, \mathcal{E}_2) \in R$ gilt:*

1. $\mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_1' \Longrightarrow \exists \overline{K'} : (\overline{K} \lhd \overline{K'} \wedge \mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow} \mathcal{E}_2' \wedge (\mathcal{E}_1', \mathcal{E}_2') \in R)$
2. $\mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow} \mathcal{E}_2' \Longrightarrow \exists \overline{K} : (\overline{K} \equiv \overline{K'} \wedge \mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_1' \wedge (\mathcal{E}_1', \mathcal{E}_2') \in R)$ □

Betrachten wir die oben angegebenen Prozesse $B1$ und $B2$, dann gilt $B1 \sqsubseteq_W B2$ und $B2 \not\sqsubseteq_W B1$.

Auch $\sqsubseteq_W$ ist keine Halbordnung, da die Antisymmetrie verletzt ist, z.B.:
$(a; b \,[]\, b; a) \,[]\, (a \,|||\, b) \sqsubseteq_W (a \,|||\, b)$ und $(a \,|||\, b) \sqsubseteq_W (a; b \,[]\, b; a) \,[]\, (a \,|||\, b)$.

Korollar 3.3 *Sei $B, C \in \mathcal{BL}$, dann gilt $B \sqsubseteq_A C \Longrightarrow B \sqsubseteq_W C$.* □

Korollar 3.4 *Sei $B, C \in \mathcal{BL}$. Gilt $B \sqsubseteq_W C$, dann gilt $B \sim C$.* □

3.3 Parallelitätsgrad

Im diesem Abschnitt wird ein vierter Parallelitätsbegriff eingeführt, der im Vergleich zu $\sqsubseteq_A$, $\sqsubseteq_P$ und $\sqsubseteq_W$ nicht die kausale Abhängigkeit der Ereignisse in jeder Konfiguration betrachet, sondern den Parallelitätsgrad eines Prozesses berücksichtigt. Informell beschrieben wird im folgenden unter Parallelitätsgrad eines Prozesses die maximale Anzahl der Aktionen verstanden, die ein Prozeß kausal unabhängig ausführen kann.

Definition 3.13 (Schritt-Transition) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine e.b.e.s und $\overline{K} := (K, \leq_K, l|_K)$ eine lposet von $\mathcal{E}$. $\mathcal{E} \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}'$ heißt eine Schritt-Transition, wenn $\leq_K$ eine Identitätsrelation ist.* □

Definition 3.14 (Notationen) *Sei $B \in \mathcal{BL}$, $g \in Act(B)$ und $\mathcal{E} = [\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$. Dann ist*

- $ST(B) := \{K \mid \exists \mathcal{E}_1, \mathcal{E}_2 \in Er(\mathcal{E}) : \mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_2$ *ist eine Schritt-Transition*$\}$.
- $ST(g, B) := \{K_g \mid K_g = K \cap \{e \,|\, l(e) = g\} \wedge K \in ST(B)\}$.

Definition 3.15 (Parallelitätsgrad) *Sei $B \in \mathcal{BL}$ und*

$$M = \{|K| \mid K \in ST(B)\}.$$

Dann ist $G(B) := max(M)$ der Parallelitätsgrad von B. Falls $\forall k \in I\!N : \exists K \in M : |K| > k$ gilt, dann sprechen wir von einem unendlichen Parallelitätsgrad. In diesem Fall schreiben wir $G(B) = \infty$. □

Beispiel 3.6

1. $G(a \,|||\, b) = 2$, während $G(a; b \,[]\, b; a) = 1$.
2. $G((a \,|||\, a) \,|[a]|\, (a \,|||\, a)) = 2$.

3. Für $P := P|||a$ gilt $G(P) = \infty$. □

Es ist manchmal hilfreich zu wissen, wie groß der Parallelitätsgrad eines Prozesses maximal sein darf (Darunter versteht man den maximalen Parallelitätsgrad, den ein äquivalenter Prozeß annehmen kann). Damit läßt sich z.B. vorhersagen, wieviele Prozessoren man mindestens benötigt, um einen Prozeß mit einem maximalen Parallelitätsgrad ausführen zu können.

Definition 3.16 (Maximaler Parallelitätsgrad) *Sei $B \in \mathcal{BL}$. Gilt*

$$\exists B' \in \mathcal{BL} : (B \sim B' \wedge \forall C \in \mathcal{BL} : (B \sim C \Longrightarrow G(C) \leq G(B'))),$$

dann ist $G_{max}(B) := G(B')$ der maximale Parallelitätsgrad von B.

In Abschnitt 3.4 werden wir zeigen, daß auf einer Teilsprache von $\mathcal{BL}$ ohne Rekursion der maximale Parallelitätsgrad sich abschätzen läßt.

Wir führen nun den vierten Parallelitätsbegriff ein.

Definition 3.17 ($\sqsubseteq_G$) *Sei $B, B' \in \mathcal{BL}$. B ist G-parallel zu B' ($B' \sqsubseteq_G B$), wenn $B \sim B'$ und $G(B) \geq G(B')$ gilt.* □

Auch hier ist $\sqsubseteq_G$ wegen der Antisymmetrie keine Halbordnung. Fordert man jedoch in Definition 3.17 $G(B) > G(B')$, so ist die reflexive Hülle von $\sqsubseteq_G$ eine Halbordnung.

Satz 3.6 *Sei $B, B' \in \mathcal{BL}$. Gilt $B \sqsubseteq_W B'$, dann gilt $B \sqsubseteq_G B'$.*

Beweis: Angenommen, es gilt $\neg(B \sqsubseteq_G B')$. Dann gibt es ein $K \in ST(B)$, so daß $|K| > |K'|$ für alle $K' \in ST(B')$ gilt. Sei nun $\mathcal{E} = [\![B]\!]$, $\mathcal{E}' = [\![B']\!]$ und $\mathcal{E}_1 \in Er(\mathcal{E})$ mit $\mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1'$. Da wegen Voraussetzung $B \sqsubseteq_W B'$ gilt, folgt $\exists \mathcal{E}_2 \in Er(\mathcal{E}') : \mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_2'$ mit $\overline{K} \lhd \overline{K'}$. Da $\mathcal{E}_1 \overset{\overline{K}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1'$ eine Schritt-Transition ist, so ist auch $\mathcal{E}_2 \overset{\overline{K'}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_2'$ eine Schritt-Transition. Also gibt es ein $K' \in ST(B')$ mit $|K'| = |K|$. Widerspruch. □

Der Satz 3.6 ermöglicht es uns also eine Ausage über $\sqsubseteq_A$, $\sqsubseteq_P$ und $\sqsubseteq_W$. Gilt nämlich $\neg(B1 \sqsubseteq_G B2)$, so gilt mit Sicherheit $\neg(B1 \sqsubseteq_A B2)$, $\neg(B1 \sqsubseteq_P B2)$ und $\neg(B1 \sqsubseteq_W B2)$. Wir werden daher im folgenden die Gültigkeit einiger Gesetze zeigen, die uns bei der Bestimmung des Parallelitätsgrades erleichtern.

Lemma 3.9 *Sei $\mathcal{E}_i$ e.b.e.s. mit $i = 1, 2$ und $\mathcal{E} = \mathcal{E}_1 \overline{|||} \mathcal{E}_2$. Dann gilt*

1. $\mathcal{E}_1 \overset{\overline{K_1}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1' \Longrightarrow \mathcal{E} \overset{\overline{K}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1' \overline{|||} \mathcal{E}_2$ *mit* $K = \{(e, *) \mid e \in K_1\}$
2. $\mathcal{E}_2 \overset{\overline{K_2}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_2' \Longrightarrow \mathcal{E} \overset{\overline{K}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1 \overline{|||} \mathcal{E}_2'$ *mit* $K = \{(*, e) \mid e \in K_2\}$
3. $\mathcal{E} \overset{\overline{K}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}' \Longrightarrow \exists K_1, K_2 : (K_1 \cap K_2 = \emptyset \wedge K = \{(e, *) \mid e \in K_1\} \cup \{(*, e) \mid e \in K_2\}$
   $\wedge\ \mathcal{E}_i \overset{\overline{K_i}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_i'$ *mit* $i = 1, 2 \wedge\ \mathcal{E}' = \mathcal{E}_1' \overline{|||} \mathcal{E}_2')$

Beweis: Im folgenden beweisen wir nur den Fall 3, da der Beweis für Fall 1. und 2. einfach ist und daher nicht ausgeführt wird.

Sei $K_1 = \{e \,|(e,*) \in K\}$ und $K_2 = K \setminus K_1$, dann bleibt es nur noch zu zeigen, daß $\mathcal{E}_i \overset{\overline{K_i}}{\Leftrightarrow} \mathcal{E}_i'$ mit $i = 1,2$ gilt. Man beachte, daß $K_2 = \{e \,|(*,e) \in K\}$ gilt, denn die Elemente in K sind vom Typ $(e,*)$ oder $(*,e)$. Wie in [Lan92] führen wir nun zwei Funktionen $\pi_1(\sigma)$ und $\pi_1(\sigma)$ ein, die auf einer Ausführungsfolge $\sigma = e_1 \ldots e_n \in PS(\mathcal{E})$ wie folgt definiert sind:

- $\pi_i(\epsilon) = \epsilon$ mit $i = 1,2$
- $\pi_1((x,y)\sigma') = \begin{cases} x\pi_1(\sigma') & \text{falls} \quad y = * \\ \pi_1(\sigma') & \text{falls} \quad \text{sonst} \end{cases}$
- $\pi_2((x,y)\sigma') = \begin{cases} y\pi_2(\sigma') & \text{falls} \quad x = * \\ \pi_2(\sigma') & \text{falls} \quad \text{sonst} \end{cases}$

Sei ferner $\widehat{\sigma} := \{e_1, \ldots, e_n\}$ mit $\sigma \in PS(\mathcal{E})$. Dann gilt offenbar $K_1 = \widehat{\pi_2(\sigma)}$ und $K_2 = \widehat{\pi_2(\sigma)}$, falls σ die Ausführungsfolge von K ist. Da nach [Lan92] (S. 124) $\pi_i(\sigma)$ eine Ausführungsfolge von K_i für $i = 1,2$ ist (Die Gültigkeit dieser Behauptung ist in der Tat leicht zu prüfen.), gilt $\mathcal{E}_i \overset{\overline{K_i}}{\Leftrightarrow} \mathcal{E}_i'$ mit $i = 1,2$. Nach Fall 1. und 2. gilt dann

$$\mathcal{E} \overset{\overline{H_1}}{\Leftrightarrow} \mathcal{E}_1' \overline{|||} \mathcal{E}_2 \overset{\overline{H_2}}{\Leftrightarrow} \mathcal{E}_1' \overline{|||} \mathcal{E}_2'$$

mit $H_1 = \{(e,*) \,|\, e \in K_1\}$ und $H_2 = \{(*,e) \,|\, e \in K_2\}$. Da $K = H_1 \cup H_2$ gilt, gilt nach Theorem 2.2 $\mathcal{E} \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}_1' \overline{|||} \mathcal{E}_2'$. □

Satz 3.7

1. $G(\mathbf{stop}) = 0$ *und* $G(\mathbf{exit}) = 1$
2. $G(g;B) = max(1, G(B))$
3. $G(B_1 \,[]\, B_2) = max(G(B_1), G(B_2))$
4. $G(B_1 \,|||\, B_2) = G(B_1) + G(B_2)$

Beweis:

1. trivial.
2. Sei $\mathcal{E} = [\![g;B]\!]$ und $\mathcal{E}_1 = [\![B]\!]$. Für den Fall $G(B) = 0$ gilt offenbar $\mathcal{E}' = \mathcal{E}$ und $|K| = 1$, wenn $\mathcal{E}' \in Er(\mathcal{E})$ und $\mathcal{E}' \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}''$ mit $K \neq \emptyset$ gilt. Daraus folgt $G(g;B) = 1$. Gilt $G(B) > 0$, dann gilt $G(g;B) = G(B)$. Denn: Daß $G(g;B) < G(B)$ nicht sein kann, ist offensichtlich. Wäre $G(g;B) > G(B)$, so gäbe es ein $\overline{K}$ mit $|K| > G(B)$, so daß $\exists \mathcal{E}' \in Er(\mathcal{E}) : \mathcal{E}' \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}''$ gilt, wobei $\mathcal{E}' \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}''$ eine Schritt-Transition ist. Da aus $\mathcal{E}' \neq \mathcal{E}$ auch $\mathcal{E}' \in Er(\mathcal{E}_1)$ folgt, gilt zunächst $\mathcal{E}' = \mathcal{E}$. Sei nun $e_1, \ldots, e_n$ eine Ausführungsfolge von K, dann gilt nach Korollar 3.1:

   $$\mathcal{E} \overset{\overline{\{e_1\}}}{\Leftrightarrow} \mathcal{E}_1 \ldots \mathcal{E}_{n-1} \overset{\overline{\{e_n\}}}{\Leftrightarrow} \mathcal{E}_n = \mathcal{E}''$$

   Nach Lemma 3.3 gilt $e_1 \in init(\mathcal{E})$ und nach Definition 2.27 $l(e_1) = g$ und $\{e_1\} \mapsto init(\mathcal{E}_1)$. Da nach Lemma 3.3 $e_2 \in init(\mathcal{E}_1)$ gilt, gilt $e_1 \leq_K e_2$. Also ist $\mathcal{E} \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}''$ keine Schritt-Transition, wenn K mehr als zwei Ereignisse enthält. Daraus folgt $K = \{e_1\}$ und somit $|K| \not> G(B)$. Widerspruch.

3. Es ist zu zeigen, daß $ST(B_1 [] B_2) = ST(B_1) \cup ST(B_2)$ gilt. Sei $\mathcal{E} = [\![B_1 [] B_2]\!]$, $\mathcal{E}_1 = [\![B_1]\!]$ und $\mathcal{E}_2 = [\![B_2]\!]$. Dann gilt zunächst nach [Lan92] (S. 123) $PS(\mathcal{E}) = PS(\mathcal{E}_1) \cup PS(\mathcal{E}_2)$ (Die Gültigkeit dieser Aussage ist leicht zu prüfen). Damit gilt:

   (a) „$\Rightarrow$“:

$$\begin{aligned}
& K \in ST(B_1 [] B_2) \\
\Rightarrow\ & \exists \mathcal{E}' \in Er(\mathcal{E}) : \mathcal{E}' \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}'' \\
\Rightarrow\ & \exists \overline{H} : \mathcal{E} \overset{\overline{H}}{\Leftrightarrow} \mathcal{E}' \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}'' \\
\Rightarrow\ & \mathcal{E} \overset{\overline{H \cup K}}{\Leftrightarrow} \mathcal{E}'' \text{ nach Theorem 2.2} \\
& \quad \text{und } \overline{H} \text{ ist ein Präfix von } \overline{H \cup K} \\
\Rightarrow\ & \exists i \in \{1,2\} : \mathcal{E}_i \overset{\overline{H}}{\Leftrightarrow}_i \mathcal{E}_i' \text{ und } \mathcal{E}_i \overset{\overline{H \cup K}}{\Leftrightarrow}_i \mathcal{E}_i''
\end{aligned}$$

   $\overline{H}$ ist ein Präfix von $\overline{H \cup K}$ in $\mathcal{E}_i$. Denn: Wenn $\overline{H}$ kein Präfix ist, dann gilt

$$\leq_H \neq \leq_{H \cup K} \cap ((H \cup K) \times H)$$

   in $\mathcal{E}_i$. D.h. es gibt ein $e \in K$ und $e' \in H$, so daß $e \leq_{H \cup K} e'$ in $\mathcal{E}_i$ gilt. Dies bedeutet $\exists X : X \mapsto_i e'$ und $e \in X$ oder $e \rightsquigarrow_i e'$. Diese Aussage gilt aber nach Definition 2.27 auch für $\mathcal{E}$. Also ist $\overline{H}$ kein Präfix von $\overline{H \cup K}$ in $\mathcal{E}$. Widerspruch.

   Somit gilt $\mathcal{E}_i' \overset{\overline{K}}{\Leftrightarrow}_i \mathcal{E}_i''$. Da $\leq_K$ eine Identitätsrelation in $\mathcal{E}$ ist, ist $\leq_K$ auch eine Identitätsrelation in $\mathcal{E}_i$. Denn: Angenommen, es gilt $e \leq_K e'$ in $\mathcal{E}_i$ mit $e \neq e'$ und $e, e' \in K$. Dann gilt $e \in X$ mit $X \mapsto_i e'$ oder $e \rightsquigarrow_i e'$. Nach Definition 2.27 gilt aber auch $e \in X$ mit $X \mapsto e'$ oder $e \rightsquigarrow e'$ in $\mathcal{E}$, was $e \leq_K e'$ in $\mathcal{E}$ bedeutet. Also ist $\mathcal{E}_i' \overset{\overline{K}}{\Leftrightarrow}_i \mathcal{E}_i''$ eine Schritt-Transition und daher $K \in ST(B_i)$.

   (b) „$\Leftarrow$“: Analog zu (a).

4. Wir konstruieren zunächst eine Konfiguration K mit $|K| = G(B_1) + G(B_2)$, so daß $\mathcal{E}' \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}''$ eine Schritt-Transition ist, wobei $\mathcal{E}' \in Er(\mathcal{E})$ gilt. Sei $K_i \in ST(B_i)$ mit $|K_i| = G(B_i)$ und $\mathcal{E}_i' \overset{\overline{K_i}}{\Leftrightarrow} \mathcal{E}_i''$ mit $\mathcal{E}_i' \in Er(\mathcal{E}_i)$, wobei $i = 1, 2$ gilt. Sei weiterhin H_i eine Konfiguration mit $\mathcal{E}_i \overset{\overline{H_i}}{\Leftrightarrow} \mathcal{E}_i'$ und $i = 1, 2$. Dann gilt nach Lemma 3.9 und Theorem 2.2

$$\mathcal{E} \overset{\overline{H}}{\Leftrightarrow} \mathcal{E}_1' \overline{|||} \mathcal{E}_2' \overset{\overline{L}}{\Leftrightarrow} \mathcal{E}_1'' \overline{|||} \mathcal{E}_2''$$

   mit $H = \{(e, *) \mid e \in H_1\} \cup \{(*, e) \mid e \in H_2\}$ und $L = \{(e, *) \mid e \in K_1\} \cup \{(e, *) \mid e \in K_2\}$. Sei nun $K = L$. Da $\leq_{K_i}$ in $\mathcal{E}_i$ mit $i = 1, 2$ eine Identitätsrelation ist, ist auch $\leq_K$ in $\mathcal{E}$ eine Identitätsrelation. Denn:

   Angenommen, es gilt $e \leq_K e'$ in $\mathcal{E}$ mit $e \neq e'$. Dann gilt $e = (\xi, *)$ und $e' = (\psi, *)$ mit $\xi, \psi \in K_1$ oder $e = (*, \xi)$ und $e' = (*, \psi)$ mit $\xi, \psi \in K_2$. O.E.d.A. gilt $(\xi, *) \leq_K (\psi, *)$. Nach Definition 2.13 gilt $(\xi, *) \in X$ mit $X \mapsto (\psi, *)$ in $\mathcal{E}$, was $Y \mapsto \psi$ mit $Y = \{e \,|\, (e, *) \in X\}$ und $\xi \in Y$ in $\mathcal{E}_1$ bedeutet, oder $(\xi, *) \rightsquigarrow (\psi, *)$, was $\xi \rightsquigarrow \psi$ in $\mathcal{E}_1$ impliziert. Daraus folgt, daß auch $\xi \leq_{K_1} \psi$ in $\mathcal{E}_1$ gilt. Widerspruch.

   Es bleibt noch zu zeigen, daß es kein $K \in ST(B_1 ||| B_2)$ mit $|K| > G(B_1) + G(B_2)$ gibt. Angenommen, es gibt so ein K und ein $\mathcal{E}' \in Er(\mathcal{E})$ mit $\mathcal{E}' \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}''$, also

$$\exists H : \mathcal{E} \overset{\overline{H}}{\Leftrightarrow} \mathcal{E}' \overset{\overline{K}}{\Leftrightarrow} \mathcal{E}''$$

Dann gilt nach Lemma 3.9 $\mathcal{E}' = \mathcal{E}_1' \overline{|||} \mathcal{E}_2'$, $\mathcal{E}'' = \mathcal{E}_1'' \overline{|||} \mathcal{E}_2''$, $\mathcal{E}_1' \overset{\overline{K_1}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1''$ und $\mathcal{E}_2' \overset{\overline{K_2}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_2''$, wobei $K = \{(e,*) \mid e \in K_1\} \cup \{(*,e) \mid e \in K_2\}$ gilt. Mit der gleichen Begründung wie oben sind dann $\leq_{K_1}$ in $\mathcal{E}_1$ und $\leq_{K_2}$ in $\mathcal{E}_2$ Identiätsrelationen, d.h. es gilt $K_1 \in ST(B_1)$ und $K_2 \in ST(B_2)$. Da aber $|K| > G(B_1) + G(B_2)$ und $K_1 \cap K_2 = \emptyset$ gilt, gilt $|K_1| > G(B_1)$ oder $|K_2| > G(B_2)$. Widerspruch. □

Mit Satz 3.7 gilt z.B. $G(a; b \,[]\, b; a) = 1 \leq G(a \,|||\, b) = 2$, womit $a \,|||\, b \not\sqsubseteq_A a; b \,[]\, b; a$ (bzw. $a \,|||\, b \not\sqsubseteq_P a; b \,[]\, b; a$ und $a \,|||\, b \not\sqsubseteq_W a; b \,[]\, b; a$) gilt.

3.4 Eingeschränkte Basic LOTOS ohne Rekursion

In diesem Abschnitt wird eine Teilsprache von $\mathcal{BL}$ ohne Rekursion betrachtet, für die gezeigt wird, daß der maximale Parallelitätsgrad sich abschätzen läßt.

Definition 3.18 ($\mathcal{BL}^e$, eingeschränkte Basic LOTOS) *Sei $g \in \mathcal{G}$ und $g_i \in \mathcal{G}$ mit $1 \leq i \leq n$. Dann ist $\mathcal{BL}^e$ durch folgende Grammatik*

$$B ::= \mathbf{stop} \quad | \quad g; B \quad | \quad B \,[]\, B \quad | \quad B \,|[g_1, \ldots, g_n]|\, B$$

definiert. □

Lemma 3.10 *Sei $B \in \mathcal{BL}^e$ und $[\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$, dann ist $|E|$ endlich.*

Beweis: Induktion über den Termaufbau. □

Im folgenden zeigen wir, daß $G_{max}(B) \leq |E|$ gilt.

Definition 3.19 ($G_{max}(g,B)$) *Sei $B, C \in \mathcal{BL}^e$ mit $B \sim C$ und $G_{max}(B) = G(C)$, $g \in Act(B)$ und $M = \{|K_g| \mid K_g \in ST(g,C)\}$. Dann ist $G_{max}(g,B) := max(M)$ der maximale Parallelitätsgrad einer Aktion g in B* □

Satz 3.8 *Sei $B \in \mathcal{BL}^e$, $[\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$, $g \in Act(B)$ und $E_g = \{e \mid e \in E \wedge l(e) = g\}$, dann gilt $G_{max}(g,B) \leq |E_g|$.*

Beweis: Angenommen, es gilt $G_{max}(g,B) \geq |E_g| + 1$. Dann gibt es nach Definition 3.19 ein $C \in \mathcal{BL}$ mit $B \sim C$, $G_{max}(B) = G(C)$ und $G_{max}(g,B) = |K_g|$, wobei $K_g \in ST(g,C)$. Nach Definition 3.14 gibt es also ein $K \in ST(C)$, so daß $K_g \subseteq K$ und $\mathcal{E}' \overset{\overline{K}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}'''$ mit $\mathcal{E}', \mathcal{E}''' \in Er(\mathcal{E}_c)$ und $\mathcal{E}_c = [\![C]\!] = (E_c, \rightsquigarrow_c, \mapsto_c, l_c)$ gilt. Da $\mathcal{E}' \overset{\overline{K}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}'''$ eine Schritt-Transition, gilt natürlich auch $\mathcal{E}' \overset{\overline{K_g}}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}''$ mit $\mathcal{E}'' \in Er(\mathcal{E}_c)$ und somit nach Korollar 3.1

$$\mathcal{E}' \overset{\overline{\{e_1\}}}{\Leftrightarrow\!\!\!\rightarrow}_l \mathcal{E}_1 \overset{\overline{\{e_2\}}}{\Leftrightarrow\!\!\!\rightarrow}_l \cdots \mathcal{E}_{n-1} \overset{\overline{\{e_n\}}}{\Leftrightarrow\!\!\!\rightarrow}_l \mathcal{E}_n = \mathcal{E}'',$$

wobei $K_g = \{e_1, \ldots, e_n\}$, $\mathcal{E}_i \in Er(\mathcal{E}_c)$ mit $i = 1, \ldots, n \Leftrightarrow 1$ und $n \geq |E_g| + 1$.

Da $\mathcal{T}(\mathcal{E}_c) \sim_s \mathcal{T}([\![B]\!])$ gilt, gibt es eine Bisimulation R und $\mathcal{E}_j^B \in Er([\![B]\!])$ mit $j = 0, \ldots, n$, so daß

$$\mathcal{E}_0^B \overset{l(\psi_1)}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1^B \overset{l(\psi_2)}{\Leftrightarrow\!\!\!\rightarrow} \cdots \mathcal{E}_{n-1}^B \overset{l(\psi_n)}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_n^B,$$

$(\mathcal{E}', \mathcal{E}_0^B) \in R, (\mathcal{E}_i, \mathcal{E}_i^B) \in R$ und $l(\psi_i) = g$ mit $i = 1, \ldots, n$ gilt. Dies ist ein Widerspruch, da sonst $|E_g| > |E_g|$ gilt. □

Korollar 3.5 *Sei $B \in \mathcal{BL}^e$ und $[\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$, dann gilt $G_{max}(B) \leq |E|$.*

Beweis: Nach Definition 3.16 gibt es ein $C \in \mathcal{BL}^e$ mit $[\![C]\!] = (E, \rightsquigarrow, \mapsto, l)$ und $G_{max}(B) = G(C)$, d.h. $\exists K \in ST(C)$ mit $G(C) = |K|$. Da

$$K = \bigcup_{g \in Act(B)} (K \cap \{e \,|\, l(e) = g\})$$

gilt und offenbar $K_i = K \cap \{e \,|\, l(e) = g_i\}$ mit $g_i \in Act(B)$ paarweise disjunkt sind, folgt

$$\begin{aligned} |K| &= \sum_{g \in Act(B)} |K \cap \{e \,|\, l(e) = g\}| \\ &\leq \sum_{g \in Act(B)} |E_g| \text{ nach Satz 3.8} \\ &= |E| \end{aligned}$$

□

Betrachtet man z.B. einen Prozeß $B1 = a; (b; (c \,[]\, d) \,[]\, c; b \,[]\, d; b) \,[]\, c; a; b \,[]\, d; a; b$, dann ist $G_{max}(B1) \leq 14$. Da aber $B2 \sim B1$ mit $B2 = a; b \,|||\, (c \,[]\, d)$ gilt, ist $G_{max}(B1) = 4$. Aus dem obigen Korollar kann man z.B. auch folgern, daß ein Prozeß wie $B = a \,|||\, a \,|||\, b$ ein Prozeß mit einem maximalen Parallelitätsgrad ist, da $G(B) = 3 = G_{max}(B)$ gilt.

Wie aus dem Prozeß $B1$ zu erkennen ist, ist die Abschätzung des Parallelitätsgrades oft ungenau. Eine gröbere Abschätzung kann dadurch gegeben werden, daß man zunächst auf der Menge der erreichbaren Zustände in einem Transitionssystem die Äquivalenzklassen bezüglich $\sim$ bildet und anschließend die Anzahl der Übergänge (definiert auf dieser Klassen) bestimmt. Diese Anzahl ist dann der abgeschätzte maximale Parallelitätsgrad. Wir wollen im folgenden dazu einen Satz zitieren, wofür wir aufgrund der Analogie zum Satz 3.8 und Korollar 3.5 keinen Beweis angeben.

Definition 3.20 (Notationen) *Sei $B \in \mathcal{BL}^e$, $g \in Act(B)$, $C, C' \in Er(B)$ und $\sim_B := \sim|_{Er(B)}$. Wir definieren*

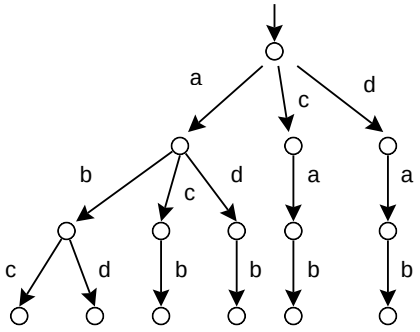
- $[C]_{/\sim_B} \overset{g}{\nrightarrow} [C']_{/\sim_B}$, *wenn* $C \overset{g}{\nrightarrow} C'$ *gilt.*
- $T(g, B) := \{[C]_{/\sim_B} \overset{g}{\nrightarrow} [C']_{/\sim_B} \mid \; C, C' \in Er(B)\}$
- $E_{min}(g, B) := \{(g, i) \,|\, i = 1, \ldots, |T(g, B)|\}$
- $E_{min}(B) := \bigcup_{g \in Act(B)} E_{min}(g, B)$ □

Satz 3.9 *Sei $B \in \mathcal{BL}^e$, $g \in Act(B)$, dann gilt $G_{max}(g, B) \leq |T(g, B)|$.*

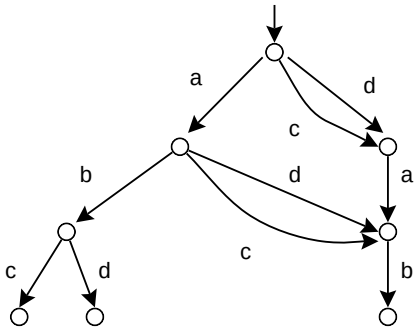
Beweis: Analog zum Satz 3.8. □

Korollar 3.6 *Sei $B \in \mathcal{BL}^e$, dann gilt $G_{max}(B) \leq |E_{min}(B)|$.* □

Betrachten wir nochmal den Prozeß $B1$, so hat $\mathcal{OS}(B1)$ folgende Struktur:



Bildet man die Äquivalenzklassen auf $\mathcal{OS}(B1)$ bezüglich $\sim$, so erhält man



Daraus ist $G_{max}(B1) \leq 10$.

3.5 Ausblick auf Parallelisierung

In diesem Abschnitt wird informell ein Ansatz zur Parallelisierung bezüglich $\sqsubseteq_W$ vorgestellt, deren Korrektheit in Rahmen dieses Berichtes nicht mehr beantwortet werden konnte. Es wird nur kurz erläutert, wie Prozesse mittels der Halbordnungssemantik parallelisiert werden können. Eine ausführliche Behandlung dieses Themas wird daher in einem weiteren Bericht zu finden sein.

Wir beschränken uns wieder auf die Sprache $\mathcal{BL}^e$. Das Parallelisierungsproblem lautet wie folgt:

- Gegeben: $B \in \mathcal{BL}^e$
- Gesucht ist ein Prozeß $C \in \mathcal{BL}^e$ mit $B \sqsubseteq_W C$, falls C existiert, so daß für alle $B' \in \mathcal{BL}^e$ gilt:

  $$B \sqsubseteq_W B' \Longrightarrow B' \sqsubseteq_W C$$

  Dabei wird gefordert, daß in C keine Synchronisation stattfindet, d.h. die Menge der Synchronsationsaktionen leer ist.

Der Ansatz zur Lösung dieses Problems basiert auf folgenden Aussagen (deren Beweis aufgrund der Einfachheit nicht angegeben wird):

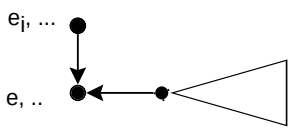
Sei $B \in \mathcal{BL}^e$, $\mathcal{E} = [\![B]\!] = (E, \rightsquigarrow, \mapsto, l)$ und $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ e.b.e.s. mit $\mathcal{E} \overset{\overline{\{e_i\}}}{\leftrightarrow} \mathcal{E}_i$ und $i = 1, \ldots, n$, wobei $init(\mathcal{E}) = \{e_1, \ldots, e_n\}$ gilt und in B keine Synchronisation vorkommt. Dann gilt

1. $E = \{e_i\} \cup E_i$ mit $i = 1, \ldots, n$.

2. In $\mathcal{E}$ gibt es kein Bündel, das mehr als zwei Elemente enthält. Z.B. $\{e_1, e_2, \ldots, e_n\} \mapsto e$ mit $n \geq 2$ ist nicht möglich.

3. Gilt $\{e_i\} \mapsto e$, so gilt $\{e_i\} \mapsto_j e$ mit $j \neq i$ und $j \in \{1, \ldots, n\}$.

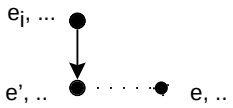
4. Gilt $\{e_i\} \mapsto e$, so gilt $e \in init(\mathcal{E}_i)$. D.h. der Fall wie



   kann nicht vorkommen.

5. Sei $e \in init(\mathcal{E}_i)$. e_i und e sind kausal unabhängig (d.h. $\neg(\{e_i\} \mapsto e \vee e \rightsquigarrow e_i \vee e_i \rightsquigarrow e)$), wenn es ein e_j mit $j \neq i$ gibt, so daß $e_j = e$ gilt.

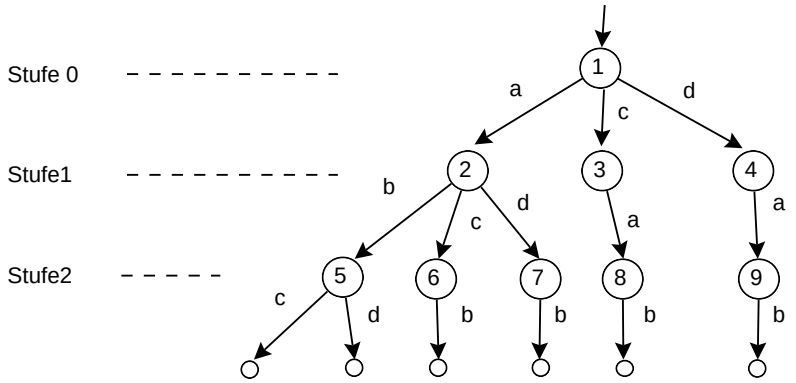
6. Sei $e, e' \in init(\mathcal{E}_i)$ und $e \rightsquigarrow e'$. Sind e_i und e kausal unabhängig, dann sind e_i und e' kausal unabhängig. D.h. der Fall wie



   kann nicht vorkommen.

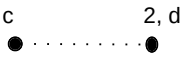
Wir werden nun die Aussagen 3., 4., 5. und 6. verwenden, um das Paralleliserungsproblem zu lösen. Der Lösungsansatz wird an einem Beispiel erläutert. Dazu wird das Beispiel aus [PHQ+92] genommen, in dem folgender Prozeß (siehe auch Abschnitt 3.4) betrachtet wird: $B = a; (b; (c \,[]\, d) \,[]\, c; b \,[]\, d; b) \,[]\, c; a; b \,[]\, d; a; b$

Wir bestimmen zunächst das Transitionssystem $\mathcal{OS}(B)$ und stellen es als ein Baum dar (D.h. ein Zustand kann dadurch mehrfach vorkommen.). Anschließend kennzeichnen wir die Zustände mit den natürlichen Zahlen. Dabei werden die Zustände, die als Blätter vorkommen, nicht gekennzeichnet. Wir bekommen daher die folgende Abbildung:



Betrachten wir die Zustände auf der Stufe 2, so kommen wir zu folgenden Prozessen, wobei TB_i den Teilprozeß angibt, der den Zustand i repräsentiert:

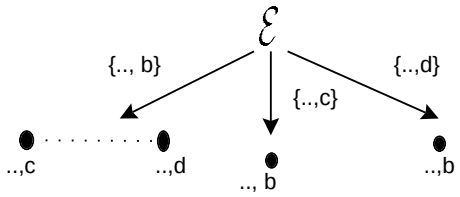
1. $TB_5 = c \,[]\, d$. c und d können nicht kausal unabhängig geordnet sein, da wir sonst einen Prozeß bekämen, der zu TB_5 nicht bisimulationsäquivalent ist. Wir haben also



2. Für Zustand $i = 6, 7, 8, 9$ gilt $TB_i = b$.

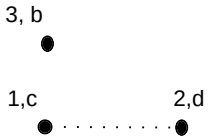
Die Prozesse für die Zustände auf der Stufe 1 lassen sich wie folgt bestimmen:

1. Zustand 2: Zunächst gilt für die Ereignisstruktur $\mathcal{E}$, die den Zustand 2 repräsentiert:

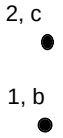


Man beachte, daß Ereignisse wie e mit $\emptyset \mapsto e$ in den Nachfolge-Ereignisstrukturen vorkommen können. Da monentan nicht bekannt ist, um welche Ereignisse es sich handelt, können diese Ereignisse nicht explizit gezeichnet werden. Sicher ist jedoch, daß $\mathcal{E}$ bis auf die Ereignisnamen und die Bündelmenge folgende Strukturen enthält:

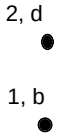
(a)



(b)



(c)



Ob bei a) $\{3\} \mapsto 1$ und $\{3\} \mapsto 2$, bei b) $\{2\} \mapsto 1$ und bei c) $\{2\} \mapsto 1$ gilt, ist noch ungeklärt. Es wäre optimal, wenn dies nicht gelten würde, denn dann ist die kausale Abhängigkeit auf Minimum reduziert, was wiederum die Parallelität erhöht.

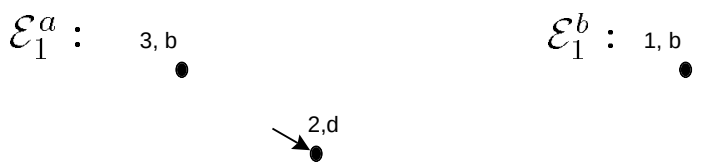
In Abhängigkeit dieser Strukturen konstruieren wir nun die e.b.e.s. $\mathcal{E}$ so, daß $\mathcal{E}$, wenn der Übergang mit $\overline{\{3\}}$ bei (a) und der Übergang mit $\overline{\{2\}}$ bei (b) bzw. bei (c) stattfindet, zu entsprechenden Nachfolge-Ereignisstrukturen führt. In $\mathcal{E}$ sollen möglichst wenig kausale Abhängigkeiten enthalten sein.

Es ist zunächst anzumerken, daß $\mathcal{E}$ im allgemeinen von der Form $\mathcal{E} = \mathcal{E}_1 \,[]\, \mathcal{E}_2 \,[]\, \cdots \,[]\, \mathcal{E}_n$ ist, wobei $\mathcal{E}_i = \mathcal{E}_i' \,|||\, \mathcal{E}_i''$ mit $i = 1, \ldots, n$ gilt. Die Nachfolge-Ereignisstrukturen können daher nur von $\mathcal{E}_i$ stammen. Es kann sogar vorkommen, daß zwei oder drei Nachfolge-Ereignisstrukturen von einer einzigen Ereignisstruktur $\mathcal{E}_i$ stammen. Aus diesen Gründen wird $\mathcal{E}$ im folgenden dadurch bestimmt, daß wir schrittweise die Ereignisstrukturen $\mathcal{E}_i$ bestimmen.

- Schritt 1:

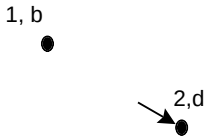
  Wir betrachten die Ereignisstruktur bei a) und prüfen, ob die Ereignisse 3 und 1 sowie 3 und 2 kausal unabhängig sind.

  Nach 5. sind 3 und 1 kausal unabhängig, wenn zunächst $Ew(\mathcal{E}_1^a) \cong Ew(\mathcal{E}_1^b)$ mit

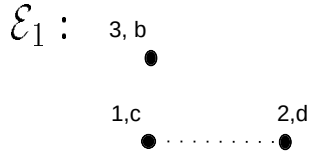


  gilt ($\mathcal{E}_1^b$ (bzw. $\mathcal{E}_1^a$) ist hier die Ereignisstruktur, die sich aus der Ereignisstruktur bei (b) (bzw. (a)) ergibt, wenn das Ereignis 2 (bzw. 1) stattfindet.). Dabei bezeichnet $Ew(\mathcal{E}_1^a)$ (bzw. $Ew(\mathcal{E}_1^b)$ die erweiterte Ereignisstruktur von $\mathcal{E}_1^a$ (bzw. $\mathcal{E}_1^b$). D.h. $\mathcal{E}_1^a$ und $\mathcal{E}_1^b$ müssen so erweitert (oder besser gesagt ergänzt) werden, so daß sie isomorph sind. Unter Erweiterung einer e.b.e.s $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ wird die Erweiterung von E um zusätzliche Ereignisse, von $\rightsquigarrow$ und $\mapsto$ um zusätzliche Relationenpaare sowie von l um die Kennzeichnung weiterer Ereignisse mit Aktionennamen verstanden.

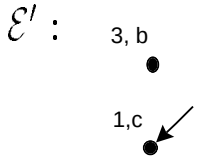
  Bei der Erweiterung soll jedoch folgendes gelten: Sei $Ev_a(g)$ die Menge der Ereignisse von $\mathcal{E}_1^a$, die eine Aktion g darstellen, und analog $Ev_b(g)$ die Menge der Ereignisse von $\mathcal{E}_1^b$. Dann dürfen sowohl in $Ew(\mathcal{E}_1^a)$ als auch in $Ew(\mathcal{E}_1^b)$ nicht mehr als $max(|Ev_a(g)|, |Ev_b(g)|)$ Ereignisse vorkommen, die eine Aktion g repräsentieren. Angewendet auf $\mathcal{E}_1^a$ und $\mathcal{E}_1^b$ gilt daher $Ew(\mathcal{E}_1^a) = \mathcal{E}_1^a$, und $Ew(\mathcal{E}_1^b)$ ist von folgender Form:



  Da $Ew(\mathcal{E}_1^a) \cong Ew(\mathcal{E}_1^b)$ offensichtlich gilt, kann man folgern, daß



  die e.b.e.s. ist, von der die Ereignisstrukturen bei a) und b) stammen könnten. Die Ereignisstrukturen bei (a) und (b) können von $\mathcal{E}_1$ nur dann stammen, wenn es einen Übergang mit d gibt, der in die Nachfolge-Ereignisstruktur endet ($2 \in init(\mathcal{E}_1)$):



  Da es einen Übergang mit d gibt, der in $\mathcal{E}'$ endet (siehe die Ereignisstruktur bei (c)), stammen die Ereignisstrukturen bei (a) und (b) von $\mathcal{E}_1$. Damit kann man dann folgern, daß die Ereignisse 3 und 1 kausal unabhängig sind.

  Da in $\mathcal{E}_1$ die Ereignisse 3 und 2 kausal unabhängig sind, ist $\mathcal{E}_1$ die gesuchte e.b.e.s.

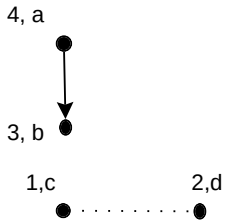
- Schritt 2: Analog zum Schritt 1 ist die gesuchte e.b.e.s. $\mathcal{E}_2$ für die Ereignisstruktur bei (b) identisch mit $\mathcal{E}_1$, d.h. $\mathcal{E}_2 = \mathcal{E}_1$.
- Schritt 3: Analog zum Schritt 1 ist die gesuchte e.b.e.s. $\mathcal{E}_3$ für die Ereignisstruktur bei (c) identisch mit $\mathcal{E}_1$, d.h. $\mathcal{E}_3 = \mathcal{E}_1$.

Daraus folgt, daß $\mathcal{E} = \mathcal{E}_1$ gilt.

Zusammengefaßt lautet die Idee zur Bestimmung von $\mathcal{E}$ wie folgt: Es werden Ereignisstrukturen mit Hife der Erweiterung der Nachfolge-Strukturen bestimmt. Auf der Menge dieser Ereignisstrukturen werden Äquivalenzklassen bezüglich $\cong$ gebildet. Die Repräsentanten dieser Klassen werden bezüglich $\overline{[]}$ zu $\mathcal{E}$ verknüpft.

2. Zustand 3 und 4: $TB_3 = TB_4 = a; b$

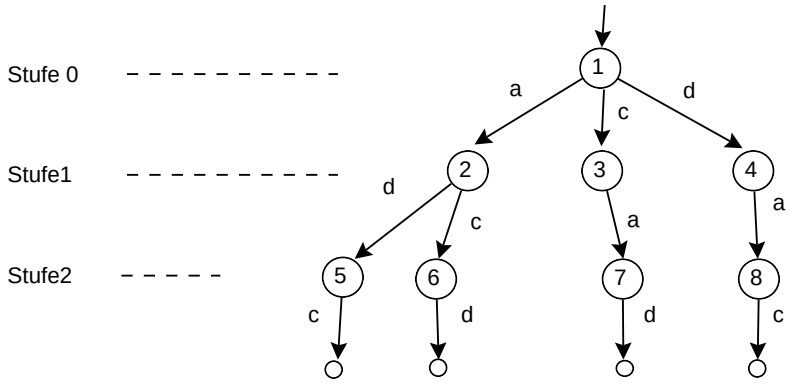
Mit der analogen Vorgehenweise ist die gesuchte e.b.e.s. für den Zustand 1 die folgende e.b.e.s.:



Dies bedeutet, daß $a; b \,|||\, (c \,[]\, d)$ der gesuchte Prozeß ist. Es gilt $B \sqsubseteq_W a; b \,|||\, (c \,[]\, d)$. Dieses Ergebnis stimmt mit dem in [PHQ+92] überein. In [PHQ+92] wird dieser Prozeß jedoch in Abhängigkeit von zwei vorgegebenen Aktionenmengen, die zwei Teilprozessen zugehören, bestimmt.

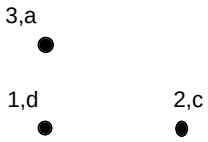
Zum besseren Verständnis des oben erläuterten Ansatzes wird im folgenden noch ein weiteres Beispiel ausführlich betrachtet.

Beispiel 3.7 Sei $B = ((a; d) \,|||\, c) \,[]\, d; a; c$, dann ist $\mathcal{OS}(B)$ das folgende Transitionssystem:

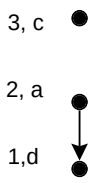


- Stufe 2: $TB_5 = TB_8 = c$, $TB_6 = TB_7 = d$
- Stufe 1: $TB_2 = d \,|||\, c$, $TB_3 = a; d$, $TB_4 = a; c$
- Stufe 0: Wir haben drei Fälle zu betrachten:

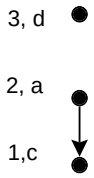
1.



2.

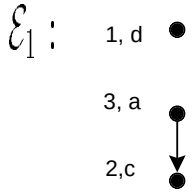


3.

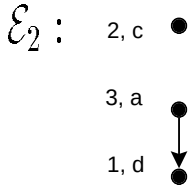


Schritt 1:

Wir betrachten die Ereignisstruktur bei 1. und prüfen, ob die Ereignisse 3 und 1 sowie 3 und 2 kausal unabhängig sind. Diese Ereignisse sind in der Tat kausal unabhängig, denn die Ereignisstruktur bei 1. stammt sowohl von



als auch von

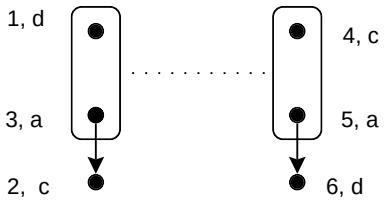


In $\mathcal{E}_1$ (bzw. $\mathcal{E}_2$) sind 3 und 1 (bzw. 3 und 2) kausal unabhängig.

Schritt 2: Die gesuchte e.b.e.s. für die Ereignisstruktur bei 2. ist $\mathcal{E}_2$.

Schritt 3: Die gesuchte e.b.e.s. für die Ereignisstruktur bei 3. ist $\mathcal{E}_1$.

Zusammengefaßt ergibt sich die gesuchte e.b.e.s. $\mathcal{E} = \mathcal{E}_1 \,\overline{[]}\, \mathcal{E}_2$ wie folgt:



Dies bedeutet, daß $C = (c \,|||\, (a;d)) \;\; [] \; (d \,|||\, (a;c))$ der gesuchte Prozeß ist. Es gilt $B \sqsubseteq_W C$. □

Beispiel 3.8 Sei $B = a;(b;b;a \,[]\, b;(b \,|||\, a)) \,[]\, b;(a;a;b \,[]\, a;(b \,|||\, a))$, dann ist der gesuchte Prozeß unter Anwendung des oben erläuterten Ansatzes $C = a;b \,|||\, b;a$. Für $B = a;a;a$ ist der gesuchte Prozeß $C = a \,|||\, a \,|||\, a$. □

Fazit: Der in diesem Abschnitt vorgestellte Ansatz zur Parallelisierung beschreibt nur eine Möglichkeit, wie Prozesse mittels der Halbordnungssemantik parallelisiert werden können. Er stellt jedoch noch keinen anwendbaren Algorithmus dar, denn es ist, wie aus obiger Erläuterung hervorgeht, nicht erkennbar, ob tatsächlich

$$\forall B' : (B \sqsubseteq_W B' \Longrightarrow B' \sqsubseteq_W C)$$

gilt, wenn C der Prozeß ist, der sich aus der Paralleliserung ergibt. Der Versuch, die Korrekheit dieses Ansatzes nachzuweisen (oder zu widerlegen), wird daher der Gegenstand der zukünftigen Untersuchung sein.

4 Zusammenfassung

In diesem Bericht werden folgende Resultate erzielt:

- Es werden Parallelitätsbegriffe basierend auf der Halbordnungssemantik für Basic LOTOS eingeführt und diskutiert. Damit können Prozesse differenziert werden, die sich in ihrer Ausführung bezüglich der Parallelität unterscheiden.
- Bei der Einführung der Parallelitätsbegriffe wird gezeigt, daß die Bisimulationsäquivalenz, die auf der operationellen Semantik definiert ist, sich mittels der Halbordnungssemantik ausdrücken läßt. Damit kann die operationelle Semantik als ein Speziallfall der Halbordnungssemantik betrachtet werden.
- Es wird ein Ansatz zur Parallelisierung vorgestellt, deren Korrektheit im Rahmen dieses Berichtes nicht mehr beantwortet werden konnte. Damit wird jedoch demonstriert, wie Prozesse sich mittels der Halbordnungssemantik parallelisieren lassen.

Zukünftigen Arbeiten können folgende Untersuchungen gewidmet werden:

- Neben dem Korrektheitsnachweis des in diesem Bericht vorgestellten Ansatzes zur Paralleliserung können weitere Methoden untersucht werden, die der Parallelisierung bezüglich der restlichen Parallelitätsbegriffe dienen.
- Bis jetzt wurde Parallelisierung unabhängig von den Hardware-Gegebenheiten diskutiert. Es wäre daher sehr interessant, Parallelisierung auf einem Rechnersystem zu untersuchen, das nur einen bestimmten Parallelitätsgrad zuläßt, d.h. nur eine maximale Anzahl der gleichzeitig ausführbaren Prozesse erlaubt.
- Die bis jetzt vernachlässigte Synchronisation kann auch noch berücksichtigt werden. Darüber hinaus kann eine architektur-abhängige Paralleliserung untersucht werden. Z.B. die Synchronisation ist nur zwischen bestimmten Prozessen erlaubt.

- Es wäre im Hinblick auf die Leistungsanalyse sehr interessant zu beobachten, welche Bedeutungen die Parallelitätsbegriffe haben, wenn jeder Aktion eine bestimmte Zeitdauer zugeordnet wird (siehe z.B. [MT91], [GHR93],[HR94], [QdFA93], [RGS93] und [RB91]).

Zum Schluß möchte ich mich bei Herrn Prof. Dr. Stetter und bei Torsten Vogt für die Lesekorrektur bedanken. Bedanken möchte ich mich auch bei Frau Prof. Dr. Majster-Cederbaum und Dr. Christel Baier für die wertvollen Literaturhinweise und bei Markus Roggenbach für die hilfreichen Diskussionen.

Literatur

[Ace91] L. Aceto. On relating concurrency and nondeterminism. In *LNCS 598*, pages 376–402. Springer-Verlag, 1991.

[BB87] T. Bolognesi and E. Brinksma. Introduction to the ISO Specification Language LOTOS. In *Computer Networks and ISDN Systems 14*, pages 25–59. Elsevier Science Publishers B.V. North-Holand, 1987.

[BC87] G. Boudol and I. Castellani. On the semantics of concurrency: partial orders and transition systems. In *LNCS 249*, pages 123–137. Springer-Verlag, 1987.

[BC88] G. Boudol and I. Castellani. Concurrency and atomicity. In *Theoretical Computer Science 59*, pages 25–84. North-Holland, 1988.

[BL95] E. Brinksma and R. Langerak. Functionality decomposition by compositional correctness preserving transformation. Memoranda informatica 95-13, TIOS 95-07, University of Twente, 1995.

[Bol92] T. Bolognesi. *Catalogue of LOTOS Correctness Preserving Transformations.* Third deliverable of Task 1.2 of project ESPRIT 2304, the LOTOSPHERE consortium, 1992.

[Bri88] E. Brinksma. A theory for the derivation of tests. In S. Aggarwal and K. Sabnani, editors, *Protocol Specification, Testing, and Verification VIII.* Elsevier Science Publishers B.V. North-Holand, 1988.

[BvdLV95] T. Bolognesi, J. van de Lagemaat, and C. Visser. *LOTOSphere: Software Development with LOTOS.* Kluwer Academic Publisher, 1995.

[CPS93] R. Cleaveland, J. Parrow, and B. Steffen. The Concurrency Workbench: A Semantics-Based Tool for the Verification of concurrent Systems. *ACM Transactions on Programming Languages and Systems*, 14(1):36–72, January 1993.

[dFO91] D. de Frutos and Y. Ortega. Step Semantics for LOTOS. ESPRIT 2304, T1.2, the LOTOSPHERE consortium, 1991.

[Do93] H. T. Do. *Entwurf von Prozeßsprachen zur Leistungsbewertung.* Diplomarbeit, Universität Erlangen-Nürnberg, 1993.

[GH94] S. Gilmore and J. Hillston. The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. In G. Harring and G. Kotkis, editors, *LNCS 794, Computer Performance Evaluation: Modelling Technique and Tools*, pages 353–368. Springer Verlag, 1994.

[GHR93] N. Götz, U. Herzog, and M. Rettelbach. Multiprocessor and Distributed System Design: The Integration of Functional Specification and Performance Analysis Using Stochastic Process Algebras. In *Arbeitsberichte des Institus für Mathematische Maschinen und Datenverarbeitung (Informatik), Workshop on Formalisms, Principles and State-Of-The-Art,* Band 26, Nummer 14, Universität Erlangen-Nürnberg, 1993.

[GL91] R. Gerber and I. Lee. Specification and Analysis of Resources-Bound Real-Time Systems. *LNCS 600, Real-Time: Theorie in Practice*, pages 571–596, 1991.

[Her93] H. Hermanns. *Semantik von Prozeßsprachen zur Leistungsbewertung.* Diplomarbeit, Universität Erlangen-Nürnberg, 1993.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall International, 1985.

[Hog89] D. Hogrefe. *ESTELLE, LOTOS und SDL.* Springer-Verlag, 1989.

[HR94] U. Herzog and M. Rettelbach. Proceedings of the 2nd workshop on process algebras and performance modelling. In *Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung (Informatik), Workshop on Formalisms, Principles and State-Of-The-Art,* Band 27, Nummer 4, Universität Erlangen-Nürnberg, 1994.

[ISO89] ISO/BS. *ISO 8807 - Information processing systems - Open Systems Interconnection - LOTOS- A formal description technique based on the temporal ordering of observational behaviour.* BSI, 1989.

[Jan85] R. Janicki. Transforming sequential systems into concurrent systems. In *Theoretical Computer Science 36*, pages 27–58. North-Holand, 1985.

[KBK89] F. Khendek, G.v. Bochmann, and C. Kant. New results on deriving protocol specifications from service specifications. *SIGGOM '89 Symposium Communications Architectures & Protocols, Computer Communications Review*, 19(4), September 1989.

[Lan92] R. Langerak. *Transformation and Semantics for LOTOS.* Dissertation, University of Twente, November 1992.

[LG91] R. Loogen and U. Goltz. Modelling Nondeterministic Concurrent Processes with Event Structures. *Fundamenta Informaticae*, 14(IV):39–73, 1991.

[Maz89] A. Mazurkiewicz. Basic notions of trace theory. In *LNCS 354*, pages 285–363. Springer-Verlag, 1989.

[Mil89] R. Milner. *Communication and Concurrency.* Prentice Hall International, 1989.

[MT91] F. Moller and C. Tofts. Relating processes with respect to speed. ECS-LFCS-91-143, Department of computer science, University of Edinburgh, 1991.

[Mü87] H. Müller. Diskrete Algebraische Strukturen. In *Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung (Informatik)*, Band 20, Nummer 5, Universität Erlangen-Nürnberg, 1987.

[NS91] X. Nicollin and J. Sifakis. An Overview and Synthesis on Timed Process Algebras. *LNCS 600, Real-Time: Theorie in Practice*, pages 526–548, 1991.

[Par89] J. Parrow. Submodule construction as equation solving in CCS. In *Theoretical Computer Science 68*, pages 175–202. Elsevier Science Publishers B.V. North-Holand, 1989.

[PHQ+92] S. Pavón, M. Hultström, J. Quemada, D. Frutos, and Y. Ortega. Inverse Expansion. In *Formal Description Technique IV*, pages 297–312. Elsevier Science Publishers B.V. North-Holand, 1992.

[QdFA93] J. Quemada, D. de Frutos, and A. Azcorra. TIC: A TImed Calculus. In *Formal Aspects of Computing 5*, pages 224–252. BCS, 1993.

[RB91] N. Rico and G. Bochmann. Performance description and analysis for distributed systems using a variant of LOTOS. In *Protocol Specification, Testing, and Verification XI*, pages 199–213. Elsevier Science Publishers B. V. North-Holland, 1991.

[Rei85] W. Reisig. *Systementwurf mit Netzen*. Springer Verlag, 1985.

[Rei86] W. Reisig. *Petrinetze, eine Einführung - zweite, überarbeitete und erweiterte Auflage*. Springer Verlag, 1986.

[Rei87] W. Reisig. *Das Verhalten verteilter Systeme*. GMD-Bericht Nr. 170. R. Oldenburg Verlag, Gesellschaft für Mathematik und Datenverarbeitung MBH, 1987.

[RGS93] M. Roccetti R. Gorrieri and E. Standcampiano. A Theory of Processes with Durational Actions. Technical Report UBLCS-93-27, Laboratory for Computer Science, University of Bologna, December 1993.

[Tof90] C. Tofts. Timed Concurrent Processes. In M. Z. Kwiatkowska, M. W. Schields, and R. M. Thomas, editors, *Workshops in Computing: Semantics for Concurrency*, pages 281–294. Springer Verlag, July 1990.

[VSSB91] C.A. Vissers, G. Scollo, M. Sinderen, and E. Brinksma. Specification styles in distributed systems design and verification. In *Theoretical Computer Science 89*, pages 179–206. Elsevier Science Publishers B.V. North-Holand, 1991.

[Win89] G. Winskel. An introcution to event strutures. In *LNCS 354*, pages 364–397. Springer-Verlag, 1989.

[Win93] G. Winskel. *The Formal Semantics of Programming Languages, An Introdution*. The MIT Press, 1993.

[WN94] G. Winskel and M. Nielsen. Models for concurrency. BRICS RS-94-12, Department of Computer Science, University of Aarhus, 1994.