REIHE INFORMATIK
12/96

Maximale Zerlegung von parallelen Prozessen ohne Rekursion

Huu Trung Do
Universität Mannheim
Fakultät für Mathematik und Informatik
Seminargebäude A5
D-68131 Mannheim

Maximale Zerlegung von parallelen Prozessen ohne Rekursion

Universität Mannheim
Fakultät für Mathematik und Informatik
Lehrstuhl für Praktische Informatik I
Seminargebäude A5
E-mail: do@pi1.informatik.uni-mannheim.de
12. Juni 1996
Huu Trung Do

Zusammenfassung

In diesem Bericht wird eine Methode zur maximalen Zerlegung von parallelen Prozessen in (eingeschränktem) Basic LOTOS ohne Rekursion vorgestellt. Darunter versteht man die Transformation eines vorgegebenen Prozesses in einen äquivalenten Prozeß bestehend aus mehreren Teilprozessen, die unabhängig voneinander parallel ablaufen. Dabei ist die Anzahl dieser Teilprozesse maximal.

Im Vergleich zur Methode „Inverse Expansion“ aus [PHQ$^+$92] ist die Anzahl der Teilprozesse nicht auf zwei beschränkt, und es werden keine Aktionenmengen zweier Teilprozesse, die in [PHQ$^+$92] erforderlich sind, als Vorgabe benötigt. Außerdem sind die Aktionenmengen der Teilprozesse im allgemeinen nicht disjunkt, d.h. die Zerlegung von nichtdeterministischen Prozessen ist auch erlaubt.

Ferner wird im Zusammenhang mit der Zerlegungsmethode eine weitere Methode für eine Teilsprache Basic LOTOS vorgestellt, die die Rückführung von Semantik, in diesem Fall einer Ereignisstruktur, zur Syntax ermöglicht.

Inhaltsverzeichnis

1 Einführung

Beim Entwurf von verteilten Systemen kommt es oft vor, daß der Weg von Spezifikation zur Implementierung über mehrere Verfeinerungsschritte erfolgt, insbesondere beim Top-Down-Entwurf. Es ist daher wünschenswert, Werkzeuge oder Methoden zu besitzen, die den Entwickler in der Entwurfsphase im Hinblick auf Verfeinerungen oder Transformationen von Spezifikationen systematisch unterstützen.

In den letzten Jahren wurden viele solche Transformationsmethoden (engl.: correctness preserving transformations) für die formale Spezifikationssprache LOTOS bereits untersucht [Bol92]. Eine von diesen Methoden ist die sogenannte *Inverse Expansion* [PHQ+92], bei der ein sequentiell beschriebener, nichtrekursiver Prozeß (synonym Spezifikation) in zwei Teilprozesse zerlegt wird, die unabhängig bzw. synchron parallel ablaufen. Wie bei vielen Transformationsmethoden stehen der Ausgangsprozeß und der mit *Inverse Expansion* bestimmte Prozeß in einer sogenannten beobachtbaren Äquivalenz zueinander, die, anschaulich gesehen, die Gleichheit des beobachtbaren Verhaltens der Systeme wiedergibt.

Sei P der Ausgangsprozeß, so wird mit *Inverse Expansion* ein Prozeß Q bestimmt, der die Form $Q = Q_1 \,|||\, Q_2$ im nicht synchronisierenden Fall und $Q = Q_1 \,|[g_1, \ldots, g_n]|\, Q_2$ im synchronisierenden Fall enthält. Bei $|||$ werden Teilprozesse Q_1 und Q_2 parallel unabhängig voneinander ausgeführt. Bei $|[g_1, \ldots, g_n]|$ müssen Q_1 und Q_2 über die Aktionen $g_1, \ldots, g_n$ synchronisieren. Es gilt $P \sim Q$ im nicht synchronisierenden Fall und $P \approx Q$ im synchronisierenden Fall. Dabei bezeichnet $\sim$ die sogenannte Strong-Bisimultionsäquivalenz und $\approx$ die beobachtbare Äquivalenz. $\sim$ unterscheidet sich von $\approx$ dadurch, daß auch interne Aktionen (das sind Aktionen, die nicht beobachtbar sind) mitberücksichtigt werden.

In der Praxis kann *Inverse Expansion* dazu verwendet werden, um z.B. folgende Probleme zu lösen (siehe [PHQ+92]): Zerlegung von Spezifikationen im Bezug auf Systemsressourcen, Ableiten einer Protokollspezifikation von einer vorgegebenen Dienstspezifikation, Modularisierung, Zerlegung des globalen Tests in sogenannte *upper tests* und *lower tests*, usw.

Damit *Inverse Expansion* im nicht synchronisierenden Fall anwendbar ist, muß zunächst die Aktionenmenge von P, als $A(P)$ bezeichnet, in zwei disjunkte Teilmengen A_1 und A_2 zerlegt werden, d.h. $A(P) = A_1 \cup A_2$. Die Mengen A_1 und A_2 entsprechen dabei den Aktionenmengen von Q_1 und Q_2, falls Q_1 und Q_2 existieren. Ist *Inverse Expansion* erfolgreich anwendbar, so wird der gesuchte Prozeß Q zurückgeliefert. Ansonsten ist eine solche Lösung für P nicht vorhanden.

Da *Inverse Expansion* von der Voraussetzung ausgeht, daß die Mengen A_1 und A_2 disjunkt sein müssen, kann damit folgendes Problem nicht gelöst werden:

> Sei P der Ausgangsprozeß. Gesucht ist ein Prozeß Q mit $P \sim Q$, der die Form $Q = Q_1 \,|||\, Q_2 \ldots |||\, Q_n$ hat, wobei n, als Zerlegungsgrad bezeichnet, maximal ist und Q_i mit $i = 1, \ldots, n$ keine leere Prozesse, d.h. $Q_i \not\sim$ **stop**, sind. Außerdem gilt, daß die Aktionenmengen von Q_i sich überschneiden dürfen (d.h. nicht disjunkt sind) und die Folgezustände von Q einen maximalen Zerlegungsgrad besitzen.

Anschaulich ausgedrückt heißt das, daß der Ausgangsprozeß in möglichst viele Teilprozesse zerlegt wird, die parallel unabhängig voneinander ablaufen. Das Ziel ist, dadurch einen Zeitgewinn bei der Prozeßausführung zu erreichen.

Die Anforderung, daß die Aktionenmengen sich überschneiden dürfen, erlaubt es, auch nicht derterministische Prozesse maximal zu zerlegen. Nichtdeterministische Prozesse kommen z.B. beim Entwurf von Spielautomaten oder von Systemen vor, deren beobachtbares Verhalten nicht vorhersagbar ist.

Aus diesem Grund wird in diesem Bericht eine Methode, als *maximale Prozeßzerlegung* bezeichnet, vorgestellt, die dieses Problem löst. Im Gegensatz zur *Inverse Expansion*, die direkt auf der Syntax der Sprache LOTOS basiert, basiert diese Methode auf Ereignisstrukturen (engl.: event strutures). Ereignisstrukturen wurden in [Lan92] von Langerak als semantisches Modell für Basic LOTOS verwendet.

Die obengenannten praktischen Probleme, die sich mit *Inverse Expansion* lösen lassen, können auch hier mit *maximale Prozeßzerlegung* gelöst werden. Allerdings ist anzumerken, daß sowohl *Inverse Expansion* als auch *maximale Prozeßzerlegung* in der Praxis nicht oft zum Einsatz gebracht werden können, da diese Methoden lediglich auf die Klasse der nicht rekursiven Prozessen eingeschränkt sind. In der Praxis kommen z.B. beim Entwurf von Kommunikationssystemen fast ausschließlich rekursive Prozesse vor. Die in diesem Bericht vorgestellte Methode *maximale Prozeßzerlegung* soll jedoch als erster Schritt für die weitere Untersuchung im Hinblick auf rekursive Prozesse verstanden werden.

Der Bericht gliedert sich wie folgt: Im Abschnitt 2 werden die Syntax, die operationelle Semantik und die True-Concurrency-Semantik für eingeschränktes Basic LOTOS aus [Lan92], [Do95] wiedergegeben. Abschnitt 3 beschäftigt sich mit der Methode der Rückführung von Semantik zur Syntax, die im nachfolgenden Abschnitt für die Zerlegungsmethode benötigt wird. Im Abschnitt 4 werden das Zerlegungsproblem und die Zerlegungsmethode vorgestellt. Abschnitt 5 stellt ein einfaches Anwendungsbeispiel vor, das die praktische Anwendbarkeit der Methode *maximale Prozeßzerlegung* demonstriert. Es handelt sich dabei um die Spezifikation eines Waren-Verkaufsautomaten. Im Abschnitt 6 erfolgt eine Zusammenfassung.

2 Eingeschränktes Basic LOTOS ohne Rekursion

Dieser Abschnitt gibt als Grundlage die formale Definition der Syntax, der operationellen Semantik und der True-Concurrency-Semantik (auch Halbordnungssemantik genannt) von (eingeschränktem) Basic LOTOS ohne Rekursion aus [Lan92] kurz wieder. Die Notationen lehnen sich an [Lan92], [Do95]. Für Einzelheiten wird der Leser auf [Lan92], [Do95] verwiesen.

2.1 Syntax

Definition 2.1 **($\mathcal{BL}$)** *Sei $\mathcal{G}$ eine Menge von Aktionsnamen, $g \in \mathcal{G}$ und $g_i \in \mathcal{G}$ mit $1 \leq i \leq n$. Dann ist $\mathcal{BL}$ durch folgende Grammatik*

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

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.

- *action prefix*: $g; B$

  Der Prozeß $g; B$ beschreibt einen Prozeß, der zunächst die Aktion g ausführt und sich anschließend wie B verhält.

- *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.

- *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.

Bemerkung 2.1

- Für $B_1|[\emptyset]|B_2$ bzw. $B_1|[\mathcal{G}]|B_2$ schreiben wir auch kurz $B_1 \,|||\, B_2$ bzw. $B_1 \,||\, B_2$.

- Ist $I = \{1, \ldots, n\}$. Dann steht $\sum\{B_i \mid\ i \in I\}$ für $B_1\ \ []\ \ B_2 \ldots []\ \ B_n$ und $\prod\{B_i \mid\ i \in I\}$ für $B_1\ \ |||\ \ B_2 \ldots |||\ \ B_n$. Gilt $I = \emptyset$, dann ist $\sum\{B_i \mid\ i \in I\} = \mathbf{stop}$ und $\prod\{B_i \mid\ i \in I\} = \mathbf{stop}$. □

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 $B \in \mathcal{BL}$. Dann ist die operationelle Semantik von B ein Transitionssystem*

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

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

1. $g; B \overset{g}{\Leftrightarrow\!\!\!\rightarrow} B$
2. $\dfrac{B_1 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_1'}{B_1 \,[]\, B_2 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_1'}$
3. $\dfrac{B_2 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_2'}{B_1 \,[]\, B_2 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_2'}$
4. $\dfrac{B_1 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_1' \wedge a \notin \{g_1, \ldots, g_n\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_1' \,|[g_1, \ldots, g_n]|\, B_2}$
5. $\dfrac{B_2 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_2' \wedge a \notin \{g_1, \ldots, g_n\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_1 \,|[g_1, \ldots, g_n]|\, B_2'}$
6. $\dfrac{B_1 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_1' \wedge B_2 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_2' \wedge a \in \{g_1, \ldots, g_n\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \overset{a}{\Leftrightarrow\!\!\!\rightarrow} B_1' \,|[g_1, \ldots, g_n]|\, B_2'}$ □

Die in Definition 2.3 angegebenen Regeln sind Regeln der Form:

$$\frac{A_1 \overset{a_1}{\Leftrightarrow\!\!\!\rightarrow} A_1', \ldots, A_n \overset{a_n}{\Leftrightarrow\!\!\!\rightarrow} A_n'}{A \overset{a}{\Leftrightarrow\!\!\!\rightarrow} A'}$$

Diese Regel ist wie folgt zu interpretieren: Sind die Vorbedingungen $A_1 \overset{a_1}{\Leftrightarrow\!\!\!\rightarrow} A_1', \ldots, A_n \overset{a_n}{\Leftrightarrow\!\!\!\rightarrow} A_n'$ erfüllt, so ist daraus $A \overset{a}{\Leftrightarrow\!\!\!\rightarrow} A'$ abzuleiten. Anders ausgedrückt: Existieren Übergänge mit den Aktionen a_i von Zuständen A_i in Folgezustände A_i' für $i = 1, \ldots, n$, so existiert ein Übergang mit der Aktion a vom Zustand A in den Folgezustand A'.

Die Regeln 1 bis 6 in Definition 2.3 haben daher folgende anschauliche Bedeutung:

1. <u>Zu 1:</u> Unabhängig von den Vorbedingungen wird ein Übergang mit g erzeugt, der vom Zustand $g; B$ ausgeht und im Folgezustand B endet.

2. <u>Zu 2 und 3:</u> Führt der Prozeß B_1 (bzw. B_2) die Aktion a aus, so führt der Prozeß $B_1 \,[]\, B_2$ die Aktion a aus und geht in den Folgezustand B_1' (bzw. B_2') über. Der Prozeß B_2 (bzw. B_1) wird aus dem weiteren Prozeßgeschehen entfernt.

3. Zu 4: Führt der Prozeß B_1 eine Aktion a aus, wobei a keine Synchronisationsaktion ist, so führt der Prozeß $B_1 \,|[g_1, \ldots, g_n]|\, B_2$ die Aktion a aus und geht anschließend in den Folgezustand $B_1' \,|[g_1, \ldots, g_n]|\, B_2$ über. Dies bedeutet, daß B_1 in $B_1 \,|[g_1, \ldots, g_n]|\, B_2$ unabhängig von B_2 stets eine Aktion ausführen kann, falls diese keine Synchronisationsaktion ist.

4. Zu 5: Symmetrisch zu 4.

5. Zu 6: Ist a eine Synchronisationsaktion, so führt $B_1 \,|[g_1, \ldots, g_n]|\, B_2$ die Aktion a genau dann aus, wenn B_1 und B_2 die Aktion a ausführen. D.h. B_1 und B_2 führen die Aktion a synchron aus.

Dadurch daß jedem Term $B \in \mathcal{BL}$ ein Transitionssystem zugeordnet wird, sind wir nun in der Lage, die Äquivalenz von zwei Prozessen zu definieren.

Definition 2.4 (Äquivalenz auf Transitionssystemen)
Sei $T_i = (Q_i, \rightarrow_i, q_i)$ 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 $p \overset{a}{\Leftrightarrow}_1 p'$, dann $\exists q' \in Q_2 : q \overset{a}{\Leftrightarrow}_2 q'$ und $(p', q') \in R$.*

2. *Gilt $q \overset{a}{\Leftrightarrow}_2 q'$, dann $\exists p' \in Q_1 : p \overset{a}{\Leftrightarrow}_1 p'$ und $(p', q') \in R$.*

Eine Relation R, die die Bedingung 1. und 2. erfüllt, heißt eine Bisimulation. $\square$

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. $\square$

Lemma 2.1 *Sei $B, C \in \mathcal{BL}$. $B \sim C$ gilt genau dann, wenn gilt:*

1. *Gilt $B \overset{a}{\Leftrightarrow} B'$, dann gilt $\exists C' : C \overset{a}{\Leftrightarrow} C'$ und $B' \sim C'$.*

2. *Gilt $C \overset{a}{\Leftrightarrow} C'$, dann gilt $\exists B' : B \overset{a}{\Leftrightarrow} B'$ und $B' \sim C'$.*

Beweis: Trivial. $\square$

Satz 2.1 *$\sim$ ist auf $\mathcal{BL}$ eine Kongruenzrelation, d.h. eine Äquivalenzrelation, für die folgendes gilt: Sei $B, B', C, C' \in \mathcal{BL}$ mit $B \sim C$ und $B' \sim C'$ sowie $op \in \{[], \ |[g_1, \ldots, g_n]|\}$, dann gilt $B \ op \ B' \sim C \ op \ C'$ und $g; B \sim g; C$.*

Beweis: siehe [Mil89], [Do95] $\square$

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

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

2.3 True-Concurrency-Semantik

Die True-Concurrency-Semantik, die im Gegensatz zur operationellen Semantik die kausale Abhängigkeit der Aktionen beschreibt, wird auf der Menge der sogenannten bundle event structures definiert [Lan92]. Diese ist eine abgewandelte Form von primes event structures, flow event structures und stable event structures, die Winskel in [Win89] eingeführt hat.

Definition 2.7 (EBES) *Ein Quartupel* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *mit*

- E *als eine Menge von Ereignissen*
- $\rightsquigarrow \subseteq E \times E$ *(Konfliktrelation)*
- $\mapsto \subseteq 2^E \times E$ *(Bündelrelation)*
- $l : E \to Act$ *(Markierungsfunktion)*

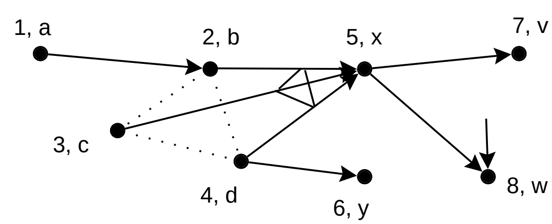
heißt eine extended bundle event structure (kurz EBES), wenn gilt:

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 EBES bezeichnen wir als $\mathcal{ES}$. □

Graphisch wird eine EBES wie folgt dargestellt: Jedes Ereignis $e \in E$ wird durch einen Knoten repräsentiert. Bei jedem Knoten 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 gezeichnet.

Beispiel 2.1 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 EBES. Anschaulich sind zwei EBES isomorph zueinander, wenn sie sich nur durch die Ereignisnamen unterscheiden.

Definition 2.8 (Isomorphismus, $\cong$) *Sei* $\mathcal{E}_i$ *eine EBES 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 von* $\mathcal{E}_1$ *auf* $\mathcal{E}_2$*, wenn gilt:*

1. *h ist bijektiv.*

2. $e \rightsquigarrow_1 e' \Longleftrightarrow h(e) \rightsquigarrow_2 h(e')$

3. $X \mapsto_1 e \Longleftrightarrow h(X) \mapsto_2 h(e)$, *wobei* $X \neq \emptyset$, *und* $\emptyset \mapsto_1 e \Longleftrightarrow \emptyset \mapsto_2 h(e)$

4. $l_1(e) = l_2(h(e))$

Wir schreiben $\mathcal{E}_1 \cong \mathcal{E}_2$ $(\cong \subseteq \ (\mathcal{ES} \times \mathcal{ES}))$, *wenn es einen Isomorphismus von* $\mathcal{E}_1$ *auf* $\mathcal{E}_2$ *gibt. Sei* $E_1' \subseteq E_1$, *dann* $h(E_1') := \{h(e) \mid \ e \in E_1'\}$. *Sei* $\mathcal{E}_1' = (E_1', \rightsquigarrow_1', \mapsto_1', l_1')$ *mit*

- $\rightsquigarrow_1' = \rightsquigarrow_1 \cap (E_1' \times E_1')$
- $\mapsto_1' = \mapsto_1 \cap (2^{E_1'} \times E_1')$
- $l_1' = l_1 \lceil E_1'$

Dann $h(\mathcal{E}_1') := (h(E_1'), \rightsquigarrow_2', \mapsto_2', l_2')$

- $\rightsquigarrow_2' = \rightsquigarrow_2 \cap (h(E_1') \times h(E_1'))$
- $\mapsto_2' = \mapsto_2 \cap (2^{h(E_1')} \times h(E_1'))$
- $l_2' = l_2 \lceil h(E_1')$ $\square$

Offenbar ist $\cong$ eine Äquivalenzrelation.

Definition 2.9 (Ausführungsfolge) *Sei* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *eine EBES. 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. $X \mapsto e_i \Longrightarrow \{e_1, \ldots, e_{i-1}\} \cap X \neq \emptyset$ $\square$

Aus dem Beispiel 2.1 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.

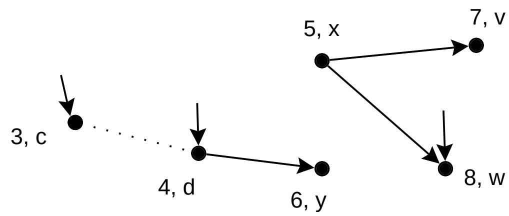
Definition 2.10 (Konfiguration) *Sei* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *eine EBES. Eine Menge* $K \subseteq E$ *heißt eine Konfiguration von* $\mathcal{E}$, *wenn es eine Ausführungsfolge* $e_1, \ldots, e_n$ *gibt, so daß* $K = \{e_1, , \ldots, e_n\}$ *gilt.* $\square$

Definition 2.11 ($\mathcal{E}[K]$) *Sei* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *eine EBES und* K *eine Konfiguration von* $\mathcal{E}$. *Dann ist* $\mathcal{E}[K] := (E', \rightsquigarrow', \mapsto', l')$, *wobei*

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

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 2.2 Sei $\mathcal{E}$ die Ereignisstruktur aus dem Beispiel 2.1, dann ist $\mathcal{E}[\{1,2\}] =$



□

Definition 2.12 ($\prec_K$) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine EBES 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.

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

Beweis: siehe [Lan92]. □

Definition 2.13 (*lposet*) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine EBES 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 $\overline{K} := (K, \leq_K, l\lceil K)$.* □

Definition 2.14 ($\overline{K}$-Transition) *Sei $\mathcal{E}$ eine EBES und $\overline{K}$ eine lposet von $\mathcal{E}$. $\mathcal{E}$ hat eine $\overline{K}$-Transition, bezeichnet als $\mathcal{E} \overset{\overline{K}}{\Longleftrightarrow} \mathcal{E}'$, wenn $\mathcal{E}' = \mathcal{E}[K]$ gilt.* □

Bemerkung 2.3 *Für $\mathcal{E} \overset{\overline{K}}{\Longleftrightarrow} \mathcal{E}'$ mit $K = \{e\}$ schreiben wir kurz $\mathcal{E} \xrightarrow{e} \mathcal{E}'$.* □

Nachdem die notwendigen Definitionen über EBES zitiert wurden, wird nun die True-Concurrency-Semantik für $\mathcal{BL}$, die jedem Term $B \in \mathcal{BL}$ eine EBES zuordnet, definiert.

Definition 2.15 (Notationen) *Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ eine EBES. Dann*

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

Definition 2.16 (Operatoren auf EBES) *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 die (abzählbare) Menge aller möglichen Ereignisnamen. Dann*

- $\overline{\textbf{stop}} := (\emptyset, \emptyset, \emptyset, \emptyset)$

- *Für* $g \in \mathcal{G}$ *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. $l = l_1 \cup \{(e, g)\}$

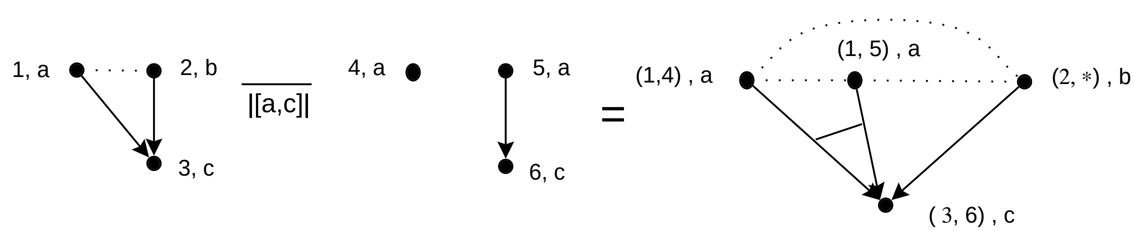
- $\mathcal{E}_1\ \overline{[]}\ \mathcal{E}_2 := (E, \rightsquigarrow, \mapsto, l)$, *wobei*

   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. $l = l_1 \cup l_2$

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

   1. $E = (E_1^f \times \{*\}) \cup (\{*\} \times E_2^f) \cup \{(e_1, e_2) \in E_1^s \times E_2^s \,|\, l_1(e_1) = l_2(e_2)\}$
   mit $E_i^s = \{e \in E_i \,|\, l_i(e) \in \{g_1, \ldots, g_n\}\}$ *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 2.3



□

Definition 2.17 (True-Concurrency-Semantik) *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* $B = \textbf{stop}$, *dann* $[\![B]\!] := [\ \overline{\textbf{stop}}\ ]_{/\cong}$.

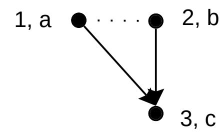
- *Sei* $B = g; B_1$, *dann* $[\![B]\!] := [\overline{g_e;}\ \mathcal{E}_1]_{/\cong}$ *mit* $e \in E_U$ *und* $e \notin E_1$.

- *Sei* $B = B_1 \,[]\, B_2$, *dann* $[\![B]\!] := [\mathcal{E}_1 \;\overline{[]}\; \mathcal{E}_2]_{/\cong}$.
- *Sei* $B = B_1 \,|[g_1, \ldots, g_n]|\, B_2$, *dann* $[\![B]\!] := [\mathcal{E}_1 \overline{|[g_1, \ldots, g_n]|}\; \mathcal{E}_2]_{/\cong}$. □

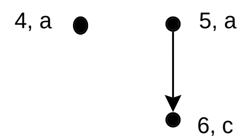
$[\![B]\!]$ ordnet also jedem Term $B \in \mathcal{BL}$ eine Isomorphieklasse zu. Es ist jedoch anzumerken, daß ein Repräsentant dieser Klasse gemeint ist, wenn über $[\![B]\!]$ gesprochen wird. Wir schreiben daher an Stelle $\mathcal{E} \in [\![B]\!]$ auch $[\![B]\!] = \mathcal{E}$ oder $\mathcal{E}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

Beispiel 2.4

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



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



- Sei $B = B1 \;|[a, c]|\; B2$, dann hat $[\![B]\!]$ die Ereignisstruktur wie im Beispiel 2.3 angegeben. □

Man beachte, daß die Definition 2.17 nur dann korrekt ist, wenn $\cong$ bezüglich $\overline{g_e;}$, $\overline{[]}$ und $\overline{|[g_1, \ldots, g_n]|}$ eine Kongruenzrelation ist. Daß dies gilt, bestätigt das nachfolgende Lemma.

Lemma 2.3 *Sei* $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ *mit* $i = 1, 2, 3, 4$, $E_1 \cap E_2 = \emptyset$, $E_3 \cap E_4 = \emptyset$, $\mathcal{E}_1 \cong \mathcal{E}_3$ *und* $\mathcal{E}_2 \cong \mathcal{E}_4$. *Sei ferner* $e1 \notin E_1$ *und* $e3 \notin E_3$. *Dann gilt:*

1. $\overline{g_{e1};}\; \mathcal{E}_1 \cong \overline{g_{e3};}\; \mathcal{E}_3$
2. $\mathcal{E}_1 \;\overline{[]}\; \mathcal{E}_2 \cong \mathcal{E}_3 \;\overline{[]}\; \mathcal{E}_4$
3. $\mathcal{E}_1 \;\overline{|[g_1, \ldots, g_n]|}\; \mathcal{E}_2 \cong \mathcal{E}_3 \;\overline{|[g_1, \ldots, g_n]|}\; \mathcal{E}_4$

Beweis: Zu 1. und 2.: trivial. Zu 3: Sei $\mathcal{E} = \mathcal{E}_1 \;\overline{|[g_1, \ldots, g_n]|}\; \mathcal{E}_2 = (E, \rightsquigarrow, \mapsto, l)$ und $\mathcal{E}' = \mathcal{E}_3 \;\overline{|[g_1, \ldots, g_n]|}\; \mathcal{E}_4 = (E', \rightsquigarrow', \mapsto', l')$. Seien $h_1 : E_1 \to E_3$ und $h_2 : E_2 \to E_4$ Isomorphismen. Sei $f : E \to E'$ wie folgt definiert:

$$f((e_1, e_2)) = \begin{cases} (h_1(e_1), *) & \text{falls} \quad e_2 = * \\ (*, h_2(e_2)) & \text{falls} \quad e_1 = * \\ (h_1(e_1), h_2(e_2)) & \text{falls} \quad e_1 \neq * \wedge e_2 \neq * \end{cases}$$

Dann läßt sich zeigen, daß f ein Isomorphismus von E auf E' ist. Es ist leicht zu prüfen, daß f die Bedingungen 1, 2 und 4 der Definition 2.8 erfüllt. Wir zeigen daher im folgenden nur die Gültigkeit der Bedingung 3.

- $\Rightarrow$:

  Sei $X \mapsto (e_1, e_2)$ und $X \neq \emptyset$, dann gilt o.E.d.A. $X = \{(e_i, e_j) \in E \mid e_i \in X_1\}$, wobei $X_1 \mapsto_1 e_1$ gilt. Aus $h_1(X_1) \mapsto_3 h_1(e_1)$ folgt $Y \mapsto' f((e_1, e_2))$ mit $Y = \{(e, e') \in E' \mid e \in h_1(X_1)\}$. Es bleibt zu zeigen, daß $f(X) = Y$ gilt. Sei $(e_i, e_j) \in X$, dann gilt $f((e_i, e_j)) = (h_1(e_i), *)$ für den Fall $e_j = *$, und $f((e_i, e_j)) = (h_1(e_i), h_2(e_j))$ für den Fall $e_j \neq *$. Da $h_1(e_i) \in h_1(X_1)$ gilt, gilt $f(X) \subseteq Y$.

  Gilt umgekehrt $(e, e') \in Y$, so sind zwei Fälle zu unterscheiden:

  - Fall 1: $e' = *$. Dann gilt $(h_1^{-1}(e), *) \in E$. Da $h_1^{-1}(e) \in X_1$ gilt, gilt $(h_1^{-1}(e), *) \in X$. Wegen $(e, *) = f((h_1^{-1}(e), *))$ gilt $(e, *) \in f(X)$.
  - Fall 2: $e' \neq *$. Mit der analogen Begründung zum Fall 1 gilt $(e, e') = f((h_1^{-1}(e), h_1^{-1}(e')))$ und $(e, e') \in f(X)$.

  Insgesamt gilt $f(X) = Y$.

  Gilt $\emptyset \mapsto (e_1, e_2)$, dann gilt $\emptyset \mapsto_1 e_1$ oder $\emptyset \mapsto_2 e_2$. Daraus folgt $\emptyset \mapsto_2 h_1(e_1)$ oder $\emptyset \mapsto_2 h_2(e_2)$ und somit $\emptyset \mapsto' f((e_1, e_2))$.

- $\Leftarrow$: Die Umkehrung gilt analog. □

Zum Schluß dieses Abschnittes wird der Begriff „Bisimulationsäquivalenz", bezeichnet als $\approx_E$, für Ereignisstrukturen eingeführt und ein wichtiges Lemma aus [Do95] zitiert, das die Äquivalenz von $\sim$ und $\approx_E$ wiedergibt.

Definition 2.18 ($Er(\mathcal{E})$) *Sei $\mathcal{E}$ eine EBES. $Er(\mathcal{E})$ ist die kleinste Menge, für die gilt:*

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

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

Definition 2.19 ($\approx_E$)

- *Eine Relation $R \subseteq \mathcal{ES} \times \mathcal{ES}$ heißt eine E-Bisimulation, wenn für alle $(\mathcal{E}_1, \mathcal{E}_2) \in R$ gilt:*

  1. $\mathcal{E}_1 \overset{e}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_1' \Longrightarrow \exists e' : (l_1(e) = l_2(e') \ \wedge \ \mathcal{E}_2 \overset{e'}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_2' \wedge (\mathcal{E}_1', \mathcal{E}_2') \in R)$
  2. $\mathcal{E}_2 \overset{e'}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_2' \Longrightarrow \exists e : (l_1(e) = l_2(e') \ \wedge \ \mathcal{E}_1 \overset{e}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_1' \wedge (\mathcal{E}_1', \mathcal{E}_2') \in R)$

  Dabei ist l_1 Markierungsfunktion in $\mathcal{E}_1$ und l_2 Markierungsfunktion in $\mathcal{E}_2$.

- *$\mathcal{E}_1$ 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.* □

Korollar 2.1 *Sei $\mathcal{E}_1, \mathcal{E}_2 \in \mathcal{ES}$. $\mathcal{E}_1 \approx_E \mathcal{E}_2$ gilt genau dann, wenn gilt:*

1. $\mathcal{E}_1 \overset{e}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_1' \Longrightarrow \exists e' : (l_1(e) = l_2(e') \ \wedge \ \mathcal{E}_2 \overset{e'}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_2' \wedge \ \mathcal{E}_1' \approx_E \mathcal{E}_2')$
2. $\mathcal{E}_2 \overset{e'}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_2' \Longrightarrow \exists e : (l_1(e) = l_2(e') \ \wedge \ \mathcal{E}_1 \overset{e}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}_1' \wedge \ \mathcal{E}_1' \approx_E \mathcal{E}_2')$ □

Lemma 2.4 *Sei $B, B' \in \mathcal{BL}$, dann gilt $[\![B]\!] \approx_E [\![B']\!] \Longleftrightarrow B \sim B'$.*

Beweis: Siehe [Do93]. □

3 Von der Ereignisstruktur zurück zur Syntax

Im vorigen Abschnitt wurde die Semantikfunktion $[\![B]\!]$ definiert, die jedem syntaktischen Gebilde B ein semantisches Objekt in Form einer Ereignisstruktur zuordnet. Im Gegensatz dazu wird nun eine Methode vorgestellt, die es ermöglicht, für eine bestimmte Klasse von Ereignisstrukturen (im folgenden als $\mathcal{ES}_\mathcal{P}$ bezeichnet) die entsprechenden syntaktischen Gebilde zu bestimmen. Formal:

- Gegeben sei eine Ereignisstruktur $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$.
- Gesucht ist ein Prozeß $B \in \mathcal{BL}$, so daß $[\![B]\!] = \mathcal{E}$ gilt.

Diese Methode wird im Abschnitt 4 für die Zerlegungsmethode benötigt. Es ist anzumerken, daß es im allgemeinen nicht gilt, daß es zu jeder endlichen Ereignisstruktur $\mathcal{E}$ (d.h. die Ereignismenge ist endlich) einen Prozeßterm $B \in BL$ mit $[\![B]\!] = \mathcal{E}$ gibt.

Sei beispielsweise $\mathcal{E} = (\{1,2,3\}, \{(1,2)\}, \{(\{1,2\},3)\}, \{(1,a),(2,b),(3,c)\})$, dann gibt es keinen Prozeßterm $B \in \mathcal{BL}$, so daß $[\![B]\!] = \mathcal{E}$ gilt. Der Grund ist, daß für $e_1, e_2 \in X$ mit $X \mapsto e$ in $[\![B]\!]$ immer $l(e_1) = l(e_2)$ gilt. Dabei ist l die Markierungsfunktion in $[\![B]\!]$. Diese Aussage läßt sich leicht mit struktureller Induktion über B zeigen.

3.1 Methode zur Bestimmung der Syntax

Im folgenden wird $\mathcal{ES}_\mathcal{P}$ formal definiert. Außerdem wird eine Teilsprache von $\mathcal{BL}$, bezeichnet als $\mathcal{BL}_\mathcal{P}$, definiert, wofür dann gezeigt wird, daß jede Ereignisstruktur $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$ einen Prozeß $B \in \mathcal{BL}_\mathcal{P}$ als zugehöriges, syntaktisches Objekt besitzt und umgekehrt. $\mathcal{BL}_\mathcal{P}$ gibt dabei anschaulich die Klasse von Prozessen an, in der keine Synchronisation zwischen Prozessen erlaubt ist.

Definition 3.1 ($\overline{\cup}$) *Sei $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ eine EBES mit $i = 1,2$ und $E_1 \cap E_2 = \emptyset$. Dann ist $\mathcal{E}_1 \,\overline{\cup}\, \mathcal{E}_2 := (E, \rightsquigarrow, \mapsto, l)$, wobei*

- $E = E_1 \cup E_2$
- $\rightsquigarrow \;=\; \rightsquigarrow_1 \cup \rightsquigarrow_2$
- $\mapsto \;=\; \mapsto_1 \cup \mapsto_2$
- $l = l_1 \cup l_2$ □

Wie im folgenden gezeigt wird, kann der Operator $\overline{|||}$ durch $\overline{\cup}$ ersetzt werden, ohne daß dabei die Bedeutung von $\overline{|||}$ verändert wird. Dazu wird eine kürzere Schreibweise für $\overline{|||}$ und $\overline{\cup}$ benötigt. Für $\mathcal{E}_1 \overline{|||} \mathcal{E}_2 \ldots \overline{|||} \mathcal{E}_n$ bzw. $\mathcal{E}_1 \overline{\cup} \mathcal{E}_2 \ldots \overline{\cup} \mathcal{E}_n$ schreiben wir auch kurz $\overline{\prod}\{\mathcal{E}_i \,|\, i \in I\}$ bzw. $\overline{\bigcup}\{\mathcal{E}_i \,|\, i \in I\}$, wobei $I = \{1, \ldots, n\}$ gilt. Ist $I = \emptyset$, dann ist $\overline{\prod}\{\mathcal{E}_i \,|\, i \in I\} = (\emptyset, \emptyset, \emptyset, \emptyset) = \overline{\bigcup}\{\mathcal{E}_i \,|\, i \in I\}$.

Lemma 3.1 *Sei $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ eine EBES mit $i = 1,2,3,4$ und $E_1 \cap E_2 = \emptyset = E_3 \cap E_4$. Sei ferner $\mathcal{E}_1 \cong \mathcal{E}_3$ und $\mathcal{E}_2 \cong \mathcal{E}_4$. Dann gilt $\mathcal{E}_1 \,\overline{\cup}\, \mathcal{E}_2 \cong \mathcal{E}_3 \,\overline{|||}\, \mathcal{E}_4$.*

Beweis: Es ist leicht zu prüfen, daß $\mathcal{E}_3 \,\overline{|||}\, \mathcal{E}_4 = \mathcal{E}'_3 \,\overline{\cup}\, \mathcal{E}'_4$ gilt, wobei $\mathcal{E}'_i = (E'_i, \rightsquigarrow'_i, \mapsto'_i, l'_i)$ mit $i = 3,4$ und

- $E_3' = \{(e,*) \mid \; e \in E_3\}$ und $E_4' = \{(*,e) \mid \; e \in E_4\}$
- $\rightsquigarrow_3' = \; \{((e,*),(e',*)) \mid \; e \rightsquigarrow_3 e'\}$ und $\rightsquigarrow_4' = \; \{((*,e),(*,e')) \mid \; e \rightsquigarrow_4 e'\}$
- $\mapsto_3' = \; \{(X,(e',*)) \mid \; X = \{(e,*) \mid \; Y \mapsto_3 e' \wedge e \in Y\}\}$ und
  $\mapsto_4' = \; \{(X,(*,e')) \mid \; X = \{(*,e) \mid \; Y \mapsto_4 e' \wedge e \in Y\}\}$
- $l_3'((e,*)) = l_3(e)$ und $l_4'((*,e)) = l_4(e)$

Es ist auch leicht zu zeigen, daß $\mathcal{E}_3 \cong \mathcal{E}_3'$ und $\mathcal{E}_4 \cong \mathcal{E}_4'$ gilt. Aus $\mathcal{E}_1 \cong \mathcal{E}_3$ und $\mathcal{E}_2 \cong \mathcal{E}_4$ folgt dann $\mathcal{E}_1 \cong \mathcal{E}_3'$ und $\mathcal{E}_2 \cong \mathcal{E}_4'$. Daraus läßt sich leicht die Gültigkeit von $\mathcal{E}_1 \,\overline{\cup}\, \mathcal{E}_2 \cong \mathcal{E}_3' \,\overline{\cup}\, \mathcal{E}_4'$ folgern. $\square$

$\overline{\cup}$ ist offenbar bezüglich $\cong$ assoziativ. Daß dies auch für $\overline{|||}$ gilt, läßt sich wie folgt zeigen: Sei $\mathcal{E}1 = \mathcal{E}_1\overline{|||}(\mathcal{E}_2\overline{|||}\mathcal{E}_3)$ und $\mathcal{E}3 = (\mathcal{E}_1\overline{|||}\mathcal{E}_2)\overline{|||}\mathcal{E}_3$, dann folgt aus Lemma 3.1 $\mathcal{E}1 \cong \mathcal{E}2$ mit $\mathcal{E}2 = \mathcal{E}_1\overline{\cup}(\mathcal{E}_2\overline{\cup}\mathcal{E}_3)$ und $\mathcal{E}3 \cong \mathcal{E}4$ mit $\mathcal{E}4 = (\mathcal{E}_1\overline{\cup}\mathcal{E}_2)\overline{\cup}\mathcal{E}_3$. Da $\mathcal{E}2 \cong \mathcal{E}4$ gilt, gilt $\mathcal{E}1 \cong \mathcal{E}3$. $\overline{|||}$ ist also bezüglich $\cong$ assoziativ.

Es ist anzumerken, daß $\overline{[]}$ bezüglich der Identitätsrelation $=$ trivialerweise auch assoziativ ist. Die Klammerung von Termen kann daher weggelassen werden.

Lemma 3.2 *Sei $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ und $\mathcal{E}_i' = (E_i', \rightsquigarrow_i', \mapsto_i', l_i')$ mit $i = 1,\ldots,n$ eine EBES, wobei $E_i \cap E_j = \emptyset$ und $E_i' \cap E_j' = \emptyset$ für $i,j \in \{1,\ldots,n\}$ gilt. Sei ferner $\mathcal{E}_i \cong \mathcal{E}_i'$ für $i = 1,\ldots,n$. Dann gilt*

$$\overline{\bigcup_{i=1}^{n}}\mathcal{E}_i \;\cong\; \overline{\prod_{i=1}^{n}}\mathcal{E}_i'$$

Beweis: Vollständige Induktion über n. Der Beweis ist einfach und wird deshalb nicht ausgeführt. $\square$

Definition 3.2 ($\mathcal{ES}_\mathcal{P}$) *Sei E_U wie in Definition 2.16 die (abzählbare)Menge aller möglichen Ereignisnamen und $e \in E_U$. Dann ist $\mathcal{ES}_\mathcal{P}$ die kleinste Menge, die folgendes erfüllt:*

- $(\emptyset,\emptyset,\emptyset,\emptyset) \in \mathcal{ES}_\mathcal{P}$
- *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$, $g \in \mathcal{G}$ und $e \in E_U$, dann gilt $\overline{g_e;}\;\mathcal{E} \in \mathcal{ES}_\mathcal{P}$.*
- *Sei $op \in \{\overline{[]}, \overline{\cup}\}$ und $\mathcal{E}_1, \mathcal{E}_2 \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i) \neq (\emptyset,\emptyset,\emptyset,\emptyset)$ für $i = 1,2$ und $E_1 \cap E_2 = \emptyset$, dann gilt $\mathcal{E}_1 \; op \; \mathcal{E}_2 \in \mathcal{ES}_\mathcal{P}$.* $\square$

Lemma 3.3 *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$. Gilt $\mathcal{E} \cong \mathcal{E}'$, dann gilt $\mathcal{E}' \in \mathcal{ES}_\mathcal{P}$.*

Beweis: Induktion über den Aufbau von $\mathcal{E}$.

- $\mathcal{E} = (\emptyset,\emptyset,\emptyset,\emptyset)$

  Dann gilt $\mathcal{E}' = (\emptyset,\emptyset,\emptyset,\emptyset)$ und somit $\mathcal{E}' \in \mathcal{ES}_\mathcal{P}$.
- $\mathcal{E} = \overline{g_e;}\mathcal{E}_1$

  Es ist leicht zu zeigen, daß $\mathcal{E}' = \overline{g_{e'};}\mathcal{E}_1'$ gilt, wobei $\mathcal{E}_1 \cong \mathcal{E}_1'$ gilt. Nach Induktionsannahme gilt $\mathcal{E}_1' \in \mathcal{ES}_\mathcal{P}$. Daraus folgt $\mathcal{E}' \in \mathcal{ES}_\mathcal{P}$.

- $\mathcal{E} = \mathcal{E}_1 \overline{[]} \mathcal{E}_2$

  Sei h der Isomorphismus von $\mathcal{E}$ auf $\mathcal{E}'$. Es ist leicht zu zeigen, daß $\mathcal{E}' = h(\mathcal{E}_1)\overline{[]}h(\mathcal{E}_2)$ gilt. Da $h(\mathcal{E}_i) \cong \mathcal{E}_i$ mit $i = 1, 2$ gilt, gilt nach Induktionsannahme $h(\mathcal{E}_i) \in \mathcal{ES}_{\mathcal{P}}$.

- Analog zu 3. □

Definition 3.3 $(\mathcal{BL}_{\mathcal{P}})$ *Sei $\mathcal{G}$ eine Menge von Aktionsnamen und $g \in \mathcal{G}$. Dann ist $\mathcal{BL}_{\mathcal{P}}$ durch folgende Grammatik*

$$B \quad ::= \quad \textbf{stop} \quad | \quad g; B \quad | \quad B \,[]\, B \quad | \quad B \,|||\, B$$

definiert. □

$\mathcal{BL}_{\mathcal{P}}$ ist also eine Teilsprache von $\mathcal{BL}$. Die Semantikfunktion $[\![.]\!]$, die für $\mathcal{BL}$ definiert ist, soll daher auch für $\mathcal{BL}_{\mathcal{P}}$ gelten. Es läßt sich leicht mit Induktion über den Termaufbau von $B \in \mathcal{BL}_{\mathcal{P}}$ und mit Lemma 3.3 zeigen, daß es immer ein $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ gibt, so daß $[\![B]\!] = \mathcal{E}$ gilt. Die Umkehrung ist etwas schwieriger. Für $\mathcal{ES}_P$ wird nun eine Methode vorgestellt, die jeder Ereignisstruktur $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ einen Prozeß $B \in \mathcal{BL}_{\mathcal{P}}$ mit $\mathcal{E} = [\![B]\!]$ zuordnet.

Die Methode basiert auf folgender Idee: $\mathcal{E}$ wird in in einer geeigneten Weise in Teilprozesse zerlegt, für die zunächst die entsprechenden, syntaktischen Objekte ermittelt werden. Anschließend werden diese Objekte in Abhängigkeit davon, wie sie zueinander in $\mathcal{E}$ stehen, entweder mit $\overline{[]}$ oder mit $\overline{\cup}$ verknüpft. Dazu definieren wir zwei Relationen, die die Zerlegung von $\mathcal{E}$ ermöglichen.

Definition 3.4 $(\bowtie_M)$ *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} = (E, \leadsto, \mapsto, l)$ und $M \subseteq E$. Zwei Ereignisse $e, e' \in M$ sind in Konflikt erreichbar (kurz $e \bowtie_M e'$, d.h. $\bowtie_M \subseteq (M \times M)$), wenn e und e' identisch sind, oder wenn es eine endliche Folge von Ereignissen $e_1, e_2, \ldots e_n \in M$ mit $e = e_1$ und $e' = e_n$ gibt, so daß $e_i \leadsto e_{i+1}$ mit $i = 1, \ldots, n \Leftrightarrow 1$ gilt.* □

Korollar 3.1 *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} = (E, \leadsto, \mapsto, l)$ und $M \subseteq E$, dann ist $\bowtie_M$ eine Äquivalenzrelation auf M.* □

Definition 3.5 $(Konflikt(\mathcal{E}))$ *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$, $M = init(\mathcal{E})$ und $e \in M$. Dann ist $Konflikt(\mathcal{E})$ genau dann wahr, wenn $M = [e]_{/\bowtie_M}$ gilt.* □

Definition 3.6 $(\smile_M)$ *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} = (E, \leadsto, \mapsto, l)$ und $M \subseteq E$. Zwei Ereignisse $e, e' \in M$ sind kausal unabhängig erreichbar (kurz $e \smile_M e'$, d.h. $\smile_M \subseteq (M \times M)$), wenn e und e' identisch sind, oder wenn es eine endliche Folge von Ereignissen $e_1, e_2, \ldots e_n \in M$ mit $e = e_1$ und $e' = e_n$ gibt, so daß $\neg(e_i \leadsto e_{i+1})$ mit $i = 1, \ldots, n \Leftrightarrow 1$ gilt.* □

Korollar 3.2 *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} = (E, \leadsto, \mapsto, l)$ und $M \subseteq E$, dann ist $\smile_M$ eine Äquivalenzrelation auf M.* □

Definition 3.7 $(Konf(\mathcal{E}), Kaus(\mathcal{E}))$ *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ und $M = init(\mathcal{E})$.*

- $Konf(\mathcal{E}) := \{[e]_{/\bowtie_M} \mid e \in M\}$
- $Kaus(\mathcal{E}) := \{[e]_{/\smile_M} \mid e \in M\}$ □

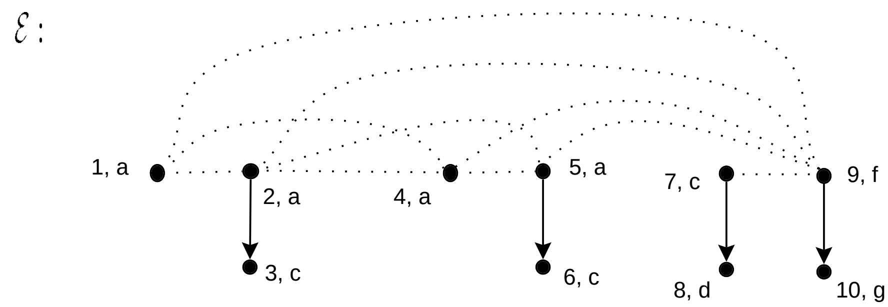


Abbildung 1: Erläuterung zu $Konf(\mathcal{E})$ und $Kaus(\mathcal{E})$

Beispiel 3.1 Gegeben sei die Ereignisstruktur $\mathcal{E}$ aus Abbildung 1. Offenbar gilt $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$. Es gilt $Konf(\mathcal{E}) = \{\{1, 2, 4, 5, 7, 9\}\}$ und $Kaus(\mathcal{E}) = \{\{1, 2, 4, 5, 7\}, \{9\}\}$. □

Lemma 3.4 *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$. Sei ferner $M = init(\mathcal{E})$ mit $|M| > 1$ und $e \in M$. Dann gilt*

$$[e]_{/\bowtie_M} = M \Longleftrightarrow [e]_{/\smile_M} \neq M.$$

Beweis: Zunächst gilt $\mathcal{E} = \mathcal{E}_1 \overline{[]} \mathcal{E}_2$ oder $\mathcal{E} = \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$ mit $\mathcal{E}_1 \neq (\emptyset, \emptyset, \emptyset, \emptyset) \neq \mathcal{E}_2$. Daraus folgt $init(\mathcal{E}_1) \neq \emptyset \neq init(\mathcal{E}_1)$ (Induktion über den Aufbau von $\mathcal{E}'$ mit $\mathcal{E}' \in \mathcal{ES}_{\mathcal{P}}$). Außerdem läßt sich auch leicht prüfen, daß $init(\mathcal{E}) = init(\mathcal{E}_1) \cup init(\mathcal{E}_2)$ gilt.

Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ und $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ mit $i = 1, 2$.

- „$\Rightarrow$:“

  1. Zu $\mathcal{E} = \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$: Zunächst gilt $e2 \in E_1$ (bzw. $e2 \in E_2$), falls $e1 \rightsquigarrow e2$ und $e1 \in E_1$ (bzw. $e1 \in E_2$) gilt. Daraus folgt $[e]_{/\bowtie_M} \neq M$, denn es gilt $[e]_{/\bowtie_M} \subseteq init(\mathcal{E}_1)$, wenn $e \in init(\mathcal{E}_1)$ gilt (bzw. $[e]_{/\bowtie_M} \subseteq init(\mathcal{E}_2)$), wenn $e \in init(\mathcal{E}_2)$ gilt.). Dieser Fall braucht also nicht weiter betrachtet zu werden.
  2. Zu $\mathcal{E} = \mathcal{E}_1 \overline{[]} \mathcal{E}_2$: In diesem Fall gilt $[e]_{/\bowtie_M} = M$. Begründung: Sei o.E.d.A. $e \in init(\mathcal{E}_1)$, dann gilt nach Definition 2.17 $e \rightsquigarrow e'$ für $e' \in init(\mathcal{E}_2)$. Sei $e'' \in init(\mathcal{E}_1)$, so gilt $e \bowtie_M e''$, da $e \rightsquigarrow e' \rightsquigarrow e''$ gilt. Also sind alle Elemente in $init(\mathcal{E})$ von e über $\bowtie_M$ erreichbar.

     Es gilt $\neg(e \smile_M e')$ für $e \in init(\mathcal{E}_1)$ und $e' \in init(\mathcal{E}_2)$. Somit gilt $[e]_{/\smile_M} \neq M$. Begründung: Offenbar gilt $\phi \in init(\mathcal{E}_2)$, falls $\neg(\phi \rightsquigarrow e')$ und $\phi \in M$ gilt. Daraus folgt $e \in init(\mathcal{E}_2)$, falls $e \smile_M e'$ gilt. Widerspruch.

- „$\Leftarrow$:“ Wir beweisen kontraproduktiv. D.h. gilt $[e]_{/\bowtie_M} \neq M$, dann gilt $[e]_{/\smile_M} = M$. Mit der analogen Begründung wie oben kommt hier nur der Fall $\mathcal{E} = \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$ in Frage. In diesem Fall gilt $\neg(e \rightsquigarrow e')$ und $\neg(e' \rightsquigarrow e)$ für $e \in init(\mathcal{E}_1)$ und $e' \in init(\mathcal{E}_2)$. Daraus folgt $e \smile_M e''$ für alle $e'' \in init(\mathcal{E})$. Daher gilt $[e]_{/\smile_M} = M$. □

$Konf(\mathcal{E})$ und $Kaus(\mathcal{E})$ bilden im Grunde genommen nichts anderes als die Zerlegung von $init(\mathcal{E})$. Wie später gezeigt wird, kann $\mathcal{E}$ mit Hilfe dieser zwei Zerlegungen in Teilprozesse zerlegt werden.

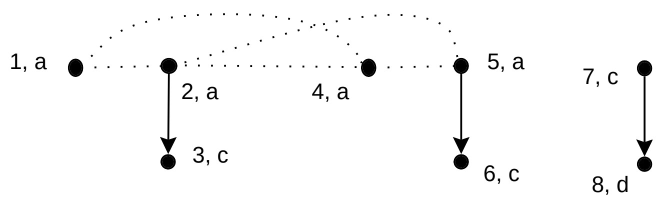
Definition 3.8 (Erreichbarkeit von Ereignissen) *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$, $e \in init(\mathcal{E})$ und $e' \in E$. e' ist von e erreichbar (kurz $e \vdash e'$), wenn $e = e'$ gilt, oder wenn es eine endliche Folge $e_1, e_2, \ldots e_n \in E$ mit $e_1 = e$ und $e_n = e'$ gibt, so daß $\{e_i\} \mapsto e_{i+1}$ für $i = 1, \ldots, n \Leftrightarrow 1$ gilt.* □

Definition 3.9 ($Er(m, \mathcal{E})$) *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ und $m \subseteq init(\mathcal{E})$.*

1. *Falls $m \neq \emptyset$, dann ist $Er(m, \mathcal{E}) = (E', \rightsquigarrow', \mapsto', l')$ mit*
   - $E' = \{e' \mid e \vdash e' \wedge e \in m\}$
   - $\rightsquigarrow' = \rightsquigarrow \cap (E' \times E')$
   - $\mapsto' = \mapsto \cap (2^{E'} \times E')$
   - $l' = l \lceil E'$
2. *Falls $m = \emptyset$, dann $Er(m, \mathcal{E}) := (\emptyset, \emptyset, \emptyset, \emptyset)$.* □

Man beachte, daß $init(Er(m, \mathcal{E})) = m$ gilt.

Beispiel 3.2 Sei $\mathcal{E}$ die Ereignisstruktur aus Abbildung 1 und $m = \{1, 2, 4, 5, 7\}$, dann hat $Er(m, \mathcal{E})$ folgende Form:



□

Mit $Konf(\mathcal{E})$, $Kaus(\mathcal{E})$ und $Er(m, \mathcal{E})$ läßt sich nun eine Methode formulieren, die die Rückführung von Ereignisstruktur zur Syntax ermöglicht.

Definition 3.10 ($Term(\mathcal{E})$) *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$, l Markierungsfunktion in $\mathcal{E}$ und $M = init(\mathcal{E})$. Dann ist $Term(\mathcal{E})$) wie folgt definiert:*

1. *Falls $M = \emptyset$ gilt, dann $Term(\mathcal{E}) :=$ **stop**.*
2. *Falls $M = \{e\}$ gilt, dann $Term(\mathcal{E}) := l(e); (Term(\mathcal{E}[\{e\}]))$.*
3. *Falls $Konflikt(\mathcal{E})$ wahr ist und $|M| > 1$ gilt, dann*
$$Term(\mathcal{E}) := \sum_{m \in Kaus(\mathcal{E})} (Term(Er(m, \mathcal{E}))).$$
4. *Falls $Konflikt(\mathcal{E})$ falsch ist und $|M| > 1$ gilt, dann*
$$Term(\mathcal{E}) := \prod_{m \in Konf(\mathcal{E})} (Term(Er(m, \mathcal{E}))).$$

□

Es handelt sich bei der Definition 3.10 also um eine induktive Definition, die nach [Win93] (S. 177) dem sogenannten wohlfundierten Rekursionsprinzip unterliegt (engl.: well-founded recursion). Dieses Prinzip besagt, daß eine induktive Definition nur dann korrekt ist, wenn die Relation $\prec$, in der die Teilargumente (in unserem Fall $\mathcal{E}[\{e\}]$ und $Er(m, \mathcal{E})$) eines Arguments (in unserem Fall $\mathcal{E}$) und das Argument selbst zueinanderstehen, wohlfundiert ist. Dabei ist eine Relation $\prec$ auf einer Menge A wohlfundiert, wenn es keine unendlich absteigende Kette $\cdots \prec a_i \prec \cdots \prec a_1 \prec a_0$ gibt. $\prec$ ist daher immer irreflexiv, d.h. $\forall a \in A : \neg(a \prec a)$. Anschaulich ausgedrückt heißt das, daß es bei rekursiver Definition ein Abbruchkriterium gibt, das erfüllt werden muß.

A ist in unserem Fall die Menge $\mathcal{ES}_P$ und $\prec$ die Teil-Ereignisstruktur-Relation, deren genaue Definition im folgenden angegeben wird.

Definition 3.11 ($\subset_\mathcal{E}$) *Sei $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ mit $i = 1, 2$. Gilt $E_1 \subset E_2$, so schreiben wir $\mathcal{E}_1 \subset_\mathcal{E} \mathcal{E}_2$.* □

Offensichtlich ist $\subset_\mathcal{E}$ auf $\mathcal{ES}_P$ wohlfundiert, da die Ereignismenge von $\mathcal{E}$ endlich ist. Um zu zeigen, daß die Definition 3.10 korrekt ist, ist es nachzuweisen, daß $\mathcal{E}[\{e\}] \subset_\mathcal{E} \mathcal{E}$ und $Er(m, \mathcal{E}) \subset_\mathcal{E} \mathcal{E}$ gilt. Daß $\mathcal{E}[\{e\}] \subset_\mathcal{E} \mathcal{E}$ gilt, ist offensichtlich. Die Gültigkeit von $Er(m, \mathcal{E}) \subset_\mathcal{E} \mathcal{E}$ bestätigt das nachfolgende Lemma.

Lemma 3.5 *Sei $\mathcal{E} \in \mathcal{ES}_P$ und $M = init(\mathcal{E})$.*

1. *Falls $Konflikt(\mathcal{E})$ wahr ist und $|M| > 1$ gilt, dann gilt $Er(m, \mathcal{E}) \subset_\mathcal{E} \mathcal{E}$ für $m \in Kaus(\mathcal{E})$.*
2. *Falls $Konflikt(\mathcal{E})$ falsch ist und $|M| > 1$ gilt, dann gilt $Er(m, \mathcal{E}) \subset_\mathcal{E} \mathcal{E}$ für $m \in Konf(\mathcal{E})$.*

Beweis: Zu 1.: Nach Lemma 3.4 gilt $m \subset M$. Daraus folgt $Er(m, \mathcal{E}) \subset_\mathcal{E} \mathcal{E}$. Zu 2.: Gilt $Konflikt(\mathcal{E}) = falsch$, dann gilt $m \subset M$ und somit $Er(m, \mathcal{E}) \subset_\mathcal{E} \mathcal{E}$. □

Man beachte, daß in Definition 3.10 $\mathcal{E}[\{e\}], Er(m, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$ gilt. Daß $\mathcal{E}[\{e\}] \in \mathcal{ES}_\mathcal{P}$ gilt, ist offensichtlich. Die Gültigkeit von $Er(m, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$ wird später gezeigt.

Beispiel 3.3 Gegeben sei die Ereignisstruktur $\mathcal{E}$ aus Abbildung 1. Dann gilt:

1. $Term(\mathcal{E}) = \sum_{m \in Kaus(\mathcal{E})}(Term(Er(m, \mathcal{E})))$, wobei $Kaus(\mathcal{E}) = \{\{1, 2, 4, 5, 7\}, \{9\}\}$.
2. Sei $m_1 = \{1, 2, 4, 5, 7\}$, $m_2 = \{9\}$, $\mathcal{E}_1 = Er(m_1, \mathcal{E})$ und $\mathcal{E}_2 = Er(m_2, \mathcal{E})$. Dann gilt
   (a) $Term(\mathcal{E}_1) = \prod_{m \in Konf(\mathcal{E}_1)}(Term(Er(m, \mathcal{E}_1)))$
   (b) $Term(\mathcal{E}_2) = f; (Term(\mathcal{E}_2[\{9\}])) = f; (g; (\mathbf{stop}))$

   Dabei gilt $Konf(\mathcal{E}_1) = \{\{1, 2, 4, 5\}, \{7\}\}$.
3. Sei $m_{11} = \{1, 2, 4, 5\}$ und $m_{12} = \{7\}$. Sei ferner $\mathcal{E}_{11} = Er(m_{11}, \mathcal{E}_1)$ und $\mathcal{E}_{12} = Er(m_{12}, \mathcal{E}_1)$. Dann gilt
   (a) $Term(\mathcal{E}_{11}) = \sum_{m \in Kaus(\mathcal{E}_{11})}(Term(Er(m, \mathcal{E}_{11})))$

(b) $Term(\mathcal{E}_{12}) = c; (Term(\mathcal{E}_{12}[\{7\}])) = c; (d; (\mathbf{stop}))$

Dabei gilt $Kaus(\mathcal{E}_{11}) = \{\{1\}, \{2\}, \{4\}, \{5\}\}$. Dies bedeutet

$$Term(\mathcal{E}_{11}) = (a; (\mathbf{stop}))\ \overline{[]}\ (a; (c; (\mathbf{stop})))\ \overline{[]}\ (a; (\mathbf{stop}))\overline{[]}(a; (c; (\mathbf{stop}))).$$

Somit ergibt sich

$$\begin{aligned} Term(\mathcal{E}) \quad = \quad & (((a; (\mathbf{stop}))\ \overline{[]}\ (a; (c; (\mathbf{stop})))\ \overline{[]}\ (a; (\mathbf{stop}))\overline{[]}(a; (c; (\mathbf{stop})))) \\ & \overline{|||}\ (c(d; (\mathbf{stop})))) \\ & \overline{[]}\ (f; (g; (\mathbf{stop}))). \end{aligned}$$

□

3.2 Korrektheitsnachweis

In diesem Abschnitt wird die Gültigkeit der Aussage $\mathcal{E} = [\![Term(\mathcal{E})]\!]$ gezeigt (siehe Satz 3.2). Der Beweis erfolgt über mehrere Schritte. Dabei wird oft von der Gültigkeit der folgenden Aussage ausgegangen, wofür wir aufgrund der Einfachheit keinen Beweis angeben: Sei $\mathcal{E}, \mathcal{E}_i \in \mathcal{ES}_\mathcal{P}$ und $i = 1, 2$. Sei ferner $\mathcal{E} = \mathcal{E}_1\overline{[]}\mathcal{E}_2$ oder $\mathcal{E} = \mathcal{E}_1\overline{\cup}\mathcal{E}_2$, dann gilt $init(\mathcal{E}) = init(\mathcal{E}_1)\overline{\cup}init(\mathcal{E}_2)$.

Zur Erinnerung wird an dieser Stelle das Well-Founded-Induktionsbeweisprinzip aus [Win93] zitiert, das im Verlauf des Beweises und im weiteren Verlauf des Berichtes oft angewendet wird.

> Sei $\prec$ eine wohlfundierte Relation auf einer Menge A. Sei P eine Eigenschaft.
> $\forall a \in A : P(a)$ gilt genau dann, wenn gilt:
>
> $$\forall a \in A : ((\forall b \in A : b \prec a \wedge P(b)) \implies P(a)).$$

Strukturelle Induktionsbeweis auf der Menge der Terme oder vollständige Induktionbeweis auf der Menge der natürlichen Zahlen ist daher nur ein Speziallfall dieses Induktionsbeweisprinzips. Näheres dazu findet man in [Win93].

Lemma 3.6 *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$. Dann gilt:*

1. *E ist endlich.*
2. *$\rightsquigarrow$ ist symmetrisch.*
3. *Gilt $e \in init(\mathcal{E})$ und $e \rightsquigarrow e'$, so gilt $e' \in init(\mathcal{E})$.*
4. *Sei $e1 \rightsquigarrow e2$, $e1 \vdash e1'$ und $e2 \vdash e2'$, wobei $e1 \neq e1'$ und $e2 \neq e2'$ gilt, dann gilt $\neg(e1 \rightsquigarrow e2')$, $\neg(e1' \rightsquigarrow e2)$ und $\neg(e1' \rightsquigarrow e2')$.*
5. *Gilt $X \mapsto e$, dann gilt $|X| = 1$.*
6. *Gilt $X \mapsto e$ und $Y \mapsto e$, so gilt $X = Y$.*
7. *Gilt $\{e\} \mapsto e'$, dann gilt $e \neq e'$.*

8. *Gilt* $\{e\} \mapsto e'$ *und* $e' \rightsquigarrow e''$*, dann gilt* $\{e\} \mapsto e''$.

9. *Für alle* $e' \in E$ *gibt es genau ein* $e \in init(\mathcal{E})$*, so daß* $e \vdash e'$ *gilt.* □

Beweis: Induktion über die Teil-Ereignisstruktur-Relation $\subset_{\mathcal{E}}$. Der Beweis ist einfach und wird daher nicht ausgeführt. □

Es ist anzumerken, daß wegen der Eigenschaft 9 des Lemmas 3.6 $Er(init(\mathcal{E}), \mathcal{E}) = \mathcal{E}$ gilt.

Satz 3.1 *Sei* $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ *mit* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *und* $M = init(\mathcal{E})$.

1. *Gilt* $Konflikt(\mathcal{E}) = wahr$ *und* $|M| > 1$*, dann gilt*

$$\mathcal{E} = \overline{\sum_{m \in Kaus(\mathcal{E})}} Er(m, \mathcal{E}).$$

2. *Gilt* $Konflikt(\mathcal{E}) = falsch$ *und* $|M| > 1$*, dann gilt*

$$\mathcal{E} = \overline{\bigcup_{m \in Konf(\mathcal{E})}} Er(m, \mathcal{E}).$$

Beweis:

1. Sei

$$\mathcal{E}' = \overline{\sum_{m \in Kaus(\mathcal{E})}} Er(m, \mathcal{E}) = (E', \rightsquigarrow', \mapsto', l').$$

   Sei $Kaus(\mathcal{E}) = \{m_1, \ldots, m_n\}$ und $Er(m_i, \mathcal{E}) = \mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ für $i = 1, \ldots, n$, dann gilt wegen $m_i \cap m_j = \emptyset$ für $i, j \in \{1, \ldots, n\}$ und $i \neq j$ auch $E_i \cap E_j = \emptyset$ (wegen Lemma 3.6(9)). Damit läßt sich folgendes zeigen:

   - $E = \bigcup_{i=1}^{n} E_i = E'$ (wegen Lemma 3.6(9))
   - Gilt $e \rightsquigarrow e'$, so sind folgende Fälle zu unterscheiden:
     (a) Fall 1: $\exists E_i : e, e' \in E_i$
     In diesem Fall folgt sofort $e \rightsquigarrow' e'$.
     (b) Fall 2: $\exists E_i, E_j : (i \neq j \wedge e \in E_i \wedge e' \in E_j)$
     Im diesem Fall gibt es 4 weitere Fälle zu unterscheiden:
       i. Fall: $e \in m_i$ und $e' \in m_j$
       In diesem Fall gilt natürlich auch $e \rightsquigarrow' e'$.
       ii. Fall: $e \in m_i$ und $e' \notin m_j$
       Aus $e' \in E_j$ gibt es ein $e_j \in m_j$, so daß $e_j \vdash e'$ gilt. Da $e \rightsquigarrow e_j$ gilt (denn sonst gilt $e \smile_M e_j$ und somit $e_j \in m_i$. Widerspruch), gilt wegen Lemma 3.6(4) $\neg(e \rightsquigarrow e')$.
       iii. Fall: $e \notin m_i$ und $e' \in m_j$
       Analog zu ii. gilt $\neg(e \rightsquigarrow e')$.
       iv. Fall: $e \notin m_i$ und $e' \notin m_j$
       Analog zu ii. gilt $\neg(e \rightsquigarrow e')$.

Die Fälle ii., iii. und iv. können also nicht vorkommen. Aus (a) und (b) folgt $\leadsto \subseteq \leadsto'$.

Gilt umgekehrt $e \leadsto' e'$, dann sind zwei Fälle zu unterscheiden:

Fall 1: $\exists E_i : (e \in E_i \wedge e' \in E_i)$. In diesem Fall folgt sofort $e \leadsto e'$.

Fall 2: $\exists E_i, E_j : (i \neq j \wedge e \in E_i \wedge e' \in E_j)$. In diesem Fall gilt $e \in m_i$ und $e' \in m_j$, was $e \leadsto e'$ impliziert.

Insgesamt gilt $\leadsto' \subseteq \leadsto$.

- Offensichtlich gilt $\{e\} \mapsto e'$, falls $\{e\} \mapsto' e'$ gilt. Es gilt somit $\mapsto' \subseteq \mapsto$. Gilt umgekehrt $\{e\} \mapsto e'$, so sind folgende Fälle zu unterscheiden:

  (a) Fall 1: $\exists E_i : e, e' \in E_i$
  In diesem Fall folgt sofort $\{e\} \mapsto' e'$.

  (b) Fall 2: $\exists E_i, E_j : (i \neq j \wedge e \in E_i \wedge e' \in E_j)$
  Es sind 2 weitere Fälle zu unterscheiden:

    i. Fall: $e' \in m_j$
    In diesem Fall kann $\{e\} \mapsto e'$ nicht gelten, denn dann gilt $e' \notin M$, was $e' \notin m_j$ bedeutet.

    ii. Fall: $e' \notin m_j$
    In diesem Fall gibt es ein $e_j \in m_j$ mit $e_j \vdash e'$, was nach Lemma 3.6(6) $e = e_j$ bedeutet. Widerspruch. Also kann $\{e\} \mapsto e'$ nicht gelten.

  Aus (a) gilt $\mapsto \subseteq \mapsto'$.

- $l = l'$ (trivial)

Daraus folgt $\mathcal{E} = \mathcal{E}'$.

2. Für diesen Fall gilt es analog zu 1. zu zeigen.

Sei also $Konf(\mathcal{E}) = \{m_1, \ldots, m_n\}$ und $Er(m_i, \mathcal{E}) = \mathcal{E}_i = (E_i, \leadsto_i, \mapsto_i, l_i)$ für $i = 1, \ldots, n$, dann gilt wegen $m_i \cap m_j = \emptyset$ für $i, j \in \{1, \ldots, n\}$ und $i \neq j$ auch $E_i \cap E_j = \emptyset$ (wegen Lemma 3.6(9)). Sei ferner

$$\mathcal{E}' = \overline{\bigcup_{i=1}^{n}} \mathcal{E}_i = (E', \leadsto', \mapsto', l').$$

Dann gilt:

- $E = \bigcup_{i=1}^{n} E_i = E'$ (wegen Lemma 3.6(9))
- Gilt $e \leadsto e'$, so sind folgende Fälle zu unterscheiden:

  (a) Fall 1: $\exists E_i : e, e' \in E_i$
  In diesem Fall folgt sofort $e \leadsto' e'$.

  (b) Fall 2: $\exists E_i, E_j : (i \neq j \wedge e \in E_i \wedge e' \in E_j)$
  Im diesem Fall gibt es 4 weitere Fälle zu unterscheiden:

    i. Fall: $e \in m_i$ und $e' \in m_j$
    In diesem Fall gilt $e, e' \in m_i$. Widerspruch. Es gilt also $\neg(e \leadsto e')$.

ii. Fall: $e \in m_i$ und $e' \notin m_j$
Aus $e' \in E_j$ gibt es ein $e_j \in m_j$, so daß $e_j \vdash e'$ gilt. Anders geschrieben heißt das, daß es ein $\phi \in E_j$ gibt, so daß $e_j \vdash \phi$ und $\{\phi\} \mapsto e'$ gilt. Da $e' \rightsquigarrow e$ wegen Symmetrie von $\rightsquigarrow$ gilt, folgt aus Lemma 3.6(8) $\{\phi\} \mapsto e$. D.h. es gilt $e \notin M$ und somit $e \notin m_i$. Widerspruch. Es gilt also $\neg(e \rightsquigarrow e')$.

iii. Fall: $e \notin m_i$ und $e' \in m_j$
Analog zu ii. gilt $\neg(e \rightsquigarrow e')$.

iv. Fall: $e \notin m_i$ und $e' \notin m_j$
Mit analoger Begründung wie bei ii. und mit Lemma 3.6(9) gilt $\neg(e \rightsquigarrow e')$.

Der Fall (b) kann also nicht vorkommen. Aus (a) folgt $\rightsquigarrow \subseteq \rightsquigarrow'$. Die Umkehrung gilt offensichtlich. Insgesamt gilt $\rightsquigarrow' = \rightsquigarrow$.

- Analog zu 1. gilt $\mapsto = \mapsto'$.
- $l = l'$ (trivial)

Daraus folgt $\mathcal{E} = \mathcal{E}'$. □

Lemma 3.7 *Sei $\mathcal{E}, \mathcal{E}_i \in \mathcal{ES}_\mathcal{P}$ mit $|init(\mathcal{E}_i)| \geq 1$, wobei $i = 1,2$ gilt. Sei ferner $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ mit $i = 1,2$.*

1. *Gilt $\mathcal{E} = \mathcal{E}_1 \overline{[]} \mathcal{E}_2$, so gilt $Kaus(\mathcal{E}) = Kaus(\mathcal{E}_1) \cup Kaus(\mathcal{E}_2)$.*
2. *Gilt $\mathcal{E} = \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$, so gilt $Konf(\mathcal{E}) = Konf(\mathcal{E}_1) \cup Konf(\mathcal{E}_2)$.*

Beweis: Sei $M = init(\mathcal{E})$, $M^i = init(\mathcal{E}_i)$ und $i = 1,2$.

- Zu 1: Zunächst gilt offenbar $e' \in M^i$, falls $e \in M^i$, $\neg(e \rightsquigarrow e')$ und $e' \in M$ gilt. Daraus folgt $e'' \in M^i$, falls $e \smile_M e''$ gilt. Da $\neg(\phi \rightsquigarrow \phi')$ auch $\neg(\phi \rightsquigarrow_i \phi')$ für $\phi, \phi' \in M^i$ impliziert und umgekehrt, gilt

$$\psi \smile_M \psi' \Longleftrightarrow \psi \smile_{M^i} \psi'$$

mit $\psi \in M^i$. Daraus läßt sich leicht folgern, daß $Kaus(\mathcal{E}) = Kaus(\mathcal{E}_1) \cup Kaus(\mathcal{E}_2)$ gilt.

- Zu 2: Symmetrisch zu 1. □

Lemma 3.8 *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ und $M = init(\mathcal{E})$. Sei ferner $|M| > 1$.*

1. *Gilt $Konflikt(\mathcal{E}) = wahr$, dann gilt*

$$\mathcal{E} = \overline{\sum_{m \in Kaus(\mathcal{E})}} Er(m, \mathcal{E})$$

und $Er(m, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$.

2. Gilt $Konflikt(\mathcal{E}) = falsch$, dann gilt

$$\mathcal{E} = \overline{\bigcup_{m \in Konf(\mathcal{E})}} Er(m, \mathcal{E})$$

und $Er(m, \mathcal{E}) \in \mathcal{ES}_{\mathcal{P}}$.

Beweis: Wegen Satz 3.1 bleibt es nur noch zu zeigen, daß $Er(m, \mathcal{E}) \in \mathcal{ES}_{\mathcal{P}}$ gilt.

Wir beweisen mittels des Well-Founded-Induktionsprinzpis. Sei $L = \{\mathcal{E} \mid \mathcal{E} \in \mathcal{ES}_{\mathcal{P}} \wedge |init(\mathcal{E})| > 1\}$. $\subset_{\mathcal{E}}$ ist natürlich auch auf L wohlfundiert. Wir können daher per Induktion annehmen, daß die Aussagen 1 und 2 für $\mathcal{E}' \in L$ mit $\mathcal{E}' \subset_{\mathcal{E}} \mathcal{E}$ und $\mathcal{E} \in L$ bereits gelten. Daß dies auch für $\mathcal{E}$ gilt, zeigen wir wie folgt:

Da $\mathcal{E} \in L$ entweder von der Form $\mathcal{E} = \mathcal{E}_1 \overline{[]} \mathcal{E}_2$ oder von der Form $\mathcal{E} = \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$ ist, wobei wobei $\mathcal{E}_i \in \mathcal{ES}_{\mathcal{P}}$, $init(\mathcal{E}_i) \geq 1$ und $i = 1, 2$ gilt, gibt es nur zwei Fälle zu unterscheiden.

- Fall $\mathcal{E} = \mathcal{E}_1 \overline{[]} \mathcal{E}_2$

  1. Zu 1.: Wir unterscheiden vier Fälle:

     (a) $|init(\mathcal{E}_1)| > 1$ und $|init(\mathcal{E}_2)| > 1$
     Wir unterscheiden vier Fälle:

       i. $Konflikt(\mathcal{E}_i) = wahr$ für $i = 1, 2$
       Nach Lemma 3.7 gilt

       $$Kaus(\mathcal{E}) = Kaus(\mathcal{E}_1) \cup Kaus(\mathcal{E}_2).$$

       Da offenbar $Er(m, \mathcal{E}) = Er(m, \mathcal{E}_i)$ für $m \in Kaus(\mathcal{E}_i)$ und somit $Er(m, \mathcal{E}) \subset_{\mathcal{E}} \mathcal{E}$ gilt, gilt nach Induktionsannahme $Er(m, \mathcal{E}) \in \mathcal{ES}_{\mathcal{P}}$.

       ii. $Konflikt(\mathcal{E}_1) = wahr$ und $Konflikt(\mathcal{E}_2) = falsch$
       Nach Lemma 3.7 gilt

       $$Kaus(\mathcal{E}) = Kaus(\mathcal{E}_1) \cup Kaus(\mathcal{E}_2).$$

       Wegen Lemma 3.4 gilt $Kaus(\mathcal{E}_2) = \{init(\mathcal{E}_2)\}$. Da $Er(init(\mathcal{E}_2), \mathcal{E}_2) = \mathcal{E}_2$ gilt, folgt $Er(init(\mathcal{E}_2), \mathcal{E}_2) \in \mathcal{ES}_{\mathcal{P}}$. Für $m \in Kaus(\mathcal{E}_1)$ gilt nach Induktionsannahme $Er(m, \mathcal{E}) \in \mathcal{ES}_{\mathcal{P}}$.

       iii. $Konflikt(\mathcal{E}_1) = falsch$ und $Konflikt(\mathcal{E}_2) = wahr$
       Analog zu ii.

       iv. $Konflikt(\mathcal{E}_1) = falsch = Konflikt(\mathcal{E}_2)$
       Analog zu ii.

     (b) $init(\mathcal{E}_1) > 1$ und $init(\mathcal{E}_2) = 1$
     In diesem Fall gilt $\mathcal{E}_2 = \overline{e_g;} \mathcal{E}'_2$, wobei $\mathcal{E}'_2 \in \mathcal{ES}_{\mathcal{P}}$ gilt. Daraus folgt $Kaus(\mathcal{E}_2) = \{\{e\}\}$. Nach Lemma 3.7 gilt

     $$Kaus(\mathcal{E}) = Kaus(\mathcal{E}_1) \cup Kaus(\mathcal{E}_2).$$

     Für $m \in Kaus(\mathcal{E}_1)$ gilt nach Induktionsannahme $Er(m, \mathcal{E}) \in \mathcal{ES}_{\mathcal{P}}$. Für $m \in Kaus(\mathcal{E}_2)$ gilt $m = \{e\}$ und somit $Er(m, \mathcal{E}_2) = \mathcal{E}_2$, d.h. $Er(m, \mathcal{E}_2) \in \mathcal{ES}_{\mathcal{P}}$.

(c) $init(\mathcal{E}_1) = 1$ und $init(\mathcal{E}_2) > 1$
Analog zu (b).

(d) $init(\mathcal{E}_1) = 1 = init(\mathcal{E}_2)$
Analog zu (b).

2. Zu 2.: Die Aussage 2 gilt immer, da $Konflikt(\mathcal{E})$ wahr ist.

- Fall $\mathcal{E} = \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$

  1. Zu 1.: Die Aussage 1 gilt immer, da $Konflikt(\mathcal{E})$ falsch ist.
  2. Zu 2.: Symmetrisch zu 1 bei $\mathcal{E} = \mathcal{E}_1 \overline{[]} \mathcal{E}_2$. □

Nun sind wir in der Lage, die Gültigkeit von $[\![Term(\mathcal{E})]\!] = \mathcal{E}$ zu zeigen.

Satz 3.2 *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$. Dann gilt $[\![Term(\mathcal{E})]\!] = \mathcal{E}$.*

Beweis: Da die Definition 3.10 eine induktive Definition ist, ist der folgende Beweis ein Induktionsbeweis über die Teil-Ereignisstruktur-Relation $\subset_\mathcal{E}$. Sei also $\mathcal{E}, \mathcal{E}' \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E}' \subset_\mathcal{E} \mathcal{E}$, dann nehmen wir an, daß $[\![Term(\mathcal{E}')]\!] = \mathcal{E}'$ bereits gilt. Sei $\mathcal{E} = (E, \leadsto, \mapsto, l)$ und $M = init(\mathcal{E})$.

1. Gilt $M = \emptyset$, so gilt $\mathcal{E} = (\emptyset, \emptyset, \emptyset, \emptyset)$ und somit $[\![Term(\mathcal{E})]\!] = \mathcal{E}$, da $Term(\mathcal{E}) = \mathbf{stop}$ gilt.

2. Für den Fall $M = \{e\}$ gilt $\mathcal{E} = \overline{l(e)_e;}\mathcal{E}[\{e\}]$, wobei $\mathcal{E}[\{e\}] \in \mathcal{ES}_\mathcal{P}$ gilt. Da nach Induktionsannahme $\mathcal{E}[\{e\}] = [\![Term(\mathcal{E}[\{e\}])]\!]$ gilt, folgt $\mathcal{E} = [\![Term(\mathcal{E})]\!]$.

3. Für den Fall, daß $Konflikt(\mathcal{E}) = wahr$ und $|M| > 1$ gilt, gilt nach Lemma 3.8

$$\mathcal{E} = \overline{\sum_{m \in Kaus(\mathcal{E})}} Er(m, \mathcal{E})$$

   und $Er(m, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$. Nach Induktionsannahme gilt $[\![Term(Er(m, \mathcal{E}))]\!] = Er(m, \mathcal{E})$ und somit $\mathcal{E} = [\![Term(\mathcal{E})]\!]$.

4. Für den Fall, daß $Konflikt(\mathcal{E}) = faslch$ und $|M| > 1$ gilt, gilt nach Lemma 3.8

$$\mathcal{E} = \overline{\bigcup_{m \in Konf(\mathcal{E})}} Er(m, \mathcal{E})$$

   und $Er(m, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$. Nach Lemma 3.2 gilt

$$\mathcal{E} \cong \overline{\prod_{m \in Konf(\mathcal{E})}} Er(m, \mathcal{E}).$$

   Nach Induktionsannahme gilt $[\![Term(Er(m, \mathcal{E}))]\!] = Er(m, \mathcal{E})$. Daraus folgt $\mathcal{E} = [\![Term(\mathcal{E})]\!]$. □

Korollar 3.3 *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$. Dann gibt es einen Prozeß $B \in \mathcal{BL}_\mathcal{P}$, so daß $[\![B]\!] = \mathcal{E}$ gilt.*

Beweis: Es ist offensichtlich, daß $Term(\mathcal{E}) \in \mathcal{BL}_\mathcal{P}$ gilt. Aus Satz 3.2 folgt sofort die Behauptung. □

4 Eine Methode zur maximalen Prozeßzerlegung

In diesem Abschnitt wird eine Methode zur maximalen Prozeßzerlegung vorgestellt. Damit wird ein Prozeß C aus einem vorgegebenen Prozeß B bestimmt, der folgendes erfüllt: 1) $C \sim B$. 2) C ist von der Form $C = C_1 \;|||\; C_2 \ldots |||\; C_n$, wobei $C_i \not\sim$ **stop** gilt und n maximal ist. 3) Alle Folgezustände von C sind nicht weiter zerlegbar, d.h. sei $D \in Er(C)$, dann ist D von der Form $D = D_1 \,|||\, D_2 \ldots |||\; D_m$, wobei $D_i \not\sim$ **stop** gilt und m maximal ist.

Da die Zerlegungsmethode auf semantischer Ebene basiert, wird im ersten Teilabschnitt das Zerlegungsproblem in Form von Ereignisstrukturen formuliert. Im zweiten Teilabschnitt wird die Zerlegungsmethode vorgestellt und anhand von Beispielen erläutert. Der dritte Teilabschnitt befaßt sich mit dem Korrektheitsnachweis der Zerlegungsmethode.

4.1 Zerlegungsproblem

Um angeben zu können, aus wievielen Teilprozessen ein Prozeß besteht, die parallel unabhängig voneinander ablaufen, führen wir zunächst den Begriff „Zerlegungsgrad“ ein. Damit lassen sich zwei Ereignisstrukturen vergleichen.

Definition 4.1 (Zerlegungsgrad) *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$, dann ist $Zer(\mathcal{E}) := |Konf(\mathcal{E})|$ der Zerlegungsgrad von $\mathcal{E}$.* □

Sei beispielsweise $B = a;\mathbf{stop} \,|||\, b;\mathbf{stop} \,|||\, c;\mathbf{stop}$. Dann gilt $Zer([\![B]\!]) = 3$. Mit Induktion über den Termaufbau von B läßt sich leicht zeigen, daß $Zer([\![B]\!])$ tatsächlich die Gesamtanzahl der Teilprozesse in B angibt, die parallel unabhängig voneinander ablaufen können.

Definition 4.2 ($\sqsubseteq$) *Sei $\mathcal{E}_1, \mathcal{E}_2 \in \mathcal{ES}_{\mathcal{P}}$.*

1. *Gilt $\mathcal{E}_1 \approx_E \mathcal{E}_2$ und $Zer(\mathcal{E}_1) \leq Zer(\mathcal{E}_2)$, so schreiben wir $\mathcal{E}_1 \sqsubseteq \mathcal{E}_2$.*
2. *Sei $B_1, B_2 \in \mathcal{BL}_{\mathcal{P}}$. Gilt $[\![B_1]\!] \sqsubseteq [\![B_2]\!]$, dann schreiben wir auch $B_1 \sqsubseteq B_2$.* □

Nach Definition 2.11 enthält $\mathcal{E}[\{e\}]$ im allgemeinen sogenannte nicht ausführbare Ereignisse, d.h. Ereignisse, die nach Ausführung von e niemals stattfinden können. Daher ist $\mathcal{E}[\{e\}]$ nicht immer ein Element aus $\mathcal{ES}_{\mathcal{P}}$. In $\mathcal{ES}_{\mathcal{P}}$ kommen nämlich Ereignisstrukturen vor, in denen nicht ausführbare Ereignisse vollkommen ausgeschlossen sind. Da der Zerlegungsgrad nur für $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ definiert ist, ist $Zer(\mathcal{E}[\{e\}])$ i.a. nicht definiert. Es ist daher erforderlich, eine Ereignisstruktur zu definieren, die zu $\mathcal{E}[\{e\}]$ im Sinne von $\approx_E$ äquivalent ist und die sowohl keine nicht ausführbare Ereignisse enthält als auch in $\mathcal{ES}_{\mathcal{P}}$ liegt. Diese Ereignisstruktur wird im folgenden durch $Rest(e, \mathcal{E})$ wiedergegeben.

Definition 4.3 ($Rest(e, \mathcal{E})$) *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ und $e \in init(\mathcal{E})$. Dann ist $Rest(e, \mathcal{E}) := (E', \rightsquigarrow', \mapsto', l')$ mit*

- *$E' = E \setminus (\{e\} \cup m \cup \{e'' \mid e' \vdash e'' \wedge e' \in m\})$, wobei $m = \{e' \mid e \rightsquigarrow e'\}$.*
- *$\rightsquigarrow' = \rightsquigarrow \cap (E' \times E')$*

- $\mapsto' = \mapsto \cap (2^{E'} \times E')$
- $l' = l\lceil E'$ $\square$

Später wird gezeigt, daß tatsächlich $Rest(e, \mathcal{E}) \approx_E \mathcal{E}[\{e\}]$ und $Rest(e, \mathcal{E}) \in \mathcal{ES}_{\mathcal{P}}$ gilt. Analog zu $Er(\mathcal{E})$ führen wir nun die Menge $REr(\mathcal{E})$ ein, in denen nur Ereignisstrukturen der Form $Rest(.,..)$ vorkommen.

Definition 4.4 ($REr(\mathcal{E})$) *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$. $REr(\mathcal{E})$ ist die kleinste Menge, für die gilt:*

- $\mathcal{E} \in REr(\mathcal{E})$
- *Sei $\mathcal{E}' \in REr(\mathcal{E})$. Gilt $\mathcal{E}' \stackrel{e}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}''$, dann gilt $Rest(e, \mathcal{E}') \in REr(\mathcal{E})$.* $\square$

Das Zerlegungsproblem lautet wie folgt:

- Gegeben sei $B \in \mathcal{BL}$.
- Gesucht ist ein Prozeß $C \in \mathcal{BL}_{\mathcal{P}}$ mit $B \sqsubseteq C$, falls C existiert, so daß gilt:
    1. $\forall B' \in \mathcal{BL}_{\mathcal{P}} : (C \sim B' \Longrightarrow B' \sqsubseteq C)$
    2. Sei $\mathcal{E} = [\![C]\!]$ und $\mathcal{E}' \in REr(\mathcal{E})$, dann gilt $\forall \mathcal{E}'' \in \mathcal{ES}_{\mathcal{P}} : (\mathcal{E}'' \approx_E \mathcal{E}' \Longrightarrow \mathcal{E}'' \sqsubseteq \mathcal{E}')$.

Die Bedingung 2 besagt anschaulich, daß jeder Folgezustand $C' \in Er(C)$, dargestellt durch $\mathcal{E}'$, genauer $\mathcal{E}' = [\![C]\!]$, auch einen maximalen Zerlegungsgrad besitzt. D.h. gilt $B'' \sim C'$, dann gilt $B'' \sqsubseteq C'$. Daß $\exists \mathcal{E}' \in REr(\mathcal{E}) : [\![C']\!] = \mathcal{E}'$ gilt, läßt sich leicht mit Induktion über den Termbaufbau von C zeigen. Im folgenden wird gezeigt, daß C stets existiert.

4.2 Zerlegungsmethode

Die Zerlegung erfolgt in drei Schritten:

1. Schritt 1:

   Der vorgegebene Prozeß B wird mit Hilfe einer Funktion namens APF in eine sogenannte Action-Prefix-Form gebracht, bezeichnet als $APF(B)$, die parallele Kompositionen nicht enthält. D.h. nur Prefix- oder Choice-Operatoren dürfen in $APF(B)$ vorkommen. Es gilt $B \sim APF(B)$.

2. Schritt 2:

   Aus $APF(B)$ wird mittels einer Funktion EZ eine Ereignisstruktur $\mathcal{E}$, d.h. $\mathcal{E} = EZ(APF(B)))$, erzeugt, die die Bedigung 1 und 2 des Zerlegungsproblems erfüllt.

3. Schritt 3:

   $Term(\mathcal{E})$ wird bestimmt. $Term(\mathcal{E})$ ist dann der gesuchte Prozeß des Zerlegungsproblems.

Dabei sind APF und EZ effektiv berechenbar. Das folgende Diagramm soll die drei Schritte anschaulich wiedergeben:

$$\begin{tikzcd}
B \arrow[r, "APF"] \arrow[d, phantom, "\sim"] & APF(B) \arrow[d, "EZ"] \\
Term(EZ(APF(B))) & EZ(APF(B)) \arrow[l, "Term"]
\end{tikzcd}$$

Zur übersichlichen Darstellung der Methode werden die grundlegenden Definitionen, aus denen die Methode sich zusammensetzt, sowie die Methode selbst mit Beispielen jeweils in einem eigenen Abschnitt zusammengefaßt. Bei den Definitionen handelt es sich vor allem um Operatoren, die auf $\mathcal{ES}_{\mathcal{P}}$ definiert sind.

4.2.1 Grundlegende Definitionen

Im folgenden wird zunächst die Funktion APF definiert. Anschließend werden die Operationen auf Ereignisstrukturen angegeben und an Beispielen anschaulich erläutert.

Definition 4.5 (Action-Prefix-Form) *Sei $B \in \mathcal{BL}$ und I eine endliche Indexmenge. B ist eine Action-Prefix-Form, wenn $B = \sum_{i \in I} a_i; B_i$ gilt, wobei B_i eine Action-Prefix-Form ist.*

Eine Action-Prefix-Form enthält also keine parallele Komposition . Zur Erinnerung gilt $B =$ **stop**, falls $I = \emptyset$ gilt. Daß es zu jedem Prozeß $B \in \mathcal{BL}$ eine Action-Prefix-Form B' mit $B \sim B'$ gibt, läßt sich mit Hilfe des sogenannten Expansionstheorems zeigen.

Theorem 4.1 (Expansionstheorem) *Sei $B, C \in \mathcal{BL}$ mit $B = \sum_{i \in I} b_i; B_i$ und $C = \sum_{j \in J} c_j; C_j$. Sei ferner G eine Menge von Aktionen. Dann gilt*

$$\begin{aligned} B \;|[G]|\; C \quad \sim \quad & \sum_{b_i \notin G} b_i; (B_i \;|[G]|\; C) \\ & [] \sum_{c_j \notin G} c_j; (B \;|[G]|\; C_j) \\ & [] \sum_{b_i = c_j \in G} a; (B_i \;|[G]|\; C_j) \ \mathit{mit}\ a = b_i = c_j \end{aligned}$$

Beweis: siehe [Mil89], [Bol92] □

Damit läßt sich die obengenannte Funktion APF leicht konstruieren. Doch zuvor ist noch eine weitere Definition notwendig.

Definition 4.6 ($PF(B)$) *Sei $B, C, D \in \mathcal{BL}$ mit $B = C \;|[G]|\; D$, wobei $C = \sum_{i \in I} c_i; C_i$ und $D = \sum_{j \in J} d_j; D_j$ Action-Prefix-Formen sind. Sei ferner $G_1 = \{c_i \mid i \in I\}$ und $G_2 = \{d_j \mid j \in J\}$. Dann*

1.

$$\begin{aligned} PF(B) \ := \ & \sum_{c_i \notin G} c_i; PF(C_i \ |[G]| \ D) \\ & [] \ \sum_{d_j \notin G} d_j; PF(C \ |[G]| \ D_j) \\ & [] \ \sum_{c_i = d_j \in G} a; PF(C_i \ |[G]| \ D_j) \ mit \ a = c_i = d_j, \end{aligned}$$

falls $G_1 \setminus G \neq \emptyset$ *und* $G_2 \setminus G \neq \emptyset \neq G_1 \cap G_2 \cap G$ *gilt.*

2. $PF(B) := \sum_{c_i \notin G} c_i; PF(C_i \ |[G]| \ D) \ [] \sum_{d_j \notin G} d_j; PF(C \ |[G]| \ D_j)$,

   falls $G_1 \setminus G \neq \emptyset$ *und* $G_2 \setminus G \neq \emptyset = G_1 \cap G_2 \cap G$ *gilt.*

3. $PF(B) := \sum_{c_i \notin G} c_i; PF(C_i \ |[G]| \ D) \ [] \sum_{c_i = d_j \in G} a; PF(C_i \ |[G]| \ D_j) \ mit \ a = c_i = d_j$,

   falls $G_1 \setminus G \neq \emptyset$ *und* $G_2 \setminus G = \emptyset \neq G_1 \cap G_2 \cap G$ *gilt.*

4. $PF(B) := \sum_{c_i \notin G} c_i; PF(C_i \ |[G]| \ D)$,

   falls $G_1 \setminus G \neq \emptyset$ *und* $G_2 \setminus G = \emptyset = G_1 \cap G_2 \cap G$ *gilt.*

5. $PF(B) := \sum_{d_j \notin G} d_j; PF(C \ |[G]| \ D_j) \ [] \sum_{c_i = d_j \in G} a; PF(C_i \ |[G]| \ D_j) \ mit \ a = c_i = d_j$,

   falls $G_1 \setminus G = \emptyset$ *und* $G_2 \setminus G \neq \emptyset \neq G_1 \cap G_2 \cap G$ *gilt.*

6. $PF(B) := \sum_{d_j \notin G} d_j; PF(C \ |[G]| \ D_j)$,

   falls $G_1 \setminus G = \emptyset$ *und* $G_2 \setminus G \neq \emptyset = G_1 \cap G_2 \cap G$ *gilt.*

7. $PF(B) := \sum_{c_i = d_j \in G} a; PF(C_i \ |[G]| \ D_j) \ mit \ a = c_i = d_j$,

   falls $G_1 \setminus G = \emptyset$ *und* $G_2 \setminus G = \emptyset \neq G_1 \cap G_2 \cap G$ *gilt.*

8. $PF(B) :=$ **stop**, *falls* $G_1 \setminus G = G_2 \setminus G = G_1 \cap G_2 \cap G = \emptyset$ *gilt.* □

Lemma 4.1

1. *$PF(B)$ ist eine Action-Prefix-Form.*

2. $PF(B) \sim B$

Beweis:

- Zu 1: Wir beweisen mit Hilfe des Well-Founded-Induktionsprinzips. Sei $L \subset \mathcal{BL}$ mit $L = \{P \ |[G]| \ Q \mid \ P$ und Q sind Action-Prefix-Formen$\}$. Sei $\prec$ auf L wie folgt definiert: $P' \prec P$, falls $P = U \ |[G]| \ V$ und $P' = U' \ |[G]| \ V'$ gilt, wobei

  – $U = \sum_{i \in I} c_i; U_i$ und $V = \sum_{j \in J} d_j; V_j$ Action-Prefix-Formen sind.

– $P' \neq P$ gilt.

– $U' \in \{U\} \cup \{U_i \mid i \in I\}$ und $V' \in \{V\} \cup \{V_j \mid j \in J\}$ gilt.

Offenbar ist $\prec$ auf L wohlfundiert. Um zu zeigen, daß $PF(P)$ mit $P \in L$ eine Aktion-Prefix-Form ist, nehmen wir per Induktion an, daß $PF(P')$ mit $P' \prec P$ und $P' \in L$ breits eine Action-Prefix-Form ist.

Es sind vier Fälle zu unterscheiden:

1. $P = U \;|[G]|\; V$ mit $U = \mathbf{stop}$ und $U = \mathbf{stop}$
2. $P = U \;|[G]|\; V$ mit $U = \mathbf{stop}$ und $U \neq \mathbf{stop}$
3. $P = U \;|[G]|\; V$ mit $U \neq \mathbf{stop}$ und $U = \mathbf{stop}$
4. $P = U \;|[G]|\; V$ mit $U \neq \mathbf{stop}$ und $U \neq \mathbf{stop}$

Zu 1: Es gilt offenbar, daß $PF(P)$ eine Action-Prefix-Form ist. Zu 2,3,4: Mit Induktionsannahmen läßt sich leicht zeigen, daß $PF(P)$ auch eine Action-Prefix-Form ist. Da $B \in L$ gilt, ist $PF(B)$ eine Action-prefix-Form

- Zu 2: Seien L und $\prec$ so definiert wie bei 1. Für $P \in L$ nehmen wir per Induktion an, daß $PF(P') \sim P'$ gilt, wenn $P' \prec P$ und $P' \in L$ gilt. Es sind wie bei 1 vier Fälle zu unterscheiden. Mit Theorem 4.1 und Satz 2.1 läßt sich dann leicht folgern, daß $PF(P) \sim P$ bei allen vier Fällen gilt. Da $B \in L$ gilt, gilt $PF(B) \sim B$. □

Definition 4.7 **(**$APF(B)$**)** *Sei $B \in \mathcal{BL}$, dann ist $APF(B)$ induktiv wie folgt definiert:*

1. *Ist $B = \mathbf{stop}$, dann $APF(B) := \mathbf{stop}$.*
2. *Ist $B = g; B'$, dann $APF(B) := g; APF(B')$.*
3. *Ist $B = B_1 \,[]\, B_2$, dann*
   (a) *$APF(B) := APF(B_1) \,[]\, APF(B_2)$, falls $APF(B_1) \neq \mathbf{stop} \neq APF(B_2)$.*
   (b) *$APF(B) := APF(B_2)$, falls $APF(B_1) = \mathbf{stop} \neq APF(B_2)$.*
   (c) *$APF(B) := APF(B_1)$, falls $APF(B_1) \neq \mathbf{stop} = APF(B_2)$.*
   (d) *$APF(B) := \mathbf{stop}$, falls $APF(B_1) = \mathbf{stop} = APF(B_2)$.*
4. *Ist $B = B_1 \;|[G]|\; B_2$, dann $APF(B) = PF(APF(B_1) \;|[G]|\; APF(B_2))$.* □

Beispiel 4.1 *Sei $B = a; b; \mathbf{stop}|[b]|c; b; a; \mathbf{stop}$, dann ist*

$$APF(B) = a; c; b; a; \mathbf{stop} \,[]\, c; a; b; a; \mathbf{stop}.$$

□

Korollar 4.1 *Sei $B \in \mathcal{BL}$. Dann gilt:*

1. *$APF(B)$ ist eine Action-Prefix-Form.*
2. *$APF(B) \sim B$*

Beweis:

- <u>Zu 1:</u> Wir beweisen mit Hilfe der Induktion über den Termaufbau von B. Für den Fall $B = \mathbf{stop}$, $B = g; B'$ und $B = B_1 \, [] \, B_2$ ist offenbar $APF(B)$ eine Action-Prefix-Form, da nach Induktionsannahme $APF(B')$, $APF(B_1)$ und $APF(B_2)$ Action-Prefix-Formen sind. Für den Fall $B = B_1 \, |[G]| \, B_2$ ist auch $APF(B)$ wegen Lemma 4.1 eine Action-Prefix-Form.

- <u>Zu 2:</u> Es läßt sich mit Hilfe der Induktion über den Termaufbau von B, des Lemmas 4.1 und des Satzes 2.1 leicht zeigen. Der Beweis ist einfach und wird deshalb nicht angegeben. □

Es ist anzumerken, daß es zu jedem Prozeß $B \in \mathcal{BL}$ unendlich viele äquivalente Action-Prefix-Formen gibt. Hat man nämlich eine Action-Prefix-Form C gefunden, so erhält man eine neue äquivalente Action-Prefix-Form durch Verknüpfen von C mit sich selbst bezüglich [], d.h. $C \, [] \, C$. Eine Action-Prefix-Form C ist i.a. also nicht minimal, d.h. sie kann Teilprozesse enthalten, deren Entfernen das beobachtbare Verhalten von C nicht verändert. Da diese Teilprozesse die Laufzeit des Zerlegungsalgorithmus maßgeblich beeinträchtigen können, ist es erforderlich, eine neue Klasse von Prozessen einzuführen, die nur die nötigen Prozesse als Teilprozesse beinhalten. Ein Prozeß, der dies erfüllt, wird im folgenden als die Normalform bezeichnet.

Definition 4.8 (Normalform) *Sei $B \in \mathcal{BL}$. B ist eine Normalform, wenn gilt:*

- $B = \sum_{i \in I} a_i; B_i$.
- *Ist $I \neq \emptyset$, d.h. $B \neq \mathbf{stop}$, dann gilt:*
  - $B_i \not\sim B_j$, *wenn $a_i = a_j$ für $i, j \in I$ gilt.*
  - B_i *ist eine Normalform für $i \in I$.* □

Eine Normalform ist also auch eine Action-Prefix-Form. Der Unterschied ist, daß es in einer Normalform keine zwei Übergänge gibt, die mit identischen Aktionen markiert sind und die in äquivalente Folgezustände enden.

Definition 4.9 ($\equiv$) *Sei $B, C \in \mathcal{BL}$ mit $B = b; B'$ und $C = c; C'$. Wir schreiben $B \equiv C$, falls $b = c$ und $B' \sim C'$ gilt.* □

Offenbar ist $\equiv$ eine Äquivalenzrelation.

Definition 4.10 ($NF(B)$) *Sei $B \in \mathcal{BL}$ eine Action-Prefix-Form mit $B = \sum_{i \in I} b_i; B_i$ und $M = \{b_i; B_i \mid i \in I\}$. Dann ist $NF(B)$ wie folgt definiert:*

- *Ist $I = \emptyset$, dann $NF(B) := \mathbf{stop}$.*
- *Ist $I \neq \emptyset$, d.h. $B \neq \mathbf{stop}$, dann $NF(B) := \sum\limits_{b_i; B_i \in M'} b_i; NF(B_i)$. Dabei ist $M' \subseteq M$ ein Repräsentensystem für M bezüglich $\equiv$.* □

Lemma 4.2

1. $NF(B)$ *ist eine Normalform.*

2. $NF(B) \sim B$.

Beweis:

- Zu 1: Wir beweisen mit Hilfe des Well-Founded-Induktionsprinzips. Sei

  $$L = \{P \mid P \text{ ist eine Action-Prefix-Form}\}$$

  Sei $\prec$ auf L wie folgt definiert: $P' \prec P$, falls $P = \sum_{i \in I} g_i; P_i$, wobei $I \neq \emptyset$ gilt, und $P' = P_i$ mit $i \in I$ gilt. Offenbar ist $\prec$ wohlfundiert. Um zu zeigen, daß $NF(P)$ für $P \in L$ eine Normalform ist, nehmen wir per Induktion an, daß $NF(P')$ mit $P' \prec P$ eine Normalform ist. Es sind zwei Fälle zu unterscheiden: 1) $P = \textbf{stop}$ 2) $P = \sum_{i \in I} g_i; P_i$ mit $I \neq \emptyset$. Im ersten Fall ist $NF(P) = \textbf{stop}$ und somit eine Normalform. Im zweiten Fall läßt sich mit Induktionsannahmen sofort folgern, daß auch $NF(P)$ eine Normalform ist. Da $B \in L$ gilt, ist $NF(B)$ eine Normalform.

- Zu 2: Sei L und $\prec$ so definiert wie bei 1. Für $P \in L$ nehmen wir per Induktion an, daß $NF(P') \sim P'$ gilt, falls $P' \prec P$ gilt. Es sind zwei Fälle zu unterscheiden: 1) $P = \textbf{stop}$. 2) $P = \sum_{i \in I} g_i; P_i$ mit $I \neq \emptyset$. Im ersten Fall gilt $NF(B) = \textbf{stop}$ und somit $NF(B) \sim \textbf{stop}$. Im zweiten Fall folgt mit Satz 2.1 sofort $NF(P) \sim P$. Da $B \in L$ gilt, gilt $NF(B) \sim B$. □

Nachdem der Weg, wie aus einem Prozeß $B \in \mathcal{BL}$ eine Action-Prefix-Form bestimmt wird, bereits erläutert wurde, ist es jetzt der nächste Schritt, die für die Zerlegungsmethode erforderlichen Operatoren zu definieren und zu erläutern. Ein davon ist die Umbenennung von Ereignissen in einer Ereignisstruktur $\mathcal{E}$.

Definition 4.11 ($e(\mathcal{E})$) *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 4.12 ($Komp(M)$) *Sei $M = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ eine nicht leere Menge von extended bundle event structures. Dann ist*

$$Komp(M) := 1(\mathcal{E}_1) \; \overline{[]} \; 2(\mathcal{E}_1) \cdots \overline{[]} \; n(\mathcal{E}_n).$$

□

Definition 4.13 ($\propto$) *Sei $\mathcal{E}_i \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ und $i = 1, 2$. Dann ist $\mathcal{E}_1 \propto \mathcal{E}_2$, wenn gilt:*

1. *Gilt* $\mathcal{E}_1 \overset{e_1}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1'$, *dann*

$$\exists e_2 \in E_2 : (l_1(e_1) = l_2(e_2) \wedge \mathcal{E}_2 \overset{e_2}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_2' \wedge Rest(e_1, \mathcal{E}_1) \cong Rest(e_2, \mathcal{E}_2)).$$

2. *Es gilt* $\exists e_2 \in E_2 : (\mathcal{E}_2 \overset{e_2}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_2'$ *und*

$$\neg(\exists e_1 \in E_1 : l_1(e_1) = l_2(e_2) \wedge \mathcal{E}_1 \overset{e_1}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}_1' \wedge Rest(e_1, \mathcal{E}_1) \cong Rest(e_2, \mathcal{E}_2))).$$

□

$\mathcal{E}_1 \propto \mathcal{E}_2$ bedeutet anschaulich, daß $\mathcal{E}_1$ von $\mathcal{E}_2$ simuliert wird, aber nicht umgekehrt. Man beachte, daß die Gültigkeit von $Rest(e_1, \mathcal{E}_1) \cong Rest(e_2, \mathcal{E}_2))$ verlangt wird und nicht von $\mathcal{E}_1' \approx_E \mathcal{E}_2'$ wie bei $\approx_E$.

Definition 4.14 ($Opt(M)$) *Sei* $M = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ *und* $M' \subseteq M$ *ein Repräsentantensystem für* M *bezüglich* $\cong$. *Dann ist*

$$Opt(M) := \{\mathcal{E} \mid \ \mathcal{E} \in M' \wedge \neg(\exists \mathcal{E}' \in M' : \mathcal{E} \propto \mathcal{E}')\}.$$

□

Mit $Opt(M)$ werden diejenigen Ereignisstrukturen in M entfernt, die bereits durch andere Ereignisstrukturen simuliert werden. Außerdem sind zwei beliebige Ereignisstrukturen in $Opt(M)$ paarweise nicht isomorph.

Definition 4.15 ($Bed1((e_1, e, \mathcal{E}_1), f, (e_2, e', \mathcal{E}_2))$, $Kons(e, \mathcal{E}_1, e', \mathcal{E}_2)$)

Sei $\mathcal{E}_i \in \mathcal{ES}_{\mathcal{P}}$ *mit* $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ *und* $i = 1, 2$. *Seien weiterhin*

- e_1 *und* e_2 *Ereignisse mit* $e_1 \neq e_2$ *und* $f : M \rightarrow \mathcal{G}$ *eine Funktion, wobei* $e_1, e_2 \in M$ *gilt.*
- $e \in init(\mathcal{E}_1)$ *und* $e' \in init(\mathcal{E}_2)$.
- $m1 \in Konf(\mathcal{E}_1)$ *mit* $e \in m1$ *und* $m2 \in Konf(\mathcal{E}_2)$ *mit* $e' \in m2$.

Dann:

1. $Bed1((e_1, e, \mathcal{E}_1), f, (e_2, e', \mathcal{E}_2))$ *ist genau dann wahr, wenn gilt:*
   (a) $f(e_1) = l_2(e') \wedge l_1(e) = f(e_2)$
   (b) *Es gibt eine Menge* $N1 \subset Konf(\mathcal{E}_1)$ *mit* $m1 \notin N1$, *so daß gilt:*

   $$\overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}_1) \cong Rest(e', Er(m2, \mathcal{E}_2))$$

   (c) *Es gibt eine Menge* $N2 \subset Konf(\mathcal{E}_2)$ *mit* $m2 \notin N2$, *so daß gilt:*

   $$\overline{\bigcup_{m \in N2}} Er(m, \mathcal{E}_2) \cong Rest(e, Er(m1, \mathcal{E}_1))$$

(d) *Sei* $L1 = Konf(\mathcal{E}_1) \setminus (m1 \cup N1)$ *und* $L2 = Konf(\mathcal{E}_2) \setminus (m2 \cup N2)$, *dann*

$$\overline{\bigcup_{m \in L1}} Er(m, \mathcal{E}_1) \cong \overline{\bigcup_{m \in L2}} Er(m, \mathcal{E}_2)$$

2. *Ist* $Bed1((e_1, e, \mathcal{E}_1), f, (e_2, e', \mathcal{E}_2))$ *wahr, dann ist*

$$Kons(e, \mathcal{E}_1, e', \mathcal{E}_2) := 1(\overline{\bigcup_{m \in H}} Er(m, \mathcal{E}_1)) \,\overline{\cup}\, 2(Er(m2, \mathcal{E}_2)),$$

wobei $H = Konf(\mathcal{E}_1) \setminus N1$. □

Die Bedingungen 1(b), 1(c) und 1(d) der Definition 4.15 haben folgende Bedeutung: Sei

- $Konf(\mathcal{E}_1) = \{init1_1, \ldots, init1_n\}$, $Konf(\mathcal{E}_2) = \{init2_1, \ldots, init2_k\}$,
- $N1 = \{init1_1, \ldots, init1_i\}$, $N2 = \{init2_{j+1}, \ldots, init2_k\}$
- $m1 = init1_n$ und $m2 = init2_1$,

so erhält man die Isomorphie-Beziehung zwischen den Ereignisstrukturen wie in Abbildung 2 dargestellt. Die gestrichelte Linie weist dabei auf die Bedingung 1(b) hin, die

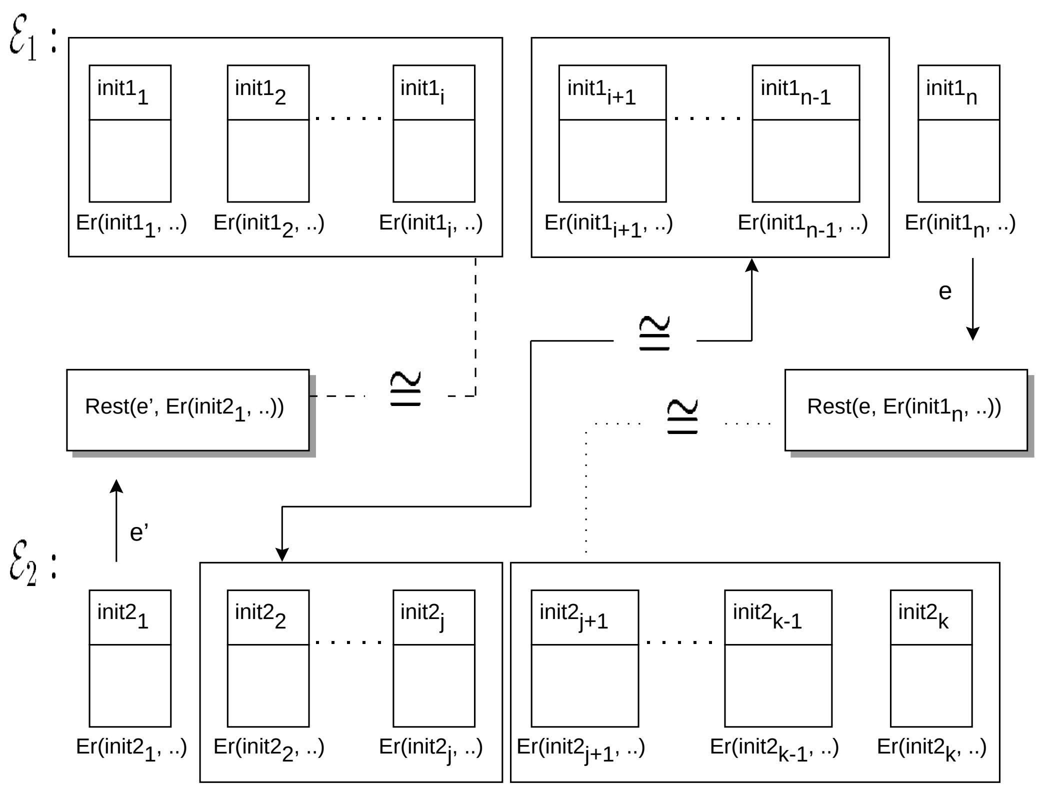
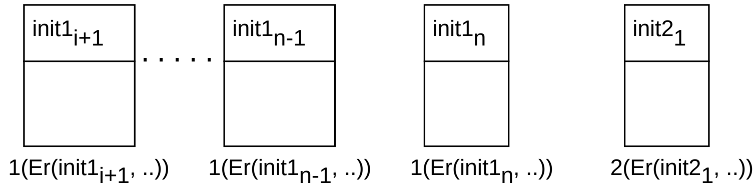
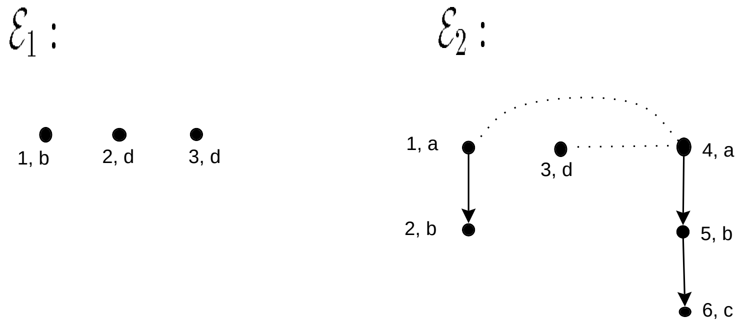


Abbildung 2: Erläuterung zur $Bed1((e_1, e, \mathcal{E}_1), f, (e_2, e', \mathcal{E}_2))$

punktierte Linie auf die Bedingung 1(c) und die Doppelpfeilkante auf die Bedingung 1(d). Ist Bedingung $Bed1((e_1, e, \mathcal{E}_1), f, (e_2, e', \mathcal{E}_2))$ wahr, so ist $Kons(e, \mathcal{E}_1, e', \mathcal{E}_2)$ die folgende Ereignisstruktur:

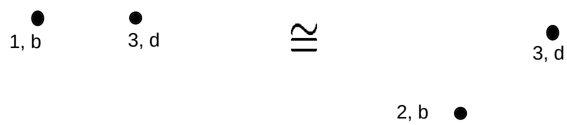


Beispiel 4.2 Gegeben seien



Seien ferner e_1 und e_2 Ereignisse mit $f(e_1) = a$ und $f(e_2) = d$. Dann ist $Bed1((e_1, 2, \mathcal{E}_1), f, (e_2, 1, \mathcal{E}_2))$ wahr, denn es gilt:

1. Wählt man $N1 = \{\{1\}, \{3\}\}$, so gilt $\overline{\bigcup}_{m \in N1} Er(m, \mathcal{E}_1) \cong Rest(1, \mathcal{E}_2)$ (siehe folgendes Bild).

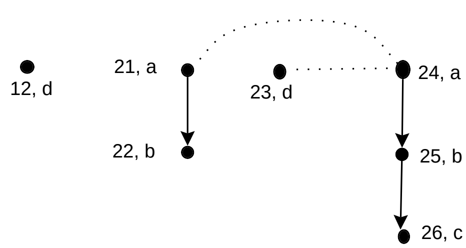


2. Sei $N2 = \emptyset$, dann gilt $Rest(2, \mathcal{E}_1) = (\emptyset, \emptyset, \emptyset, \emptyset) = \overline{\bigcup}_{m \in N2} Er(m, \mathcal{E}_2)$.

3. Es gilt $L1 = \emptyset$ und $L2 = \emptyset$, denn es gilt $L1 = Konf(\mathcal{E}_1) \setminus (\{2\} \cup N1)$ und $L2 = Konf(\mathcal{E}_2) \setminus (\{1, 3, 4\})$. Daraus folgt

$$\overline{\bigcup_{m \in L1}} Er(m, \mathcal{E}_1) = (\emptyset, \emptyset, \emptyset, \emptyset) = \overline{\bigcup_{m \in L2}} Er(m, \mathcal{E}_2)$$

$Kons(2, \mathcal{E}_1, 1, \mathcal{E}_2)$ ist die folgende Ereignisstruktur:



Man beachte, daß $Bed1((e_1, 2, \mathcal{E}_1), f, (e_2, 4, \mathcal{E}_2))$ falsch ist. □

Definition 4.16 ($Ver(f,M)$, $Bilde(f,M)$, $Erz(f,M)$)
Sei $\mathcal{E}_i \in \mathcal{ES}_{\mathcal{P}}$ *mit* $\mathcal{E}_i = (E_i, \leadsto_i, \mapsto_i, l_i)$ *und* $i = 1, \ldots, n$. *Sei weiterhin*

$$M = \{(e_1, \mathcal{E}_1), \ldots, (e_n, \mathcal{E}_n)\},$$

wobei $e_i \neq e_j$ *mit* $i \neq j$ *für* $i, j \in \{1, \ldots n\}$ *und* $e_i \notin E_i$ *gilt, und* $f : \{e_1, \ldots, e_n\} \to \mathcal{G}$ *eine Funktion.*

1. *Sei* $e \in init(\mathcal{E}_i)$ *und* $e' \in init(\mathcal{E}_j)$ *mit* $i, j \in \{1, \ldots, n\}$. *Dann ist* $Bed2((e_i, e, \mathcal{E}_i), f, (e_j, e', \mathcal{E}_j), M)$ *genau dann wahr, wenn folgendes gilt:*

   (a) $Bed1((e_i, e, \mathcal{E}_i), f, (e_j, e', \mathcal{E}_j))$ *ist wahr.*

   (b) *Es gilt*

   $$\forall e \in init(\mathcal{E}) : \exists h \in \{1, \ldots, n\} : l(e) = f(e_h) \wedge Rest(e, \mathcal{E}) \cong \mathcal{E}_h,$$

   wobei $\mathcal{E} = Kons(e, \mathcal{E}_i, e', \mathcal{E}_j)$ *gilt und* l *die Markierungsfunktion in* $\mathcal{E}$ *ist.*

2. *Sei* $i, j \in \{1, \ldots, n\}$ *und* $e \in init(\mathcal{E}_i)$. *Dann ist*

$$\begin{aligned} Komb((e_i, e, \mathcal{E}_i), f, \mathcal{E}_j, M) \;\; &:= \;\; \{Kons(e, \mathcal{E}_i, e', \mathcal{E}_j) \,|\, \exists e' \in init(\mathcal{E}_j) : \\ &\qquad Bed2((e_i, e, \mathcal{E}_i), f, (e_j, e', \mathcal{E}_j), M) = wahr\} \end{aligned}$$

3. *Für* $e \in init(\mathcal{E}_i)$ *ist*

$$\begin{aligned} \mathcal{M}((e_i, e, \mathcal{E}_i), f, M) \;\; &:= \;\; \{\mathcal{E}_j \,|\; \exists (e_j, \mathcal{E}_j) \in M : \exists\exists e' \in init(\mathcal{E}_j) : \\ &\qquad Bed1((e_i, e, \mathcal{E}_i), f, (e_j, e', \mathcal{E}_j)) = wahr\} \end{aligned}$$

4. $Bes((e_i, e, \mathcal{E}_i), f, M) := \bigcup_{\mathcal{E} \in \mathcal{M}((e_i, e, \mathcal{E}_i), f, M)} Komb((e_i, e, \mathcal{E}_i), f, \mathcal{E}, M)$

5. *Gilt* $\bigcup_{e \in init(\mathcal{E}_i)} Bes((e_i, e, \mathcal{E}_i), f, M) \neq \emptyset$, *dann ist*

$$Ver((e_i, \mathcal{E}_i), f, M) := \bigcup_{e \in init(\mathcal{E}_i)} Bes((e_i, e, \mathcal{E}_i), f, M).$$

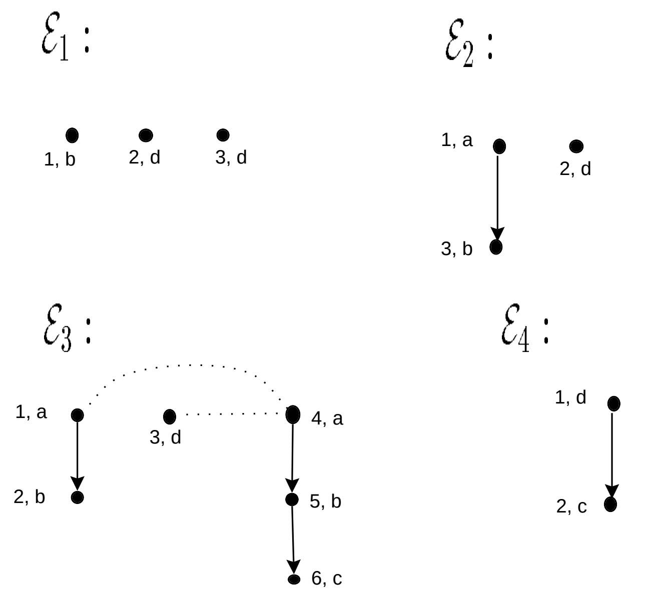
   Ansonsten ist $Ver((e_i, \mathcal{E}_i), f, M) := \overline{f(e_i)_{e_i}};\mathcal{E}_i$.

6. $Bilde(f, M) := \bigcup_{i=1}^{n} Ver((e_i, \mathcal{E}_i), f, M)$

7. $Erz(f, M) := Komp(Opt(Bilde(f, M)))$ □

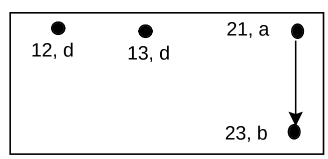
Mit $Erz(f,M)$ wird eine Ereignisstruktur aus der Menge M erzeugt. Dabei werden zunächst alle Elemente aus M paarweise bezüglich $Kons(\ldots)$ miteinander oder mit sich selbst verknüpft, falls die Bedingung $Bed2(\ldots)$ erfüllt ist. Aus jedem Element $\mathcal{E}_i$, das sich mit keinem anderen Element oder nicht mit sich selbst in dieser Weise verknüpfen läßt, wird $\overline{f(e_i)_{e_i}};\mathcal{E}_i$ erzeugt (siehe $Ver((e_i, \mathcal{E}_i), f, M)$). Die sich daraus ergebenen Ereignisstrukturen sind genau die der Menge $Bilde(f, M)$. $Erz(f, M)$ ist dann die Ereignisstruktur, die mit $Komp$ und Opt, angewendet auf $Bilde(f, M)$, erzeugt wird.

Beispiel 4.3 Sei $M = \{(e_1, \mathcal{E}_1), (e_2, \mathcal{E}_2), (e_3, \mathcal{E}_3), (e_4, \mathcal{E}_4)\}$, wobei

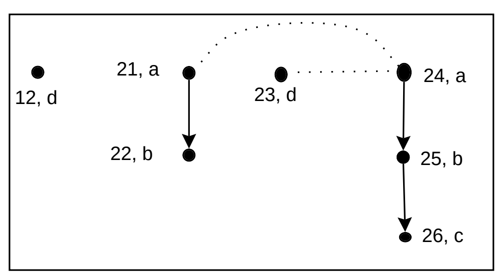


und $f(e_1) = a$, $f(e_2) = d$, $f(e_3) = d$, $f(e_4) = a$. Es gilt:

1. $Ver((e_1, \mathcal{E}_1), f, M)$ enthält folgende Ereignisstruktur:

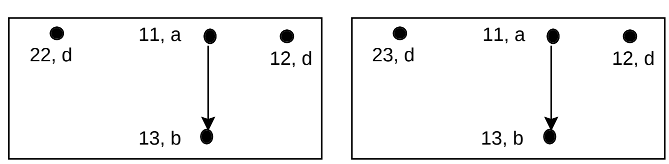


Man beachte, daß $Kons(2, \mathcal{E}_1, 1, \mathcal{E}_3)$, d.h. die folgende Ereignisstruktur



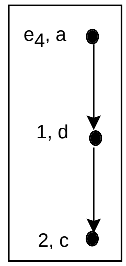
nicht zu $Ver((e_1, \mathcal{E}_1), f, M)$ gehört, obwohl $Bed1((e_1, 1, \mathcal{E}_1), f, (e_3, 1, \mathcal{E}_3))$ wahr ist. Der Grund ist, daß $Bed2((e_1, 1, \mathcal{E}_1), f, (e_3, 1, \mathcal{E}_3), M)$ verletzt ist, denn es gilt $\neg(\exists i \in \{1, \ldots, 4\} : \mathcal{E}_i \cong Rest(24, Kons(\ldots)))$.

2. $Ver((e_2, \mathcal{E}_2), f, M)$ enthält folgende 2 Ereignisstrukturen:

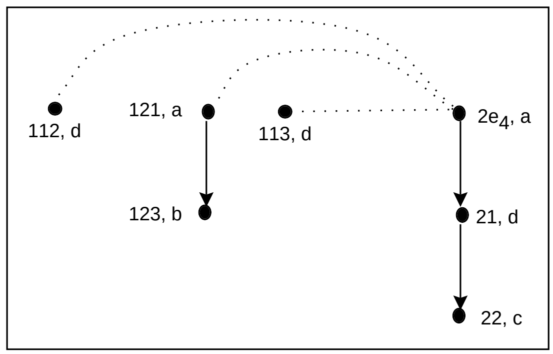


3. Es gilt $Ver((e_3, \mathcal{E}_3), f, M) = \emptyset$.

4. $Ver((e_4, \mathcal{E}_4), f, M)$ erhält folgende Ereignisstruktur:



$Erz(f, M)$ ergibt sich wie folgt:



□

Bemerkung 4.1 Hat man $Kons(e, \mathcal{E}_1, e', \mathcal{E}_2)$ bereits bestimmt, so ist es nicht nötig, $Kons(e', \mathcal{E}_2, e, \mathcal{E}_1)$ zu bestimmen. Wählt man nämlich auf beiden Seiten die identischen Mengen $N1$ und $N2$ (siehe Definition 4.15, 1(a), 1(b)), so gilt $Kons(e, \mathcal{E}_1, e', \mathcal{E}_2) \cong Kons(e', \mathcal{E}_2, e, \mathcal{E}_1)$. Diese Aussage kann dazu verwendet werden, um das Erzeugen von isomorphen Ereignisstrukturen zu vermeiden. □

4.2.2 Definition der Zerlegungsmethode

Nachdem die notwendigen Operatoren auf Ereignisstrukturen definiert und erläutert wurden, sind wir nun in der Lage, die Zerlegungsmethode in einer kurzen Form formal anzugeben.

Definition 4.17 ($EZ(B)$) *Sei $B \in \mathcal{BL}$ eine Action-Prefix-Form. Dann ist $EZ(B)$ induktiv wie folgt definiert:*

1. *$EZ(B) := (\emptyset, \emptyset, \emptyset, \emptyset)$, falls $B =$ **stop**.*

2. *Falls $B = \sum_{i=1}^{n} b_i; B_i$, dann $EZ(B) := Erz(f, M)$ mit*

$$M = \{(e_1, EZ(B_1)), \ldots, (e_n, EZ(B_n))\},$$

wobei

- *$e_i \neq e_j$ mit $i \neq j$ für $i, j \in \{1, \ldots n\}$.*
- *$e_i \notin E_i$ mit $EZ(B_i) = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ für $i = 1, \ldots, n$.*

- $f : \{e_1, \ldots, e_n\} \rightarrow \mathcal{G}$ *mit* $f(e_i) = b_i$. □

Definition 4.18 ($PZ(B)$) *Sei* $B \in \mathcal{BL}$. *Dann ist* $PZ(B) := Term(EZ(APF(B)))$. □

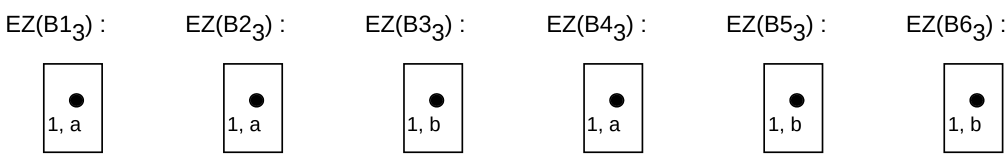
Im Abschnitt 4.3 wird gezeigt, daß $PZ(B)$ der Prozeß ist, der die Bedingung 1 und 2 des Zerlegungsproblems erfüllt.

Beispiel 4.4 Im folgenden und im weiteren Verlauf des Berichtes wird **stop** ab und zu nicht explizit angegeben. Anstatt $a;$ **stop** schreiben wir auch kurz a.

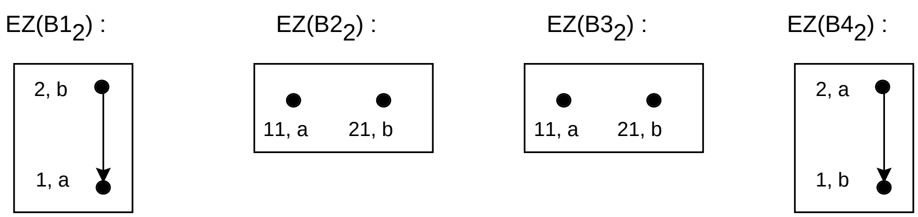
Sei $B = a; (b; b; a \,[]\, b; (b; a \,[]\, a; b)) \,[]\, b; (a; (b; a \,|||\, a; b) \,[]\, a; a; b)$, d.h.

1. $B = a; B1_1 \,[]\, b; B2_1$
2. $B1_1 = b; B1_2 \,[]\, b; B2_2$ und $B2_1 = a; B3_2 \,[]\, a; B4_2$
3. $B1_2 = b; B1_3$, $B2_2 = b; B2_3 \,[]\, a; B3_3$, $B3_2 = b; B4_3 \,[]\, a; B5_3$ und $B4_2 = a; B6_3$
4. $B1_3 = a;$ **stop**, $B2_3 = a;$ **stop**, $B3_3 = b;$ **stop**, $B4_3 = a;$ **stop**, $B5_3 = b;$ **stop** und $B6_3 = b;$ **stop**

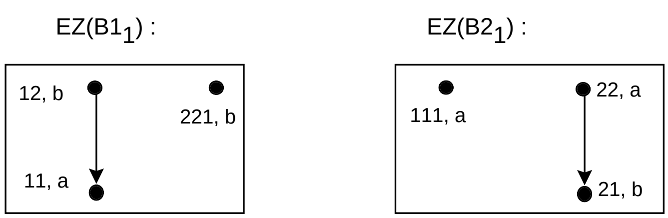
<u>Zu 4:</u> Es gilt



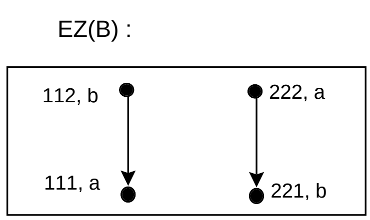
<u>Zu 3:</u> Es gilt



<u>Zu 2:</u> Es gilt



<u>Zu 1:</u> Es gilt

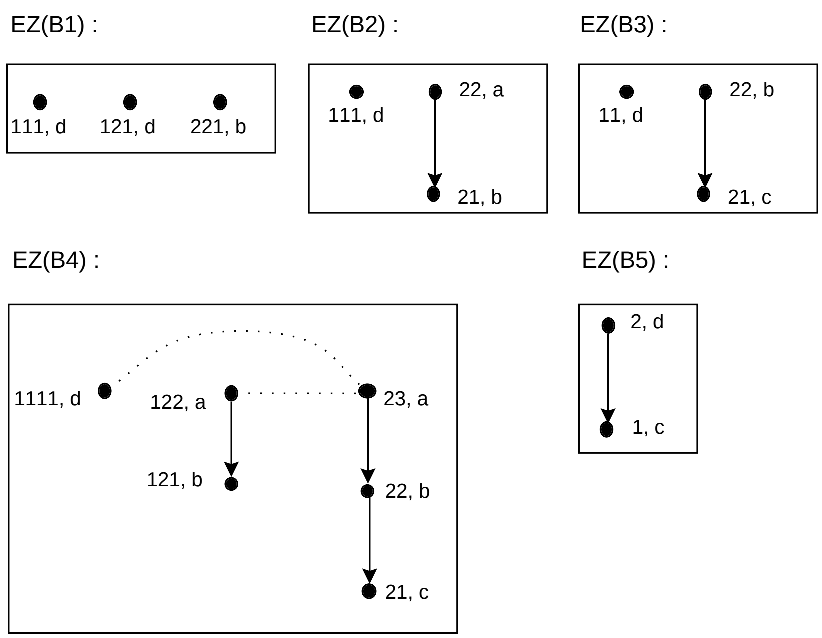


Wendet man $Term$ auf $EZ(B)$ an, so erhält man $PZ(B) = a; b; \mathbf{stop} \,|||\, b; a; \mathbf{stop}$. $\square$

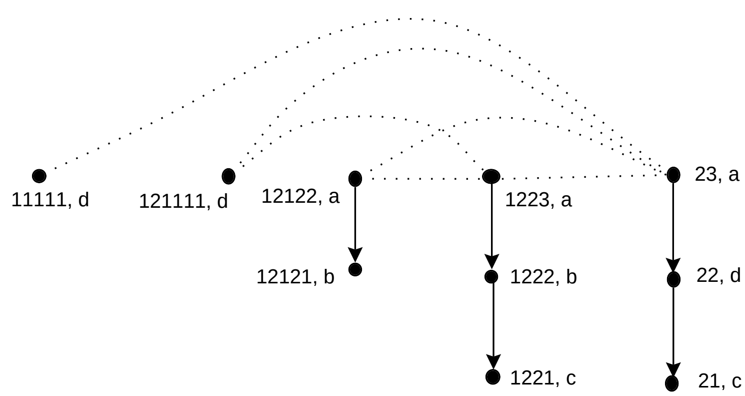
Beispiel 4.5 Sei $B = a; B1 \,[]\, d; B2 \,[]\, a; B3 \,[]\, d; B4 \,[]\, a; B5$, wobei

1. $B1 = b; d; d; \mathbf{stop} \,[]\, d; (b; d; \mathbf{stop} \,[]\, d; b; \mathbf{stop})$
2. $B2 = a; (b; d; \mathbf{stop} \,[]\, d; b; \mathbf{stop}) \,[]\, d; a; b; \mathbf{stop}$
3. $B3 = b; (c; d; \mathbf{stop} \,[]\, d; c; \mathbf{stop}) \,[]\, d; b; c; \mathbf{stop}$
4. $B4 = a; (b; d; \mathbf{stop} \,[]\, d; b; \mathbf{stop}) \,[]\, d; a; b; \mathbf{stop} \,[]\, a; b; c; \mathbf{stop}$
5. $B5 = d; c; \mathbf{stop}$

Es ist leicht zu prüfen, daß folgendes gilt:



Daraus ergibt sich $EZ(B)$ wie folgt:



Es gilt $PZ(B) = (d; \mathbf{stop} \,|||((d; \mathbf{stop} \,|||\, a; b; \mathbf{stop}) \,[]\, a; b; c; \mathbf{stop})) \,[]\, a; d; c; \mathbf{stop}$. $\square$

Beispiel 4.6 Sei $B1 = a; (b; (c \,[]\, d) \,[]\, c; b \,[]\, d; b) \,[]\, c; a; b \,[]\, d; a; b$ und $B2 = a; a; a$. Dann gilt $PZ(B1) = a; b \,|||(c \,[]\, d)$ und $PZ(B2) = a \,|||\, a \,|||\, a$. $\square$

Bemerkung 4.2

- Später wird gezeigt, daß $EZ(B) \cong EZ(C)$ mit B und C als Action-Prefix-Formen gilt, falls $B \sim C$ gilt. Dies hat zur Folge, daß auch $EZ(APF(B)) \cong EZ(NF(B))$ gilt. Es liegt daher kein Unterschied vor, ob aus B eine Action-Prefix-Form oder eine Normalform erzeugt wird, bevor EZ angewendet wird. Allerdings ist es zu empfehlen, bei komplexen Prozessen die Normalform zu verwenden, da diese mit größerer Wahrscheinlichkeit mehr unnötiges Erzeugen von isomorphen Ererignisstrukturen vermeidet. Dafür ist leider die Bestimmmung der Normalform aufwendiger als die der Action-Prefix-Form.

- In [Do93] wurde ein Verfahren $F(B,C)$ vorgestellt, das in Abhängigkeit von zwei Prozessen B und C die folgende Menge berechnet:

  $$F(B,C) = \{(B',C') \mid \; B',C' \in Er(B) \cup Er(C) \;\wedge\; B \sim C\}$$

  Mit $F(B,C)$ kann die Menge $N1$ (analog dazu $N2$) in Definition 4.15 wie folgt bestimmt werden: Sei $Konf(\mathcal{E}_1) = \{m1_1, \ldots, m1_h\}$, $m1 = m1_h$ und $Konf(Rest(e', Er(m2, \mathcal{E}_2))) = \{m2_1, \ldots, m2_k\}$.

  1. Bestimme $C_i = Term(Er(m1_i, \mathcal{E}_1))$ mit $i = 1, \ldots, h \Leftrightarrow 1$ und $D_j = Term(Er(m2_j, Rest(e' Er(m2, \mathcal{E}_2))))$ mit $j = 1, \ldots, k$.
  2. Bestimme $F(B,C)$, wobei $B1 = \sum_{i=1}^{h-1} a; C_i$, $B2 = \sum_{j=1}^{k} a; D_j$ und a beliebiger Aktionsname.
  3. Bestimme die Menge N, wobei

     $$N = \{i \mid \; \exists (C_i, D_j) \in F(B1, B2)\} \cap \{i, \ldots, h \Leftrightarrow 1\}.$$

  4. Gilt $|N| = k$, dann ist $N1 = \{m1_i \mid i \in N\}$. Ansonsten ist $N1$ nicht vorhanden.

  Das aufwendige Prüfen der Isomorphie-Eigenschaft kann also vermieden werden. Daß diese Vorgehensweise korrekt ist, folgt aus dem Korrektheitsnachweis der Zerlegungsmethode im Abschnitt 4.3. □

4.3 Korrektheitsnachweis der Zerlegungsmethode

Der Korrektheitsnachweis der Zerlegungsmethode wird zur Übersichtlichkeit in drei Teilabschnitte zerlegt. Im ersten Teil werden die grundlegenden (zum Teil trivialen) Lemmata zitiert und bewiesen. Im zweiten Teil wird gezeigt, daß $EZ(B)$ und $EZ(C)$ zueinander isomorph sind, wenn B und C Action-Prefix-Formen sind und $B \sim C$ gilt. Im dritten Teil wird nachgewiesen, daß $PZ(B)$ der Prozeß ist, der die Bedingung 1 und 2 des Zerlegungsproblems erfüllt.

4.3.1 Teil I

In diesem Teil werden die erforderlichen Lemmata zitiert und nachgewiesen, die für den Nachweis im dritten Teil benötigt werden. Teilweise sind dabei auch triviale Lemmata, wofür dann kein Beweis angegeben wird.

Lemma 4.3 *Sei $\mathcal{E}, \mathcal{E}' \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E} \cong \mathcal{E}'$. Dann gilt $\mathcal{E} \approx_E \mathcal{E}'$.*

Beweis: Trivial. □

Lemma 4.4 *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$ und $e \in E_U$, dann gilt*

1. *$e(\mathcal{E}) \in \mathcal{ES}_\mathcal{P}$*
2. *$e(\mathcal{E}) \cong \mathcal{E}$*
3. *$e(\mathcal{E}) \approx_E \mathcal{E}$*

Beweis: Zu 1: Induktionsbeweis über den Aufbau von $\mathcal{E}$. Zu 2 und 3: Trivial. □

Lemma 4.5 *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E} = \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$ und $e \in init(\mathcal{E}_1)$. Dann gilt $Rest(e, \mathcal{E}) = Rest(e, \mathcal{E}_1) \overline{\cup} \mathcal{E}_2$.*

Beweis: Trivial. □

Lemma 4.6 *Sei $\mathcal{E}, \mathcal{E}1, \mathcal{E}2 \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E} = \mathcal{E}1 \overline{\cup} \mathcal{E}2$ und $\mathcal{E} \cong \mathcal{E}'$. Sei ferner der Isomorphismus von $\mathcal{E}$ auf $\mathcal{E}'$, dann gilt $h(\mathcal{E}_1) \cong \mathcal{E}1$.*

Beweis: Trivial. □

Lemma 4.7 *Sei $\mathcal{E}, \mathcal{E}' \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E} \cong \mathcal{E}'$. Dann gilt $Zer(\mathcal{E}) = Zer(\mathcal{E}')$.*

Beweis: Trivial. □

Lemma 4.8 *Sei $\mathcal{E}, \mathcal{E}_1, \mathcal{E}_2 \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E} = \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$ und $init(\mathcal{E}_1) \neq \emptyset \neq init(\mathcal{E}_2)$. Dann gilt $Zer(\mathcal{E}) = Zer(\mathcal{E}1) + Zer(\mathcal{E}2)$.*

Beweis: Es folgt aus Lemma 3.7. □

Lemma 4.9 *Sei $\mathcal{E}_i \in \mathcal{ES}_\mathcal{P}$ mit $i = 1, \ldots, 4$. Sei ferner $\mathcal{E}_1 \approx_E \mathcal{E}_2$ und $\mathcal{E}_3 \approx_E \mathcal{E}_4$. Dann gilt $\mathcal{E}_1 \overline{\cup} \mathcal{E}_2 \approx_E \mathcal{E}_3 \overline{\cup} \mathcal{E}_4$.*

Beweis: Seien $R_1 \subseteq (Er(\mathcal{E}_1) \times Er(\mathcal{E}_3))$ und $R_2 \subseteq (Er(\mathcal{E}_2) \times Er(\mathcal{E}_4))$ E-Bisimulationen mit $(\mathcal{E}_1, \mathcal{E}_3) \in R_1$ und $(\mathcal{E}_2, \mathcal{E}_4) \in R_2$. Sei $R \subseteq (Er(\mathcal{E}_1 \overline{\cup} \mathcal{E}_2) \times Er(\mathcal{E}_3 \overline{\cup} \mathcal{E}_4))$ wie folgt definiert:

$$\begin{aligned} R \;=\; \{(\mathcal{E}_1' \overline{\cup} \mathcal{E}_2', \mathcal{E}_3' \overline{\cup} \mathcal{E}_4') \mid\; & \mathcal{E}_1' \in Er(\mathcal{E}_1) \wedge \mathcal{E}_2' \in Er(\mathcal{E}_2) \\ & \wedge\, \mathcal{E}_3' \in Er(\mathcal{E}_3) \wedge \mathcal{E}_4' \in Er(\mathcal{E}_4) \\ & \wedge\, (\mathcal{E}_1', \mathcal{E}_3') \in R_1 \wedge (\mathcal{E}_2', \mathcal{E}_4') \in R_2\} \end{aligned}$$

Dann läßt sich leicht prüfen, daß R eine E-Bisimulation ist. Da $(\mathcal{E}_1 \overline{\cup} \mathcal{E}_2, \mathcal{E}_3 \overline{\cup} \mathcal{E}_4) \in R$ gilt, gilt $\mathcal{E}_1 \overline{\cup} \mathcal{E}_2 \approx_E \mathcal{E}_3 \overline{\cup} \mathcal{E}_4$. □

Lemma 4.10 *Sei $\mathcal{E} = Kons(e, \mathcal{E}_1, e', \mathcal{E}_2)$. Dann gilt*

$$Rest(1e, \mathcal{E}) \cong \mathcal{E}_2 \text{ und } Rest(2e', \mathcal{E}) \cong \mathcal{E}_1.$$

Beweis: Nach Definition 4.15 gilt

$$\mathcal{E} = 1(\overline{\bigcup}_{m \in H} Er(m, \mathcal{E}_1)) \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_2)),$$

wobei $H = Konf(\mathcal{E}_1) \setminus N1$, $m1 \notin N1$, $e \in m1$, $m2 \in Konf(\mathcal{E}_2)$ und $e' \in m2$ gilt.

Nach Definition 4.15 gilt aber auch

$$Rest(e', Er(m2, \mathcal{E}_2)) \cong \overline{\bigcup}_{m \in N1} Er(m, \mathcal{E}_1)$$

und somit

$$\begin{aligned} 2(Rest(e', Er(m2, \mathcal{E}_2))) &\cong \overline{\bigcup}_{m \in N1} Er(m, \mathcal{E}_1) \\ &\cong 2(\overline{\bigcup}_{m \in N1} Er(m, \mathcal{E}_1)). \end{aligned}$$

Daraus folgt

$$\begin{aligned} Rest(2e', \mathcal{E}) &= 1(\overline{\bigcup}_{m \in H} Er(m, \mathcal{E}_1)) \mathbin{\overline{\cup}} 2(Rest(e', Er(m2, \mathcal{E}_2))) \\ &\cong 1(\overline{\bigcup}_{m \in H} Er(m, \mathcal{E}_1)) \mathbin{\overline{\cup}} 2(\overline{\bigcup}_{m \in N1} Er(m, \mathcal{E}_1)) \\ &\cong \mathcal{E}_1. \end{aligned}$$

Andererseits läßt sich $\mathcal{E}$ in folgender Form umschreiben:

$$\mathcal{E} = 1(Er(m1, \mathcal{E}1)) \mathbin{\overline{\cup}} 1(\overline{\bigcup}_{m \in (H \setminus m1)} Er(m, \mathcal{E}_1)) \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_2))$$

Es gilt

$$\begin{aligned} Rest(1e, \mathcal{E}) &= 1(Rest(e, Er(m1, \mathcal{E}1))) \mathbin{\overline{\cup}} 1(\overline{\bigcup}_{m \in (H \setminus m1)} Er(m, \mathcal{E}_1)) \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_2)) \\ &\cong 3(\overline{\bigcup}_{m \in N2} Er(m, \mathcal{E}_2)) \mathbin{\overline{\cup}} 1(\overline{\bigcup}_{m \in (H \setminus m1)} Er(m, \mathcal{E}_1)) \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_2)) \\ &\cong 3(\overline{\bigcup}_{m \in N2} Er(m, \mathcal{E}_2)) \mathbin{\overline{\cup}} 1(\overline{\bigcup}_{m \in L2} Er(m, \mathcal{E}_2)) \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_2)) \\ &\cong \mathcal{E}_2, \end{aligned}$$

wobei $N2 \subset Konf(\mathcal{E}_2)$ mit $m2 \notin N2$ und $L2 = Konf(\mathcal{E}_2) \setminus (m2 \cup N2)$ gilt. □

Lemma 4.11 *Sei $\mathcal{E}, \mathcal{E}_i \in \mathcal{ES}_\mathcal{P}$ mit $i = 1, 2$, $\mathcal{E} = \mathcal{E}_1 \overline{[]} \mathcal{E}_2$ und $init(\mathcal{E}_i) \neq \emptyset$. Sei ferner $e \in init(\mathcal{E}_1)$ und $\mathcal{E}_2 = (E_2, \leadsto_2, \mapsto_2, l_2)$, dann gilt*

1. $\mathcal{E}[\{e\}] \;=\; \mathcal{E}_1[\{e\}] \mathbin{\overline{\cup}} \mathcal{E}_3$, *wobei* $\mathcal{E}_3 = (E_2, \leadsto_2, \mapsto_3, l_2)$ *mit*

$$\mapsto_3 \;=\; \mapsto_2 \;\cup\; \{(\emptyset, e) \mid e \in init(\mathcal{E}_2)\}$$

2. $Rest(e, \mathcal{E}) = Rest(e, \mathcal{E}_1)$

Beweis: Zu 1: Trivial. Zu 2: Sei $Rest(e, \mathcal{E}) = (E', \rightsquigarrow', \mapsto', l')$, $\mathcal{E}_i = (E_i, \rightsquigarrow_i, \mapsto_i, l_i)$ mit $i = 1, 2$ und $Rest(e, \mathcal{E}_1) = (E_1', \rightsquigarrow_1', \mapsto_1', l_1')$. Es genügt offenbar zu zeigen, daß $E' = E_1'$ gilt.

- „$\subseteq$"

  Sei $m = \{e' \mid e \rightsquigarrow e'\}$, dann folgt aus Definition 4.3 $init(\mathcal{E}_2) \subseteq m$. Mit Lemma 3.6(9) läßt sich folgern, daß $E_2 \cap E' = \emptyset$ gilt. Damit gilt $E' \subseteq E_1$.

  Sei $\phi \in E'$, dann gilt nach Definition 4.3 $\phi \neq e$ und $\neg(e' \vdash \phi)$, wobei $e \rightsquigarrow e'$ und $e' \in E_1$ gilt. Da $\phi \in E_1$ gilt, gilt $\phi \in E_1'$.

- „$\supseteq$"

  Sei $\phi \in E_1'$, dann gilt $\phi \neq e$ und $\neg(e' \vdash \phi)$, wobei $e \rightsquigarrow e'$ und $e' \in E_1$ gilt. Gilt angenommen $\phi \notin E'$, dann gibt es ein $e'' \in E_2$ mit $e \rightsquigarrow e''$, so daß $e'' \vdash \phi$ gilt. Da aus Lemma 3.6(3) $e'' \in init(\mathcal{E}_2)$ folgt, gilt nach Lemma 3.6(9) $\phi \in E_2$. Widerspruch. $\square$

Lemma 4.12 *Sei $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ und $init(\mathcal{E}) \neq \emptyset$, dann gilt*

$$Rest(e, \mathcal{E}) \approx_E \mathcal{E}[\{e\}].$$

Beweis: Wir beweisen mit Hilfe des Well-Founded-Induktionsprinzips. Sei

$$L = \{\mathcal{E} \mid \mathcal{E} \in \mathcal{ES}_{\mathcal{P}} \ \wedge init(\mathcal{E}) \neq \emptyset\}.$$

Für $\mathcal{E} \in L$ nehmen wir per Induktion an, daß $Rest(e', \mathcal{E}') \approx_E \mathcal{E}'[\{e'\}]$ gilt, falls $\mathcal{E}' \in L$ und $\mathcal{E}' \subset_{\mathcal{E}} \mathcal{E}$ gilt. Es sind zwei Fälle zu unterscheiden:

1. $|init(\mathcal{E})| = 1$

   In diesem Fall gilt $\mathcal{E} = \overline{g_e;}\mathcal{E}'$. Daraus folgt sofort die Behauptung.

2. $|init(\mathcal{E})| > 1$

   In diesem Fall gilt entweder $\mathcal{E} = \mathcal{E}_1 \ \overline{[]}\ \mathcal{E}_2$ oder $\mathcal{E} = \mathcal{E}_1 \ \overline{\cup}\ \mathcal{E}_2$, wobei $init(\mathcal{E}_1) \neq \emptyset \neq init(\mathcal{E}_2)$ gilt. Wir unterscheiden daher zwei Fälle:

   - $\mathcal{E} = \mathcal{E}_1 \ \overline{[]}\ \mathcal{E}_2$

     Sei o.E.d.A. $e \in init(\mathcal{E}_1)$. Nach Lemma 4.11 gilt $Rest(e, \mathcal{E}) = Rest(e, \mathcal{E}_1)$. Nach Induktionsannahme gilt $Rest(e, \mathcal{E}_1) \approx_E \mathcal{E}_1[\{e\}]$. Nach Lemma 4.11 gilt $\mathcal{E}[\{e\}] = \mathcal{E}_1[\{e\}] \ \overline{\cup}\ \mathcal{E}_3$, wobei $init(\mathcal{E}_3) = \emptyset$ gilt. Es ist leicht zu zeigen, daß $\mathcal{E}[\{e\}] \approx_E \mathcal{E}_1[\{e\}]$ gilt. Somit gilt $Rest(e, \mathcal{E}_1) \approx_E \mathcal{E}[\{e\}]$.

   - $\mathcal{E} = \mathcal{E}_1 \ \overline{\cup}\ \mathcal{E}_2$

     Sei o.E.d.A. $e \in init(\mathcal{E}_1)$. Nach Lemma 4.5 gilt $Rest(e, \mathcal{E}) = Rest(e, \mathcal{E}_1) \ \overline{\cup}\ \mathcal{E}_2$. Da nach Induktionsannahme $Rest(e, \mathcal{E}_1) \approx_E \mathcal{E}_1[\{e\}]$ gilt, gilt mit Lemma 4.9 $Rest(e, \mathcal{E}) \approx_E \mathcal{E}[\{e\}]$. $\square$

Lemma 4.13 *Sei* $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$ *mit* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *und* $init(\mathcal{E}) \neq \emptyset$*. Dann gilt* $Rest(e, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$*.*

Beweis: Wir beweisen mit Hilfe des Well-Founded-Induktionsprinzips. Sei

$$L = \{\mathcal{E} \mid \mathcal{E} \in \mathcal{ES}_\mathcal{P} \ \wedge init(\mathcal{E}) \neq \emptyset\}.$$

Für $\mathcal{E} \in L$ nehmen wir per Induktion an, daß $Rest(e', \mathcal{E}') \in \mathcal{ES}_\mathcal{P}$ gilt, falls $\mathcal{E}' \in L$ und $\mathcal{E}' \subset_\mathcal{E} \mathcal{E}$ gilt. Es sind zwei Fälle zu unterscheiden:

1. $|init(\mathcal{E})| = 1$

   In diesem Fall gilt $\mathcal{E} = \overline{g_e;}\mathcal{E}'$. Nach Induktionsannahme gilt $\mathcal{E}' \in \mathcal{ES}_\mathcal{P}$ und somit auch $\mathcal{E}' \in \mathcal{ES}_\mathcal{P}$.

2. $|init(\mathcal{E})| > 1$

   In diesem Fall gilt entweder $\mathcal{E} = \mathcal{E}_1 \ \overline{[]}\ \mathcal{E}_2$ oder $\mathcal{E} = \mathcal{E}_1 \ \overline{\cup}\ \mathcal{E}_2$, wobei $init(\mathcal{E}_1) \neq \emptyset \neq init(\mathcal{E}_2)$ gilt. Wir unterscheiden daher zwei Fälle:

   - $\mathcal{E} = \mathcal{E}_1 \ \overline{[]}\ \mathcal{E}_2$
     Sei o.E.d.A. $e \in init(\mathcal{E}_1)$. Nach Lemma 4.11 gilt $Rest(e, \mathcal{E}) = Rest(e, \mathcal{E}_1)$. Nach Induktionsannahme gilt $Rest(e, \mathcal{E}_1) \in \mathcal{ES}_\mathcal{P}$ und somit $Rest(e, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$.
   - $\mathcal{E} = \mathcal{E}_1 \ \overline{\cup}\ \mathcal{E}_2$
     Sei o.E.d.A. $e \in init(\mathcal{E}_1)$. Nach Lemma 4.5 gilt $Rest(e, \mathcal{E}) = Rest(e, \mathcal{E}_1) \ \overline{\cup}\ \mathcal{E}_2$. Da nach Induktionsannahme $Rest(e, \mathcal{E}_1) \in \mathcal{ES}_\mathcal{P}$ gilt, gilt $Rest(e, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$. □

Lemma 4.14 *Sei* $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$ *mit* $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ *und* $E \neq \emptyset$*. Dann gibt es eine endliche Folge von Ereignissen* $e_1, \ldots, e_n$*, so daß gilt:*

$$\mathcal{E} \overset{e_1}{\Leftrightarrow\!\!\to} \mathcal{E}^1,\ \mathcal{E}_R^1 \overset{e_2}{\Leftrightarrow\!\!\to} \mathcal{E}^2,\ \mathcal{E}_R^2 \overset{e_3}{\Leftrightarrow\!\!\to} \mathcal{E}^3, \cdots,\ \mathcal{E}_R^{n-1} \overset{e_n}{\Leftrightarrow\!\!\to} \mathcal{E}^n,$$

wobei $\mathcal{E}_R^1 = Rest(e_1, \mathcal{E})$*,* $\mathcal{E}_R^i = Rest(e_i, \mathcal{E}^{i-1})$ *mit* $i = 2, \ldots, n \Leftrightarrow 1$ *und* $Rest(e_n, \mathcal{E}_R^{n-1}) = (\emptyset, \emptyset, \emptyset, \emptyset)$ *gilt.*

Beweis: Es ist leicht zu zeigen, daß für $\mathcal{E}' \in \mathcal{ES}_\mathcal{P}$ gilt: Sei $\mathcal{E}' = (E', \rightsquigarrow', \mapsto', l')$ und $E' \neq \emptyset$, dann gilt $init(\mathcal{E}') \neq \emptyset$ (Induktion über den Aufbau von $\mathcal{E}'$).

Sei $e_1 \in init(\mathcal{E})$ und $Rest(e_1, \mathcal{E}) = (E_1, \rightsquigarrow_1, \mapsto_1, l_1)$, dann gilt nach Lemma 4.13 $Rest(e_1, \mathcal{E}) \in \mathcal{ES}_\mathcal{P}$. Außerdem gilt $E_1 \subset E$. Gilt $Rest(e_1, \mathcal{E}) \neq (\emptyset, \emptyset, \emptyset, \emptyset)$, so gibt es ein $e_2 \in init(Rest(e_1, \mathcal{E}))$, so daß $Rest(e_2, Rest(e_1, \mathcal{E})) \in \mathcal{ES}_\mathcal{P}$ gilt (D.h. wenn $Rest(e_1, \mathcal{E}) \neq (\emptyset, \emptyset, \emptyset, \emptyset)$ gilt, dann läßt sich immer ein $e_2 \in init(Rest(e_1, \mathcal{E}))$ finden.). Da E nach Lemma 3.6 endlich ist, gibt es also eine endliche Folge $e_1, \ldots, e_n$ mit $n \leq |E|$, so daß $\mathcal{E}_R^1 = Rest(e_1, \mathcal{E})$, $\mathcal{E}_R^i = Rest(e_i, \mathcal{E}^{i-1})$ mit $i = 2, \ldots, n \Leftrightarrow 1$ und $Rest(e_n, \mathcal{E}_R^{n-1}) = (\emptyset, \emptyset, \emptyset, \emptyset)$ gilt. □

Lemma 4.15 *Sei* $\mathcal{E}_i \in \mathcal{ES}_\mathcal{P}$ *mit* $i = 1, 2, 3, 4$*. Sei ferner* $\mathcal{E}_1 \cup \mathcal{E}_2 \approx_E \mathcal{E}_3 \cup \mathcal{E}_4$ *und* $\mathcal{E}_1 \approx_E \mathcal{E}_3$*, dann gilt* $\mathcal{E}_2 \approx_E \mathcal{E}_4$*.*

Beweis: Nach Lemma 4.9 gilt $\mathcal{E}_3 \cup \mathcal{E}_4 \approx_E \mathcal{E}_1 \cup \mathcal{E}_4$. Wir zeigen daher die Behauptung: Gilt $\mathcal{E}_1 \cup \mathcal{E}_2 \approx_E \mathcal{E}_1 \cup \mathcal{E}_4$, dann gilt $\mathcal{E}_2 \approx_E \mathcal{E}_4$. Wir beweisen kontraproduktiv. D.h. gilt $\neg(\mathcal{E}_2 \approx_E \mathcal{E}_4)$, dann gilt $\neg(\mathcal{E}_1 \cup \mathcal{E}_2 \approx_E \mathcal{E}_1 \cup \mathcal{E}_4)$. Zur einfachen Schreibweise wird l als Markierungsfunktion für alle Ereignisstrukturen verwendet.

Wir beweisen mit Hilfe des Well-Founded-Induktionsprinzips. Sei

$$L = \{\mathcal{E}\overline{\cup}\mathcal{E}' \mid \mathcal{E}, \mathcal{E}' \text{ sind EBES und endlich}\}.$$

Dabei heißt eine Ereignisstruktur endlich, wenn die zugehörige Ereignismenge endlich ist. Da jede Ereignisstruktur in $\mathcal{ES}_{\mathcal{P}}$ endlich ist, gilt $\mathcal{E}_1 \cup \mathcal{E}_2, \mathcal{E}_3 \cup \mathcal{E}_4 \in L$.

Definiere auf L die Relation $\prec$ wie folgt: $\mathcal{E}' \prec \mathcal{E}$, falls gilt:

- $\mathcal{E} = \mathcal{E}_1\overline{\cup}\mathcal{E}_2$, $\mathcal{E}' = \mathcal{E}_1'\overline{\cup}\mathcal{E}_2'$
- $\mathcal{E}_1' \in Er(\mathcal{E}_1)$, $\mathcal{E}_2' \in Er(\mathcal{E}_2)$
- $\mathcal{E}_1' \neq \mathcal{E}_1$ oder $\mathcal{E}_2' \neq \mathcal{E}_2$

Offenbar ist $\prec$ wohlfundiert, da $\mathcal{E}' \prec \mathcal{E} \Rightarrow \mathcal{E}' \subset_{\mathcal{E}} \mathcal{E}$ gilt. Für $\mathcal{E} \in L$ nehmen wir per Induktion an, daß für $\mathcal{E}'$ mit $\mathcal{E}' \prec \mathcal{E}$ gilt: Sei $\mathcal{E}' = \mathcal{E}_1'\overline{\cup}\mathcal{E}_2'$, dann gilt $\mathcal{E}_1'\overline{\cup}\mathcal{E}_2' \not\approx_E \mathcal{E}_1'\overline{\cup}\mathcal{E}_3'$, falls $\mathcal{E}_2' \not\approx_E \mathcal{E}_3'$ gilt. Um zu zeigen, daß dies auch für $\mathcal{E}$ gilt, sind es zwei Fälle zu unterscheiden.

1. $\neg(\exists \mathcal{E}' : \mathcal{E}' \prec \mathcal{E})$, d.h. $\mathcal{E} = \mathcal{E}_1\overline{\cup}\mathcal{E}_2$ mit $\mathcal{E}_1 = (\emptyset,\emptyset,\emptyset,\emptyset) = \mathcal{E}_2$. In diesem Fall gilt offenbar $\mathcal{E}_1\overline{\cup}\mathcal{E}_2 \not\approx_E \mathcal{E}_1\overline{\cup}\mathcal{E}_3$, falls $\mathcal{E}_2 \not\approx_E \mathcal{E}_3$ gilt.

2. $\exists \mathcal{E}' : \mathcal{E}' \prec \mathcal{E}$. Sei $\mathcal{E} = \mathcal{E}_1\overline{\cup}\mathcal{E}_2$ mit $\mathcal{E}_1 \neq (\emptyset,\emptyset,\emptyset,\emptyset)$ oder $\mathcal{E}_2 \neq (\emptyset,\emptyset,\emptyset,\emptyset)$.

   In diesem Fall unterscheiden wir drei weitere Fälle:

   (a) $\mathcal{E}_1 \neq (\emptyset,\emptyset,\emptyset,\emptyset) = \mathcal{E}_2$
   (b) $\mathcal{E}_1 = (\emptyset,\emptyset,\emptyset,\emptyset) \neq \mathcal{E}_2$
   (c) $\mathcal{E}_1 \neq (\emptyset,\emptyset,\emptyset,\emptyset) \neq \mathcal{E}_2$

   Zu (a):

   Es gilt $\mathcal{E}_1\overline{\cup}\mathcal{E}_2 \approx_E \mathcal{E}_1 \not\approx_E \mathcal{E}_1\overline{\cup}\mathcal{E}_3$, falls $\mathcal{E}_2 \not\approx_E \mathcal{E}_3$ gilt. Gilt nämlich $\mathcal{E}_2 \not\approx_E \mathcal{E}_3$, dann gilt $init(\mathcal{E}_3) \neq \emptyset$. Sei $e \in init(\mathcal{E}_3)$. Sei ferner $e_1, e_2, \ldots, e_n$ eine Folge von Ereignissen in $\mathcal{E}_1$, für die gilt:

   - $\mathcal{E}_1 \overset{e_1}{\Longleftrightarrow} \mathcal{E}_1^1 \overset{e_2}{\Longleftrightarrow} \ldots \mathcal{E}_1^{n-1} \overset{e_n}{\Longleftrightarrow} \mathcal{E}_1^n$
   - $init(\mathcal{E}_1^n) = \emptyset$
   - n ist maximal.

   Eine solche Folge von Ereignissen können wir natürlich immer annehmen, da die Ereignismenge von $\mathcal{E}$ endlich ist. Dann gilt

   $$\mathcal{E}_1\overline{\cup}\mathcal{E}_3 \overset{e_1}{\Longleftrightarrow} \mathcal{E}_1^1\overline{\cup}\mathcal{E}_3 \overset{e_2}{\Longleftrightarrow} \ldots \mathcal{E}_1^{n-1}\overline{\cup}\mathcal{E}_3 \overset{e_n}{\Longleftrightarrow} \mathcal{E}_1^n\overline{\cup}\mathcal{E}_3 \overset{e}{\Longleftrightarrow} \mathcal{E}_1^n\overline{\cup}\mathcal{E}_3[\{e\}],$$

   was mit Lemma 2.1 $\mathcal{E}_1 \not\approx_E \mathcal{E}_1\overline{\cup}\mathcal{E}_3$ impliziert.

   Zu (b):

Es gilt $\mathcal{E}_1\overline{\cup}\mathcal{E}_2 \approx_E \mathcal{E}_2 \not\approx_E \mathcal{E}_1\overline{\cup}\mathcal{E}_3 \approx_E \mathcal{E}_3$. Wegen Transitivität von $\approx_E$ gilt $\mathcal{E}_1\overline{\cup}\mathcal{E}_2 \not\approx_E \mathcal{E}_1\overline{\cup}\mathcal{E}_3$.

Zu (c):

Es gilt $init(\mathcal{E}_1) \neq \emptyset \neq init(\mathcal{E}_2)$. Gilt $init(\mathcal{E}_3) = \emptyset$, dann gilt offenbar $\mathcal{E}_1\overline{\cup}\mathcal{E}_2 \not\approx_E \mathcal{E}_1\overline{\cup}\mathcal{E}_3$. Wir betrachten daher $init(\mathcal{E}_3) \neq \emptyset$. Sei $init(\mathcal{E}_1) = \{e_1, \ldots, e_n\}$, $init(\mathcal{E}_2) = \{\phi_1, \ldots, \phi_m\}$ und $init(\mathcal{E}_3) = \{\psi_1, \ldots, \psi_k\}$. Dann haben wir

- $\mathcal{E}_1\overline{\cup}\mathcal{E}_2 \overset{e_i}{\Leftrightarrow\!\!\!\to} \mathcal{E}_1^i\overline{\cup}\mathcal{E}_2$ und $\mathcal{E}_1\overline{\cup}\mathcal{E}_3 \overset{e_i}{\Leftrightarrow\!\!\!\to} \mathcal{E}_1^i\overline{\cup}\mathcal{E}_3$ mit $i = 1, \ldots, n$, wobei $\mathcal{E}_1 \overset{e_i}{\Leftrightarrow\!\!\!\to} \mathcal{E}_1^i$ gilt.
- $\mathcal{E}_1\overline{\cup}\mathcal{E}_2 \overset{\phi_j}{\Leftrightarrow\!\!\!\to} \mathcal{E}_1\overline{\cup}\mathcal{E}_2^j$ mit $j = 1, \ldots, m$, wobei $\mathcal{E}_1 \overset{\phi_j}{\Leftrightarrow\!\!\!\to} \mathcal{E}_2^j$ gilt.
- $\mathcal{E}_1\overline{\cup}\mathcal{E}_3 \overset{\psi_h}{\Leftrightarrow\!\!\!\to} \mathcal{E}_1\overline{\cup}\mathcal{E}_3^h$ mit $h = 1, \ldots, k$, wobei $\mathcal{E}_1 \overset{\psi_h}{\Leftrightarrow\!\!\!\to} \mathcal{E}_3^h$ gilt.

Da $\mathcal{E}_2 \not\approx_E \mathcal{E}_3$ gilt, muß ein von folgenden zwei Fällen gelten:

I) $\exists\phi_j : \neg(\exists\psi_h : \mathcal{E}_3 \overset{\psi_h}{\Leftrightarrow\!\!\!\to} \mathcal{E}_3^h \;\wedge l(\phi_j) = l(\psi_h) \;\wedge\; \mathcal{E}_2^j \approx_E \mathcal{E}_3^h)$

II) $\exists\psi_h : \neg(\exists\phi_j : \mathcal{E}_2 \overset{\phi_j}{\Leftrightarrow\!\!\!\to} \mathcal{E}_2^j \;\wedge l(\phi_j) = l(\psi_h) \;\wedge\; \mathcal{E}_2^i \approx_E \mathcal{E}_3^h)$

Zu I):

Wegen Induktionsannahme gilt für alle $h \in \{1, \ldots, k\}$ mit $l(\phi_j) = l(\psi_h)$:

$$\mathcal{E}_1\overline{\cup}\mathcal{E}_2^j \not\approx_E \mathcal{E}_1\overline{\cup}\mathcal{E}_3^h$$

Gilt angenommen $\mathcal{E}_1\overline{\cup}\mathcal{E}_2 \approx_E \mathcal{E}_1\overline{\cup}\mathcal{E}_3$, so muß es ein e_i geben, so daß gilt:

- $l(e_i) = l(\phi_j)$
- $\mathcal{E}_1\overline{\cup}\mathcal{E}_2^j \approx_E \mathcal{E}_1^i\overline{\cup}\mathcal{E}_3$

Dies ist ein Widerspruch, denn es gilt $\mathcal{E}_1\overline{\cup}\mathcal{E}_2^j \not\approx_E \mathcal{E}_1^i\overline{\cup}\mathcal{E}_3$, wie im folgenden gezeigt wird. Gilt angenommen $\mathcal{E}_1\overline{\cup}\mathcal{E}_2^j \approx_E \mathcal{E}_1^i\overline{\cup}\mathcal{E}_3$ und sei $\mathcal{E}' = \mathcal{E}_1^i\overline{\cup}\mathcal{E}_3$, dann gilt

$$\exists e^1 \in init(\mathcal{E}') : (l(e^1) = l(e_i) \;\wedge\; \mathcal{E}_1^i\overline{\cup}\mathcal{E}_2^j \approx \mathcal{E}_1'),$$

wobei $\mathcal{E}_1' = \mathcal{E}'[\{e^1\}]$. Es gilt $e^1 \in init(\mathcal{E}_1^i)$, denn gilt $e^1 \in init(\mathcal{E}_3)$, dann gilt wegen $\mathcal{E}_2^j \not\approx_E \mathcal{E}_3[\{e^1\}]$ und Induktionsannahme

$$\mathcal{E}_1^i\overline{\cup}\mathcal{E}_2^j \not\approx_E \mathcal{E}_1^i\overline{\cup}\mathcal{E}_3[\{e^1\}]$$

Widerspruch. D.h. es gilt $\mathcal{E}_1^i\overline{\cup}\mathcal{E}_2^j \approx_E (\mathcal{E}_1^i)^1\overline{\cup}\mathcal{E}_3$ mit $(\mathcal{E}_1^i)^1 = \mathcal{E}_1^i[\{e\}]$. Daraus können wir folgern, daß es eine Folge von Ereignissen $e^2, \ldots, e^z$ mit $l(e^1) = l(e^x)$ und $x = 1, \ldots, z$ gibt, so daß gilt:

- $(\mathcal{E}_1^i)^1\overline{\cup}\mathcal{E}_3 \overset{e^2}{\Leftrightarrow\!\!\!\to} (\mathcal{E}_1^i)^2\overline{\cup}\mathcal{E}_3 \overset{e^2}{\Leftrightarrow\!\!\!\to} \ldots (\mathcal{E}_1^i)^{z-1}\overline{\cup}\mathcal{E}_3 \overset{e^z}{\Leftrightarrow\!\!\!\to} (\mathcal{E}_1^i)^z\overline{\cup}\mathcal{E}_3$, wobei
$$\neg(\exists e \in init((\mathcal{E}_1^i)^z) : l(e^1) = l(e)).$$
- $\mathcal{E}_1^i\overline{\cup}\mathcal{E}_2^j \overset{e^1}{\Leftrightarrow\!\!\!\to} (\mathcal{E}_1^i)^1\overline{\cup}\mathcal{E}_2^j \overset{e^2}{\Leftrightarrow\!\!\!\to} \ldots (\mathcal{E}_1^i)^{z-2}\overline{\cup}\mathcal{E}_2^j \overset{e^{z-1}}{\Leftrightarrow\!\!\!\to} (\mathcal{E}_1^i)^{z-1}\overline{\cup}\mathcal{E}_2^j$
- $(\mathcal{E}_1^i)^x\overline{\cup}\mathcal{E}_2^j \approx_E (\mathcal{E}_1^i)^{x+1}\overline{\cup}\mathcal{E}_3$ mit $x = 2, \ldots, z \Leftrightarrow 1$.

Daraus folgt

$$\exists \psi_h \in init(\mathcal{E}_3) : (l(\psi_h) = l(e^z) \ \wedge (\mathcal{E}_1^i)^z \overline{\cup} \mathcal{E}_2^j \approx_E (\mathcal{E}_1^i)^z \overline{\cup} \mathcal{E}_3^h).$$

Nach Induktionsannahme gilt somit $\mathcal{E}_2^j \approx_E \mathcal{E}_3^h$. Widerspruch. Es gilt also $\mathcal{E}_1 \overline{\cup} \mathcal{E}_2^j \not\approx_E \mathcal{E}_1^i \overline{\cup} \mathcal{E}_3$. Somit ist die Annahme $\mathcal{E}_1 \overline{\cup} \mathcal{E}_2 \approx_E \mathcal{E}_1 \overline{\cup} \mathcal{E}_3$ falsch.

Zu II): Analog zu I). □

Definition 4.19 (zerlegbar) *Sei $\mathcal{E} \in \mathcal{ES}_\mathcal{P}$. $\mathcal{E}$ ist zerlegbar genau dann, wenn es $\mathcal{E}_1, \mathcal{E}_2 \in \mathcal{ES}_\mathcal{P}$ mit $init(\mathcal{E}_i) \neq \emptyset$ für $i = 1, 2$ gibt, so daß $\mathcal{E} \approx_E \mathcal{E}_1 \overline{\cup} \mathcal{E}_2$ gilt.* □

Lemma 4.16 *Sei $\mathcal{E}, \mathcal{E}', \mathcal{E}_i, \mathcal{E}'_j \in \mathcal{ES}_\mathcal{P}$ mit $init(\mathcal{E}_i) \neq \emptyset \neq init(\mathcal{E}_j)$, $i = 1, \dots, n$ und $j = 1, \dots, m$. Sei ferner $\mathcal{E}_i$ und $\mathcal{E}'_j$ nicht zerlegbar. Gilt $\mathcal{E} \approx_E \mathcal{E}'$ und*

$$\mathcal{E} = \overline{\bigcup_{i=1}^{n}} \mathcal{E}_i \wedge \mathcal{E}' = \overline{\bigcup_{j=1}^{m}} \mathcal{E}'_j,$$

dann gilt $\forall i \in \{1, \dots, n\} : \exists j \in \{1, \dots, m\} : \mathcal{E}_i \approx_E \mathcal{E}'_j$.

Beweis: Wir führen zunächst die folgende Notation ein: Für $\mathcal{E}1, \mathcal{E}2 \in \mathcal{ES}_\mathcal{P}$ schreiben wir $\mathcal{E}1 \prec \mathcal{E}2$, falls $\exists \mathcal{E}2' \in Er(\mathcal{E}2) : (\mathcal{E}2' \neq \mathcal{E}2 \wedge \mathcal{E}1 \approx_E \mathcal{E}2')$ gilt. Offenbar ist $\prec$ nicht reflexiv, nicht symmetrisch und transitiv. $\prec$ wird im folgenden zur Übersichtlichkeit des Beweises verwendet.

Wir beweisen mittels des Well-Founded-Induktionsprinzips. Sei

$$L_k = \{\mathcal{E} \mid \ \mathcal{E} = \overline{\bigcup_{i=1}^{k}} \mathcal{E}_i \ \wedge \mathcal{E}_i \in \mathcal{ES}_\mathcal{P} \text{ nicht zerlegbar } \wedge init(\mathcal{E}_i) \neq \emptyset\},$$

wobei $k \geq 1$ gilt, und $\mathcal{L} = \{L_k \,|\, k \geq 1\}$. Definiere S auf $\mathcal{L}$ wie folgt: $(L_h, L_k) \in S$, falls $h = k \Leftrightarrow 1$ gilt. Offenbar ist S wohlfundiert. Für L_n nehmen wir per Induktion an, daß für $\mathcal{E} \in L_{n-1}$ gilt: Sei

$$\mathcal{E} = \overline{\bigcup_{i=1}^{n-1}} \mathcal{E}_i \wedge \mathcal{E}' = \overline{\bigcup_{j=1}^{l}} \mathcal{E}'_j,$$

wobei $\mathcal{E}'_j \in \mathcal{ES}_\mathcal{P}$ und $init(\mathcal{E}'_j) \neq \emptyset$ gilt und $\mathcal{E}'_j$ nicht zerlegbar ist. Gilt $\mathcal{E} \approx_E \mathcal{E}'$, dann gilt $\forall i \in \{1, \dots, n \Leftrightarrow 1\} : \exists j \in \{1, \dots, l\} : \mathcal{E}_i \approx_E \mathcal{E}'_j$.

Wir zeigen nun, daß diese Aussage auch für L_n zutrifft. Sei $\mathcal{E} \in L_n$ mit $\mathcal{E} = \overline{\bigcup_{i=1}^{n}} \mathcal{E}_i$ und $\mathcal{E}' = \overline{\bigcup_{j=1}^{m}} \mathcal{E}'_j$, wobei $\mathcal{E}'_j \in \mathcal{ES}_\mathcal{P}$ und $init(\mathcal{E}'_j) \neq \emptyset$ gilt und $\mathcal{E}'_j$ nicht zerlegbar ist. Sei ferner $\mathcal{E} \approx_E \mathcal{E}'$, $\mathcal{E}_i = (E_i, \leadsto_i, \mapsto_i, l_i)$, $\mathcal{E}'_j = (E'_j, \leadsto'_j, \mapsto'_j, l'_j)$, $i = 1, \dots, n$ und $j = 1, \dots, m$.

Es sind zwei Fälle zu unterscheiden:

- Fall 1: $n = 1$. Trivial.
- Fall 2: $n > 1$

  Da E_i endlich ist, gibt es eine endliche Folge $e_i^1, e_i^2, \dots, e_i^{n_i}$, so daß gilt:

$$\mathcal{E}_i \overset{e_i^1}{\Leftrightarrow\!\!\rightarrow_i} \mathcal{E}_i[\{e_i^1\}] \cdots \Leftrightarrow\!\!\rightarrow_i \mathcal{E}_i[\{e_i^{n_i-1}\}] \overset{e_i^{n_i}}{\Leftrightarrow\!\!\rightarrow_i} \mathcal{E}_i[\{e_i^{n_i}\}]$$

und $init(\mathcal{E}[\{e_i^{n_i}\}]) = \emptyset$. Daraus können wir folgern, daß für eine beliebige Ereignisstruktur $\mathcal{E}_h$ gilt:

$$\mathcal{E}_h \approx_E (\overline{\bigcup_{i=1, i\neq h}^{n}} \mathcal{E}_i[\{e_i^{n_i}\}]) \mathbin{\overline{\cup}} \mathcal{E}_h,$$

da $(\overline{\bigcup_{i=1,i\neq h}^{n}}\mathcal{E}_i[\{e_i^{n_i}\}]) \approx_E (\emptyset, \emptyset, \emptyset, \emptyset)$ gilt. Da $(\overline{\bigcup_{i=1,i\neq h}^{n}}\mathcal{E}_i[\{e_i^{n_i}\}]) \mathbin{\overline{\cup}} \mathcal{E}_h \in Er(\mathcal{E})$ und $\mathcal{E} \approx_E \mathcal{E}'$ gilt, gibt es ein $\mathcal{E}'' \in Er(\mathcal{E})$, so daß $\mathcal{E}_h \approx \mathcal{E}''$ gilt. $\mathcal{E}''$ ist i.a. von der Form

$$\mathcal{E}'' = \overline{\bigcup_{j=1}^{m}} \mathcal{E}_j'',$$

wobei gilt:

$$\mathcal{E}_j' \overset{e_j^1}{\not\Leftrightarrow\!\!\rightarrow}{}_j' \mathcal{E}_j[\{e_j^1\}] \cdots \overset{}{\not\Leftrightarrow\!\!\rightarrow}{}_j' \mathcal{E}_j[\{e_j^{m_j-1}\}] \overset{e_j^{m_j}}{\not\Leftrightarrow\!\!\rightarrow}{}_j' \mathcal{E}_i[\{e_j^{m_j}\}] = \mathcal{E}_j'' \text{ oder } \mathcal{E}_j'' = \mathcal{E}_j'$$

Aus Lemma 4.12 läßt sich folgern, daß es zu jeder $\mathcal{E}_j''$ eine Ereignisstruktur $\mathcal{E}_j^R \in \mathcal{ES}_{\mathcal{P}}$ gibt, so daß $\mathcal{E}_j'' \approx_E \mathcal{E}_j^R$ gilt. D.h. es gilt

$$\mathcal{E}'' \approx_E \overline{\bigcup_{j=1}^{m}} \mathcal{E}_j^R.$$

Da $\mathcal{E}_h$ nicht zerlegbar ist, folgt daraus, daß es genau ein $l \in \{1, \ldots, m\}$ gibt, so daß $init(\mathcal{E}_l'') \neq \emptyset$ gilt. Dies bedeutet $\mathcal{E}_h \approx_E \mathcal{E}_l''$. Wir unterscheiden nun zwei Fälle:

1. $\mathcal{E}_l' = \mathcal{E}_l''$
   In diesem Fall gilt $\mathcal{E}_h \approx_E \mathcal{E}_l'$.
2. $\mathcal{E}_l' \neq \mathcal{E}_l''$
   Es gilt also $\mathcal{E}_h \prec \mathcal{E}_l'$.

Somit können wir folgern, daß folgendes gilt:

$$\forall i \in \{1, \ldots, n\} : \exists j \in \{1, \ldots, m\} : (\mathcal{E}_i \approx_E \mathcal{E}_j' \oplus \mathcal{E}_i \prec \mathcal{E}_j') \qquad (*)$$

($\oplus$ bedeutet „entweder .. oder ..“). Analog dazu gilt natürlich auch

$$\forall j \in \{1, \ldots, m\} : \exists i \in \{1, \ldots, n\} : (\mathcal{E}_j' \approx_E \mathcal{E}_i \oplus \mathcal{E}_j' \prec \mathcal{E}_i). \qquad (**)$$

Damit behaupten wir nun, daß folgendes gilt:

$$\exists i \in \{1, \ldots, \} : \exists j \in \{1, \ldots, m\} : \mathcal{E}_i \approx_E \mathcal{E}_j'$$

Angenommen, diese Aussage gilt nicht, d.h. es gilt

$$\forall i \in \{1, \ldots, \} : \forall j \in \{1, \ldots, m\} : \neg(\mathcal{E}_i \approx_E \mathcal{E}_j'). \qquad (+)$$

Damit folgt zusammen mit Aussagen $(*)$ und $(**)$, daß es eine Folge $h_1, \ldots, h_{n+1} \in \{1, \ldots, n\}$ und eine Folge $l_1, \ldots, l_n \in \{1, \ldots, m\}$ gibt, so daß gilt:

- $\mathcal{E}_{h_i} \prec \mathcal{E}'_{l_i}$ mit $i = 1, \ldots, n$
- $\mathcal{E}'_{l_i} \prec \mathcal{E}_{h_{i+1}}$ mit $i = 1, \ldots, n$

Da es zwei Zahlen x und y gibt, so daß $h_x = h_y$ gilt, gilt $\mathcal{E}_{h_x} = \mathcal{E}_{h_y}$. Sei o.E.d.A. $x < y$, dann gilt wegen Transitivität $\mathcal{E}_{h_x} \prec \mathcal{E}_{h_y}$. Dies ist ein Widerspruch, da $\prec$ nicht reflexiv ist.

Somit ist gezeigt, daß die Annahme (+) falsch ist.

Sei also $\mathcal{E}_h \approx_E \mathcal{E}_l$. Eliminiert man $\mathcal{E}_h$ in $\mathcal{E}$ und $\mathcal{E}'_l$ in $\mathcal{E}'$, dann erhält man mit Lemma 4.15

$$\overline{\bigcup_{i=1, i \neq h}^{n}} \mathcal{E}_i \approx_E \overline{\bigcup_{j=1, j \neq l}^{m}} \mathcal{E}'_j$$

Nach Induktionsannahme gilt

$$\forall i \in \{1, \ldots, n\} \setminus \{h\} : \exists j \in \{1, \ldots, m\} \setminus \{l\} : \mathcal{E}_i \approx_E \mathcal{E}'_j.$$

Daraus folgt $\forall i \in \{1, \ldots, n\} : \exists j \in \{1, \ldots, m\} : \mathcal{E}_i \approx_E \mathcal{E}'_j$. □

4.3.2 Teil II

In diesem Teil zeigen wir die Gültigkeit der folgenden Aussage: Seien B und C Action-Prefix-Formen. Dann gilt $EZ(B) \cong EZ(C)$, wenn $B \sim C$ gilt (siehe Satz 4.1). Diese Aussage ist für die Richtigkeit der Zerlegungsansmethode, wie später zu sehen wird, sehr entscheidend.

Lemma 4.17 *Sei $\mathcal{E}, \mathcal{E}', \mathcal{E}1, \mathcal{E}2 \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} \cong \mathcal{E}'$. Sei $\mathcal{E} = \mathcal{E}1 \overline{\cup} \mathcal{E}2$ mit $\mathcal{E}1 \neq (\emptyset, \emptyset, \emptyset, \emptyset) \neq \mathcal{E}2$ und h der Isomorphismus von $\mathcal{E}$ auf $\mathcal{E}'$. Dann gilt*

$$\exists N1 \subset Konf(\mathcal{E}') : h(\mathcal{E}1) = \overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}')$$

und $h(\mathcal{E}2) = \overline{\bigcup_{m \in N2}} Er(m, \mathcal{E}')$, wobei $N2 = Konf(\mathcal{E}) \setminus N1$.

Beweis: Sei $M = Konf(\mathcal{E})$ und $M' = Konf(\mathcal{E}')$. Wir zeigen zunächst, daß gilt:

$$\forall m \in M : h(m) \in M'$$

Es ist leicht zu sehen, daß $h(e) \in init(\mathcal{E}')$ genau dann gilt, wenn $e \in init(\mathcal{E}')$ gilt. Daraus folgt wegen der Isomorphie-Eigenschaft

$$\forall e_1, e_2 \in m : h(e_1) \bowtie_{M'} h(e_2).$$

Gilt andererseits $h(e_1) \bowtie_{M'} e'_3$, wobei $e_1 \in m$ gilt, so läßt sich wegen der Isomorphie folgern, daß es ein $e_3 \in init(\mathcal{E})$ mit $h(e_3) = e'_3$ gibt, so daß auch $e_1 \bowtie_M e_3$ gilt. Somit gilt $e_3 \in m$. Zusammengefaßt heißt das, daß $[h(e)]_{/\bowtie_{M'}} = h(m)$ gilt, wobei $e \in m$ gilt, d.h. $h(m) \in M'$.

Nach Lemma 3.8 gilt $\mathcal{E}1 = \overline{\bigcup_{m \in Konf(\mathcal{E}1)}} Er(m, \mathcal{E}1)$. Sei $N1 = \{h(m) \mid m \in Konf(\mathcal{E}1)\}$. Damit zeigen wir, daß

$$h(\mathcal{E}1) = \overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}')$$

gilt. Sei $\mathcal{E}1 = (E_1, \rightsquigarrow_1, \mapsto_1, l_1)$ und

$$\overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}') = (E_h, \rightsquigarrow_h, \mapsto_h, l_h).$$

Es genügt offenbar zu zeigen, daß $h(E_1) = E_h$ gilt, denn daraus folgt sofort $h(\mathcal{E}1) = \overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}')$.

Sei $e \in E_1$, dann gibt es wegen Lemma 3.6 genau ein $m \in Konf(\mathcal{E}1)$ und genau ein $e' \in m$, so daß $e' \vdash e$ gilt. Da $h(m) \in N1$ gilt, gilt $h(e') \in h(m)$. Wegen Isomorphie läßt sich folgern, daß auch $h(e') \vdash h(e)$ gilt. Somit gilt $h(e) \in E_h$. D.h. es gilt $h(E_1) \subseteq E_h$. Umgekehrt läßt sich analog zeigen, daß $\mathcal{E}_h \subseteq h(E_1)$ gilt. Daß $N1 \subset Konf(\mathcal{E}')$ gilt, ist offensichtlich.

Sei $\mathcal{E}2 = (E_2, \rightsquigarrow_2, \mapsto_2, l_2)$, dann gilt wegen der injektiven Eigenschaft $h(E_2) = (E' \setminus E_h)$. Da die Ereignismenge von $\overline{\bigcup_{m \in N2}} Er(m, \mathcal{E}')$ genau die Menge $(E' \setminus E_h)$ ist, gilt

$$h(\mathcal{E}2) = \overline{\bigcup_{m \in N2}} Er(m, \mathcal{E}')$$

$\square$

Korollar 4.2 *Sei $\mathcal{E}, \mathcal{E}' \in \mathcal{ES}_\mathcal{P}$ mit $\mathcal{E} \cong \mathcal{E}'$ und h der Isomorphismus von $\mathcal{E}$ auf $\mathcal{E}'$.*

1. *Sei $N \subseteq Konf(\mathcal{E})$, dann gibt es ein $M \subseteq Konf(\mathcal{E}')$, so daß*

$$h(\overline{\bigcup_{m \in N}} Er(m, \mathcal{E})) = \overline{\bigcup_{m \in M}} Er(m, \mathcal{E}')$$

   und

$$h(\overline{\bigcup_{m \in Konf(\mathcal{E}) \setminus N}} Er(m, \mathcal{E})) = \overline{\bigcup_{m \in Konf(\mathcal{E}') \setminus M}} Er(m, \mathcal{E}')$$

   gilt.

2. *Sei $n \in Konf(\mathcal{E})$, dann gilt $\exists m \in Konf(\mathcal{E}') : h(Er(n, \mathcal{E})) = Er(m, \mathcal{E}')$.*

Beweis:

- Zu 1: Für $N \subset Konf(\mathcal{E})$ folgt es aus Lemma 4.17. Für $N = Konf(\mathcal{E})$ gilt die Aussage offensichtlich.

- Zu 2: Nach 1 gibt es ein $M \subseteq Konf(\mathcal{E}')$, so daß $h(Er(n, \mathcal{E})) = \overline{\bigcup_{m \in M}} Er(m, \mathcal{E}')$ gilt. Wir behaupten, daß $|M| = 1$ gilt. Es sei angenommen $|M| > 1$ und $M = \{m_1, \ldots, m_k\}$. Wir wählen nun zwei Elemente $e_1 \in m_1$ und $e_2 \in m_2$. Dann gibt es $e'_1, e'_2 \in n$, so daß $h(e'_1) = e_1$ und $h_2(e'_2) = e_2$ gilt. Da $\neg(e_1 \bowtie_{init(\mathcal{E}_2)} e_2)$ gilt, gilt wegen Isomorphie $\neg(e'_1 \bowtie_{init(\mathcal{E}_1)} e'_2)$. Widerspruch. Also gilt $|M| = 1$. $\square$

Lemma 4.18 *Sei $\mathcal{E}, \mathcal{E}' \in \mathcal{ES}_{\mathcal{P}}$ mit $\mathcal{E} \cong \mathcal{E}'$. Sei ferner $e \in init(\mathcal{E})$ und h der Isomophismus von $\mathcal{E}$ auf $\mathcal{E}'$. Dann gilt*

$$Rest(e, \mathcal{E}) \cong Rest(h(e), \mathcal{E}')$$

Beweis: Sei $\mathcal{E} = (E, \rightsquigarrow, \mapsto, l)$ und $\mathcal{E}' = (E', \rightsquigarrow', \mapsto', l')$. Nach Definition 4.3 gilt:

- $Rest(e, \mathcal{E}) = (E_1, \rightsquigarrow_1, \mapsto_1, l_1)$ mit
  - $E_1 = E \setminus (\{e\} \cup m \cup \{e'' \mid\ e' \vdash e'' \wedge e' \in m\})$, wobei $m = \{e' \mid\ e \rightsquigarrow e'\}$.
  - $\rightsquigarrow_1 = \rightsquigarrow \cap (E_1 \times E_1)$
  - $\mapsto_1 = \mapsto \cap (2^{E_1} \times E_1)$
  - $l_1 = l \lceil E_1$
- $Rest(h(e), \mathcal{E}') = (E_2, \rightsquigarrow_2, \mapsto_2, l_2)$ mit
  - $E_2 = E \setminus (\{h(e)\} \cup m_h \cup \{e'' \mid\ e' \vdash e'' \wedge e' \in m_h\})$, wobei $m_h = \{e' \mid\ h(e) \rightsquigarrow' e'\}$.
  - $\rightsquigarrow_2 = \rightsquigarrow \cap (E_2 \times E_2)$
  - $\mapsto_2 = \mapsto \cap (2^{E_2} \times E_2)$
  - $l_2 = l' \lceil E_2$

Es genügt also zu zeigen, daß $h(E_1) = E_2$ gilt, denn daraus folgt sofort $Rest(e, \mathcal{E}) \cong Rest(h(e), \mathcal{E}')$.

- „$\subseteq$:“ Sei $\phi \in E_1$, dann gilt $h(\phi) \neq h(e)$ und $\neg(\psi \vdash h(\phi))$ mit $\psi \in m_h$. Gilt nämlich $\psi \vdash h(\phi)$, dann gibt es ein Ereignis $\psi' \in E$, so daß $h(\psi') = \psi$ und $\psi' \vdash \phi$ gilt. Da wegen Isomorphie $e \rightsquigarrow \psi'$ gilt und nach Lemma 3.6 ψ' das einzige Ereignis ist, so daß $\psi' \vdash \phi$ gilt, gilt $\phi \notin E_1$. Widerspruch. Also gilt $h(\phi) \in E_2$ und somit $h(E_1) \subseteq E_2$
- „$\supseteq$:“ Die Umkehrung gilt analog. □

Lemma 4.19 *Sei $\mathcal{E}_i \in \mathcal{ES}_{\mathcal{P}}$ mit $i = 1, 2, 3, 4$, $\mathcal{E}_1 \cong \mathcal{E}_3$ und $\mathcal{E}_2 \cong \mathcal{E}_4$. Seien ferner*

- *$f_1 : \{\phi_1, \ldots, \phi_n\} \to \mathcal{G}$ und $f_2 : \{\psi_1, \ldots, \psi_m\} \to \mathcal{G}$ Funktionen mit $f_1(\phi_1) = f_2(\psi_1)$ und $f_1(\phi_2) = f_2(\psi_2)$.*
- *h_1 der Isomorphismus von $\mathcal{E}_1$ auf $\mathcal{E}_3$ und h_2 der Isomorphismus von $\mathcal{E}_2$ auf $\mathcal{E}_4$.*

Dann gilt

1. *$Bed1((\phi_1, \phi, \mathcal{E}_1), f_1, (\phi_2, \phi', \mathcal{E}_2)) = Bed1((\psi_1, h_1(\phi), \mathcal{E}_3), f_2, (\psi_2, h_2(\phi'), \mathcal{E}_4))$.*
2. *$Kons(\phi, \mathcal{E}_1, \phi', \mathcal{E}_2) \cong Kons(h_1(\phi), \mathcal{E}_3, h_2(\phi'), \mathcal{E}_4)$*

Beweis:

1. <u>Zu 1:</u> Wir zeigen im folgenden nur die folgende Behauptung: Gilt

$$Bed1((\phi_1, \phi, \mathcal{E}_1), f_1, (\phi_2, \phi', \mathcal{E}_2)) = wahr,$$

dann gilt $Bed1((\psi_1, h_1(\phi), \mathcal{E}_3), f_2, (\psi_2, h_2(\phi'), \mathcal{E}_4)) = wahr$. Die umgekehrte Richtung gilt analog.

Es ist also zu zeigen, daß die Bedingungen 1(b), 1(c) und 1(d) der Definition 4.15 erfüllt sind.

(a) <u>Zu 1(b) :</u> Sei $m1 \in Konf(\mathcal{E}_1)$ mit $\phi \in m1$ und $m2 \in Konf(\mathcal{E}_2)$ mit $\phi' \in m2$. Aus

$$\overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}_1) \cong Rest(\phi', Er(m2, \mathcal{E}_2)),$$

wobei $m1 \notin N1$ gilt, folgt mit Korollar 4.2 und Lemma 4.18, daß es eine Menge $N3 \subseteq Konf(\mathcal{E}_3)$ und ein $m4 \in Konf(\mathcal{E}_4)$ gibt, so daß gilt:

$$\overline{\bigcup_{m \in N3}} Er(m, \mathcal{E}_3) \cong Rest(h_2(\phi'), Er(m4, \mathcal{E}_4))$$

Dabei gilt $h_1(\overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}_1)) = \overline{\bigcup_{m \in N3}} Er(m, \mathcal{E}_3)$ und $h_2(Er(m2, \mathcal{E}_2)) = Er(m4, \mathcal{E}_4)$. Sei $m3 \in Konf(\mathcal{E}3)$ und $h_1(\phi) \in m3$, dann gilt offenbar $m3 \notin N3$. Damit ist gezeigt, daß die Bedingung 1(b) erfüllt ist.

(b) <u>Zu 1(c) :</u> Analog zu (a).

(c) <u>Zu 1(d) :</u> Die Erfüllbarkeit der Bedingung 1(d) folgt leicht aus (a) und (b).

2. <u>Zu 2:</u> Trivial. □

Lemma 4.20 *Sei B eine Action-Prefix-Form und $\mathcal{E} = EZ(B)$. Dann gilt*

*1. $\mathcal{E} \neq (\emptyset, \emptyset, \emptyset, \emptyset)$, falls $B \neq$ **stop**.*

2. $\mathcal{E} \in \mathcal{ES}_{\mathcal{P}}$.

Beweis:

1. <u>Zu 1:</u> Trivial.

2. <u>Zu 2:</u> Wir beweisen mittels des Well-Founded-Induktionsprinzips. Sei

$$L = \{P \mid \ P \text{ ist eine Action-Prefix-Form}\}.$$

Dann gilt also $B \in L$. Sei $\prec$ auf L wie folgt definiert: $P' \prec P$, falls $P = \sum_{i \in I} b_i; P_i$, $P' \in \{P_i \mid i \in I\}$ und $I \neq \emptyset$ gilt. $\prec$ ist offenbar wohlfundiert. Für $B \in L$ nehmen wir per Induktion an, daß die Aussage 2. für B' mit $B' \prec B$ bereits gilt. Es sind zwei Fälle zu unterscheiden:

- $B =$ **stop**
  Es gilt $EZ(B) = (\emptyset, \emptyset, \emptyset, \emptyset)$ und somit $EZ(B) \in \mathcal{ES}_{\mathcal{P}}$.

- $B \neq \mathbf{stop}$

  Sei $B = \sum_{i=1}^{n} b_i; B_i$ und

  $$M = \{(e_1, EZ(B_1)), \ldots, (e_n, EZ(B_n))\},$$

  wobei $e_i \notin init(\mathcal{E}_i)$ und $\mathcal{E}_i = EZ(B_i)$ mit $i = 1, \ldots, n$ gilt. Sei ferner $f : \{e_1, \ldots, e_n\} \to \mathcal{G}$ mit $f(e_i) = b_i$.

  Dann gilt per Induktionsannahme $EZ(B_i) \in \mathcal{ES}_{\mathcal{P}}$.

  Offenbar genügt es zu zeigen, daß $\mathcal{E}' \in \mathcal{ES}_{\mathcal{P}}$ für $\mathcal{E}' \in Bilde(f, M)$ gilt, denn daraus folgt sofort $EZ(B) \in \mathcal{ES}_{\mathcal{P}}$. Dazu sind es zwei Fälle zu unterscheiden:

  a) Es gibt ein $i \in \{1, \ldots, n\}$, so daß $\mathcal{E}' = \overline{f(e_i)_{e_i}}; \mathcal{E}_i$, d.h. $Ver((e_i, \mathcal{E}_i), f, M) = \{\mathcal{E}'\}$.

     In diesem Fall unterscheiden wir zwei Fälle.

     – $\mathcal{E}_i \neq (\emptyset, \emptyset, \emptyset, \emptyset)$

       Es gilt aso $B_i \neq \mathbf{stop}$. Da nach Induktionsannahme $\mathcal{E}_i \in \mathcal{ES}_{\mathcal{P}}$ gilt, gilt offenbar auch $\mathcal{E}' \in \mathcal{ES}_{\mathcal{P}}$.

     – $\mathcal{E}_i = (\emptyset, \emptyset, \emptyset, \emptyset)$

       Es gilt also $\mathcal{E}' = (\{e_i\}, \emptyset, \emptyset, \{(e_i, f(e_i))\})$ und somit $\mathcal{E}' \in \mathcal{ES}_{\mathcal{P}}$.

  b) Es gibt $i1, i2 \in \{1, \ldots, n\}$, so daß $\mathcal{E}'' = Kons(e1, \mathcal{E}_{i1}, e2, \mathcal{E}_{i2})$ gilt. Nach Definition 4.15 gilt

  $$\begin{aligned}
  \mathcal{E}' &= 1(\bigcup_{k=1}^{\overline{K}} Er(m_k, \mathcal{E}_{i1})) \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_{i2})) \\
  &= 1(Er(m_1, \mathcal{E}_{i1})) \mathbin{\overline{\cup}} 1(Er(m_2, \mathcal{E}_{i1})) \cdots \mathbin{\overline{\cup}} 1(Er(m_K, \mathcal{E}_{i1})) \\
  &\quad \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_{i2})),
  \end{aligned}$$

  Dabei gilt $m_k \in Konf(\mathcal{E}_{i1})$ mit $k = 1, \ldots, K$ und $m2 \in Konf(\mathcal{E}_{i2})$. Da $\mathcal{E}_{i1} \neq (\emptyset, \emptyset, \emptyset, \emptyset) \neq \mathcal{E}_{i2}$ gilt, gilt $B_{i1} \neq \mathbf{stop} \neq B_{i2}$. Induktionsannahme ist daher anwendbar.

  Da nach Induktionsannahme $\mathcal{E}_{i1}, \mathcal{E}_{i2} \in \mathcal{ES}_{\mathcal{P}}$ gilt, gilt nach Lemma 3.8 $Er(m, \mathcal{E}_{i1}) \in \mathcal{ES}_{\mathcal{P}}$ mit $m \in Konf(\mathcal{E}_{i1})$ (entsprechend $Er(m, \mathcal{E}_{i2}) \in \mathcal{ES}_{\mathcal{P}}$ mit $m \in Konf(\mathcal{E}_{i2})$). Daraus folgt mit Lemma 4.4 $\mathcal{E}' \in \mathcal{ES}_{\mathcal{P}}$. □

Satz 4.1 *Seien B und C Action-Prefix-Formen mit $B \sim C$. Dann gilt $EZ(B) \cong EZ(C)$.*

Beweis: Wir zeigen die Gültigkeit von $EZ(B) \cong EZ(C)$ mit Hilfe des Well-Founded-Induktionsprinzips. Sei $L = \{P \mid P \text{ ist eine Action-Prefix-Form}\}$. Sei $\prec$ auf L wie folgt definiert: $P' \prec P$, falls $P = \sum_{i \in I} b_i; P_i$ mit $I \neq \emptyset$ und $P' \in \{P_i \mid i \in I\}$. Offenbar ist $\prec$ wohldefiniert.

Für $B \in L$ nehmen wir per Induktion an, daß $EZ(B') \cong EZ(Q)$ gilt, falls $B' \prec B$ und $Q \sim B'$ gilt und Q eine Action-Prefix-Form ist. Es sind zwei Fälle zu unterscheiden:

- $B = \mathbf{stop}$

  In diesem Fall gilt $C = \mathbf{stop}$ und somit $EZ(B) = (\emptyset, \emptyset, \emptyset, \emptyset) = EZ(C)$.

- $B \neq \textbf{stop}$

  Sei $B = \sum_{i \in I} b_i; B_i$, dann gilt auch $C = \sum_{j \in J} c_j; C_j$. Sei $I = \{1, \ldots, n\}$ und $J = \{1, \ldots, m\}$. Nach Definition 4.17 gilt $EZ(B) = Erz(f_B, M_B)$ mit

  $$M_B = \{(\phi_1, \mathcal{E}_i^B), \ldots, (\phi_n, \mathcal{E}_n^B)\},$$

  wobei $f_B : \{\phi_1, \ldots, \phi_n\} \rightarrow \mathcal{G}$, $f_B(\phi_i) = b_i$ und $\mathcal{E}_i^B = EZ(B_i)$ gilt. Entsprechend gilt $EZ(C) = Erz(f_C, M_C)$ mit

  $$M_C = \{(\psi_1, \mathcal{E}_i^C), \ldots, (\psi_n, \mathcal{E}_m^C)\},$$

  wobei $f_C : \{\psi_1, \ldots, \psi_m\} \rightarrow \mathcal{G}$, $f_C(\psi_j) = c_j$ und $\mathcal{E}_j^C = EZ(C_j)$ gilt.

  Da $C \sim B$ gilt, gibt es zu jedem $b_i; B_i$ ein $c_j; C_j$, so daß $b_i; B_i \equiv c_j; C_j$ gilt und umgekehrt. Nach Induktionsannahme gilt $\mathcal{E}_i^B \cong \mathcal{E}_j^C$. Nach Lemma 4.20 gilt $\mathcal{E}_i^B, \mathcal{E}_j^C \in \mathcal{ES}_{\mathcal{P}}$. Somit ist das Lemma 4.19 anwendbar. Daher gilt:

  - $Bed1((\phi_i, \phi, \mathcal{E}_i^B), f_B, (\phi_h, \phi', \mathcal{E}_h^B)) = Bed1((\psi_j, h_1(\phi), \mathcal{E}_j^C), f_C, (\psi_k, h_2(\phi'), \mathcal{E}_k^C))$
  - $Kons(\phi, \mathcal{E}_i^B, \phi', \mathcal{E}_h^B) \cong Kons(h_1(\phi), \mathcal{E}_j^C, h_2(\phi'), \mathcal{E}_k^C)$

  Dabei

  - ist h_1 der Isomorphismus von $\mathcal{E}_i^B$ auf $\mathcal{E}_j^C$ und h_2 der Isomorphismus von $\mathcal{E}_h^B$ auf $\mathcal{E}_k^C$.
  - gilt $b_i; B_i \equiv c_j; C_j$ und $b_h; B_h \equiv c_k; C_k$.

  Mit Lemma 4.18 läßt sich folgern, daß $Bed2((\phi_i, \phi, \mathcal{E}_i^B), f_B, (\phi_h, \phi', \mathcal{E}_h^B), M_B)$ $= Bed2((\psi_j, h_1(\phi), \mathcal{E}_j^C), f_C, (\psi_k, h(\phi'), \mathcal{E}_k^C), M_C)$ gilt. Daraus folgt, daß es zu jeder Ereignisstruktur $\mathcal{E} \in Bilde(f_B, M_B)$ eine Ereignisstruktur $\mathcal{E}' \in Bilde(f_C, M_C)$ gibt, so daß $\mathcal{E} \cong \mathcal{E}'$ gilt und umgekehrt. Somit gilt $EZ(B) \cong EZ(C)$. □

4.3.3 Teil III

Nachdem die erforderlichen Lemmata bereits nachgewiesen wurden, sind wir nun in der Lage, zu zeigen, daß $PZ(B)$ der gesuchte Prozeß ist, der das Zerlegungsproblem löst (siehe Korollar 4.5).

Definition 4.20 ($FB(B)$) *Sei B eine Action-Prefix-Form. Dann ist $FB(B)$ induktiv wie folgt definiert:*

- *Sei $B = \textbf{stop}$, dann ist $FB(B) := \{\textbf{stop}\}$.*
- *Sei $B \neq \textbf{stop}$, dann ist $FB(B) := Er(B) \setminus \{B\}$.* □

Definition 4.21 ($|B|$) *Sei $B \in \mathcal{BL}_{\mathcal{P}}$. Dann ist $|B|$ induktiv wie folgt definiert ist:*

- *Ist $B = \textbf{stop}$, dann $|B| := 1$.*
- *Ist $B = g; B'$, dann $|B| := 1 + |B'|$.*

- *Ist* $B = B_1 \,[]\, B_2$*, dann* $|B| := |B_1| + |B_2|$.
- *Ist* $B = B_1 \,|||\, B_2$*, dann* $|B| := |B_1| + |B_2|$. □

Lemma 4.21 *Sei* B *eine Action-Prefix-Form, dann gilt*

$$B' \in FB(B) \Longrightarrow |B'| \leq |B|.$$

Beweis: Sei $L = \{P \mid P \text{ ist eine Action-Prefix-Form}\}$. Definiere $\prec$ auf L wie folgt: $B' \prec B$, falls $B = \sum_{i=1}^{n} b_i; B_i$ und $\exists i : B' = B_i$ gilt. $\prec$ ist offenbar wohlfundiert. Für $B \in L$ nehmen wir per Induktion an, daß für $B' \prec B$ gilt: $B'' \in FB(B') \Longrightarrow |B''| \leq |B'|$. Um zu zeigen daß dies auch für B gilt, sind es zwei Fälle zu unterscheiden.

1. $\neg(\exists B' : B' \prec B)$, d.h. $B = \textbf{stop}$. Trivial.
2. $\exists B' : B' \prec B$, d.h. $B \neq \textbf{stop}$. Sei $B = \sum_{i=1}^{n} b_i; B_i$. Da $|B| = \sum_{i=1}^{n}(1 + |B_i|)$ und nach Induktionsannahme

$$B'_i \in FB(B_i) \Longrightarrow |B'_i| \leq |B_i|$$

   gilt, gilt $|B'| \leq |B|$ für $B' \in FB(B)$. □

Korollar 4.3 *Sei* $B \neq \textbf{stop}$ *eine Action-Prefix-Form, dann gilt*

$$\forall B' \in FB(B) \Longrightarrow |B'| < |B|.$$

□

Lemma 4.22 *Sei* $B \neq \textbf{stop}$ *eine Action-Prefix-Form mit* $B = \sum_{i=1}^{n} b_i; B_i$ *und*

$$M = \{(e_1, \mathcal{E}_1), \ldots, (e_n, \mathcal{E}_n)\},$$

wobei $\mathcal{E}_i = EZ(B_i)$ *und* $e_i \notin \mathcal{E}_i$ *für* $i = 1, \ldots, n$ *gilt. Sei ferner* $\mathcal{E} = EZ(B)$, $\mathcal{E}_B = [\![B]\!]$ *und* $f : \{e_1, \ldots, e_n\} \to \mathcal{G}$ *mit* $f(e_i) = b_i$*. Dann gilt:*

1. *Sei* $e \in init(\mathcal{E})$*, dann gilt* $\exists i : (Rest(e, \mathcal{E}) \cong \mathcal{E}_i \wedge b_i = l(e))$.
2. $\mathcal{E} \approx_E \mathcal{E}_B$
3. *Sei* $\mathcal{E}', \mathcal{E}'' \in Bilde(f, M)$ *mit* $\mathcal{E}' \approx_E \mathcal{E}''$*. Dann gilt* $\mathcal{E}' \cong \mathcal{E}''$.
4. *Sei* $\mathcal{E}_C, \mathcal{E}1_C, \mathcal{E}2_C \in \mathcal{ES}_\mathcal{P}$ *mit* $\mathcal{E}_C = \mathcal{E}1_C \,\overline{\cup}\, \mathcal{E}2_C$ *und* $init(\mathcal{E}1_C) \neq \emptyset \neq init(\mathcal{E}2_C)$*. Gilt* $\mathcal{E}_C \approx_E \mathcal{E}$*, dann gilt*

$$\exists \mathcal{E}1, \mathcal{E}2 \in \mathcal{ES}_\mathcal{P} : (\mathcal{E} = \mathcal{E}1 \,\overline{\cup}\, \mathcal{E}2 \wedge \mathcal{E}1_C \sqsubseteq \mathcal{E}1 \wedge \mathcal{E}2_C \sqsubseteq \mathcal{E}2).$$

5. $Er(m, \mathcal{E})$ *mit* $m \in Konf(\mathcal{E})$ *ist nicht zerlegbar. Außerdem gilt* $\exists B' \in Er(B) : EZ(B') \cong Er(m, \mathcal{E})$.
6. $\forall \mathcal{E}' \in \mathcal{ES}_\mathcal{P} : (\mathcal{E}' \approx_E \mathcal{E} \Longrightarrow \mathcal{E}' \sqsubseteq \mathcal{E})$

Beweis: Wir beweisen mittels des Well-Founded-Induktionsprinzips. Sei

$$L = \{P \mid \ P \neq \textbf{stop} \text{ ist eine Action-Prefix-Form}\}.$$

B ist also in L. Definiere $\prec$ auf L wie folgt: Sei $B \in L$. $B' \prec B$, falls $B' \in FB(B)$ und $B' \neq \textbf{stop}$ gilt. Nach Korollar 4.3 gilt $B' \prec B \Rightarrow |B'| < |B|$. Da $<$ auf der Menge der natürlichen Zahlen wohlfundiert ist, ist $\prec$ wohlfundiert. Für $B \in L$ nehmen wir daher per Induktion an, daß die Aussagen 1 bis 6 für B' mit $B' \prec B$ und $B \in L$ gelten. Zur einfachen Schreibweise wird im folgenden l als Markierungsfunktion für alle Ereignisstrukturen verwendet. Es sind zwei Fälle zu unterscheiden:

Fall 1: $\neg(\exists B' \in L : B' \prec B)$. In diesem Fall gilt $B = \sum_{i=1}^{n} b_i; \textbf{stop}$.

- Zu 1 und 2: Trivial.
- Zu 3: Für $\mathcal{E}' \in Bilde(f, M)$ gilt $\mathcal{E}' = \overline{f(e_i)_{e_i}}; (\emptyset, \emptyset, \emptyset, \emptyset)$ mit $i \in \{1, \ldots, n\}$. Daraus folgt sofort die Behauptung.
- Zu 4: $\mathcal{E}_C$ kann nicht existieren, da $\mathcal{E}_C \not\approx_E \mathcal{E}$ gilt. Der Grund ist, daß es in $\mathcal{E}$ keine Ereignisse ϕ_1 und ϕ_2 gibt, so daß $\mathcal{E} \overset{\phi_1}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}'_B \overset{\phi_2}{\Leftrightarrow\!\!\!\rightarrow} \mathcal{E}''_B$ gilt. Dagegen existieren in $\mathcal{E}_C$ immer zwei solche Ereignisse. Die Aussage 4 gilt also immer.
- Zu 5: Aus 4 ist $\mathcal{E}$ nicht zerlegbar. Da $Konf(\mathcal{E}) = \{\{e_1, \ldots, e_n\}\}$ gilt, ist $Er(m, \mathcal{E}) = \mathcal{E}$ mit $m \in Konf(\mathcal{E})$ und somit nicht zerlegbar. Es gilt offenbar $Er(m, \mathcal{E}) = EZ(B)$.
- Zu 6: Da $\mathcal{E}$ nicht zerlegbar ist, gilt $Zer(\mathcal{E}) = Zer(\mathcal{E}') = 1$, falls $\mathcal{E} \approx_E \mathcal{E}'$ gilt.

Fall 2: $\exists B' \in L : B' \prec B$. In diesem Fall gilt $B = \sum_{i=1}^{n} b_i; B_i$, wobei es mindestens ein $i \in \{1, \ldots, n\}$ gibt, so daß $B_i \neq \textbf{stop}$ gilt. Induktionsannahmen können also angewendet werden.

- Zu 1: Es ist zu zeigen, daß für $\mathcal{E}_P \in Bilde(f, M)$ und $e \in init(\mathcal{E}_P)$ gilt:

  $$\exists B_i : (Rest(e, \mathcal{E}_P) \cong EZ(B_i) \wedge b_i = l(e))$$

  Gilt nämlich diese Aussage, dann folgt offenbar auch die Gültigkeit dieser Aussage für $\mathcal{E}$ (siehe Lemma 4.11(2)). Wir unterscheiden zwei Fälle:

  a) Es gibt ein $i \in \{1, \ldots, n\}$, so daß $\mathcal{E}_P = \overline{f(e_i)_{e_i}}; \mathcal{E}_i$ gilt, d.h. $Ver((e_i, \mathcal{E}_i), f, M) = \{\mathcal{E}_P\}$. Offenbar gilt $Rest(e, \mathcal{E}_P) = \mathcal{E}_i$ und $b_i = l(e)$.

  b) Es gibt $i1, i2 \in \{1, \ldots, n\}$, so daß $\mathcal{E}'' = Kons(e1, \mathcal{E}_{i1}, e2, \mathcal{E}_{i2})$ gilt.
  In diesem Fall ist die Aussage wegen der Bedingung $Bed2((e_i, e1, \mathcal{E}_i), f, (e_j, e2, \mathcal{E}_j)))$ erfüllt.

- Zu 2: Es folgt aus Aussage 1 und Lemma 4.10.
- Zu 3: O.E.d.A. sind es drei Fälle zu unterscheiden:

  a) Es gibt ein $i \in \{1, \ldots, n\}$, so daß $\mathcal{E}' = \overline{f(e_i)_{e_i}}; \mathcal{E}_i$ gilt. Entsprechend gibt es ein $j \in \{1, \ldots, n\}$, so daß $\mathcal{E}'' = \overline{f(e_j)_{e_j}}; \mathcal{E}_j$ gilt.
  In diesem Fall sind es zwei Fälle zu unterscheiden:

- $\mathcal{E}_i \neq (\emptyset, \emptyset, \emptyset, \emptyset)$
  Es gilt wegen Lemma 4.20 $B_i \neq \textbf{stop}$. Da B_i eine Action-Prefix-Form ist, gilt $B_i \not\sim \textbf{stop}$. Nach Induktionsannahme 2 gilt $\mathcal{E}_i \approx_E [\![B_i]\!]$ und $\mathcal{E}_j \approx_E [\![B_j]\!]$. Da $\mathcal{E}_i \approx_E \mathcal{E}_j$ gilt, gilt $[\![B_i]\!] \approx_E [\![B_j]\!]$ und nach Lemma 2.4 $B_i \sim B_j$, was nach Satz 4.1 $\mathcal{E}_i \cong \mathcal{E}_j$ bedeutet. Damit läßt sich leicht eine Funktion konstruieren, die der Isomorphismus von $\mathcal{E}'$ auf $\mathcal{E}''$ ist.
- $\mathcal{E}_i = (\emptyset, \emptyset, \emptyset, \emptyset)$
  In diesem Fall gilt $B_i = \textbf{stop}$. Da $\mathcal{E}_i \approx_E \mathcal{E}_j$ gilt, gilt $\mathcal{E}_j = (\emptyset, \emptyset, \emptyset, \emptyset)$, denn sonst gilt $B_j \neq \textbf{stop}$, was $B_j \not\sim \textbf{stop}$ bedeutet. Nach Induktionsannahme 2 gilt dann $\mathcal{E}_j \approx_E [\![B_j]\!]$, woraus $\mathcal{E}_i \not\approx_E \mathcal{E}_j$ folgt. Widerspruch.
  Zusammengefaßt heißt das, daß $\mathcal{E}' = (\{e_i\}, \emptyset, \emptyset, \{(e_i, f(e_i))\})$ und $\mathcal{E}'' = (\{e_j\}, \emptyset, \emptyset, \{(e_j, f(e_j))\})$ gilt. Also gilt $\mathcal{E}' \cong \mathcal{E}''$.

b) Es gibt ein $i \in \{1, \ldots, n\}$, so daß $\mathcal{E}' = \overline{f(e_i)_{e_i}};\mathcal{E}_i$ gilt. Es gibt $i1, i2 \in \{1, \ldots, n\}$, so daß $\mathcal{E}'' = Kons(e, \mathcal{E}_{i1}, e', \mathcal{E}_{i2})$ gilt.

   Nach Lemma 4.10 gilt $Rest(1e, \mathcal{E}'') \cong \mathcal{E}_{i2}$ und $Rest(2e', \mathcal{E}'') \cong \mathcal{E}_{i1}$. Aus $\mathcal{E}' \approx_E \mathcal{E}''$ folgt mit Lemma 4.12 $l(e_i) = l(1e) = l(2e')$ und $\mathcal{E}_i \approx_E \mathcal{E}_{i1} \approx_E \mathcal{E}_{i2}$. Wegen Induktionsannahme 2 und Lemma 2.4 gilt $B_i \sim B_{i1} \sim B_{i2}$ und somit $\mathcal{E}_i \cong \mathcal{E}_{i1} \cong \mathcal{E}_{i2}$.

   Sei h_1 der Isomorphismus von $\mathcal{E}_{i1}$ auf $\mathcal{E}_i$ und h_2 der Isomorphismus von $\mathcal{E}_{i2}$ auf $\mathcal{E}_i$, dann gilt nach Lemma 4.19

$$Kons(h_1(e), \mathcal{E}_i, h_2(e), \mathcal{E}_i) \cong Kons(e, \mathcal{E}_{i1}, e', \mathcal{E}_{i2}).$$

   Somit gilt mit Lemma 4.18 $Bed2((e_i, h_1(e), \mathcal{E}_i), f, (e_i, h_2(e'), \mathcal{E}_i), ..) = wahr$, was

$$Kons(h_1(e), \mathcal{E}_i, h_2(e'), \mathcal{E}_i) \in \bigcup_{e \in init(\mathcal{E}_i)} Bes((e_i, e, \mathcal{E}_i), f, M),$$

   impliziert, d.h. $\bigcup_{e \in init(\mathcal{E}_i)} Bes((e_i, e, \mathcal{E}_i), f, M) \neq 0$. Dies bedeutet aber $\mathcal{E}' \neq \overline{f(e_i)_{e_i}};\mathcal{E}_i$. Widerspruch. D.h. dieser Fall kann nicht vorkommen und wird daher nicht weiter betrachtet.

c) Es gibt $i1, i2, i3, i4 \in \{1, \ldots, n\}$, so daß $\mathcal{E}'' = Kons(e1, \mathcal{E}_{i1}, e2, \mathcal{E}_{i2})$ und $\mathcal{E}'' = Kons(e3, \mathcal{E}_{i3}, e4, \mathcal{E}_{i4})$ gilt. Nach Definition 4.15 gilt

$$\begin{aligned}
\mathcal{E}' &= 1(\bigcup_{k=1}^{\overline{K1}} Er(m_k, \mathcal{E}_{i1})) \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_{i2})) \\
&= 1(Er(m_1, \mathcal{E}_{i1})) \mathbin{\overline{\cup}} 1(Er(m_2, \mathcal{E}_{i1})) \cdots \mathbin{\overline{\cup}} 1(Er(m_{K1}, \mathcal{E}_{i1})) \\
&\quad \mathbin{\overline{\cup}} 2(Er(m2, \mathcal{E}_{i2}))
\end{aligned}$$

   und

$$\begin{aligned}
\mathcal{E}'' &= 1(\bigcup_{j=1}^{\overline{K3}} Er(n_j, \mathcal{E}_{i3})) \mathbin{\overline{\cup}} 2(Er(m4, \mathcal{E}_{i4})) \\
&= 1(Er(n_1, \mathcal{E}_{i1})) \mathbin{\overline{\cup}} 1(Er(n_2, \mathcal{E}_{i1})) \cdots \mathbin{\overline{\cup}} 1(Er(n_{K3}, \mathcal{E}_{i1})) \\
&\quad \mathbin{\overline{\cup}} 2(Er(m4, \mathcal{E}_{i4})).
\end{aligned}$$

Dabei gilt $m_k \in Konf(\mathcal{E}_{i1})$ mit $k = 1, \dots, K1$, $n_j \in Konf(\mathcal{E}_{i3})$ mit $j = 1, \dots, K3$, $m2 \in Konf(\mathcal{E}_{i2})$ und $m4 \in Konf(\mathcal{E}_{i4})$.

Zunächst ist es offensichtlich , daß $B_{i1} \neq \mathbf{stop} \neq B_{i2}$ und $B_{i3} \neq \mathbf{stop} \neq B_{i4}$ gilt. Somit können Induktionsannahmen auf B_{i1}, B_{i2}, B_{i3} und B_{i4} angewendet werden.

Aus Induktionsannahme 5. sind $1(Er(m_k, \mathcal{E}_{i1}))$ und $1(Er(n_j, \mathcal{E}_{i3}))$ mit $k = 1, \dots, K1$ und $j = 1, \dots, K3$ nicht zerlegbar. Ebenfalls sind $2(Er(m2, \mathcal{E}_{i2}))$ und $2(Er(m4, \mathcal{E}_{i4}))$ nicht zerlegbar. Außerdem gilt mit Lemma 3.8

$$1(Er(m_k, \mathcal{E}_{i1})), 1(Er(n_j, \mathcal{E}_{i3})), 2(Er(m2, \mathcal{E}_{i2})), 2(Er(m4, \mathcal{E}_{i4})) \in \mathcal{ES}_{\mathcal{P}}.$$

Sei $\mathcal{E}'_k = 1(Er(m_k, \mathcal{E}_{i1}))$, wobei $k = 1, \dots, K1$, und $\mathcal{E}''_j = 1(Er(n_j, \mathcal{E}_{i3}))$, wobei $j = 1, \dots, K3$, sowie $\mathcal{E}'_{K1+1} = 2(Er(m2, \mathcal{E}_{i2}))$ und $\mathcal{E}''_{K3+1} = 2(Er(m4, \mathcal{E}_{i4}))$.
Dann gilt nach Lemma 4.16

$$\forall k \in \{1, \dots, K1+1\} : \exists j \in \{1, \dots, K3+1\} : \mathcal{E}'_k \approx_E \mathcal{E}''_j.$$

Aus Induktionsannahme 5 folgt $\exists B' \in Er(B_{i1}) : \mathcal{E}'_k \cong EZ(B')$ für $k = 1, \dots, K1$ und $\exists B' \in Er(B_{i2}) : \mathcal{E}'_{K1+1} \cong EZ(B')$. Analog dazu gilt natürlich auch $\exists B'' \in Er(B_{i3}) : \mathcal{E}''_j \cong EZ(B'')$ für $j = 1, \dots, K3$ und $\exists B'' \in Er(B_{i4}) : \mathcal{E}''_{K3+1} \cong EZ(B'')$.
Da nach Induktionsannahme 2 $EZ(B') \cong [\![B']\!]$ (bzw. $EZ(B'') \cong [\![B'']\!]$) gilt, folgt mit Lemma 2.4 und Satz 4.1 $\mathcal{E}'_k \cong \mathcal{E}''_j$, falls $\mathcal{E}'_k \approx_E \mathcal{E}''_j$ gilt.
Aus

$$\forall k \in \{1, \dots, K1+1\} : \exists j \in \{1, \dots, K3+1\} : \mathcal{E}'_k \cong \mathcal{E}''_j$$

läßt sich dann leicht mit Lemma 4.15 folgern, daß $\mathcal{E}' \cong \mathcal{E}''$ gilt. Man braucht nur das Lemma 4.15 und Lemma 4.16 $K1+1$-mal anzuwenden.

- Zu 4: Sei $\mathcal{E}1_C \overset{\phi_1}{\Longleftrightarrow} \mathcal{E}1'_C$ und $\mathcal{E}2_C \overset{\phi_2}{\Longleftrightarrow} \mathcal{E}2'_C$, dann gilt

$$\mathcal{E}_C \overset{\phi_1}{\Longleftrightarrow} \mathcal{E}1'_C \mathbin{\overline{\cup}} \mathcal{E}2_C \ \wedge \ \mathcal{E}_C \overset{\phi_2}{\Longleftrightarrow} \mathcal{E}1_C \mathbin{\overline{\cup}} \mathcal{E}2'_C.$$

Da $\mathcal{E}_C \approx_E \mathcal{E}$ gilt, gilt $\mathcal{E} \overset{\psi_1}{\Longleftrightarrow} \mathcal{E}1$ und $\mathcal{E} \overset{\psi_2}{\Longleftrightarrow} \mathcal{E}2$ mit

  - $l(\phi_1) = l(\psi_1)$ und $l(\phi_2) = l(\psi_2)$
  - $\mathcal{E}1 \approx_E Rest(\psi_1, \mathcal{E}) \approx_E \mathcal{E}1'_C \mathbin{\overline{\cup}} \mathcal{E}2_C \approx_E Rest(\phi_1, \mathcal{E}1_C) \mathbin{\overline{\cup}} \mathcal{E}2_C$
  - $\mathcal{E}2 \approx_E Rest(\psi_2, \mathcal{E}) \approx_E \mathcal{E}1_C \mathbin{\overline{\cup}} \mathcal{E}2'_C \approx_E \mathcal{E}1_C \mathbin{\overline{\cup}} Rest(\phi_2, \mathcal{E}2_C)$

Nach Aussage 1 gibt es i und j, so daß gilt:

  - $l(\psi_1) = b_i$ und $l(\psi_2) = b_j$
  - $Rest(\psi_1, \mathcal{E}) \approx_E \mathcal{E}_i$ und $Rest(\psi_2, \mathcal{E}) \approx_E \mathcal{E}_j$.

Damit wird im folgenden gezeigt, daß es eine Ereignisstruktur $\mathcal{E}_P \in Bilde(f, M)$ gibt, so daß gilt:

  1. $\mathcal{E}_P \approx_E \mathcal{E}_C$

2. $\exists \mathcal{E}1_P, \mathcal{E}2_P \in \mathcal{ES_P} : (\mathcal{E}_P = \mathcal{E}1_P \;\overline{\cup}\; \mathcal{E}2_P \wedge \mathcal{E}1_C \sqsubseteq \mathcal{E}1_P \wedge \mathcal{E}2_C \sqsubseteq \mathcal{E}2_P)$

Wir unterscheiden vier Fälle:

a) $\mathit{init}(\mathit{Rest}(\phi_1, \mathcal{E}1_C)) \neq \emptyset \neq \mathit{init}(\mathit{Rest}(\phi_2, \mathcal{E}2_C))$

b) $\mathit{init}(\mathit{Rest}(\phi_1, \mathcal{E}1_C)) \neq \emptyset = \mathit{init}(\mathit{Rest}(\phi_2, \mathcal{E}2_C))$

c) $\mathit{init}(\mathit{Rest}(\phi_1, \mathcal{E}1_C)) = \emptyset \neq \mathit{init}(\mathit{Rest}(\phi_2, \mathcal{E}2_C))$

d) $\mathit{init}(\mathit{Rest}(\phi_1, \mathcal{E}1_C)) = \emptyset = \mathit{init}(\mathit{Rest}(\phi_2, \mathcal{E}2_C))$

Wir betrachten nur den Fall a). Für die restlichen Fälle gilt es analog zu zeigen. Nach Induktionsannahme 4. gilt:

- $\mathcal{E}_i = \mathcal{E}1_i \;\overline{\cup}\; \mathcal{E}2_i$, wobei $\mathit{Rest}(\phi_1, \mathcal{E}1_C) \sqsubseteq \mathcal{E}1_i$ und $\mathcal{E}2_C \sqsubseteq \mathcal{E}2_i$.
- $\mathcal{E}_j = \mathcal{E}1_j \;\overline{\cup}\; \mathcal{E}2_j$, wobei $\mathcal{E}1_C \sqsubseteq \mathcal{E}1_j$ und $\mathit{Rest}(\phi_2, \mathcal{E}2_C) \sqsubseteq \mathcal{E}2_j$.

Dabei gilt $\mathcal{E}1_i, \mathcal{E}2_i, \mathcal{E}1_j, \mathcal{E}2_j \in \mathcal{ES_P}$. Daraus folgt:

- $\exists \xi_1 : (\mathcal{E}1_j \overset{\xi_1}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}1'_j \;\wedge\; l(\phi_1) = l(\xi_1) \;\wedge\; \mathcal{E}1'_j \approx_E \mathcal{E}1'_C \approx_E \mathcal{E}1_i)$
- $\exists \xi_2 : (\mathcal{E}2_i \overset{\xi_2}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}2'_i \;\wedge\; l(\phi_2) = l(\xi_2) \;\wedge\; \mathcal{E}2'_i \approx_E \mathcal{E}2'_C \approx_E \mathcal{E}2_j)$

Somit gilt

$$\mathcal{E}1_i \approx_E \mathit{Rest}(\xi_1, \mathcal{E}1_j) \text{ und } \mathcal{E}2_j \approx_E \mathit{Rest}(\xi_2, \mathcal{E}2_i).$$

Wir zeigen nun, daß gilt:

$$\mathcal{E}1_i \cong \mathit{Rest}(\xi_1, \mathcal{E}1_j) \text{ und } \mathcal{E}2_j \cong \mathit{Rest}(\xi_2, \mathcal{E}2_i)$$

Nach Lemma 4.5 gilt $\mathit{Rest}(\xi_1, \mathcal{E}_j) = \mathit{Rest}(\xi_1, \mathcal{E}1_j) \;\overline{\cup}\; \mathcal{E}2_j$
und $\mathit{Rest}(\xi_2, \mathcal{E}_i) = \mathcal{E}1_i \;\overline{\cup}\; \mathit{Rest}(\xi_2, \mathcal{E}2_i)$. Nach Lemma 4.14 gibt es eine endliche Folge $\phi^1, \ldots, \phi^k$, so daß gilt:

$$\mathcal{E}2_j \overset{\phi^1}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}^1,\; \mathcal{E}^1_R \overset{\phi^2}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}^2,\; \mathcal{E}^2_R \overset{\phi^3}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}^3, \cdots,\; \mathcal{E}^{k-1}_R \overset{\phi^k}{\Leftrightarrow\!\!\rightarrow} \mathcal{E}^k,$$

wobei $\mathcal{E}^1_R = \mathit{Rest}(\phi^1, \mathcal{E}2_j)$, $\mathcal{E}^i_R = \mathit{Rest}(\phi^i, \mathcal{E}^{i-1}_R)$ mit $i = 2, \ldots, k \Leftrightarrow 1$ und

$$\mathit{Rest}(\phi^k, \mathcal{E}^{k-1}_R) = (\emptyset, \emptyset, \emptyset, \emptyset)$$

gilt. Dies bedeutet also

$$\begin{gathered}
\mathit{Rest}(\xi_1, \mathcal{E}_j) \overset{\phi^1}{\Leftrightarrow\!\!\rightarrow} \mathit{Rest}(\phi_1, \mathcal{E}1_j) \;\overline{\cup}\mathcal{E}^1, \\
\mathit{Rest}(\xi_1, \mathcal{E}1_j) \;\overline{\cup}\mathcal{E}^1_R \overset{\phi^2}{\Leftrightarrow\!\!\rightarrow} \mathit{Rest}(\xi_1, \mathcal{E}1_j) \;\overline{\cup}\mathcal{E}^2, \\
\cdots\cdots \\
\mathit{Rest}(\xi_1, \mathcal{E}1_j) \;\overline{\cup}\mathcal{E}^{k-1}_R \overset{\phi^k}{\Leftrightarrow\!\!\rightarrow} \mathit{Rest}(\phi_1, \mathcal{E}1_j) \;\overline{\cup}\mathcal{E}^k \text{ und} \\
\mathit{Rest}(\phi^k, \mathit{Rest}(\xi_1, \mathcal{E}1_j) \;\overline{\cup}\mathcal{E}^{k-1}) = \mathit{Rest}(\xi_1, \mathcal{E}1_j).
\end{gathered}$$

Nach Induktionsannahme 1 gibt es eine Folge $B_j^1, B_j^2, \ldots, B_j^{k+1} \in FB(B)$ mit

$$B_j^1 \overset{l(\phi^1)}{\Longleftrightarrow} B_j^2 \overset{l(\phi^2)}{\Longleftrightarrow} \cdots \overset{l(\phi^k)}{\Longleftrightarrow} B_j^{k+1},$$

so daß $Rest(\xi_1, \mathcal{E}_j) \cong EZ(B_j^1)$, $Rest(\xi_1, \mathcal{E}1_j) \,\overline{\cup} \mathcal{E}_R^t \cong EZ(B_j^{t+1})$, wobei $t = 1 \ldots, k \Leftrightarrow 1$, und $Rest(\xi_1, \mathcal{E}1_j) \cong EZ(B_j^{k+1})$ gilt.

Analog dazu gibt es ein $B_i' \in FB(B)$, so daß $\mathcal{E}1_i \cong EZ(B_i')$ gilt. Somit gilt mit Induktionsannahme 2 $[\![B_j^{k+1}]\!] \approx_E [\![B_j']\!]$, was also $B_j^{k+1} \sim B_j'$ bedeutet. Daraus folgt $EZ(B_j^{k+1}) \cong EZ(B_j')$ und somit $Rest(\xi_1, \mathcal{E}1_j) \cong \mathcal{E}1_i$. Analog dazu gilt natürlich auch $\mathcal{E}2_j \cong Rest(\xi_2, \mathcal{E}2_i)$.

Damit kann nun die gesuchte Ereignisstruktur $\mathcal{E}_P$ wie folgt konstruiert werden:

Nach Lemma 3.8 gilt:

- $\mathcal{E}1_i = \mathcal{E}1_i^1 \,\overline{\cup}\, \cdots \overline{\cup}\, \mathcal{E}1_i^{h1}$ und $\mathcal{E}2_i = \mathcal{E}2_i^1 \,\overline{\cup} \cdots \overline{\cup}\, \mathcal{E}2_i^{h2}$
- $\mathcal{E}1_j = \mathcal{E}1_j^1 \,\overline{\cup} \cdots \overline{\cup}\, \mathcal{E}1_j^{l1}$ und $\mathcal{E}2_j = \mathcal{E}2_j^1 \,\overline{\cup} \cdots \overline{\cup}\, \mathcal{E}2_j^{l2}$

Dabei gilt

- $\mathcal{E}1_i^t = Er(m1_t, \mathcal{E}1_i)$ mit $t = 1, \ldots, h1$ und $Konf(\mathcal{E}1_i) = \{m1_1, \ldots, m1_{h1}\}$.
- $\mathcal{E}2_i^t = Er(m2_t, \mathcal{E}2_i)$ mit $t = 1, \ldots, h2$ und $Konf(\mathcal{E}2_i) = \{m2_1, \ldots, m2_{h2}\}$.
- $\mathcal{E}1_j^t = Er(n1_t, \mathcal{E}1_j)$ mit $t = 1, \ldots, l1$ und $Konf(\mathcal{E}1_j) = \{n1_1, \ldots, n1_{l1}\}$.
- $\mathcal{E}2_j^t = Er(n2_t, \mathcal{E}2_j)$ mit $t = 1, \ldots, l2$ und $Konf(\mathcal{E}2_j) = \{n2_1, \ldots, n2_{l2}\}$.

Sei o.E.d.A. $\xi_1 \in n1_1$ und $\xi_2 \in m2_1$. Dann gilt:

- $Rest(\xi_1, \mathcal{E}1_j^1) \,\overline{\cup}\, \mathcal{E}1_j^2 \,\overline{\cup} \cdots \overline{\cup}\, \mathcal{E}1_j^{l1} \cong \mathcal{E}1_i$
- $Rest(\xi_2, \mathcal{E}2_i^1) \,\overline{\cup}\, \mathcal{E}2_i^2 \,\overline{\cup} \cdots \overline{\cup}\, \mathcal{E}2_i^{h2} \cong \mathcal{E}2_j$

Wir betrachten nun die Bedingung $Bed1((e_i, \xi_2, \mathcal{E}_i), f, (e_j, \xi_1, \mathcal{E}_j))$. Es gilt:

(a) $f(e_i) = l(\xi_1)$ und $f(e_j) = l(\xi_2)$ (trivial)

(b) Es gibt eine Menge $N1 \subset Konf(\mathcal{E}_i)$ mit $m1 \notin N1$ und $\xi_2 \in m1$, so daß gilt:

$$\overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}_i) \cong Rest(\xi_1, Er(m2, \mathcal{E}_j)),$$

wobei $m2 \in Konf(\mathcal{E}_j)$ und $\xi_1 \in m2$.

Wir zeigen die Gültigkeit dieser Aussage wie folgt:

Zunächst gilt $Rest(\xi_1, Er(m2, \mathcal{E}_j)) = Rest(\xi_1, \mathcal{E}1_j^1)$, da $m2 = n1_1 = init(\mathcal{E}1_j^1)$ gilt. Nach Korollar 4.2 gibt es eine Menge $N1 \subseteq Konf(\mathcal{E}1_i)$, so daß gilt:

$$\overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}1_i) = h1(Rest(\xi_1, \mathcal{E}1_j^1))$$

Dabei ist $h1$ der Isomorphismus von $Rest(\xi_1, \mathcal{E}1_j)$ auf $\mathcal{E}1_i$. Anders geschrieben heißt das:

$$\overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}_i) = h1(Rest(\xi_1, \mathcal{E}1_j^1)).$$

Da nach Lemma 4.6 $h1(Rest(\xi_1, \mathcal{E}1_j^1)) \cong Rest(\xi_1, \mathcal{E}1_j^1)$ gilt, gilt

$$\overline{\bigcup_{m \in N1}} Er(m, \mathcal{E}_i) \cong Rest(\xi_1, \mathcal{E}1_j^1).$$

Es gilt offenbar $N1 \subset Konf(\mathcal{E}_i)$ und $m1 \notin N1$ mit $\xi_2 \in m1$.

(c) Analog zu (b) gibt es eine Menge $N2 \subset Konf(\mathcal{E}_j)$ mit $m2 \notin N2$ und $\xi_1 \in m2$, so daß gilt:

$$\overline{\bigcup_{m \in N2}} Er(m, \mathcal{E}_j) \cong Rest(\xi_2, Er(m1, \mathcal{E}_i)),$$

wobei $m1 \in Konf(\mathcal{E}_i)$ und $\xi_2 \in m1$. Es gilt ferner

$$\begin{aligned}\overline{\bigcup_{m \in N2}} Er(m, \mathcal{E}_j) &\cong h2(Rest(\xi_2, \mathcal{E}2_i^1)) \\ &\cong Rest(\xi_2, \mathcal{E}2_i^1)\end{aligned}$$

Dabei ist $h2$ der Isomorphismus von $Rest(\xi_2, \mathcal{E}2_i)$ auf $\mathcal{E}2_j$.

(d) Sei $L1 = Konf(\mathcal{E}_i) \setminus (m1 \cup N1)$ und $L2 = Konf(\mathcal{E}_j) \setminus (m2 \cup N2)$, dann gilt

$$\overline{\bigcup_{m \in L1}} Er(m, \mathcal{E}_i) \cong \overline{\bigcup_{m \in L2}} Er(m, \mathcal{E}_j).$$

Denn es gilt mit Korollar 4.2 und Lemma 4.6:

$$\begin{aligned}\overline{\bigcup_{m \in L1}} Er(m, \mathcal{E}_i) &= h1(\mathcal{E}1_j^2 \,\overline{\cup} \cdots \mathcal{E}1_j^{l1}) \,\overline{\cup}\, \mathcal{E}2_i^2 \,\overline{\cup} \cdots \overline{\cup} \mathcal{E}2_i^{h2} \\ &\cong h2(\mathcal{E}2_i^2 \,\overline{\cup} \cdots \overline{\cup} \mathcal{E}2_i^{h2}) \,\overline{\cup}\, \mathcal{E}1_j^2 \,\overline{\cup} \cdots \mathcal{E}1_j^{l1} \\ &= \overline{\bigcup_{m \in L2}} Er(m, \mathcal{E}_j)\end{aligned}$$

Damit ist gezeigt, daß $Bed1((e_i, \xi_2, \mathcal{E}_i), f, (e_j, \xi_1, \mathcal{E}_j))$ wahr ist.

Es ist leicht zu prüfen, daß gilt:

$$\begin{aligned}Kons(\xi_2, \mathcal{E}_i, \xi_1, \mathcal{E}_j) &= 1(h1(\mathcal{E}1_j^2 \,\overline{\cup} \cdots \overline{\cup}\, \mathcal{E}1_j^{l1}) \,\overline{\cup}\, \mathcal{E}2_i) \,\overline{\cup}\, 2(\mathcal{E}1_j^1) \\ &= 1(h1(\mathcal{E}1_j^2 \,\overline{\cup} \cdots \overline{\cup}\, \mathcal{E}1_j^{l1})) \,\overline{\cup}\, 2(\mathcal{E}1_j^1) \,\overline{\cup}\, 1(\mathcal{E}2_i) \\ &= 1(h1(\mathcal{E}1_j^2)) \,\overline{\cup} \cdots \overline{\cup}\, 1(h1(\mathcal{E}1_j^{l1})) \,\overline{\cup}\, 2(\mathcal{E}1_j^1) \,\overline{\cup}\, 1(\mathcal{E}2_i) \\ &\cong 1(\mathcal{E}1_j) \,\overline{\cup}\, 2(\mathcal{E}2_i)\end{aligned}$$

Sei also $\mathcal{E}_P = Kons(\xi_2, \mathcal{E}_i, \xi_1, \mathcal{E}_j)$, dann gilt $\mathcal{E}_P = \mathcal{E}1_P \,\overline{\cup}\, \mathcal{E}2_P$, wobei

$$\mathcal{E}1_P = 1(h1(\mathcal{E}1_j^2)) \,\overline{\cup} \cdots \overline{\cup}\, 1(h1(\mathcal{E}1_j^{l1})) \,\overline{\cup}\, 2(\mathcal{E}1_j^1) \text{ und } \mathcal{E}2_P = 1(\mathcal{E}2_i)$$

gilt. Aus Korollar 4.2 folgt $h1(\mathcal{E}1_j^2), \ldots, h1(\mathcal{E}1_j^{l1}) \in \mathcal{ES}_\mathcal{P}$. Somit gilt mit Lemma 4.4 $\mathcal{E}1_P, \mathcal{E}2_P \in \mathcal{ES}_\mathcal{P}$.

Da $\mathcal{E}1_P \cong 1(\mathcal{E}1_j)$ und $\mathcal{E}2_P \cong 2(\mathcal{E}2_i)$ gilt, gilt mit Lemma 4.7 $Zer(\mathcal{E}1_C) \leq Zer(\mathcal{E}1_P)$ und $Zer(\mathcal{E}2_C) \leq Zer(\mathcal{E}2_P)$.

Zum Schluß bleibt es noch zu zeigen, daß $Bed2((e_i, \xi_2, \mathcal{E}_i), f, (e_j, \xi_1, \mathcal{E}_j), M)$ wahr ist. Es ist also zu zeigen, daß folgendes gilt:

$$\forall e \in init(\mathcal{E}_P) : \exists h : (l(e) = f(e_h) \wedge Rest(e, \mathcal{E}_P) \cong \mathcal{E}_h).$$

Sei o.E.d.A. $e \in init(1(h1(\mathcal{E}1_j^2)))$.

Nach Induktionsannahme 5 gibt es einen Prozeß $P \in Er(B_j)$, so daß $\mathcal{E}1_j^2 \cong EZ(P)$ und somit $1(h1(\mathcal{E}1_j^2)) \cong EZ(P)$ gilt.

Nach Induktionsannahme 1 gilt $Rest(e', \mathcal{E}1_j^2) \cong EZ(P1)$ mit $e' \in init(\mathcal{E}1_j^2)$, wobei $P \overset{l(e')}{\Leftrightarrow} P1$ gilt. Da $e = 1h1(e')$ gilt, bedeutet dies mit Lemma 4.18

$$Rest(e, 1(h1(\mathcal{E}1_j^2))) \cong EZ(P1).$$

Im folgenden betrachten wir nur $EZ(P1) \neq (\emptyset, \emptyset, \emptyset, \emptyset)$. Für $EZ(P1) = (\emptyset, \emptyset, \emptyset, \emptyset)$ gilt es analog zu zeigen.

Nach Lemma 3.8 gilt $EZ(P1) = \mathcal{E}_{P1}^1 \overline{\cup} \cdots \overline{\cup}\, \mathcal{E}_{P1}^g$, wobei

$$\mathcal{E}_{P1}^t = Er(m_t, EZ(P1))$$

mit $t = 1, \ldots, g$, $m_t \in Konf(EZ(P1))$ und $Konf(EZ(P1)) = \{m_1, \ldots, m_g\}$ gilt. Nach Induktionsannahme 5. ist $\mathcal{E}_{P1}^t$ mit $t = 1, \ldots, g$ nicht zerlegbar.

Sei $h3$ der Isomorphismus von $EZ(P1)$ auf $Rest(e, 1(h1(\mathcal{E}1_j^2)))$, dann gilt

$$Rest(e, 1(h1(\mathcal{E}1_j^2))) = h3(\mathcal{E}_{P1}^1) \overline{\cup} \cdots \overline{\cup}\, h3(\mathcal{E}_{P1}^g).$$

($h3(\mathcal{E}_{P1}^t)$ mit $t = 1, \ldots, g$ ist nicht zerlegbar.) Es gilt also

$$\begin{aligned} Rest(e, \mathcal{E}_P) &= h3(\mathcal{E}_{P1}^1) \overline{\cup} \cdots \overline{\cup}\, h3(\mathcal{E}_{P1}^g) \\ &\quad \overline{\cup}\, 1(h1(\mathcal{E}1_j^2)) \cdots \overline{\cup}\, 1(h1(\mathcal{E}1_j^{l1})) \overline{\cup}\, 2(\mathcal{E}1_j^1) \overline{\cup}\, 1(\mathcal{E}2_i) \\ &= h3(\mathcal{E}_{P1}^1) \overline{\cup} \cdots \overline{\cup}\, h3(\mathcal{E}_{P1}^g) \\ &\quad \overline{\cup}\, 1(h1(\mathcal{E}1_j^2)) \cdots \overline{\cup}\, 1(h1(\mathcal{E}1_j^{l1})) \overline{\cup}\, 2(\mathcal{E}1_j^1) \\ &\quad \overline{\cup}\, 1(\mathcal{E}2_i^1) \overline{\cup} \cdots \overline{\cup}\, 1(\mathcal{E}2_i^{h2}) \end{aligned}$$

$1(h1(\mathcal{E}1_j^t))$ mit $t = l, \ldots, l1$, $2(\mathcal{E}1_j^1)$ und $1(\mathcal{E}2_i^t)$ mit $t = l, \ldots, h2$ sind wegen Induktionsannahme 5. nicht zerlegbar.

Aus Induktionsannahme 5 folgt:

- $\forall i = 1, \ldots, g : \exists Q \in Er(P1) : EZ(Q) \cong h3(\mathcal{E}_{P1}^i)$
- $\forall i = 2, \ldots, l1 : \exists Q \in Er(B_j) : EZ(Q) \cong 1(h1(\mathcal{E}1_j^i))$ und
  $\exists Q \in Er(B_j) : EZ(Q) \cong 2(\mathcal{E}1_j^1)$
- $\forall i = 1, \ldots, h2 : \exists Q \in Er(B_i) : EZ(Q) \cong 1(\mathcal{E}2_i^i)$

Da offenbar $\mathcal{E}_P \approx_E \mathcal{E}_C$ und nach Voraussetzung $\mathcal{E}_C \approx_E \mathcal{E}$ gilt, gibt es ein h, so daß $l(e) = f(e_h)$ und $Rest(e, \mathcal{E}_P) \approx_E \mathcal{E}_h$ gilt.

Nach Lemma 3.8 gilt $\mathcal{E}_h = \mathcal{E}_h^1 \,\overline{\cup}\, \cdots \overline{\cup} \mathcal{E}_h^t$, wobei $\mathcal{E}_h^k = Er(m_t, \mathcal{E}_h)$ mit $k = 1, \ldots, t$, $m_t \in Konf(\mathcal{E}_h)$ und

$$Konf(\mathcal{E}_h) = \{m_1, \ldots, m_k\}$$

$\mathcal{E}_h^k$ mit $k = 1, \ldots, t$ ist nach Induktionsannahme 5 nicht zerlegbar.

Außerdem gilt $\forall t = 1, \ldots, k : \exists Q \in Er(B_h) : EZ(Q) \cong \mathcal{E}_h^t$. Mit Lemma 4.16 und Lemma 4.15 läßt sich dann folgern, daß $Rest(e, \mathcal{E}_P) \cong \mathcal{E}_h$ gilt (analog dazu siehe Beweis zu 3(c)).

Damit ist gezeigt, daß $\mathcal{E}_P \in Bilde(f, M)$ gilt. Da $\mathcal{E}_P \approx_E \mathcal{E}_B$ wegen der Aussage 2 gilt, läßt sich leicht folgern, daß

$$\neg(\exists \mathcal{E}'_P \in Bilde(f, M) : \mathcal{E}_P \propto \mathcal{E}'_P)$$

gilt. D.h. es gilt $\exists \mathcal{E}'_P \in Opt(Bilde(f, M)) : \mathcal{E}_P \cong \mathcal{E}'_P$. Leicht zu folgern ist es auch, daß

$$\forall \mathcal{E}'_P \in Opt(Bilde(f, M)) : \mathcal{E}'_P \approx_E \mathcal{E}_P$$

gilt. Nach Aussage 3 gilt somit

$$\forall \mathcal{E}'_P \in Opt(Bilde(f, M)) : \mathcal{E}'_P \cong \mathcal{E}_P,$$

was $\mathcal{E} \cong \mathcal{E}_P$ bedeutet. Mit Korollar 4.2 und Lemma 4.7 folgt, daß die Aussage 4 auch für $\mathcal{E}$ gilt.

- Zu 5: Zu „$Er(m, \mathcal{E})$ ist nicht zerlegbar" unterscheiden wir zwei Fälle:

  1. $|Opt(Bilde(f, M))| \geq 2$
     In diesem Fall gibt es $\mathcal{E}^1, \mathcal{E}^2, \ldots, \mathcal{E}^t \in Opt(Bilde(f, M))$, wobei $t \geq 2$, so daß gilt:
     $$\mathcal{E} = 1(\mathcal{E}^1) \,\overline{[]}\, 2(\mathcal{E}^2) \cdots \overline{[]}\, t(\mathcal{E}^t)$$
     In diesem Fall gilt $Konf(\mathcal{E}) = \{m\}$ und $m = init(\mathcal{E})$, d.h. $Er(m, \mathcal{E}) = \mathcal{E}$. $Er(m, \mathcal{E})$ ist wegen der Aussage 4. nicht zerlegbar.
  2. $|Opt(Bilde(f, M))| = 1$
     Sei $\mathcal{E}_P \in Opt(Bilde(f, M))$, dann gilt in diesem Fall $\mathcal{E} = \mathcal{E}_P$. Es sind zwei weitere Fälle zu unterscheiden:
     (a) Es gibt ein $j \in \{1, \ldots, n\}$, so daß $\overline{f(e_j)_{e_j}; \mathcal{E}_j}$ gilt. D.h. es gilt $Konf(\mathcal{E}) = \{\{e_i\}\}$ und $Er(\{e_i\}, \mathcal{E}) = \mathcal{E}_P$. Wegen der Aussage 4. ist $Er(m, \mathcal{E})$ nicht zerlegbar.
     (b) Es gibt $i1, i2 \in \{1, \ldots, n\}$, so daß $\mathcal{E}_P = Kons(e, \mathcal{E}_{i1}, e', \mathcal{E}_{i2})$ gilt. Nach Definition 4.15 gilt

$$\begin{aligned} \mathcal{E}_P &= 1(\bigcup_{k=1}^{K} Er(m_k, \mathcal{E}_{i1})) \,\overline{\cup}\, 2(Er(m2, \mathcal{E}_{i2})) \\ &= 1(Er(m_1, \mathcal{E}_{i1})) \,\overline{\cup}\, 1(Er(m_2, \mathcal{E}_{i1})) \cdots \overline{\cup}\, 1(Er(m_K, \mathcal{E}_{i1})) \\ &\quad \overline{\cup}\, 2(Er(m2, \mathcal{E}_{i2})). \end{aligned}$$

Dabei gilt $m_k \in Konf(\mathcal{E}_{i1})$ mit $k = 1, \ldots, K$ und $m2 \in Konf(\mathcal{E}_{i2})$. Sei $m \in Konf(\mathcal{E}_P)$, dann gibt es offenbar ein $m' \in \{m_1, \ldots, m_k\}$, so daß

$$Er(m, \mathcal{E}_P) = 1(Er(m', \mathcal{E}_{i1})) \text{ oder } Er(m, \mathcal{E}_P) = 2(Er(m2, \mathcal{E}_{i2}))$$

gilt. Da aber nach Induktionsannahme 5. $Er(m_k, \mathcal{E}_{i1})$ als auch $Er(m2, \mathcal{E}_{i2})$ nicht zerlegbar ist, ist $Er(m, \mathcal{E}_P)$ natürlich nicht zerlegbar.

Zu $\exists B' \in Er(B) : EZ(B') \cong Er(m, \mathcal{E})$:

Für $|Konf(\mathcal{E})| = 1$ gilt offenbar die Behauptung. Sei also $|Konf(\mathcal{E})| > 1$. Wir betrachten $m_k \in Konf(\mathcal{E})$. Nach Lemma 3.8 gilt

$$\mathcal{E} = Er(m_k, \mathcal{E}) \ \overline{\cup} \ (\overline{\bigcup}_{m \in (Konf(\mathcal{E}) \setminus m_k)} Er(m, \mathcal{E})).$$

Sei

$$\mathcal{E}' = \overline{\bigcup}_{m \in (Konf(\mathcal{E}) \setminus m_k)} Er(m, \mathcal{E}),$$

dann gilt wegen Lemmas 3.8 $\mathcal{E}' \in \mathcal{ES}_{\mathcal{P}}$.
Nach Lemma 4.14 gibt es eine endliche Folge von Ereignissen $\phi^1, \ldots, \phi^t$, so daß gilt:

$$\mathcal{E}' \overset{\phi^1}{\Leftrightarrow} \mathcal{E}^1,\ \mathcal{E}_R^1 \overset{\phi^2}{\Leftrightarrow} \mathcal{E}^2,\ \mathcal{E}_R^2 \overset{\phi^3}{\Leftrightarrow} \mathcal{E}^3, \cdots,\ \mathcal{E}_R^{t-1} \overset{\phi^t}{\Leftrightarrow} \mathcal{E}^t,$$

wobei $\mathcal{E}_R^1 = Rest(\phi^1, \mathcal{E}')$, $\mathcal{E}_R^i = Rest(\phi^i, \mathcal{E}^{i-1})$ mit $i = 2, \ldots, t \Leftrightarrow 1$ und $Rest(\phi^t, \mathcal{E}_R^{t-1}) = (\emptyset, \emptyset, \emptyset, \emptyset)$ gilt. Dies bedeutet also

$$\begin{aligned}
\mathcal{E} &\overset{\phi^1}{\Leftrightarrow} Er(m_k, \mathcal{E}) \ \overline{\cup} \mathcal{E}^1, \\
Er(m_k, \mathcal{E}) \ \overline{\cup} \mathcal{E}_R^1 &\overset{\phi^2}{\Leftrightarrow} Er(m_k, \mathcal{E}) \ \overline{\cup} \mathcal{E}^2, \\
&\cdots\cdots \\
Er(m_k, \mathcal{E}) \ \overline{\cup} \mathcal{E}_R^{t-1} &\overset{\phi^t}{\Leftrightarrow} Er(m_k, \mathcal{E}) \ \overline{\cup} \mathcal{E}^t \text{ und} \\
Rest(\phi^t, Er(m_k, \mathcal{E}) \ \overline{\cup} \mathcal{E}_R^{t-1}) &= Er(m_k, \mathcal{E}).
\end{aligned}$$

Nach Aussage 1 gilt $Er(m_k, \mathcal{E}) \overline{\cup} \mathcal{E}_R^1 \cong EZ(B^1)$, wobei $B \overset{l(\phi^1)}{\Leftrightarrow} B^1$ gilt.
Nach Induktionsannahme 1 gibt es dann eine Folge $B^2, \ldots, B^t \in FB(B)$ mit

$$B^1 \overset{l(\phi^2)}{\Leftrightarrow} B^2 \overset{l(\phi^3)}{\Leftrightarrow} \cdots \overset{l(\phi^t)}{\Leftrightarrow} B^t,$$

so daß $Er(m_k, \mathcal{E}) \overline{\cup} \mathcal{E}_R^i \cong EZ(B^i)$ mit $i = 2, \ldots, t \Leftrightarrow 1$ und $Er(m_k, \mathcal{E}) \cong EZ(B^t)$ gilt.

- <u>Zu 6:</u> Es sind zwei Fälle zu unterscheiden.

  1. $|Konf(\mathcal{E}')| > 1$
     Sei $Konf(\mathcal{E}') = \{m_1, \ldots, m_k\}$. Dann läßt sich $\mathcal{E}'$ wie folgt darstellen:

$$\mathcal{E}' = Er(m_1, \mathcal{E}') \ \overline{\cup} \ \overline{\bigcup_{i=2}^{k}} Er(m, \mathcal{E}')$$

Sei $\mathcal{E}1' = Er(m_1, \mathcal{E}')$ und $\mathcal{E}2' = \overline{\bigcup_{i=2}^{k}} Er(m, \mathcal{E}')$, dann gilt wegen Lemma 3.8 $\mathcal{E}1', \mathcal{E}1'' \in \mathcal{ES}_{\mathcal{P}}$. Somit gilt nach Aussage 4.

$$\exists \mathcal{E}1, \mathcal{E}2 \in \mathcal{ES}_{\mathcal{P}} : (\mathcal{E} = \mathcal{E}1 \mathbin{\overline{\cup}} \mathcal{E}2 \land \mathcal{E}1' \sqsubseteq \mathcal{E}1 \land \mathcal{E}2' \sqsubseteq \mathcal{E}2)$$

Nach Lemma 4.8 gilt $Zer(\mathcal{E}') = Zer(\mathcal{E}1') + Zer(\mathcal{E}2')$ und $Zer(\mathcal{E}) = Zer(\mathcal{E}1) + Zer(\mathcal{E}2)$, was $Zer(\mathcal{E}') \leq Zer(\mathcal{E})$ bedeutet. Daraus folgt $\mathcal{E}' \sqsubseteq \mathcal{E}$.

2. $|Konf(\mathcal{E}')| = 1$
   In diesem Fall gilt $Zer(\mathcal{E}') = 1$ und somit $Zer(\mathcal{E}') \leq Zer(\mathcal{E})$, d.h. $\mathcal{E}' \sqsubseteq \mathcal{E}$. □

Korollar 4.4 *Sei $B \in \mathcal{BL}$ eine Action-Prefix-Form, dann gilt*

1. $PZ(B) \in \mathcal{BL}_{\mathcal{P}}$.
2. $B \sim PZ(B)$. □

Korollar 4.5 *Sei $B \in \mathcal{BL}$ und $\mathcal{E} = [\![PZ(B)]\!]$, dann gilt:*

1. $\forall B' \in \mathcal{BL}_P : (B \sim B' \Longrightarrow B' \sqsubseteq PZ(B))$
2. *Sei* $\mathcal{E}' \in REr(\mathcal{E})$, *dann gilt* $\forall \mathcal{E}'' \in \mathcal{ES}_{\mathcal{P}} : (\mathcal{E}'' \approx_E \mathcal{E}' \Longrightarrow \mathcal{E}'' \sqsubseteq \mathcal{E}')$. □

Somit ist gezeigt, daß $PZ(B)$ der Prozeß ist, der die Bedingung 1 und 2 des Zerlegungsproblems erfüllt.

5 Ein einfaches Anwendungsbeispiel

Dieser Abschnitt stellt ein einfaches Anwendungsbeispiel vor, das die Anwendbarkeit der Methode *maximale Prozeßzerlegung* demonstriert. Dazu wird zunächst die Sprache Basic LOTOS $\mathcal{BL}_{\mathcal{L}}$, also eine Erweiterung von $\mathcal{BL}$, aus [Do95] zitiert, die es erlaubt, das rekursive Verhalten von Prozessen, das Verhalten zweier Prozesse B_1 und B_2, wobei B_1 von B_2 unterbrochen wird, und die sequentielle Ausführung von zwei Prozessen zu beschreiben.

Bei dem Anwendungsbeispiel handelt es sich um die Spezifikation eines Waren-Verkaufsautomaten, der nach dem Münzeneinwurf und nach dem Betätigen einer Taste die Ware der ausgewählten Sorte ausgibt. Anfangs wird die Spezifikation so angegeben, daß nur die temporale Ordnung, d.h. die Ordnung, nach der die Ereignisse zeitlich auftreten, beschrieben wird. Kausale Unabhängigkeit von Ereignissen wird dabei nicht berücksichtigt .

Wendet man *maximale Prozeßzerlegung* auf die vorgegebene Spezifikation an, so erhält man eine äquivalente Spezifikation, bei der die Teilkomponenten, die unabhängig voneinander ablaufen können, explizit angegeben werden.

Definition 5.1 (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 $\mathcal{BL}_\mathcal{L}$, durch folgende Grammatik

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

definiert. □

Im folgenden wird nur die Bedeutung der neu hinzukommenden Operatoren erläutert. Die Semantik der restlichen Operatoren ist dem Abschnitt 2.1 zu entnehmen.

- *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.

- *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 **hide**-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.

- *process instantion*: P

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

Definition 5.2 (operationelle Semantik) *Sei $B \in \mathcal{BL}_\mathcal{L}$, $Act := \mathcal{G} \cup \{\mathbf{i}, \delta\}$ 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. $\mathbf{exit} \stackrel{\delta}{\Leftrightarrow\!\!\rightarrow} \mathbf{stop}$

2. $g; B \stackrel{g}{\Leftrightarrow\!\!\rightarrow} B$

3. $\dfrac{B_1 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 \,[]\, B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1'}$ 4. $\dfrac{B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_2'}{B_1 \,[]\, B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_2'}$

5. $\dfrac{B_1 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \neq \delta}{B_1 \gg B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1' \gg B_2}$ 6. $\dfrac{B_1 \stackrel{\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 \gg B_2 \stackrel{\mathbf{i}}{\Leftrightarrow\!\!\rightarrow} B_2}$

7. $\dfrac{B_1 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \neq \delta}{B_1 \,[>\, B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1' \,[>\, B_2}$ 8. $\dfrac{B_1 \stackrel{\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}{B_1 \,[>\, B_2 \stackrel{\delta}{\Leftrightarrow\!\!\rightarrow} B_1'}$ 9. $\dfrac{B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_2'}{B_1 \,[>\, B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_2'}$

10. $\dfrac{B \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B' \wedge a \in \{g_1, \ldots, g_n\}}{\mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B \stackrel{\mathbf{i}}{\Leftrightarrow\!\!\rightarrow} \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B'}$

11. $\dfrac{B \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B' \wedge a \notin \{g_1, \ldots, g_n\}}{\mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B \stackrel{a}{\Leftrightarrow\!\!\rightarrow} \mathbf{hide}\ g_1, \ldots, g_n\ \mathbf{in}\ B'}$

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

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

14. $\dfrac{B_1 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge a \notin \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1' \,|[g_1, \ldots, g_n]|\, B_2}$

15. $\dfrac{B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_2' \wedge a \notin \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1 \,|[g_1, \ldots, g_n]|\, B_2'}$

16. $\dfrac{B_1 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1' \wedge B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_2' \wedge a \in \{g_1, \ldots, g_n, \delta\}}{B_1 \,|[g_1, \ldots, g_n]|\, B_2 \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B_1' \,|[g_1, \ldots, g_n]|\, B_2'}$

17. $\dfrac{P := B \wedge B \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B'}{P \stackrel{a}{\Leftrightarrow\!\!\rightarrow} B'}$ $\square$

Mit $\mathcal{BL}_{\mathcal{L}}$ sind wir nun in der Lage, den Waren-Verkaufsautomaten zu spezifizieren, dessen Funktionsweise informell wie folgt beschrieben wird:

Der Verkaufsautomat, kurz VA, verfügt über die Tasten 1 bis n und den Münzeneingang, über die VA mit der Umgebung kommuniziert (siehe Abbildung 3). Zur Vereinfachung betrachten wir im folgenden $n = 2$. Für $n > 2$ erfolgt die Vorgehensweise analog.

Wird nach dem Münzeneinwurf eine von den Tasten 1 und 2 betätigt, so wird die ausgewählte Ware ausgegeben. Der Preis einer Ware jeder Sorte beträgt 1 DM. Die Kapazität, d.h. die maximale Anzahl der Waren im Automaten, jeder Sorte, bezeichnet als

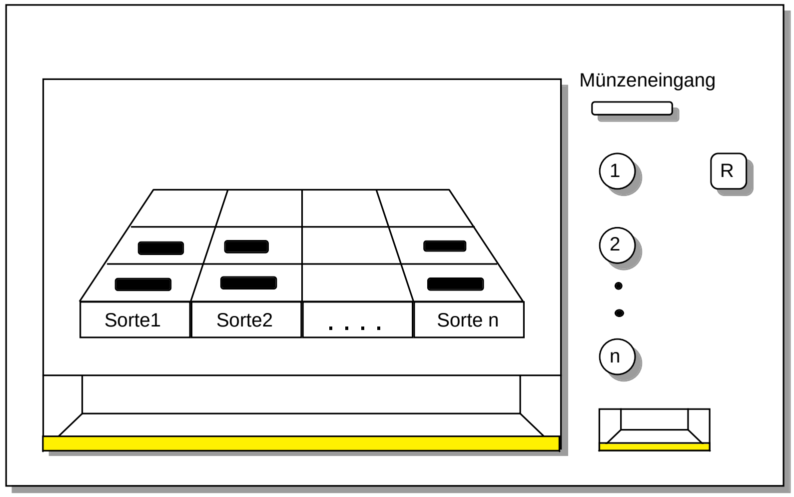


Abbildung 3: Waren-Verkaufsautomat

$K(i)$, wobei $i = 1, 2$ die Nummer der entsprechenden Sorte angibt, beträgt zur Vereinfachung 2, d.h. $K(1) = 2 = K(2)$. $K(i)$ kann natürlich im allgemeinen auf eine beliebige natürliche Zahl k erweitert werden. Allerdings hängt die Spezifikation, wie im folgenden zu sehen wird, von $K(i)$ ab.

Beim Münzeneinwurf sind nur Mark-Stücke erlaubt. Es dürfen höchstens vier Mark-Stücke eingeworfen werden. Genauer gesagt entspricht die Höchstanzahl der Mark-Stücke, die eingeworfen werden dürfen, der Anzahl der Waren, die sich noch im Automaten befinden. Das Betätigen der Taste R bewirkt die Ausgabe bzw. Rückgabe des Restbetrages.

Wir nehmen an, daß der Automat intern aus folgenden zwei Komponenten besteht:

1. Zähler

   Der Zähler merkt sich den Restgeldbetrag, der, falls die R-Taste betätigt wird, ausgezahlt wird.

2. Kontroller

   Der Kontroller hat die Aufgabe, die von der Umgebung kommenden Signale entgegenzunehmen und diese zu bearbeiten.

Die Abbildung 4 zeigt die Kopplung der Komponenten 1 und 2. Der Verkaufsautomat VA wird wie folgt spezifiziert (Vereinbarung: Prefixoperator „;“ ist stärker gebunden als der Choice-Operator „[]“.):

- $VA = \textbf{hide}\ zahlen\ \textbf{in}\ (Kontroller|[1, 2, 1DM, zahlen]|Zähler)$

- $Zähler = 1DM; Zähler_1$

  - $Zähler_1 = 1; Zähler\ []\ 2; Zähler\ []\ zahlen; Zähler\ []\ 1DM; Zähler_2$
  - $Zähler_i = 1; Zähler_{i-1}\ []\ 2; Zähler_{i-1}\ []\ zahlen; Zähler_{i-1}\ []\ 1DM; Zähler_{i+1}$ mit $i = 2, 3$.
  - $Zähler_4 = 1; Zähler_3\ []\ 2; Zähler_3\ []\ zahlen; Zähler_3$

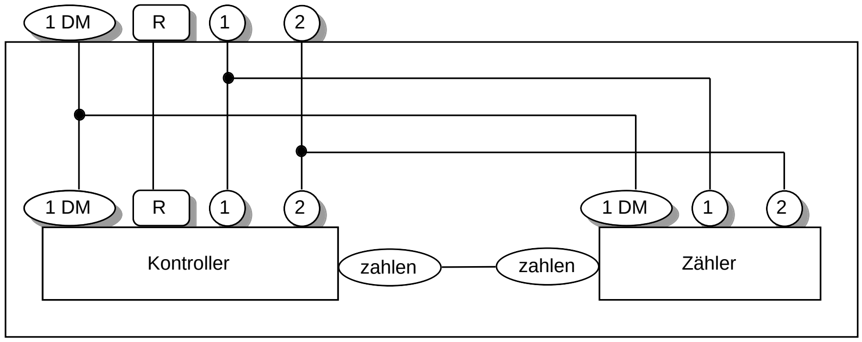


Abbildung 4: Interne Struktur des Waren-Verkaufsautomaten

- $Kontroller = Modul1|[1, 2, 1DM]|Modul2$, wobei
  - $Modul1 = 1DM; (1; B_1 \ [] \ 2; B_1[1/2, 2/1] \ [] \ 1DM; B_2)$
    - $B_1 = 1DM; B_3$
      $B_3 = 1DM; B_4 \ [] \ 1; 1DM; B_5 \ [] \ 2; 1DM; B_6$
      $B_4 = 1DM; (1; 2; 2 \ [] \ 2; (1; 2 \ [] \ 2; 1)) \ [] \ 1; B_5 \ [] \ 2; B_6$
      $B_5 = 1DM; 2; 2 \ [] \ 2; 1DM; 2$
      $B_6 = 1DM; (1; 2 \ [] \ 2; 1) \ [] \ 1; 1DM; 2 \ [] \ 2; 1DM; 1$
    - $B_2 = 1; B_3 \ [] \ 2; B_3[1/2, 2/1] \ [] \ 1DM; (1; B_4 [] 2; B_4[1/2, 2/1] [] 1DM; B_7)$
      $B_7 = 1; (1; 2; 2 \ [] \ 2; (1; 2 \ [] \ 2; 1)) \ [] \ 2; (1; (1; 2 \ [] \ 2; 1) \ [] \ 2; 1; 1)$
  - $Modul2 = Erlaubnis\,[>(R; GA \gg Modul2)$
    - $Erlaubnis = (1; Erlaubnis + 2; Erlaubnis + 1DM; Erlaubnis)$
    - $GA = zahlen; 1DM$-ausgabe; $GA + timeout$; **exit**

Der Kontroller besteht aus zwei Prozessen $Modul1$ und $Modul2$. $Modul1$ ist der Prozeß, der für die Geldannahme und Warenausgabe zuständig ist. Anfangs darf der Käufer ein Mark-Stück einwerfen. Danach stehen ihm zwei Wahlmöglichkeiten zur Verfügung:

1. Der Käufer hat die Wahl, entweder eine von den Tasten 1 und 2 zu betätigen oder ein weiteres Mark-Stück einzuzahlen. Wird eine von den Tasten 1 und 2 betätigt, so wird die ausgewählte Ware ausgegeben. $Modul1$ geht dann entweder in den Zustand B_1 oder in $B_1[1/2, 2, 1]$ über. Ansonsten erreicht $Modul1$ den Zustand B_2. Dieser Ablauf läßt sich solange fortsetzen, bis alle Waren verkauft sind, d.h. der Automat in einem leeren Zustand endet. Dabei wird jede Betätigung der Taste 1 oder 2 und jedes Einwerfen der Geldmünze dem Prozeß $Modul2$, genauer dem Prozeß $Erlaubnis$, mitgeteilt.

2. Der Käufer kann die Taste R betätigen, um den Geldbetrag, den er eingezahlt hat, zurückzufordern. Nach dem Betätigen der Taste R wird der Prozeß $Erlaubnis$ unterbrochen, was somit die Kommunikation des Prozesses $Modul1$ mit der Umgebung unterbricht. Erst nachdem eine Geldausgabe stattgefunden hat, ist $Modul1$ wieder bereit, die von der Umgebung kommenden Signale entgegenzunehmen und sie zu bearbeiten.

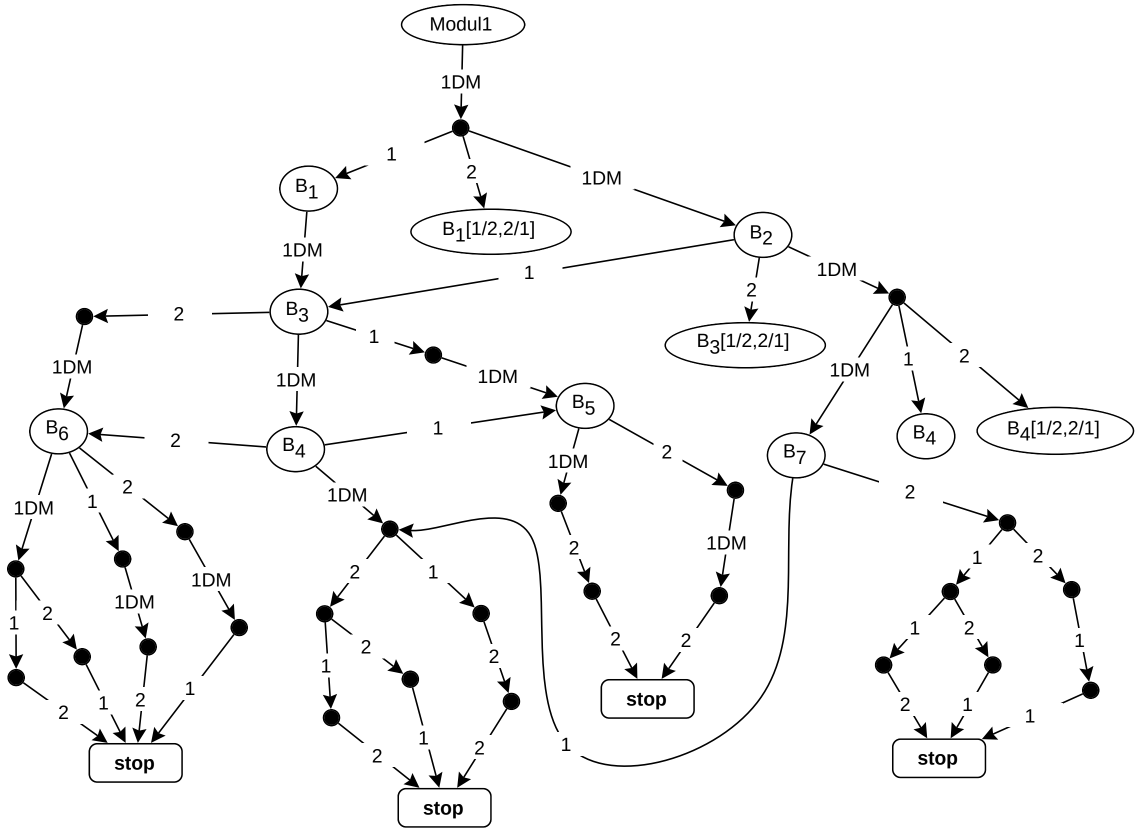


Abbildung 5: Beobachtbares Verhalten von B

Das Transitionssystem $\mathcal{OS}(Modul1)$, das also das beobachtbare Verhalten von $Modul1$ wiedergibt, ist der Abbildung 5 zu entnehmen.

$Modul2$ ist der Prozeß, der die Erlaubnis an $Modul1$ für die Geldannahme und Warenausgabe vergibt. Wird die Taste R von der Umgebung betätigt, so wird die Erlaubnis-Zuteilung unterbrochen. $Modul2$ geht dann in den Prozeß GA über.

GA ist der Prozeß, der für die Ausgabe des Restgeldbetrags zuständig ist. Der Restgeldbetrag wird durch die Kommunikation mit *Zähler* bestimmt. *timeout* repräsentiert hier eine Aktion, die nach einer bestimmten Zeitdauer stattfindet. Diese muß größer sein als die Ausführungszeit der Aktion *zahlen*. *zahlen* ist eine nicht beobachtbare Aktion, die den Prozeß *Zähler* in den entsprechenden Zustand versetzt, falls ein Mark-Stück ausgegeben wird. Falls also der Restgeldbetrag bereits ausgezahlt ist, so findet *timeout* statt, was somit das Terminieren des Prozesses GA bewirkt. Der Prozeß $Modul2$ geht dann in seinen Anfangszustand zurück.

Da in $Modul1$ keine Rekursion vorkommt, läßt sich die Methode *maximale Prozeßzerlegung* auf $Modul1$ anwenden. Es gilt:

- $$\begin{aligned} PZ(Modul1) \;=\; & 1DM;(1;1DM;PZ(B_3)\ \ []\ \ 2;1DM;PZ(B_3)[1/2,2/1] \\ & []\ \ 1DM;PZ(B_2)) \end{aligned}$$

- $$\begin{aligned} PZ(B_2) \;=\; & 1;PZ(B_3)\ \ []\ \ 2;PZ(B_3)[1/2,2/1]\ \ []\ \ 1DM;(1;PZ(B_4) \\ & []\ \ 2;PZ(B_4)[1/2,2/1]\ \ []\ \ 1DM;(1\,|||\,1\,|||\,2\,|||\,2)) \end{aligned}$$

- $PZ(B_3) = 1DM; PZ(B_4)\ []\ 1; 1DM; PZ(B_5)\ []\ 2; 1DM; PZ(B_6)$
- $PZ(B_4) = 1DM; (1\,|||\,2\,|||\,2)\ []\ 1; PZ(B_5)\ []\ 2; PZ(B_6)$
- $PZ(B_5) = 2\,|||(1DM; 2)$
- $PZ(B_6) = (1\,|||(1DM; 2))\ []\ (2\,|||(1DM; 1))$

$Modul1$ kann also durch $PZ(Modul1)$ ersetzt werden, um dadurch eine Beschleunigung der Prozeß-Ausführungsgeschwindigkeit zu erreichen.

6 Zusammenfassung

Dieser Bericht stellt eine Methode zur maximalen Zerlegung von parallelen Prozessen in Basic LOTOS ohne Rekursion vor. Damit kann aus einem vorgegebenen Prozeß B ein äquivalenter Prozeß B' bestimmt werden, deren Anzahl der parallel unabhängig voneinander ablaufenden Teilprozesse (genannt Zerlegungsgrad) maximal ist. Die Methode basiert auf Ereignisstrukturen, die in [Lan92] als semantisches Modell für Basic LOTOS definiert wurden. Von „inverse expansion“ in [PHQ+92] unterscheidet sich die Methode in folgender Hinsicht:

- Die Anzahl der parallel unabhängig voneinander ablaufenden Teilprozesse ist nicht auf zwei beschränkt.
- Die Aktionmengen der Teilprozesse sind i.a. nicht disjunkt und werden als Vorgabe nicht benötigt. Dies hat „inverse expansion“ gegenüber den entscheidenen Vorteil, daß auch nichtdeterministische Prozesse sich hiermit zerlegen lassen, die jedoch für „inverse expansion“ unlösbar sind. Bei Prozessen, deren Verhalten vom Zufall abhängt, kann die Methode daher angewendet werden.
- Auch die Folgezustände besitzen einen maximalen Zerlegungsgrad.

Im Hinblick auf die praktische Anwendbarkeit kann aus den obengenannten Unterschieden angenommen werden, daß die in diesem Bericht vorgestellte Methode genau so mächtig ist wie „inverse expansion“. D.h. praktische Probleme, die sich mit „inverse expansion“ lösen lassen, lassen sich auch hier lösen.

Da die Methode nur auf die Klasse der nichtrekursiven Prozesse eingeschränkt ist, hat sie einen Nachteil, daß sie in der Praxis z.B. beim Entwurf von Kommunikationssystemen nicht oft eingesetzt werden kann. Offen ist daher die Frage, wie diese Methode sich auf die Klasse der rekursiven Prozesse erweitern läßt.

Danksagung

Zum Schluß möchte ich mich bei Torsten Vogt für die deutsche Lesekorrektur bedanken.

Literatur

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

[AKH92] S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. In *Acta Informatica 29*, pages 737–760. Springer-Verlag, 1992.

[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.

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

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

[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.

[Do95] H. T. Do. Parallelität in Basic LOTOS. Manuskripte der Reihe Informatik Nr. 35/95, Fakultät für Mathematik und Informatik der Universität Mannheim, November 1995.

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

[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.

[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. Pav*ón*, 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.

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

[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.