INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Diplomarbeit

Eine verteilte operationale Semantik für UML 2.0-Interaktionen

vorgelegt von
Heribert Mühlberger

Aufgabensteller Professor Dr. Alexander Knapp
Betreuer Professor Dr. Alexander Knapp und Dr. María Victoria Cengarle
Abgabetermin 9. November 2007

Inhaltsverzeichnis

Kapitel 1

Kurze Einleitung

UML Interaktionen beschreiben die Zusammenarbeit und den möglichen Nachrichtenaustausch von Objekten. In UML 2.0 ersetzt eine Variante der High-Level Message Sequence Charts (HMSC) die vergleichsweise wenig ausdrucksmächtigen UML 1.x Interaktionen. Eine alltagssprachliche Semantik für UML 2.0 Interaktionen wurde von der Object Management Group 2005 spezifiziert. Eine Trace-basierte, denotationelle Semantik für diese UML 2.0 Interaktionen wurde von Störrle 2003 und Cengarle/Knapp 2004 entwickelt. Eine entsprechende operationale Semantik wurde von Cengarle/Knapp 2005 angegeben. Ein Gleichungskalkül für äquivalente positive UML 2.0 Interaktionen wurde von Knapp/Störrle 2006 entwickelt. Schließlich leisteten Cengarle/Knapp 2007 durch die Spezifikation einer verteilten operationalen Semantik für UML 2.0 Interaktionen eine direkte Vorarbeit für die hier vorgelegte Diplomarbeit.

Eine verteilte operationale Semantik wird über denjenigen Kommunikationsakten spezifiziert, welche ein einzelnes Objekt mit seiner Umgebung ausführt. Diese "Umgebung" eines Objekts kann aus einem gemeinsam genutzten Kommunikationsbereich ("shared memory") und aus gepufferten, asynchronen Kanälen bestehen. Die Grundidee der verteilten operationalen Semantik ist es, die globale Konfigurationsinformation des Gesamtsystems auf die verschiedenen beteiligten Lifelines der einzelnen Objekte zu verteilen. Für jede Lifeline wird hierbei ein eigener lokaler Konfigurationsterm verwaltet, wobei die verschiedenen Lifelines nebenläufig ausgeführt werden (Interleaving-Modell). Die Bedeutung bestimmter Interaktionsfragmente (strict, loop, alt) kann in einer solchen verteilten Semantik allerdings nur dann korrekt ausgedrückt werden, wenn ein impliziter Mechanismus zur Synchronisation und Kommunikation zwischen den Lifelines vorhanden ist. Dies macht die Anwesenheit eines gemeinsam genutzen Kommunikationsbereichs in der Objektumgebung unvermeidlich. Das wesentliche Designziel einer verteilten operationalen Semantik ist—neben Korrektheit und Vollständigkeit bezüglich der denotationellen Semantik—die Minimierung dieser Menge an gemeinsam genutzter Information.

Ich möchte mich an dieser Stelle bei Herrn Professor Dr. Alexander Knapp für die ausgezeichnete Betreuung meiner Diplomarbeit und die stets angenehme und im besten Sinne spielerische Arbeitsatmosphäre bedanken. Der ganze Lehrstuhl PST zeichnet sich m. E. durch hohe Kompetenz und Freundlichkeit aus. Weil ich bereits ein Hochschulstudium abgeschlossen habe, weiß ich aus eigener schmerzhafter Erfahrung, dass eine solches angenehmes Arbeitsumfeld keine Selbstverständlichkeit ist. Mein Dank gilt auch Frau Dr. María Victoria Cengarle, die zahlreiche Ideen und Anregungen eingebracht hat. Ich danke meinen Eltern, ohne deren geduldige finanzielle Unterstützung ich dieses Zweitstudium nicht mit dem gewünschen Erfolg hätte abschließen können. Schließlich danke ich Frau Dr. Margret Popp, die mich bei der Gestaltung des (kurzen) Literaturverzeichnisses stilistisch beraten hat. Wir haben das Literaturverzeichnis in einer Variante der Harvard Notation gehalten, die auf den Vorschlägen von Standop/Meyer 2004 und Day/Gastel 2006 basiert.

Kapitel 2

Abstrakte Syntax und denotationelle Semantik

2.1 Abstrakte Syntax

Die abstrakte Syntax eines Fragments der Sprache der positiven UML 2.0-Interaktionen wird in Übereinstimmung zu Cengarle/Knapp 2007 definiert. Hierzu sei eine endliche Menge $\mathbb{I}$ von *Instanzen* und eine endliche Menge $\mathbb{M}$ von *Nachrichten* vorgegeben, deren syntaktische Gestalt unspezifiziert bleibe. Eine *Aktion* a (über $\mathbb{I}$ und $\mathbb{M}$) habe entweder die Form $\mathsf{snd}(s,r,m)$ oder die Form $\mathsf{rcv}(s,r,m)$ und bedeute das Versenden bzw. den Empfang einer Nachricht $m \in \mathbb{M}$ mit Senderinstanz $s \in \mathbb{I}$ und Empfängerinstanz $r \in \mathbb{I}$. Es sei $\mathbb{A}$ die Menge aller Aktionen a über $\mathbb{I}$ und $\mathbb{M}$. Wir definieren:

$$\begin{aligned}
&\alpha : \mathbb{A} \longrightarrow \wp(\mathbb{I}) && \mu : \mathbb{A} \longrightarrow \mathbb{M} \\
&\alpha(\mathsf{snd}(s,r,m)) := \{s\} && \mu(\mathsf{snd}(s,r,m)) := m \\
&\alpha(\mathsf{rcv}(s,r,m)) := \{r\} && \mu(\mathsf{rcv}(s,r,m)) := m
\end{aligned}$$

Im Falle $\alpha(a) = \{l\}$ heiße die Instanz l *aktiv* für Aktion a. Wir identifizieren Instanzen mit ihren Lifelines und nennen daher α die Lifelinefunktion. Ist l die aktive Instanz der Aktion a, so sagen wir auch, dass *a auf der Lifeline l liegt.* Wir definieren eine binäre, symmetrische Konfliktrelation $\lessgtr \subseteq \mathbb{A} \times \mathbb{A}$ wie folgt: $a_1 \lessgtr a_2 :\Leftrightarrow \alpha(a_1) \cap \alpha(a_2) \neq \emptyset$. Zwei Aktionen stehen demnach genau dann in Konflikt, wenn sie auf derselben Lifeline liegen.

Die abstrakte Syntax des von uns behandelten Teils der UML 2.0-Interaktionssprache ist in Tafel 2.1 angegeben. Dabei ist a Metavariable mit Bereich $\mathbb{A}$. Wir haben das behandelte Sprachfragment absichtlich klein gehalten. Als elementare Interaktionen (Basic Interactions) lassen wir zunächst nur die leere Interaktion Empty und einelementige Interaktionen a zu. Wie üblich gibt es zwei Arten

$$
\begin{array}{rcl}
I \in \mathrm{IFrag} & ::= & \mathsf{Empty} \\
& | & a \\
& | & \mathsf{strict}(I_1, I_2) \\
& | & \mathsf{seq}(I_1, I_2) \\
& | & \mathsf{par}(I_1, I_2) \\
& | & \mathsf{loop}(I) \\
& | & \mathsf{alt}(I_1, I_2)
\end{array}
$$

Tafel 2.1: Abstrakte Syntax der Interaktionen (Fragment)

der sequentiellen Komposition von Interaktionen: Bei der *strikten* Komposition $\mathsf{strict}(I_1, I_2)$ findet jede Aktionen von I_1 echt früher als jede Aktion von I_2 statt. Bei der *schwachen* oder *lokalen* Komposition $\mathsf{seq}(I_1, I_2)$ gilt nur für jede Lifeline l (jeweils für sich), dass jede auf l liegende Aktion von I_1 echt früher als jede auf l liegende Aktion von I_2 stattfindet. Bei der parallelen Komposition $\mathsf{par}(I_1, I_2)$ werden die Interaktionen I_1 und I_2 nebenläufig ausgeführt (Interleaving-Modell). Wir arbeiten mit einer reduzierten Schleifenform $\mathsf{loop}(I)$, die semantisch äquivalent zu $\mathsf{loop}(0, \infty, I)$ aus Cengarle/Knapp 2004 und 2005 ist, d. h. es wird nichtdeterministisch eine Zahl $n \in \mathbb{N}$ bestimmt und die Interaktion I mit sich selbst n-mal schwach komponiert. Unsere Schleifensemantik impliziert, dass immer irgendwann ein Schleifenabbruch eintritt. Die Disjunktion $\mathsf{alt}(I_1, I_2)$ führt in nichtdeterministischer Weise entweder die Interaktion I_1 oder die Interaktion I_2 aus.

Wir setzen die Lifelinefunktion α rekursiv auf IFrag fort:

$$
\alpha : \mathrm{IFrag} \longrightarrow \wp(\mathbb{I})
$$

$$
\begin{array}{lll}
\alpha(a) & \text{ist schon definiert} & \\
\alpha(\mathsf{Empty}) & := \emptyset & \\
\alpha(uop(I)) & := \alpha(I) & \text{mit } uop \in \{\mathsf{loop}\} \\
\alpha(bop(I_1, I_2)) & := \alpha(I_1) \cup \alpha(I_2) & \text{mit } bop \in \{\mathsf{strict}, \mathsf{seq}, \mathsf{par}, \mathsf{alt}\}
\end{array}
$$

Für jedes $L \subseteq \mathbb{I}$ setzen wir $\mathrm{IFrag}^L := \{I \in \mathrm{IFrag} \,|\, \alpha(I) \subseteq L\}$. Es ist IFrag^L die Menge aller Interaktionen, die *über der Lifelinemenge* L definiert sind.

2.2 Pomsets

Pomsets (partially ordered, labelled multisets) wurden im Jahre 1986 von Pratt zur Modellierung nebenläufiger Systeme eingeführt. Einige Begriffe zu diesem flexiblen Konzept seien im Folgenden rekapituliert. Eine *gelabelte* (d. h. mit Label versehene) *partielle Ordnung* (Abkürzung: g. p. O.) ist ein Tripel $(X, \leq_X, \lambda_X)$

bestehend aus einer Menge X, einer partiellen Ordnung $\leq_X$ auf X und einer Labelingfunktion λ_X, die auf X total definiert ist. Ein *Isomorphismus* zwischen zwei g. p. O.en $(X, \leq_X, \lambda_X)$ und $(Y, \leq_Y, \lambda_Y)$ ist eine bijektive Abbildung $I : X \to Y$, die monoton ist bez. $\leq_X$ und $\leq_Y$, deren Umkehrabbildung ebenfalls monoton ist und die labelerhaltend ist, d. h. es gilt $\lambda_X(x) = \lambda_Y(I(x))$ für alle $x \in X$. Ein *Pomset* ist die Isomorphieklasse $[(X, \leq_X, \lambda_X)]$ einer g. p. O. $(X, \leq_X, \lambda_X)$. Ein Pomset p heiße *endlich*, falls ein Repräsentant $(X, \leq_X, \lambda_X)$ von p existiert, dessen Grundmenge X endlich ist. In diesem Falle sind die Grundmengen aller Repräsentanten von p endlich. Ein Pomset p heiße *linear* oder *total geordnet* oder *Trace* oder *Tomset* (totally ordered, labelled multiset), falls ein Repräsentant $(X, \leq_X, \lambda_X)$ von p existiert, dessen Ordnung $\leq_X$ auf seiner Grundmenge X total ist. In diesem Falle sind die Ordnungen aller Repräsentanten von p total auf ihrer jeweiligen Grundmenge. Es heiße $(X, \leq'_X, \lambda_X)$ eine Linearisierung der g. p. O. $(X, \leq_X, \lambda_X)$, falls gilt: $\leq_X \subseteq \leq'_X$ und $\leq'_X$ ist auf X total. Ein Pomset p heiße eine *Linearisierung* eines Pomsets q, falls p die Isomorphieklasse einer Linearisierung eines Repräsentanten von q ist. Wir schreiben $q\!\downarrow$ für die Menge aller Linearisierungen von q.

Das *leere Pomset*, das von $(\emptyset, \emptyset, \emptyset)$ repräsentiert wird, notieren wir als ε. Ein Pomset p heiße *einelementig* oder *Atom*, falls es die Isomorphieklasse einer g. p. O. $(X, \leq_X, \lambda_X)$ mit einelementiger Grundmenge $X = \{x\}$ ist. Wir identifizieren das Atom p mit dem Label $\lambda_X(x)$. Seien $p = [(X, \leq_X, \lambda_X)]$ und $q = [(Y, \leq_Y, \lambda_Y)]$ zwei Pomsets und o. B. d. A. gelte $X \cap Y = \emptyset$. Die *Parallelkomposition* (concurrence) von p und q, die wir als $p \parallel q$ notieren, ist definiert als $[(X \cup Y, \leq_X \cup \leq_Y, \lambda_X \cup \lambda_Y)]$. Die *Konkatenation* (concatenation) von p und q, die wir als $p\,;q$ notieren, ist definiert als $[(X \cup Y, \leq_X \cup \leq_Y \cup (X \times Y), \lambda_X \cup \lambda_Y)]$. Für jede binäre, symmetrische Relation $\lessgtr$ auf Labels ist die $\lessgtr$-*Konkatenation* von p und q, die wir als $p\,;_{\lessgtr}\,q$ notieren, definiert als $[(X \cup Y, (\leq_X \cup \leq_Y \cup \{(x, y) \in X \times Y \mid \lambda_X(x) \lessgtr \lambda_Y(y)\})^*, \lambda_X \cup \lambda_Y)]$. Das Symbol * bildet die reflexive transitive Hülle[1]. Es ist einfach zu sehen, dass die obigen Begriffe wohldefiniert sind, d. h. sie hängen nur von den Pomsets p und q ab—nicht jedoch von der Wahl der Repräsentanten $(X, \leq_X, \lambda_X) \in p$ und $(Y, \leq_Y, \lambda_Y) \in q$. Es sei noch angemerkt, dass die Konkatenation und die $\lessgtr$-Konkatenation assoziativ sind und dass die Parallelkomposition assoziativ und kommutativ ist.

Ein *Prozess* ist eine Menge von Pomsets. Bildet eine n-stellige Funktion f Pomsets auf Pomsets ab, so definiert man die *Hebung* von f auf Prozesse $P_1, \ldots, P_n$ wie folgt: $f(P_1, \ldots, P_n) := \{f(p_1, \ldots, p_n) \mid p_1 \in P_1, \ldots, p_n \in P_n\}$. Z. B. ist $P_1 ;_{\lessgtr} P_2 := \{p_1 ;_{\lessgtr} p_2 \mid p_1 \in P_1, \ldots, p_n \in P_n\}$. Bildet eine n-stellige Funktion f Pomsets auf Prozesse ab, so "klopft" man die Bildelemente der Hebung von f "flach", d. h. man setzt $f(P_1, \ldots, P_n) := \bigcup\{f(p_1, \ldots, p_n) \mid p_1 \in P_1, \ldots, p_n \in P_n\}$.

[1]Die reflexive transitive Hülle einer Relation R stellt sich dar als $R^* := \bigcup_{n \in \mathbb{N}_0} R^n$, wobei aus dem Kontext klar sein muss, auf welcher Menge die Identität $R^0 = \mathrm{id}$ definiert ist. Im obigen Fall ist dies die Grundmenge $X \cup Y$ des Repräsentanten des zu definierenden Pomsets.

Z. B. ist $P{\downarrow} := \bigcup\{p{\downarrow} \mid p \in P\}$. Abschließend sei noch die *n-te Potenz* eines Prozesses definiert. Ist P ein Prozess, so ist $P^{0\lessgtr} := \{\varepsilon\}$ und $P^{(n+1)\lessgtr} := P \,;_{\lessgtr}\, P^{n\lessgtr}$.

2.3 Semantische Bereiche

Der Bereich $\mathbb{D}$ enthalte alle Pomsets $[(E, \leq_E, \lambda_E)]$ mit $\mathrm{Ran}(\lambda_E) \subseteq \mathbb{A}$, wobei $\mathrm{Ran}(\lambda_E)$ die Bildmenge von λ_E bezeichnet. Die Elemente $e \in E$ der Grundmenge eines Repräsentanten $(E, \leq_E, \lambda_E)$ eines Pomsets $p \in \mathbb{D}$ nennen wir *Ereignisse*. Das Ereignis e bezeichnet eine Ausführung der Aktion $\lambda_E(e)$.

Ein Pomset $p \in \mathbb{D}$ heiße *lokal linear*, falls ein Repräsentant $(E, \leq_E, \lambda_E)$ von p die Eigenschaft $\forall\, e_1, e_2 \in E.\, (\lambda_E(e_1) \lessgtr \lambda_E(e_2) \Longrightarrow e_1 \leq_E e_2 \vee e_2 \leq_E e_1)$ hat, wobei $\lessgtr$ die Konfliktrelation aus Abschnitt 2.1 ist. In diesem Fall hat *jeder* Repräsentant von p die genannte Eigenschaft. Wir setzen $\mathbb{P} := \{p \in \mathbb{D} \mid p \text{ ist lokal linear}\}$ und $\mathbb{T} := \{p \in \mathbb{D} \mid p \text{ ist eine Trace}\}$. Offenkundig ist $\mathbb{A} \subseteq \mathbb{T} \subseteq \mathbb{P} \subseteq \mathbb{D}$ und $\varepsilon \in \mathbb{T}$. Sind $a_1, a_2, \ldots, a_n \in \mathbb{A}$ mit $n \geq 1$, dann schreiben wir die endliche Trace $a_1 \,;\, a_2 \,;\, \ldots \,;\, a_n$ auch in der Form $a_1 a_2 \ldots a_n$.

Satz 2.1 Sei $p \in \mathbb{P}$ beliebig und $(E, \leq_E, \lambda_E)$ ein Repräsentant von p. Es bezeichne Min_E die Menge der Minima aus E bezüglich $\leq_E$. Dann ist $\lambda_E \restriction \mathrm{Min}_E$ injektiv.

Beweis. Seien p und $(E, \leq_E, \lambda_E)$ wie gefordert. Seien $e_1, e_2 \in \mathrm{Min}_E$ und gelte $\lambda_E(e_1) = \lambda_E(e_2)$. Weil $p \in \mathbb{P} \subseteq \mathbb{D}$ ist, gilt $\mathrm{Ran}(\lambda_E) \subseteq \mathbb{A}$. Somit gilt $\lambda_E(e_1) = \lambda_E(e_2) \in \mathbb{A}$. Weil für jede Aktion $a \in \mathbb{A}$ nach Definition $\alpha(a) \neq \emptyset$ ist, folgt $\alpha(\lambda_E(e_1)) \cap \alpha(\lambda_E(e_2)) = \alpha(\lambda_E(e_1)) \neq \emptyset$. Mithin gilt $\lambda_E(e_1) \lessgtr \lambda_E(e_2)$. Weil $p \in \mathbb{P}$ ist, folgt $e_1 \leq_E e_2 \vee e_2 \leq_E e_1$. Wegen $e_1, e_2 \in \mathrm{Min}_E$ folgt $e_1 = e_2$. □

Der Satz 2.1 ermöglicht auf der Menge der Minima eines lokal linearen Pomsets eine Identifikation von Ereignissen e mit den zugehörigen Aktionen $\lambda_E(e)$, was die Formulierung von operationalen Semantiken vereinfachen kann. Cengarle/Knapp 2005 verwenden diese Identifikation stillschweigend, was sich darin äußert, dass Ereignisse und Aktionen gelegentlich durch ein und dieselbe Metavariable e dargestellt werden. Wir folgen diesem Vorgehen nicht und benutzen stattdessen getrennte Metavariablen a für Aktionen und e für Ereignisse.

Die Lifelinefunktion α werde auf $\mathrm{IFrag} \cup \mathbb{D}$ fortgesetzt, indem für jedes $p = [(E, \leq_E, \lambda_E)] \in \mathbb{D}$ per definitionem $\alpha(p) = \bigcup_{e \in E} \alpha(\lambda_E(e))$ gelte. Ist $P \subseteq \mathbb{D}$ ein Prozess, dann setzen wir $\alpha(P) := \bigcup_{p \in P} \alpha(p)$. Für jede Lifelinemenge $L \subseteq \mathbb{I}$ ist die *Restriktionsfunktion* $\mathit{restr}(L) : \wp(\mathbb{D}) \to \wp(\mathbb{D})$ definiert durch $\mathit{restr}(L)(P) := \{p \in P \mid \alpha(p) \cap L = \emptyset\}$. Die Restriktionsfunktion $\mathit{restr}(L)$ entfernt aus einem Prozess $P \in \wp(\mathbb{D})$ alle diejenigen Pomsets, die eine Aktion enthalten, welche auf einer der Lifelines aus L liegt. Wir schreiben auch $P[L]$ für $\mathit{restr}(L)(P)$. Es gilt $\alpha(P[L]) \subseteq \alpha(P) \setminus L$.

2.4 Denotationelle Semantik

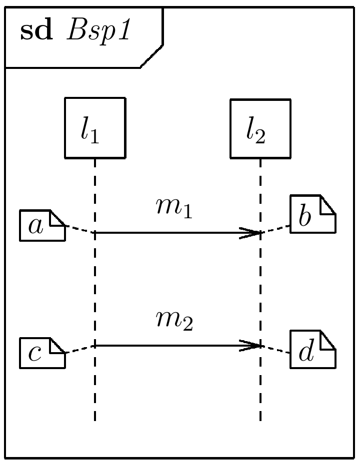
Wir rekapitulieren eine Trace-basierte, denotationelle Semantik für Interaktionsfragmente, die von Störrle 2003 und Cengarle/Knapp 2004 entwickelt worden ist, und stellen sie in einer konzisen Form[2] dar:

$$
\begin{array}{lcl}
\multicolumn{3}{l}{\mathcal{D}[\![-]\!] : \mathrm{IFrag} \longrightarrow \wp(\mathbb{T})} \\
\mathcal{D}[\![\mathsf{Empty}]\!] & := & \{\varepsilon\} \\
\mathcal{D}[\![a]\!] & := & \{a\} \\
\mathcal{D}[\![\mathsf{strict}(I_1, I_2)]\!] & := & \mathcal{D}[\![I_1]\!] \;;\; \mathcal{D}[\![I_2]\!] \\
\mathcal{D}[\![\mathsf{seq}(I_1, I_2)]\!] & := & (\mathcal{D}[\![I_1]\!] \;;_{\asymp}\; \mathcal{D}[\![I_2]\!])\!\downarrow \\
\mathcal{D}[\![\mathsf{par}(I_1, I_2)]\!] & := & (\mathcal{D}[\![I_1]\!] \parallel \mathcal{D}[\![I_2]\!])\!\downarrow \\
\mathcal{D}[\![\mathsf{loop}(I)]\!] & := & \bigcup_{n \in \mathbb{N}_0} ((\mathcal{D}[\![I]\!])^{n\asymp})\!\downarrow \\
\mathcal{D}[\![\mathsf{alt}(I_1, I_2)]\!] & := & \mathcal{D}[\![I_1]\!] \cup \mathcal{D}[\![I_2]\!]
\end{array}
$$

Für jede Interaktion $I \in \mathrm{IFrag}$ ist $\mathcal{D}[\![I]\!]$ die positive Erfüllungsmenge von I, d. h. es ist die Menge aller Traces $t \in \mathbb{T}$, welche die Interaktion I positiv erfüllen. Durch strukturelle Induktion über I sieht man leicht, dass für alle $I \in \mathrm{IFrag}$ gilt $\alpha(\mathcal{D}[\![I]\!]) = \alpha(I)$.

Anmerkung: Das Pomset $\mathcal{D}[\![I_1]\!] \parallel \mathcal{D}[\![I_2]\!]$ liegt i. A. nicht in $\mathbb{P}$. Doch zumindest im Bereich $\mathbb{D}$ sind alle Teilausdrücke wohldefiniert.

Beispiel 1 Seien $l_1, l_2 \in \mathbb{I}$, $l_1 \neq l_2$, $m_1, m_2 \in \mathbb{M}$. Wir setzen $a := \mathsf{snd}(l_1, l_2, m_1)$, $b := \mathsf{rcv}(l_1, l_2, m_1)$, $c := \mathsf{snd}(l_1, l_2, m_2)$, $d := \mathsf{rcv}(l_1, l_2, m_2)$. Die abstrakte Syntax des rechts angegebenen Interaktionsdiagramms ist $Bsp1 := \mathsf{seq}(\mathsf{strict}(a, b), \mathsf{strict}(c, d))$.



Es gilt: $\mathcal{D}[\![Bsp1]\!] =$

$$
\begin{array}{l}
= (\mathcal{D}[\![\mathsf{strict}(a, b)]\!] \;;_{\asymp}\; \mathcal{D}[\![\mathsf{strict}(c, d)]\!])\!\downarrow \\
= ((\mathcal{D}[\![a]\!] \;;\; \mathcal{D}[\![b]\!]) \;;_{\asymp}\; (\mathcal{D}[\![c]\!] \;;\; \mathcal{D}[\![d]\!]))\!\downarrow \\
= ((\{a\} \,;\, \{b\}) \;;_{\asymp}\; (\{c\} \,;\, \{d\}))\!\downarrow \\
= (ab \;;_{\asymp}\; cd)\!\downarrow \\
= (a \,;\, (b \parallel c) \,;\, d)\!\downarrow \\
= \{abcd, acbd\}.
\end{array}
$$

2.5 Prozesse und Lemmata

Für jeden Prozess $P \subseteq \mathbb{D}$ und jede Aktion $a \in \mathbb{A}$ ist der *linke Quotient* $P \,/\, a$ definiert als der Prozess $\{p \in \mathbb{D} \mid a \,;\, p \in P\}$.

[2] Die hier angegebene konzise Form der Semantik stammt aus Cengarle/Knapp 2007.

Lemma 2.2 Seien $P, P_1, P_2 \subseteq \mathbb{D}$ Prozesse, $a \in \mathbb{A}$ und $L \subseteq \mathbb{I}$. Es gilt:

1. $(P_1 \,;\, P_2){\downarrow} = (P_1{\downarrow}) \,;\, (P_2{\downarrow})$
2. $(P_1 \,;_{\lessgtr}\, P_2){\downarrow} = (P_1 \,;_{\lessgtr}\, (P_2{\downarrow})){\downarrow} = ((P_1{\downarrow}) \,;_{\lessgtr}\, P_2){\downarrow} = ((P_1{\downarrow}) \,;_{\lessgtr}\, (P_2{\downarrow})){\downarrow}$
3. $(P_1 \parallel P_2){\downarrow} = (P_1 \parallel (P_2{\downarrow})){\downarrow} = ((P_1{\downarrow}) \parallel P_2){\downarrow} = ((P_1{\downarrow}) \parallel (P_2{\downarrow})){\downarrow}$
4. $(P[L]){\downarrow} = (P{\downarrow})[L]$
5. $(P_1 \cup P_2){\downarrow} = (P_1{\downarrow}) \cup (P_2{\downarrow})$
6. $(\mathbb{D} \setminus P){\downarrow} \subseteq (\mathbb{D} \setminus (P{\downarrow})){\downarrow}$
7. $(P \,/\, a){\downarrow} \subseteq (P{\downarrow}) \,/\, a$
8. $P \subseteq \mathbb{T} \Rightarrow P{\downarrow} = P$

Beweis. Wir wollen hier nur für zwei Teilaussagen die Beweise ausführen.
Zu Teilaussage 2, erstes Gleichheitszeichen:
Z. z.: $(P_1 \,;_{\lessgtr}\, P_2){\downarrow} = (P_1 \,;_{\lessgtr}\, (P_2{\downarrow})){\downarrow}$
Z. z.: $\bigcup\{p{\downarrow} \mid p \in P_1 \,;_{\lessgtr}\, P_2\} = \bigcup\{p{\downarrow} \mid p \in P_1 \,;_{\lessgtr}\, (P_2{\downarrow})\}$
Z. z.: $\bigcup\{(p_1 \,;_{\lessgtr}\, p_2){\downarrow} \mid p_1 \in P_1 \wedge p_2 \in P_2\} = \bigcup\{(p_1 \,;_{\lessgtr}\, t_2){\downarrow} \mid p_1 \in P_1 \wedge t_2 \in P_2{\downarrow}\}$
Die rechte Seite der letzten Gleichheitsbeziehung formen wir weiter um. Es gilt:

$$\begin{aligned}
&\textstyle\bigcup\{(p_1 \,;_{\lessgtr}\, t_2){\downarrow} \mid p_1 \in P_1 \wedge t_2 \in P_2{\downarrow}\} = \\
&\textstyle\bigcup\{(p_1 \,;_{\lessgtr}\, t_2){\downarrow} \mid p_1 \in P_1 \wedge t_2 \in \bigcup\{p_2{\downarrow} \mid p_2 \in P_2\}\} = \\
&\textstyle\bigcup\{(p_1 \,;_{\lessgtr}\, t_2){\downarrow} \mid p_1 \in P_1 \wedge p_2 \in P_2 \wedge t_2 \in p_2{\downarrow}\} = \\
&\textstyle\bigcup\{\bigcup\{(p_1 \,;_{\lessgtr}\, t_2){\downarrow} \mid t_2 \in p_2{\downarrow}\} \mid p_1 \in P_1 \wedge p_2 \in P_2\}.
\end{aligned}$$

Bei einem Vergleich mit der linken Seite der zu zeigenden Gleichheitsbeziehung erkennt man, dass es genügt, die folgende Aussage zu zeigen:
Z. z.: $\forall\, p_1, p_2 \in \mathbb{D} \,.\, (p_1 \,;_{\lessgtr}\, p_2){\downarrow} = \bigcup\{(p_1 \,;_{\lessgtr}\, t_2){\downarrow} \mid t_2 \in p_2{\downarrow}\}$
Seien $p_1, p_2 \in \mathbb{D}$ beliebig. Für die Pomsets p_1 und p_2 wählen wir Repräsentanten, so dass gilt: $p_1 = [(E_1, \leq_1, \lambda_1)]$ und $p_2 = [(E_2, \leq_2, \lambda_2)]$ und $E_1 \cap E_2 = \emptyset$. Es gilt $p_1 \,;_{\lessgtr}\, p_2 = [(E, \leq_{1;\lessgtr 2}, \lambda)]$, wobei definiert ist $E := E_1 \cup E_2$ und $\lambda := \lambda_1 \cup \lambda_2$ und $\leq_{1;\lessgtr 2} := (\leq_1 \cup \leq_2 \cup \{(e_1, e_2) \in E_1 \times E_2 \mid \lambda_1(e_1) \lessgtr \lambda_2(e_2)\})^*$.
"$\subseteq$" Sei $t \in (p_1 \,;_{\lessgtr}\, p_2){\downarrow}$, d. h. es gilt $t = [(E, \leq, \lambda)]$ mit $\leq_{1;\lessgtr 2} \subseteq \leq$ und $\leq$ total auf E. Zu zeigen: $\exists\, t_2 \in p_2{\downarrow} \,.\, t \in (p_1 \,;_{\lessgtr}\, t_2){\downarrow}$. Wir setzen $t_2 := [(E_2, \leq_2', \lambda_2)]$ mit $\leq_2' := \leq \cap\, (E_2 \times E_2)$. Dann ist $\leq_2'$ eine totale Ordnung auf E_2. Es gilt $\leq_2 \subseteq \leq_{1;\lessgtr 2} \subseteq \leq$. Somit gilt $\leq_2 = \leq_2 \cap\, (E_2 \times E_2) \subseteq \leq \cap\, (E_2 \times E_2) = \leq_2'$ und daraus folgt $t_2 \in p_2{\downarrow}$. Es bleibt zu zeigen: $t \in (p_1 \,;_{\lessgtr}\, t_2){\downarrow}$
Z. z.: $(\leq_1 \cup \leq_2' \cup \{(e_1, e_2) \in E_1 \times E_2 \mid \lambda_1(e_1) \lessgtr \lambda_2(e_2)\})^* \subseteq \leq$
Das ist klar. Denn falls ein Schritt $(e_i, e_{i+1}) \in \leq_1 \cup \leq_2' \cup \ldots$ nicht aus $\leq_1 \cup \leq_2 \cup \ldots$ ist, dann ist er aus $\leq_2'$ und somit aus $\leq$.
"$\supseteq$" Sei $t \in \bigcup\{(p_1 \,;_{\lessgtr}\, t_2){\downarrow} \mid t_2 \in p_2{\downarrow}\}$. Es gibt $t_2 \in p_2{\downarrow}$ mit $t \in (p_1 \,;_{\lessgtr}\, t_2){\downarrow}$. Aus $t_2 \in p_2{\downarrow}$ folgt: $t_2 = [(E_2, \leq_2', \lambda_2)]$ mit $\leq_2 \subseteq \leq_2'$ und $\leq_2'$ total auf E_2. Aus $t \in (p_1 \,;_{\lessgtr}\, t_2){\downarrow}$ folgt: $t = [(E, \leq, \lambda)]$ mit $(\leq_1 \cup \leq_2' \cup \ldots)^* \subseteq \leq$ und $\leq$ total auf E.

Aus $\leq_2 \subseteq \leq_2'$ folgt $(\leq_1 \cup \leq_2 \cup \ldots)^* \subseteq (\leq_1 \cup \leq_2' \cup \ldots)^* \subseteq \leq$. Mithin gilt $t \in (p_1 \,;_{\lessgtr}\, p_2)\!\downarrow$.

Zu Teilaussage 6:

Z. z.: $\forall t \in (\mathbb{D} \setminus P)\!\downarrow . \, t \in (\mathbb{D} \setminus (P\!\downarrow))\!\downarrow$

Sei $t \in (\mathbb{D} \setminus P)\!\downarrow$. Also gibt es $p \in \mathbb{D} \setminus P$ mit $t \in p\!\downarrow$.

Z. z.: $\exists p' \in \mathbb{D} \setminus (P\!\downarrow) . \, t \in p'\!\downarrow$.

Fall 1: $p \notin P\!\downarrow$: Dann setzen wir $p' := p$. Fertig.

Fall 2: $p \in P\!\downarrow$: Dann ist p eine Trace und somit $t = p$. Außerdem kann p weder gleich ε noch einelementig sein, denn andernfalls wäre $p \in P$. Widerspruch. Also gilt $p = [(E, \leq, \lambda)]$ mit $|E| \geq 2$ und $\leq$ total auf E. Wir setzen $\leq' := \{(e, e) \mid e \in E\}$ und $p' := [(E, \leq', \lambda)]$. Weil $\leq' \subseteq \leq$ und $\leq$ total auf E ist, folgt $t = p \in p'\!\downarrow$. Weil $|E| \geq 2$ ist, ist p' keine Trace und somit $p' \notin P\!\downarrow$. □

Anmerkung: Die Gegenrichtung der Teilaussage 6 von Lemma 2.2 gilt nicht. Für ein Gegenbeispiel setze man $P := \{a \,\|\, b, ab, ba\}$. Dann ist $P\!\downarrow = \{ab, ba\}$ und somit $a \,\|\, b \in \mathbb{D} \setminus (P\!\downarrow)$. Folglich ist $ab \in (\mathbb{D} \setminus (P\!\downarrow))\!\downarrow$. Doch andererseits ist $ab \notin (\mathbb{D} \setminus P)\!\downarrow$. □

Lemma 2.3 Seien $P, P_1, P_2 \subseteq \mathbb{D}$ Prozesse, $a \in \mathbb{A}$ und $L \subseteq \mathbb{I}$. Es gilt:

1. $(P_1 \,;\, P_2) \,/\, a = ((P_1 \,/\, a) \,;\, P_2) \cup ((P_1 \cap \{\varepsilon\}) \,;\, (P_2 \,/\, a))$
2. $P_1, P_2 \subseteq \mathbb{T} \;\Rightarrow\; (P_1 \,;_{\lessgtr}\, P_2)\!\downarrow / \, a = ((P_1 \,/\, a) \,;_{\lessgtr}\, P_2)\!\downarrow \cup (P_1[\alpha(a)] \,;_{\lessgtr}\, (P_2 \,/\, a))\!\downarrow$
3. $P_1, P_2 \subseteq \mathbb{T} \;\Rightarrow\; (P_1 \,\|\, P_2)\!\downarrow / \, a = ((P_1 \,/\, a) \,\|\, P_2)\!\downarrow \cup (P_1 \,\|\, (P_2 \,/\, a))\!\downarrow$
4. $(P[L]) \,/\, a = \{\varepsilon \mid \alpha(a) \cap L = \emptyset\} \,;\, (P \,/\, a)[L]$
5. $(P_1 \cup P_2) \,/\, a = (P_1 \,/\, a) \cup (P_2 \,/\, a)$
6. $(\mathbb{D} \setminus P) \,/\, a = \mathbb{D} \setminus (P \,/\, a)$

Beweis. Bewiesen sei hier nur die zweite Teilaussage[3].

Z. z.: $P_1, P_2 \subseteq \mathbb{T} \;\Rightarrow\; (P_1 \,;_{\lessgtr}\, P_2)\!\downarrow / \, a = ((P_1 \,/\, a) \,;_{\lessgtr}\, P_2)\!\downarrow \cup (P_1[\alpha(a)] \,;_{\lessgtr}\, (P_2 \,/\, a))\!\downarrow$

Z. z.: $\forall t_1, t_2 \in \mathbb{T} . \, (t_1 \,;_{\lessgtr}\, t_2)\!\downarrow / \, a = ((\{t_1\} \,/\, a) \,;_{\lessgtr}\, \{t_2\})\!\downarrow \cup (\{t_1\}[\alpha(a)] \,;_{\lessgtr}\, (\{t_2\} \,/\, a))\!\downarrow$

Seien $t_1, t_2 \in \mathbb{T}$ beliebig. Für die Pomsets t_1 und t_2 wählen wir Repräsentanten, so dass gilt: $t_1 = [(E_1, \leq_1, \lambda_1)]$ und $t_2 = [(E_2, \leq_2, \lambda_2)]$ und $E_1 \cap E_2 = \emptyset$. Es ist $\leq_1$ total auf E_1 und $\leq_2$ total auf E_2. Es gilt $t_1 \,;_{\lessgtr}\, t_2 = [(E, \leq_{1;\lessgtr 2}, \lambda)]$, wobei definiert ist $E := E_1 \cup E_2$ und $\leq_{1;\lessgtr 2} := (\leq_1 \cup \leq_2 \cup \{(e_1, e_2) \in E_1 \times E_2 \mid \lambda_1(e_1) \lessgtr \lambda_2(e_2)\})^*$ und $\lambda := \lambda_1 \cup \lambda_2$.

"$\subseteq$" Sei $t \in (t_1 \,;_{\lessgtr}\, t_2)\!\downarrow / \, a$ bzw. $a \,;\, t \in (t_1 \,;_{\lessgtr}\, t_2)\!\downarrow$. Es folgt $a \,;\, t = [(E, \leq, \lambda)]$ mit $\leq_{1;\lessgtr 2} \subseteq \leq$ und $\leq$ total auf E. In der Grundmenge E des Repräsentanten $(E, \leq, \lambda)$ des Pomsets $a \,;\, t$ gibt es ein Ereignis $e_0 \in E$, so dass gilt: $\lambda(e_0) = a$

[3] Der Beweis der dritten Teilaussage ist methodisch ähnlich. Wegen der Beweise der restlichen Aussagen siehe Cengarle/Knapp 2005.

und $e_0 \leq e$ für jedes $e \in E$. Ferner gilt $t = [(E', \leq', \lambda')]$ mit $E' := E \setminus \{e_0\}$ und $\leq' := \leq \cap (E' \times E')$ und $\lambda' := \lambda \restriction E'$.

Fall 1: $e_0 \in E_1$:

Z. z.: $t \in ((\{t_1\} / a) \mathbin{;_{\between}} \{t_2\})\downarrow$

Z. z.: $t \in (t'_1 \mathbin{;_{\between}} t_2)\downarrow$ mit einem $t'_1 \in \{t_1\} / a$

Z. z.: $\exists\, t'_1 \in \mathbb{T} \,.\, a \,; t'_1 = t_1 \;\wedge\; t \in (t'_1 \mathbin{;_{\between}} t_2)\downarrow$

Wir setzen $t'_1 := [(E'_1, \leq'_1, \lambda'_1)]$ mit $E'_1 := E_1 \setminus \{e_0\}$ und $\leq'_1 := \leq_1 \cap (E'_1 \times E'_1)$ und $\lambda'_1 := \lambda_1 \restriction E'_1$. Weil $\leq_1$ totale Ordnung auf E_1 ist, ist $\leq'_1$ totale Ordnung auf E'_1.

(1) Zu zeigen: $a \,; t'_1 = t_1$

Die Aktion a wird mit dem Atom $[(\{e_0\}, \{(e_0, e_0)\}, \{(e_0, a)\})]$ identifiziert. Somit gilt $a \,; t'_1 = [(E''_1, \leq''_1, \lambda''_1)]$, wobei definiert ist $E''_1 := \{e_0\} \cup E'_1 = E_1$ und $\lambda''_1 := \{(e_0, a)\} \cup \lambda'_1 = \{(e_0, a)\} \cup \lambda_1 \restriction E'_1 = \lambda_1$ und $\leq''_1 := \{(e_0, e_0)\} \cup \leq'_1 \cup \{e_0\} \times E'_1$. Es genügt zu zeigen, dass gilt $\leq''_1 \subseteq \leq_1$: Weil $\leq_1$ total auf E_1 ist, folgt mit Widerspruchsbeweis, dass gilt $\leq \cap (E_1 \times E_1) = \leq_1$. Es folgt $\{e_0\} \times E_1 \subseteq \leq_1$ und $\leq_1 \cap (E'_1 \times \{e_0\}) = \emptyset$. Also gilt $\leq''_1 = \{e_0\} \times E_1 \cup \leq'_1 = \{e_0\} \times E_1 \cup (\leq_1 \cap (E'_1 \times E'_1))$ $= (\leq_1 \cap (\{e_0\} \times E_1)) \cup (\leq_1 \cap (E'_1 \times E'_1)) \cup (\leq_1 \cap (E'_1 \times \{e_0\})) = \leq_1$.

(2) Zu zeigen: $t \in (t'_1 \mathbin{;_{\between}} t_2)\downarrow$

Es gilt $t = [(E', \leq', \lambda')]$. Dabei ist $\leq'$ totale Ordnung auf E'. Es gilt $t'_1 \mathbin{;_{\between}} t_2 = [(E', \leq'_{1;\between 2}, \lambda')]$ mit $\leq'_{1;\between 2} := (\leq'_1 \cup \leq_2 \cup \{(e_1, e_2) \in E'_1 \times E_2 \mid \lambda'_1(e_1) \between \lambda_2(e_2)\})^*$. Zu zeigen bleibt $\leq'_{1;\between 2} \subseteq \leq'$: Es gilt (i) $\leq'_1 \subseteq \leq_1 \cap (E' \times E') \subseteq \leq_{1;\between 2} \cap (E' \times E')$ $\subseteq \leq \cap (E' \times E') = \leq'$. Es gilt (ii) $\leq_2 = \leq_2 \cap (E' \times E') \subseteq \leq_{1;\between 2} \cap (E' \times E') \subseteq \leq \cap$ $(E' \times E') = \leq'$. Ferner gilt (iii) $\{(e_1, e_2) \in E'_1 \times E_2 \mid \lambda'_1(e_1) \between \lambda_2(e_2)\} = \{(e_1, e_2) \in E_1 \times E_2 \mid \lambda_1(e_1) \between \lambda_2(e_2)\} \cap (E' \times E') \subseteq \leq_{1;\between 2} \cap (E' \times E') \subseteq \leq \cap (E' \times E') = \leq'$. Aus (i), (ii), (iii) und der Tatsache, dass $\leq'$ reflexiv und transitiv ist, folgt $\leq'_{1;\between 2} \subseteq \leq'$.

Fall 2: $e_0 \in E_2$:

Z. z.: $t \in (\{t_1\}[\alpha(a)] \mathbin{;_{\between}} (\{t_2\} / a))\downarrow$

Z. z.: $\alpha(a) \cap \alpha(t_1) = \emptyset \;\wedge\; \exists\, t'_2 \in \mathbb{T} \,.\, a \,; t'_2 = t_2 \;\wedge\; t \in (t_1 \mathbin{;_{\between}} t'_2)\downarrow$

Angenommen es wäre $\alpha(a) \cap \alpha(t_1) \neq \emptyset$. D. h. es ist $\alpha(a) \cap \bigcup_{e_1 \in E_1} \alpha(\lambda_1(e_1)) \neq \emptyset$. Also gibt es ein $e_1 \in E_1$, so dass $\alpha(a) \cap \alpha(\lambda_1(e_1)) \neq \emptyset$. Weil $a = \lambda_2(e_0)$ ist, folgt $\alpha(\lambda_2(e_0)) \cap \alpha(\lambda_1(e_1)) \neq \emptyset$. Es folgt $\lambda_1(e_1) \between \lambda_2(e_0)$. Hieraus folgt $e_1 \leq_{1;\between 2} e_0$ und somit $e_1 \leq e_0$. Andererseits gilt $e_0 \leq e_1$. Mit der Antisymmetrie von $\leq$ folgt $e_1 = e_0$. Demnach wäre $E_1 \cap E_2 \neq \emptyset$. Widerspruch. Der restliche Beweis zu Fall 2 ist analog zu Fall 1.

"$\supseteq$" Es sei o. B. d. A. $t \in (\{t_1\}[\alpha(a)] \mathbin{;_{\between}} (\{t_2\} / a))\downarrow$. Dann gilt $\alpha(a) \cap \alpha(t_1) = \emptyset$ und es gibt $t'_2 \in \mathbb{T}$ mit (1) $a \,; t'_2 = t_2$ und (2) $t \in (t_1 \mathbin{;_{\between}} t'_2)\downarrow$. Aufgrund von (1) und $t_2 = [(E_2, \leq_2, \lambda_2)]$ gibt es $e_0 \in E_2$, so dass gilt $\lambda_2(e_0) = a$ und $e_0 \leq_2 e$ für jedes $e \in E_2$. Es gilt $t'_2 = [(E'_2, \leq'_2, \lambda'_2)]$ mit $E'_2 := E_2 \setminus \{e_0\}$ und $\lambda'_2 := \lambda_2 \restriction E'_2$ und $\leq'_2 := \leq_2 \cap (E'_2 \times E'_2)$. Folglich ist $t_1 \mathbin{;_{\between}} t'_2 = [(E', \leq'_{1;\between 2}, \lambda')]$ mit $E' := E_1 \cup E'_2$

und $\lambda' := \lambda_1 \cup \lambda'_2$ und $\leq'_{1;\lessgtr 2} := (\leq_1 \cup \leq'_2 \cup \{(e_1, e_2) \in E_1 \times E'_2 \mid \lambda_1(e_1) \lessgtr \lambda'_2(e_2)\})^*$. Wegen (2) gilt $t = [(E', \leq', \lambda')]$ mit $\leq'_{1;\lessgtr 2} \subseteq \leq'$ und $\leq'$ totale Ordnung auf E'.

Z. z.: $a\,; t \in (t_1 \,;_{\lessgtr}\, t_2)\!\downarrow$
Es gilt $a\,; t = [(E, \leq, \lambda)]$ mit $\leq := \{(e_0, e_0)\} \cup \leq' \cup \{e_0\} \times E'$. Es ist $\leq$ total auf E. Zu zeigen bleibt $\leq_{1;\lessgtr 2} \subseteq \leq$: Es gilt (i) $\leq_1 \subseteq \leq'_{1;\lessgtr 2} \subseteq \leq' \subseteq \leq$. Es gilt (ii) $\leq_2 \subseteq \{(e_0, e_0)\} \cup (\{e_0\} \times E'_2) \cup (\leq_2 \cap (E'_2 \times E'_2)) \subseteq \{(e_0, e_0)\} \cup (\{e_0\} \times E') \cup \leq'_2 = \ldots \cup \leq'_{1;\lessgtr 2} \subseteq \ldots \cup \leq' = \leq$. Sei nun $(e_1, e_2) \in E_1 \times E_2$ und gelte $\lambda_1(e_1) \lessgtr \lambda_2(e_2)$. Weil $\alpha(a) \cap \alpha(t_1) = \emptyset$ ist, folgt $e_2 \neq e_0$ bzw. $e_2 \in E'_2$. Daraus folgt $(e_1, e_2) \in \leq'_{1;\lessgtr 2}$ und somit $(e_1, e_2) \in \leq$. Also gilt (iii) $\{(e_1, e_2) \in E_1 \times E_2 \mid \lambda_1(e_1) \lessgtr \lambda_2(e_2)\} \subseteq \leq$. Aus (i), (ii) und (iii) und der Tatsache, dass $\leq$ reflexiv und transitiv ist, folgt $\leq_{1;\lessgtr 2} \subseteq \leq$. ☐

Lemma 2.4 Seien $P, P_1, P_2 \subseteq \mathbb{D}$ Prozesse und $L \subseteq \mathbb{I}$. Es gilt:

1. $\varepsilon \in (P_1\,; P_2) \iff (\varepsilon \in P_1) \wedge (\varepsilon \in P_2)$
2. $\varepsilon \in (P_1 \,;_{\lessgtr}\, P_2) \iff (\varepsilon \in P_1) \wedge (\varepsilon \in P_2)$
3. $\varepsilon \in (P_1 \parallel P_2) \iff (\varepsilon \in P_1) \wedge (\varepsilon \in P_2)$
4. $\varepsilon \in (P[L]) \iff \varepsilon \in P$
5. $\varepsilon \in P \iff \varepsilon \in P\!\downarrow$

Lemma 2.5 Seien $P, P', P_1, P'_1, P_2, P'_2 \subseteq \mathbb{D}$ Prozesse und $L, L' \subseteq \mathbb{I}$. Es gilt:

1. $P_1 \subseteq P'_1 \wedge P_2 \subseteq P'_2 \Rightarrow (P_1\,; P_2) \subseteq (P'_1\,; P'_2)$
2. $P_1 \subseteq P'_1 \wedge P_2 \subseteq P'_2 \Rightarrow (P_1 \,;_{\lessgtr}\, P_2) \subseteq (P'_1 \,;_{\lessgtr}\, P'_2)$
3. $P_1 \subseteq P'_1 \wedge P_2 \subseteq P'_2 \Rightarrow (P_1 \parallel P_2) \subseteq (P'_1 \parallel P'_2)$
4. $P \subseteq P' \wedge L \supseteq L' \Rightarrow (P[L]) \subseteq (P'[L'])$
5. $P_1 \subseteq P'_1 \wedge P_2 \subseteq P'_2 \Rightarrow (P_1 \cup P_2) \subseteq (P'_1 \cup P'_2)$
6. $P \supseteq P' \Rightarrow (\mathbb{D} \setminus P) \subseteq (\mathbb{D} \setminus P')$
7. $P \subseteq P' \Rightarrow P\!\downarrow \subseteq P'\!\downarrow$

Die Beweise der Lemmata 2.4 und 2.5 sind trivial.

Lemma 2.6 Seien $P, P_1, P_2 \subseteq \mathbb{D}$ Prozesse, $L \subseteq \mathbb{I}$ und $n \in \mathbb{N}_0$. Es gilt:

1. $(P_1 \cap P_2)\,; P \subseteq (P_1\,; P) \cap (P_2\,; P)$
2. $(P_1\,; P_2)[L] = (P_1[L])\,; (P_2[L])$
3. $(P_1 \,;_{\lessgtr}\, P_2)[L] = (P_1[L]) \,;_{\lessgtr}\, (P_2[L])$
4. $(P_1 \parallel P_2)[L] = (P_1[L]) \parallel (P_2[L])$
5. $(P_1 \cup P_2)[L] = (P_1[L]) \cup (P_2[L])$
6. $(P^{n\lessgtr})[L] = (P[L])^{n\lessgtr}$

Auch dieses Lemma sei hier nicht bewiesen, sondern lediglich dargetan, dass die Gegenrichtung der ersten Teilaussage nicht gilt. Für ein Gegenbeispiel setze man $P_1 := \{b\}$ und $P_2 := \{bc\}$ und $P := \{c, cc\}$. Dann gilt $(P_1\,; P) \cap (P_2\,; P) = \{bc, bcc\} \cap \{bcc, bccc\} = \{bcc\}$. Andererseits ist $P_1 \cap P_2 = \emptyset$ und folglich auch $(P_1 \cap P_2)\,; P = \emptyset$. ☐

Die fehlende volle Distributivität von Konkatenation und Schnittbildung von Prozessen hat leider zur Folge, dass das Lemma 6 aus Cengarle/Knapp 2005 und das daraus resultierende Vollständigkeitsresultat zusammenbricht. Für ein Gegenbeispiel setze man $S := \mathsf{not}(\mathsf{strict}(\mathsf{not}(\mathsf{alt}(ab, abc)), \mathsf{alt}(c, cc)))$ mit paarweise verschiedenen Aktionen a, b, c und betrachte die Trace $t = abcc$.

Kapitel 3

Globale operationale Semantik

Wir setzen $\mathbb{A}_\tau := \mathbb{A} \dot{\cup} \{\tau\}$ wobei τ die *stille Aktion* ist. Das Symbol $\dot{\cup}$ bedeutet die Vereinigung zweier disjunkter Mengen. Der Bereich $\mathbb{D}_\tau$ enthalte alle Pomsets $[(E, \leq_E, \lambda_E)]$ mit $\mathrm{Ran}(\lambda_E) \subseteq \mathbb{A}_\tau$. Wir setzen $\alpha(\tau) := \emptyset$ und fassen die Konfliktrelation $\lessgtr$ aus Abschnitt 2.1 als Relation auf $\mathbb{A}_\tau$ auf. Somit ist der Begriff der lokalen Linearität aus Abschnitt 2.3 für alle Pomsets aus $\mathbb{D}_\tau$ wohldefiniert. Wir setzen $\mathbb{P}_\tau := \{p \in \mathbb{D}_\tau \,|\, p \text{ ist lokal linear}\}$ und $\mathbb{T}_\tau := \{p \in \mathbb{D}_\tau \,|\, p \text{ ist eine Trace}\}$. Die Lifelinefunktion α fassen wir nun als Funktion auf $\mathrm{IFrag} \cup \mathbb{D}_\tau$ auf.

3.1 Konfigurationen und Transitionen

Eine Konfiguration der globalen operationalen Small-Step-Semantik ist eine Interaktion $I \in \mathrm{IFrag}$. Die einzige Endkonfiguration ist Empty. Transitionen haben die allgemeine Form $I \xrightarrow{\bar{a}}_{\mathrm{Gu}} I'$ mit $I, I' \in \mathrm{IFrag}$, $I \neq \mathsf{Empty}$ und $\bar{a} \in \mathbb{A}_\tau$. Die Regeln der globalen operationalen Semantik werden in Abschnitt 3.3 behandelt.

3.2 Syntaktische Restriktionsfunktion

Diejenigen Regeln einer operationalen Semantik, die die schwache Komposition $\mathsf{seq}(I_1, I_2)$ betreffen, können nur dann bezüglich der denotationellen Semantik korrekt sein, wenn sie gewährleisten, dass nach Ausführung einer Aktion a der Interaktion I_2 keine mit a in Konflikt stehenden Aktionen a' der Interaktion I_1 mehr ausgeführt werden können. Wenn in der Interaktion I_1 ein nichtdeterministisches Entscheidungskonstrukt (alt oder loop) enthalten ist, dann kann es geschehen, dass nach Ausführung einer Aktion a der Interaktion I_2 in der positiven Erfüllungsmenge von I_1 sowohl Traces existieren, die eine mit a in Konflikt stehende Aktion a' enthalten (Typ 1), als auch solche Traces, die keine solche Aktion enthalten (Typ 2). Vollständigkeit des Regelsystems setzt voraus, dass die Traces vom Typ 2 nicht verworfen werden, nur weil Traces vom Typ 1 existieren.

Eine elegante Methode zur Formulierung eines korrekten und vollständigen Regelsystems zur schwachen Komposition beruht auf einer Funktion R_L, die einen gegebenen Interaktionsterm I syntaktisch so umformt, dass aus seiner positiven Erfüllungsmenge alle Traces beseitigt werden, die mindestens eine Aktion enthalten, die auf einer der Lifelines einer gegebenen Lifelinemenge L liegt. Üblicherweise wird dies erreicht, indem man $\mathrm{R}_L(I) := \mathsf{restr}(L, I)$ setzt, wobei $\mathsf{restr}(L, I)$ ein besonderes Kapselungskonstrukt ist, dessen denotationelle Semantik gemäß $\mathcal{D}[\![\mathsf{restr}(L, I)]\!] := \mathcal{D}[\![I]\!][L]$ definiert wird. Weil wir das Design unserer Interaktionssprache absichtlich schmal gehalten haben, steht uns ein solches Kapselungskonstrukt nicht zur Verfügung. Eine Funktion R_L mit der o. g. Eigenschaft kann aber auch ohne ein solches Kapselungskonstrukt definiert werden. Hierzu gibt es verschiedene Möglichkeiten, die in Abschnitt 3.5 diskutiert werden.

Zur Formulierung der globalen operationalen Semantik wählen wir eine Variante, die keine Erweiterung der Interaktionssprache IFrag erforderlich macht. Für jedes $L \subseteq \mathbb{I}$ definieren wir:

$$\mathrm{R}_L : \mathrm{IFrag} \mathbin{\dot\cup} \{\mathsf{None}\} \longrightarrow \mathrm{IFrag} \mathbin{\dot\cup} \{\mathsf{None}\}$$

$$\begin{array}{lll}
\mathrm{R}_L(\mathsf{None}) & := & \mathsf{None} \\
\mathrm{R}_L(\mathsf{Empty}) & := & \mathsf{Empty} \\
\mathrm{R}_L(a) & := & \begin{cases} a & \text{f. } \alpha(a) \cap L = \emptyset \\ \mathsf{None} & \text{andernfalls} \end{cases} \\
\mathrm{R}_L(\mathsf{strict}(I_1, I_2)) & := & \begin{cases} \mathsf{strict}(\mathrm{R}_L(I_1), \mathrm{R}_L(I_2)) & \text{f. } \mathrm{R}_L(I_1) \neq \mathsf{None} \wedge \mathrm{R}_L(I_2) \neq \mathsf{None} \\ \mathsf{None} & \text{andernfalls} \end{cases} \\
\mathrm{R}_L(\mathsf{seq}(I_1, I_2)) & & \text{analog zu } \mathsf{strict} \\
\mathrm{R}_L(\mathsf{par}(I_1, I_2)) & & \text{analog zu } \mathsf{strict} \\
\mathrm{R}_L(\mathsf{loop}(I)) & := & \begin{cases} \mathsf{loop}(\mathrm{R}_L(I)) & \text{f. } \mathrm{R}_L(I) \neq \mathsf{None} \\ \mathsf{Empty} & \text{andernfalls} \end{cases} \\
\mathrm{R}_L(\mathsf{alt}(I_1, I_2)) & := & \begin{cases} \mathsf{alt}(\mathrm{R}_L(I_1), \mathrm{R}_L(I_2)) & \text{f. } \mathrm{R}_L(I_1) \neq \mathsf{None} \wedge \mathrm{R}_L(I_2) \neq \mathsf{None} \\ \mathrm{R}_L(I_1) & \text{f. } \mathrm{R}_L(I_1) \neq \mathsf{None} \wedge \mathrm{R}_L(I_2) = \mathsf{None} \\ \mathrm{R}_L(I_2) & \text{f. } \mathrm{R}_L(I_1) = \mathsf{None} \wedge \mathrm{R}_L(I_2) \neq \mathsf{None} \\ \mathsf{None} & \text{andernfalls} \end{cases}
\end{array}$$

Das Symbol None, das in der Definition von R_L auftritt, ist kein Bestandteil der Interaktionssprache IFrag. Es handelt sich lediglich um einen besonderen Rückgabewert, der von der Funktion R_L geliefert wird, falls die positive Erfüllungsmenge der transformierte Interaktion $\mathrm{R}_L(I)$ leer ist.

Lemma 3.1 Sei $I \in \mathrm{IFrag} \dot{\cup} \{\mathsf{None}\}$ und $L \subseteq \mathbb{I}$. Setze nur für die Zwecke dieses Lemmas die semantische Funktion $\mathcal{D}[\![-]\!]$ mittels $\mathcal{D}[\![\mathsf{None}]\!] := \emptyset$ auf $\mathrm{IFrag}\dot{\cup}\{\mathsf{None}\}$ fort. Dann gilt: $\mathcal{D}[\![\mathrm{R}_L(I)]\!] = \mathcal{D}[\![I]\!][L]$.

Beweis. Wir führen den Beweis durch strukturelle Induktion über I:

$I = \mathsf{None}$: Es folgt: $\mathcal{D}[\![\mathrm{R}_L(\mathsf{None})]\!] = \mathcal{D}[\![\mathsf{None}]\!] = \emptyset = \emptyset[L] = \mathcal{D}[\![\mathsf{None}]\!][L]$.

$I = \mathsf{Empty}$: Es folgt: $\mathcal{D}[\![\mathrm{R}_L(\mathsf{Empty})]\!] = \mathcal{D}[\![\mathsf{Empty}]\!] = \{\varepsilon\} = \{\varepsilon\}[L] = \mathcal{D}[\![\mathsf{Empty}]\!][L]$.

$I = a$: Wir unterscheiden zwei Fälle:

Fall 1: $\alpha(a) \cap L = \emptyset$. Es folgt: $\mathcal{D}[\![\mathrm{R}_L(a)]\!] = \mathcal{D}[\![a]\!] = \{a\} = \{a\}[L] = \mathcal{D}[\![a]\!][L]$.

Fall 2: $\alpha(a) \cap L \neq \emptyset$. Es folgt: $\mathcal{D}[\![\mathrm{R}_L(a)]\!] = \mathcal{D}[\![\mathsf{None}]\!] = \emptyset = \{a\}[L] = \mathcal{D}[\![a]\!][L]$.

$I = \mathsf{strict}(I_1, I_2)$: Wir unterscheiden zwei Fälle:

Fall 1: $\mathrm{R}_L(I_1) \neq \mathsf{None}$ und $\mathrm{R}_L(I_2) \neq \mathsf{None}$. Dann gilt $\mathcal{D}[\![\mathrm{R}_L(\mathsf{strict}(I_1, I_2))]\!] = \mathcal{D}[\![\mathsf{strict}(\mathrm{R}_L(I_1), \mathrm{R}_L(I_2))]\!] = \mathcal{D}[\![\mathrm{R}_L(I_1)]\!] \,;\, \mathcal{D}[\![\mathrm{R}_L(I_2)]\!] =$ (mit I. V. für I_1 und I_2) $\mathcal{D}[\![I_1]\!][L] \,;\, \mathcal{D}[\![I_2]\!][L] = (\mathcal{D}[\![I_1]\!] \,;\, \mathcal{D}[\![I_2]\!])[L] = \mathcal{D}[\![\mathsf{strict}(I_1, I_2)]\!][L]$, wobei im vorletzten Schritt das Lemma 2.6.2 angewendet wurde.

Fall 2: $\mathrm{R}_L(I_1) = \mathsf{None}$ oder $\mathrm{R}_L(I_2) = \mathsf{None}$. Mit der I. V. für I_1 und I_1 folgt, dass $\mathcal{D}[\![I_1]\!][L] = \emptyset$ oder $\mathcal{D}[\![I_2]\!][L] = \emptyset$ ist. Es folgt: $\mathcal{D}[\![\mathrm{R}_L(\mathsf{strict}(I_1, I_2))]\!] = \mathcal{D}[\![\mathsf{None}]\!] = \emptyset = \mathcal{D}[\![I_1]\!][L] \,;\, \mathcal{D}[\![I_2]\!][L] = (\mathcal{D}[\![I_1]\!] \,;\, \mathcal{D}[\![I_2]\!])[L] = \mathcal{D}[\![\mathsf{strict}(I_1, I_2)]\!][L]$.

$I = \mathsf{seq}(I_1, I_2)$: Analog zu Fall $I = \mathsf{strict}(I_1, I_2)$ mit Lemma 2.6.3 und 2.2.4.

$I = \mathsf{par}(I_1, I_2)$: Analog zu Fall $I = \mathsf{strict}(I_1, I_2)$ mit Lemma und 2.2.4.

$I = \mathsf{loop}(I_1)$: Wir unterscheiden zwei Fälle:

Fall 1: $\mathrm{R}_L(I_1) \neq \mathsf{None}$. Folglich ist $\mathcal{D}[\![\mathrm{R}_L(\mathsf{loop}(I_1))]\!] = \mathcal{D}[\![\mathsf{loop}(\mathrm{R}_L(I_1))]\!] = \bigcup_{n \in \mathbb{N}_0}((\mathcal{D}[\![\mathrm{R}_L(I_1)]\!])^{n\between})\!\downarrow$. Mit der I. V. für I_1 folgt: $\bigcup_{n \in \mathbb{N}_0}((\mathcal{D}[\![\mathrm{R}_L(I_1)]\!])^{n\between})\!\downarrow = \bigcup_{n \in \mathbb{N}_0}((\mathcal{D}[\![I_1]\!][L])^{n\between})\!\downarrow$. Vermöge der Lemmata 2.6.6 und 2.2.4 und 2.6.5 (analog) folgt: $\bigcup_{n \in \mathbb{N}_0}((\mathcal{D}[\![I_1]\!][L])^{n\between})\!\downarrow = (\bigcup_{n \in \mathbb{N}_0}((\mathcal{D}[\![I_1]\!])^{n\between})\!\downarrow)[L] = \mathcal{D}[\![\mathsf{loop}(I_1)]\!][L]$.

Fall 2: $\mathrm{R}_L(I_1) = \mathsf{None}$. Es gilt $\mathcal{D}[\![\mathrm{R}_L(\mathsf{loop}(I_1))]\!] = \mathcal{D}[\![\mathsf{Empty}]\!] = \{\varepsilon\} = \{\varepsilon\} \cup \emptyset = \{\varepsilon\}\!\downarrow \cup \bigcup_{n \geq 1} \emptyset\!\downarrow = (\emptyset^{0\between})\!\downarrow \cup \bigcup_{n \geq 1}(\emptyset^{n\between})\!\downarrow = \bigcup_{n \in \mathbb{N}_0}(\emptyset^{n\between})\!\downarrow = \bigcup_{n \in \mathbb{N}_0}((\mathcal{D}[\![\mathsf{None}]\!])^{n\between})\!\downarrow = \bigcup_{n \in \mathbb{N}_0}((\mathcal{D}[\![\mathrm{R}_L(I_1)]\!])^{n\between})\!\downarrow = \ldots$ (wie in Fall 1) $\ldots = \mathcal{D}[\![\mathsf{loop}(I_1)]\!][L]$.

$I = \mathsf{alt}(I_1, I_2)$: Wir unterscheiden vier Fälle:

Fall 1: $\mathrm{R}_L(I_1) \neq \mathsf{None}$ und $\mathrm{R}_L(I_2) \neq \mathsf{None}$. Dann gilt $\mathcal{D}[\![\mathrm{R}_L(\mathsf{alt}(I_1, I_2))]\!] = \mathcal{D}[\![\mathsf{alt}(\mathrm{R}_L(I_1), \mathrm{R}_L(I_2))]\!] = \mathcal{D}[\![\mathrm{R}_L(I_1)]\!] \cup \mathcal{D}[\![\mathrm{R}_L(I_2)]\!] =$ (mit der I. V. für I_1 und I_2) $\mathcal{D}[\![I_1]\!][L] \cup \mathcal{D}[\![I_2]\!][L] = (\mathcal{D}[\![I_1]\!] \cup \mathcal{D}[\![I_2]\!])[L] = \mathcal{D}[\![\mathsf{alt}(I_1, I_2)]\!][L]$.

Fall 2: $\mathrm{R}_L(I_1) \neq \mathsf{None}$ und $\mathrm{R}_L(I_2) = \mathsf{None}$. Dann gilt $\mathcal{D}[\![\mathrm{R}_L(\mathsf{alt}(I_1, I_2))]\!] = \mathcal{D}[\![\mathrm{R}_L(I_1)]\!] = \mathcal{D}[\![\mathrm{R}_L(I_1)]\!] \cup \emptyset = \mathcal{D}[\![\mathrm{R}_L(I_1)]\!] \cup \mathcal{D}[\![\mathsf{None}]\!] = \mathcal{D}[\![\mathrm{R}_L(I_1)]\!] \cup \mathcal{D}[\![\mathrm{R}_L(I_2)]\!] =$ (wie in Fall 1) $\ldots = \mathcal{D}[\![\mathsf{alt}(I_1, I_2)]\!][L]$.

Fall 3: $\mathrm{R}_L(I_1) = \mathsf{None}$ und $\mathrm{R}_L(I_2) \neq \mathsf{None}$. Analog zu Fall 2.

Fall 4: $\mathrm{R}_L(I_1) = \mathsf{None}$ und $\mathrm{R}_L(I_2) = \mathsf{None}$. Dann gilt $\mathcal{D}[\![\mathrm{R}_L(\mathsf{alt}(I_1, I_2))]\!] =$

$\mathcal{D}[\![\mathsf{None}]\!] = \mathcal{D}[\![\mathsf{None}]\!] \cup \emptyset = \mathcal{D}[\![\mathsf{None}]\!] \cup \mathcal{D}[\![\mathsf{None}]\!] = \mathcal{D}[\![\mathrm{R}_L(I_1)]\!] \cup \mathcal{D}[\![\mathrm{R}_L(I_2)]\!] =$ (wie in Fall 1) $\ldots = \mathcal{D}[\![\mathsf{alt}(I_1, I_2)]\!][L]$. $\square$

3.3 Globale operationale Semantik

Das Regelsystem Gu der globalen operationalen Semantik ist in Tafel 3.1 angegeben. Die Metavariablen der Regelschemata von Gu haben die folgenden Bereiche: $I \in \mathrm{IFrag}$, $a \in \mathbb{A}$ und $\bar{a} \in \mathbb{A}_\tau$. Anmerkung: Das "u" im Symbol Gu steht für "unnummeriert". In Abschnitt 4.3 wird eine zweite, nummerierte Fassung der globalen operationalen Semantik angeben, die von Gu zu unterscheiden ist.

Die Ableitungsbäume des Regelsystems Gu sind linear gebaut, das heißt sie haben jeweils nur einen Blattknoten B, der einer Axiomeninstanz $T \xrightarrow{\bar{a}}_{\mathrm{Gu}} T'$ entspricht. Je nachdem, welches Axiomenschema für diesen Blattknoten B instantiiert wurde, ist die Art des abgeleiteten semantischen Schrittes zu unterscheiden:

1. Wurde B mit ($\mathrm{basic}_{\mathrm{Gu}}$) instantiiert, so sprechen wir von der Ausführung einer *echten Aktion* $\bar{a} = a$.
2. Wurde B mit ($\mathrm{alt}^1_{\mathrm{Gu}}$), ($\mathrm{alt}^2_{\mathrm{Gu}}$), ($\mathrm{loop}^1_{\mathrm{Gu}}$) oder ($\mathrm{loop}^2_{\mathrm{Gu}}$) instantiiert, dann sprechen wir von einem *Entscheidungsschritt*. Im Falle von ($\mathrm{loop}^1_{\mathrm{Gu}}$) heißt dieser Entscheidungsschritt auch *Schleifenabbruch*. Im Falle von ($\mathrm{loop}^2_{\mathrm{Gu}}$) sprechen wir von einem *Ausrollschritt*.
3. Wurde B mit ($\mathrm{strict}^2_{\mathrm{Gu}}$) instantiiert, so sprechen wir von einem *Synchronisationsschritt*.
4. Wurde B mit ($\mathrm{seq}^2_{\mathrm{Gu}}$), ($\mathrm{par}^3_{\mathrm{Gu}}$) oder ($\mathrm{par}^4_{\mathrm{Gu}}$) instantiiert, dann sprechen wir von einem *Aufräumschritt*.

3.4 Korrektheit und Vollständigkeit

Das verallgemeinerte Judgement $I \xrightarrow{\bar{t}}_{\mathrm{Gu}} I'$ ist für alle Interaktionen $I, I' \in \mathrm{IFrag}$ und alle endlichen Traces $\bar{t} \in \mathbb{T}_\tau$ als folgende Aussage definiert: Es gibt ein $n \in \mathbb{N}_0$ und Aktionen $\bar{a}_1, \ldots, \bar{a}_n \in \mathbb{A}_\tau$ und Interaktionen $I_0, \ldots, I_n \in \mathrm{IFrag}$, so dass gilt $\bar{t} = \bar{a}_1 \,;\, \ldots \,;\, \bar{a}_n$ und $I_0 = I$ und $I_n = I'$ und $\forall\, i \in \{1, \ldots, n\}.\, I_{i-1} \xrightarrow{\bar{a}_i}_{\mathrm{Gu}} I_i$. Dabei ist die Konkatenation von $n = 0$ Aktionen als ε definiert. Für jede Trace $\bar{t} \in \mathbb{T}_\tau$ sei $\lfloor \bar{t} \rfloor$ als diejenige Trace aus $\mathbb{T}$ definiert, die man erhält, wenn man aus $\bar{t}$ jedes Vorkommen einer stillen Aktion τ entfernt. Wir definieren

$$\mathcal{G}_\tau[\![-]\!] : \mathrm{IFrag} \to \wp(\mathbb{T}_\tau),\ \ \mathcal{G}_\tau[\![I]\!] := \{\bar{t} \in \mathbb{T}_\tau \mid I \xrightarrow{\bar{t}}_{\mathrm{Gu}} \mathsf{Empty}\} \text{ und}$$
$$\mathcal{G}[\![-]\!] : \mathrm{IFrag} \to \wp(\mathbb{T}),\ \ \mathcal{G}[\![I]\!] := \{\lfloor \bar{t} \rfloor \mid \bar{t} \in \mathcal{G}_\tau[\![I]\!]\}.$$

(basic$_{\text{Gu}}$)	$a \xrightarrow{a}_{\text{Gu}} \mathsf{Empty}$		
(strict$^1_{\text{Gu}}$)	$\dfrac{I_1 \xrightarrow{\bar a}_{\text{Gu}} I_1'}{\mathsf{strict}(I_1, I_2) \xrightarrow{\bar a}_{\text{Gu}} \mathsf{strict}(I_1', I_2)}$	(strict$^2_{\text{Gu}}$)	$\mathsf{strict}(\mathsf{Empty}, I_2) \xrightarrow{\tau}_{\text{Gu}} I_2$
(seq$^1_{\text{Gu}}$)	$\dfrac{I_1 \xrightarrow{\bar a}_{\text{Gu}} I_1'}{\mathsf{seq}(I_1, I_2) \xrightarrow{\bar a}_{\text{Gu}} \mathsf{seq}(I_1', I_2)}$	(seq$^2_{\text{Gu}}$)	$\mathsf{seq}(\mathsf{Empty}, I_2) \xrightarrow{\tau}_{\text{Gu}} I_2$
(seq$^3_{\text{Gu}}$)	$\dfrac{I_2 \xrightarrow{\bar a}_{\text{Gu}} I_2'}{\mathsf{seq}(I_1, I_2) \xrightarrow{\bar a}_{\text{Gu}} \mathsf{seq}(\mathrm{R}_{\alpha(\bar a)}(I_1), I_2')}$	falls $\mathrm{R}_{\alpha(\bar a)}(I_1) \neq \mathsf{None}$	
(par$^1_{\text{Gu}}$)	$\dfrac{I_1 \xrightarrow{\bar a}_{\text{Gu}} I_1'}{\mathsf{par}(I_1, I_2) \xrightarrow{\bar a}_{\text{Gu}} \mathsf{par}(I_1', I_2)}$	(par$^2_{\text{Gu}}$)	$\dfrac{I_2 \xrightarrow{\bar a}_{\text{Gu}} I_2'}{\mathsf{par}(I_1, I_2) \xrightarrow{\bar a}_{\text{Gu}} \mathsf{par}(I_1, I_2')}$
(par$^3_{\text{Gu}}$)	$\mathsf{par}(\mathsf{Empty}, I_2) \xrightarrow{\tau}_{\text{Gu}} I_2$	(par$^4_{\text{Gu}}$)	$\mathsf{par}(I_1, \mathsf{Empty}) \xrightarrow{\tau}_{\text{Gu}} I_1$
(loop$^1_{\text{Gu}}$)	$\mathsf{loop}(I) \xrightarrow{\tau}_{\text{Gu}} \mathsf{Empty}$	(loop$^2_{\text{Gu}}$)	$\mathsf{loop}(I) \xrightarrow{\tau}_{\text{Gu}} \mathsf{seq}(I, \mathsf{loop}(I))$
(alt$^1_{\text{Gu}}$)	$\mathsf{alt}(I_1, I_2) \xrightarrow{\tau}_{\text{Gu}} I_1$	(alt$^2_{\text{Gu}}$)	$\mathsf{alt}(I_1, I_2) \xrightarrow{\tau}_{\text{Gu}} I_2$

Tafel 3.1: Globale operationale Semantik

Im verbleibenden Teil dieses Abschnitts beweisen wir die Äquivalenz der globalen operationalen Semantik aus Abschnitt 3.3 und der denotationellen Semantik aus Abschnitt 2.4, d. h. wir zeigen $\mathcal{G}[\![I]\!] = \mathcal{D}[\![I]\!]$ für alle $I \in \text{IFrag}$.

Lemma 3.2 Seien $I, I' \in \text{IFrag}$ und $a \in \mathbb{A}$. Es gilt:

1. Falls $I \xrightarrow{a}_{\text{Gu}} I'$, dann $\mathcal{D}[\![I']\!] \subseteq \mathcal{D}[\![I]\!] / a$.
2. Falls $I \xrightarrow{\tau}_{\text{Gu}} I'$, dann $\mathcal{D}[\![I']\!] \subseteq \mathcal{D}[\![I]\!]$.

Beweis. Der Beweis erfolgt durch Ableitungsinduktion für Judgements $I \xrightarrow{\bar a}_{\text{Gu}} I'$:

(basic$_{\text{Gu}}$) Dann ist $I = \bar a = a \in \mathbb{A}$ und $I' = \mathsf{Empty}$. Es folgt $\mathcal{D}[\![I']\!] = \mathcal{D}[\![\mathsf{Empty}]\!] = \{\varepsilon\} = \{a\} \,/\, a = \mathcal{D}[\![a]\!] / a = \mathcal{D}[\![I]\!] / a$.

(strict$^1_{\text{Gu}}$) Dann ist $I = \mathsf{strict}(I_1, I_2)$ und $I' = \mathsf{strict}(I_1', I_2)$ und $I_1 \xrightarrow{\bar a}_{\text{Gu}} I_1'$.
Fall 1: $\bar a = a \in \mathbb{A}$: Nach I. V. gilt $\mathcal{D}[\![I_1']\!] \subseteq \mathcal{D}[\![I_1]\!] / a$. Mit Lemma 2.5.1 folgt: $\mathcal{D}[\![I']\!] = \mathcal{D}[\![\mathsf{strict}(I_1', I_2)]\!] = \mathcal{D}[\![I_1']\!] \,;\, \mathcal{D}[\![I_2]\!] \subseteq (\mathcal{D}[\![I_1]\!] / a) \,;\, \mathcal{D}[\![I_2]\!]$. Mit Lemma 2.3.1 folgt: $(\mathcal{D}[\![I_1]\!] / a) \,;\, \mathcal{D}[\![I_2]\!] \subseteq (\mathcal{D}[\![I_1]\!] \,;\, \mathcal{D}[\![I_2]\!]) \,/\, a = \mathcal{D}[\![\mathsf{strict}(I_1, I_2)]\!] / a = \mathcal{D}[\![I]\!] / a$.
Fall 2: $\bar a = \tau$: Nach I. V. gilt $\mathcal{D}[\![I_1']\!] \subseteq \mathcal{D}[\![I_1]\!]$. Mit Lemma 2.5.1 folgt: $\mathcal{D}[\![I']\!] = \mathcal{D}[\![I_1']\!] \,;\, \mathcal{D}[\![I_2]\!] \subseteq \mathcal{D}[\![I_1]\!] \,;\, \mathcal{D}[\![I_2]\!] = \mathcal{D}[\![I]\!]$.

(strict$^2_{\text{Gu}}$) Dann ist $I = \mathsf{strict}(\mathsf{Empty}, I_2)$ und $I' = I_2$ und $\bar a = \tau$. Somit gilt $\mathcal{D}[\![I']\!] = \mathcal{D}[\![I_2]\!] = \{\varepsilon\} \,;\, \mathcal{D}[\![I_2]\!] = \mathcal{D}[\![\mathsf{Empty}]\!] \,;\, \mathcal{D}[\![I_2]\!] = \mathcal{D}[\![\mathsf{strict}(\mathsf{Empty}, I_2)]\!] = \mathcal{D}[\![I]\!]$.

(seq$^1_{\text{Gu}}$) Dann ist $I = \mathsf{seq}(I_1, I_2)$ und $I' = \mathsf{seq}(I_1', I_2)$ und $I_1 \xrightarrow{\bar a}_{\text{Gu}} I_1'$.

Fall 1: $\bar{a} = a \in \mathbb{A}$: Nach I. V. gilt $\mathcal{D}[\![I_1']\!] \subseteq \mathcal{D}[\![I_1]\!]\,/a$. Mit Lemma 2.5.2 folgt: $\mathcal{D}[\![I']\!] = \mathcal{D}[\![\mathsf{seq}(I_1', I_2)]\!] = (\mathcal{D}[\![I_1']\!] \;;_{\between} \mathcal{D}[\![I_2]\!])\!\downarrow \subseteq ((\mathcal{D}[\![I_1]\!]\,/a) \;;_{\between} \mathcal{D}[\![I_2]\!])\!\downarrow$. Mit Hilfe von Lemma 2.3.2 folgt $((\mathcal{D}[\![I_1]\!]\,/a) \;;_{\between} \mathcal{D}[\![I_2]\!])\!\downarrow \;\subseteq\; (\mathcal{D}[\![I_1]\!] \;;_{\between} \mathcal{D}[\![I_2]\!])\!\downarrow \,/\, a = \mathcal{D}[\![\mathsf{seq}(I_1, I_2)]\!]\,/a$.

Fall 2: $\bar{a} = \tau$: Nach I. V. gilt $\mathcal{D}[\![I_1']\!] \subseteq \mathcal{D}[\![I_1]\!]$. Mit Lemma 2.5.2 folgt: $\mathcal{D}[\![I']\!] = (\mathcal{D}[\![I_1']\!] \;;_{\between} \mathcal{D}[\![I_2]\!])\!\downarrow \subseteq (\mathcal{D}[\![I_1]\!] \;;_{\between} \mathcal{D}[\![I_2]\!])\!\downarrow = \mathcal{D}[\![I]\!]$.

($\mathrm{seq}^2_{\mathrm{Gu}}$) Analog zu Fall ($\mathrm{strict}^2_{\mathrm{Gu}}$).

($\mathrm{seq}^3_{\mathrm{Gu}}$) Dann gilt $I = \mathsf{seq}(I_1, I_2)$ und $I' = \mathsf{seq}(\mathrm{R}_{\alpha(\bar{a})}(I_1), I_2')$ und $I_2 \xrightarrow{\bar{a}}_{\mathrm{Gu}} I_2'$ und $\mathrm{R}_{\alpha(\bar{a})}(I_1) \neq \mathsf{None}$.

Fall 1: $\bar{a} = a \in \mathbb{A}$: Nach I. V. gilt $\mathcal{D}[\![I_2']\!] \subseteq \mathcal{D}[\![I_2]\!]\,/a$. Somit gilt $\mathcal{D}[\![I']\!] = \mathcal{D}[\![\mathsf{seq}(\mathrm{R}_{\alpha(a)}(I_1), I_2')]\!] = (\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] \;;_{\between} \mathcal{D}[\![I_2']\!])\!\downarrow \subseteq$ (mit I. V. und Lemma 2.5.2) $(\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] \;;_{\between} (\mathcal{D}[\![I_2]\!]\,/a))\!\downarrow$. Nach Lemma 3.1 gilt $\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] = \mathcal{D}[\![I_1]\!][\alpha(a)]$. Es folgt $\mathcal{D}[\![I']\!] \subseteq (\mathcal{D}[\![I_1]\!][\alpha(a)] \;;_{\between} (\mathcal{D}[\![I_2]\!]\,/a))\!\downarrow$. Mit Lemma 2.3.2 folgt $(\mathcal{D}[\![I_1]\!][\alpha(a)] \;;_{\between} (\mathcal{D}[\![I_2]\!]\,/a))\!\downarrow \subseteq (\mathcal{D}[\![I_1]\!] \;;_{\between} \mathcal{D}[\![I_2]\!])\!\downarrow \,/\, a = \mathcal{D}[\![\mathsf{seq}(I_1, I_2)]\!]\,/a = \mathcal{D}[\![I]\!]\,/a$.

Fall 2: $\bar{a} = \tau$: Dann ist $\alpha(\bar{a}) = \emptyset$ und somit $\mathrm{R}_{\alpha(\bar{a})}(I_1) = I_1$. Aus der I. V. folgt $\mathcal{D}[\![I_2']\!] \subseteq \mathcal{D}[\![I_2]\!]$. Mit Lemma 2.5.2 folgt: $\mathcal{D}[\![I']\!] = (\mathcal{D}[\![\mathrm{R}_{\alpha(\bar{a})}(I_1)]\!] \;;_{\between} \mathcal{D}[\![I_2']\!])\!\downarrow = (\mathcal{D}[\![I_1]\!] \;;_{\between} \mathcal{D}[\![I_2']\!])\!\downarrow \subseteq (\mathcal{D}[\![I_1]\!] \;;_{\between} \mathcal{D}[\![I_2]\!])\!\downarrow = \mathcal{D}[\![I]\!]$.

($\mathrm{par}^1_{\mathrm{Gu}}$) Analog zu Fall ($\mathrm{seq}^1_{\mathrm{Gu}}$) mit Lemma 2.5.3 und 2.3.3.

($\mathrm{par}^2_{\mathrm{Gu}}$) Analog zu Fall ($\mathrm{seq}^3_{\mathrm{Gu}}$) mit Lemma 2.5.3 und 2.3.3—ohne die zusätzliche Schwierigkeit mit der Restriktionsfunktion.

($\mathrm{par}^3_{\mathrm{Gu}}$) und ($\mathrm{par}^4_{\mathrm{Gu}}$) Analog zu Fall ($\mathrm{strict}^2_{\mathrm{Gu}}$).

($\mathrm{loop}^1_{\mathrm{Gu}}$) Es gilt $I = \mathsf{loop}(I_1)$, $I' = \mathsf{Empty}$ und $\bar{a} = \tau$. Es folgt: $\mathcal{D}[\![I']\!] = \mathcal{D}[\![\mathsf{Empty}]\!] = \{\varepsilon\} = \{\varepsilon\}\!\downarrow = ((\mathcal{D}[\![I_1]\!])^{0\between})\!\downarrow \subseteq \bigcup_{n\in\mathbb{N}_0}((\mathcal{D}[\![I_1]\!])^{n\between})\!\downarrow = \mathcal{D}[\![\mathsf{loop}(I_1)]\!] = \mathcal{D}[\![I]\!]$.

($\mathrm{loop}^2_{\mathrm{Gu}}$) Es gilt $I = \mathsf{loop}(I_1)$, $I' = \mathsf{seq}(I_1, \mathsf{loop}(I_1))$ und $\bar{a} = \tau$. Es folgt: $\mathcal{D}[\![I']\!] = \mathcal{D}[\![\mathsf{seq}(I_1, \mathsf{loop}(I_1))]\!] = (\mathcal{D}[\![I_1]\!] \;;_{\between} \mathcal{D}[\![\mathsf{loop}(I_1)]\!])\!\downarrow = (\mathcal{D}[\![I_1]\!] \;;_{\between} \bigcup_{n\in\mathbb{N}_0}((\mathcal{D}[\![I_1]\!])^{n\between})\!\downarrow)\!\downarrow = \bigcup_{n\in\mathbb{N}_0}(\mathcal{D}[\![I_1]\!] \;;_{\between} ((\mathcal{D}[\![I_1]\!])^{n\between})\!\downarrow)\!\downarrow = \bigcup_{n\in\mathbb{N}_0}(\mathcal{D}[\![I_1]\!] \;;_{\between} (\mathcal{D}[\![I_1]\!])^{n\between})\!\downarrow$, wobei im letzten Schritt das Lemma 2.2.2 angewendet worden ist. Folglich gilt $\mathcal{D}[\![I']\!] = \bigcup_{n\in\mathbb{N}_0}((\mathcal{D}[\![I_1]\!])^{(n+1)\between})\!\downarrow = \bigcup_{n\geq 1}((\mathcal{D}[\![I_1]\!])^{n\between})\!\downarrow \subseteq \bigcup_{n\in\mathbb{N}_0}((\mathcal{D}[\![I_1]\!])^{n\between})\!\downarrow = \mathcal{D}[\![\mathsf{loop}(I_1)]\!] = \mathcal{D}[\![I]\!]$.

($\mathrm{alt}^1_{\mathrm{Gu}}$) und ($\mathrm{alt}^2_{\mathrm{Gu}}$) Trivial. □

Satz 3.3 Sei $I \in \mathrm{IFrag}$ und $\bar{t} \in \mathbb{T}_\tau$. Es gilt: $I \xrightarrow{\bar{t}}_{\mathrm{Gu}} \mathsf{Empty} \implies \lfloor \bar{t} \rfloor \in \mathcal{D}[\![I]\!]$.

Der Satz 3.3 besagt, dass das Regelsystem Gu der globalen operationalen Semantik bezüglich der denotationellen Semantik korrekt ist. Es gilt $\mathcal{G}[\![I]\!] \subseteq \mathcal{D}[\![I]\!]$ für jedes $I \in \mathrm{IFrag}$.

Beweis. Wir führen den Beweis durch vollständige Induktion nach der Länge der Trace t. Anmerkung: Die Länge $|t|$ einer endlichen Trace t ist die wohlbestimmte Mächtigkeit der Grundmenge eines beliebigen Repräsentanten von t.

$|\bar{t}| = 0$: Dann ist $I = \mathsf{Empty}$ und $\lfloor \bar{t} \rfloor = \bar{t} = \varepsilon \in \mathcal{D}[\![\mathsf{Empty}]\!] = \mathcal{D}[\![I]\!]$.

$|\bar{t}| \geq 1$: Dann ist $\bar{t} = \bar{a}\,;\bar{t}'$ mit $\bar{a} \in \mathbb{A}_\tau$, $\bar{t}' \in \mathbb{T}_\tau$ und $|\bar{t}'| < |\bar{t}|$. Nach I. V. für $|\bar{t}'|$ gilt (1) $\forall\, I' \in \mathrm{IFrag}\,.\,(I' \xrightarrow{\bar{t}'}_{\mathrm{Gu}} \mathsf{Empty} \Rightarrow \lfloor \bar{t}' \rfloor \in \mathcal{D}[\![I']\!])$. Aus $I \xrightarrow{\bar{t}}_{\mathrm{Gu}} \mathsf{Empty}$ und $\bar{t} = \bar{a}\,;\bar{t}'$ folgt (2) $I \xrightarrow{\bar{a}}_{\mathrm{Gu}} I'$ und (3) $I' \xrightarrow{\bar{t}'}_{\mathrm{Gu}} \mathsf{Empty}$ mit einem $I' \in \mathrm{IFrag}$. Aus (3) und (1) folgt $\lfloor \bar{t}' \rfloor \in \mathcal{D}[\![I']\!]$. Auf die Aussage (2) wenden wir das Lemma 3.2 an:
Fall 1: $\bar{a} = a \in \mathbb{A}$: Dann gilt $\mathcal{D}[\![I']\!] \subseteq \mathcal{D}[\![I]\!]\,/a$. Es folgt $\lfloor \bar{t}' \rfloor \in \mathcal{D}[\![I]\!]\,/a$ und somit $\lfloor \bar{t} \rfloor = \lfloor a\,;\bar{t}' \rfloor = a\,;\lfloor \bar{t}' \rfloor \in \mathcal{D}[\![I]\!]$.
Fall 2: $\bar{a} = \tau$: Dann gilt $\mathcal{D}[\![I']\!] \subseteq \mathcal{D}[\![I]\!]$. Es folgt $\lfloor \bar{t}' \rfloor \in \mathcal{D}[\![I]\!]$. Somit gilt $\lfloor \bar{t} \rfloor = \lfloor \tau\,;\bar{t}' \rfloor = \lfloor \bar{t}' \rfloor \in \mathcal{D}[\![I]\!]$. □

Die Notation $I \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$ bedeute, dass ein $n \in \mathbb{N}_0$ existiert mit $I \xrightarrow{\tau^n}_{\mathrm{Gu}} \mathsf{Empty}$. Hierbei ist definiert $\tau^0 := \varepsilon$ und $\tau^{n+1} := \tau\,;\tau^n$ für jedes $n \in \mathbb{N}_0$.

Lemma 3.4 Seien $I_1, I_2 \in \mathrm{IFrag}$ und $bop \in \{\mathsf{strict}, \mathsf{seq}, \mathsf{par}\}$. Dann gilt:

$$bop(I_1, I_2) \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty} \iff I_1 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty} \wedge I_2 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}.$$

Beweis. "$\Leftarrow$" Aus $I_1 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$ und Regel ($\mathrm{strict}^1_{\mathrm{Gu}}$) bzw. ($\mathrm{seq}^1_{\mathrm{Gu}}$) bzw. ($\mathrm{par}^1_{\mathrm{Gu}}$) folgt $bop(I_1, I_2) \xrightarrow{\tau^*}_{\mathrm{Gu}} bop(\mathsf{Empty}, I_2)$. Mit Axiom ($\mathrm{strict}^2_{\mathrm{Gu}}$) bzw. ($\mathrm{seq}^2_{\mathrm{Gu}}$) bzw. ($\mathrm{par}^3_{\mathrm{Gu}}$) folgt $bop(\mathsf{Empty}, I_2) \xrightarrow{\tau}_{\mathrm{Gu}} I_2$. Mit $I_2 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$ folgt die Behauptung.
"$\Rightarrow$" In dieser Richtung kann der Beweis leicht durch vollständige Induktion über die Anzahl n der τ-Schritte im Judgement $bop(I_1, I_2) \xrightarrow{\tau^n}_{\mathrm{Gu}} \mathsf{Empty}$ geführt werden. Wir überlassen diese Induktion dem Leser. □

Die Notation $I \xrightarrow{\tau^*\,;\,a\,;\,\tau^*}_{\mathrm{Gu}} I'$ bedeute, dass $n, m \in \mathbb{N}_0$ existieren mit $I \xrightarrow{\tau^n\,;\,a\,;\,\tau^m}_{\mathrm{Gu}} I'$.

Lemma 3.5 Sei $I \in \mathrm{IFrag}$ und $a \in \mathbb{A}$. Es gilt:

1. $\mathcal{D}[\![I]\!]\,/a = \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^*\,;\,a\,;\,\tau^*}_{\mathrm{Gu}} I'\}$.
2. $\varepsilon \in \mathcal{D}[\![I]\!] \iff I \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$.

Beweis. Wir beginnen mit dem Beweis der zweiten Teilaussage.
Ad 2. Der Beweis erfolgt durch strukturelle Induktion über $I \in \mathrm{IFrag}$:

$I = \mathsf{Empty}$: Es gilt $\varepsilon \in \{\varepsilon\} = \mathcal{D}[\![I]\!]$ und $I = \mathsf{Empty} \xrightarrow{\tau^0}_{\mathrm{Gu}} \mathsf{Empty}$.

$I = a' \in \mathbb{A}$: Es gilt $\varepsilon \notin \{a'\} = \mathcal{D}[\![I]\!]$. Für $I = a'$ sind im Regelsystem Gu keine τ-Übergange ableitbar.

$I = \mathsf{strict}(I_1, I_2)$: Es gilt $\mathcal{D}[\![I]\!] = \mathcal{D}[\![I_1]\!] \mathbin{;} \mathcal{D}[\![I_2]\!]$. Nach Lemma 2.4.1 gilt $\varepsilon \in \mathcal{D}[\![I]\!]$ genau dann, wenn gilt $\varepsilon \in \mathcal{D}[\![I_1]\!]$ und $\varepsilon \in \mathcal{D}[\![I_2]\!]$. Letzteres gilt nach I. V. genau dann, wenn gilt $I_1 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$ und $I_2 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$. Letzteres gilt nach Lemma 3.4 genau dann, wenn gilt $I = \mathsf{strict}(I_1, I_2) \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$.

$I = \mathsf{seq}(I_1, I_2)$: Analog zu Fall $I = \mathsf{strict}(I_1, I_2)$ mit Lemma 2.4.2, 2.4.5 und 3.4.

$I = \mathsf{par}(I_1, I_2)$: Analog zu Fall $I = \mathsf{strict}(I_1, I_2)$ mit Lemma 2.4.3, 2.4.5 und 3.4.

$I = \mathsf{loop}(I_1)$: Es gilt $\varepsilon \in \{\varepsilon\} = \{\varepsilon\}\!\downarrow = ((\mathcal{D}[\![I_1]\!])^{0\lessgtr})\!\downarrow \subseteq \bigcup_{n\in\mathbb{N}_0}((\mathcal{D}[\![I_1]\!])^{n\lessgtr})\!\downarrow = \mathcal{D}[\![\mathsf{loop}(I_1)]\!] = \mathcal{D}[\![I]\!]$. Nach Axiom ($\mathrm{loop}^1_{\mathrm{Gu}}$) gilt $I = \mathsf{loop}(I_1) \xrightarrow{\tau}_{\mathrm{Gu}} \mathsf{Empty}$.

$I = \mathsf{alt}(I_1, I_2)$: Es gilt $\mathcal{D}[\![I]\!] = \mathcal{D}[\![I_1]\!] \cup \mathcal{D}[\![I_2]\!]$. Somit gilt $\varepsilon \in \mathcal{D}[\![I]\!]$ genau dann, wenn $\varepsilon \in \mathcal{D}[\![I_1]\!]$ oder $\varepsilon \in \mathcal{D}[\![I_2]\!]$ gilt. Letzteres gilt nach I. V. genau dann, wenn $I_1 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$ oder $I_2 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$ gilt. Aufgrund der Axiome ($\mathrm{alt}^1_{\mathrm{Gu}}$) und ($\mathrm{alt}^2_{\mathrm{Gu}}$) gilt letzteres genau dann, wenn gilt $I = \mathsf{alt}(I_1, I_2) \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$.

Ad 1. Aufgrund von Lemma 3.2 genügt es zu zeigen:

$$\mathcal{D}[\![I]\!] \,/ a \subseteq \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^*;\, a\,;\tau^*}_{\mathrm{Gu}} I'\}$$

Wir zeigen dies durch eine strukturelle Induktion über $I \in \mathrm{IFrag}$:

$I = \mathsf{Empty}$: Es gilt $\mathcal{D}[\![I]\!] \,/ a = \mathcal{D}[\![\mathsf{Empty}]\!] \,/ a = \{\varepsilon\} \,/\, a = \emptyset \subseteq M$ für jedes M.

$I = a' \in \mathbb{A}$: Falls $a' = a$ ist, gilt $\mathcal{D}[\![I]\!] \,/ a = \{a\}/a = \{\varepsilon\} = \mathcal{D}[\![\mathsf{Empty}]\!]$. Mit Axiom ($\mathrm{basic}_{\mathrm{Gu}}$) folgt $I = a \xrightarrow{a}_{\mathrm{Gu}} \mathsf{Empty}$. Falls $a' \neq a$ ist, gilt $\mathcal{D}[\![I]\!] \,/ a = \{a'\}/a = \emptyset \subseteq M$ für jedes M.

$I = \mathsf{strict}(I_1, I_2)$: Dann gilt $\mathcal{D}[\![I]\!] \,/ a = (\mathcal{D}[\![I_1]\!] \mathbin{;} \mathcal{D}[\![I_2]\!]) \,/\, a =$ (mit Lemma 2.3.1) $((\mathcal{D}[\![I_1]\!] \,/ a) \mathbin{;} \mathcal{D}[\![I_2]\!]) \cup ((\mathcal{D}[\![I_1]\!] \cap \{\varepsilon\}) \mathbin{;} (\mathcal{D}[\![I_2]\!] \,/ a))$.

Z. z.: $(\mathcal{D}[\![I_1]\!] \,/ a) \mathbin{;} \mathcal{D}[\![I_2]\!] \subseteq \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^*;\, a\,;\tau^*}_{\mathrm{Gu}} I'\}$

Mit der I. V. für I_1 folgt $\mathcal{D}[\![I_1]\!] \,/ a \subseteq \bigcup \{\mathcal{D}[\![I_1']\!] \mid I_1 \xrightarrow{\tau^*;\, a\,;\tau^*}_{\mathrm{Gu}} I_1'\}$. Mittels Lemma 2.5.1 folgt $(\mathcal{D}[\![I_1]\!] \,/ a) \mathbin{;} \mathcal{D}[\![I_2]\!] \subseteq (\bigcup \{\mathcal{D}[\![I_1']\!] \mid I_1 \xrightarrow{\tau^*;\, a\,;\tau^*}_{\mathrm{Gu}} I_1'\}) \mathbin{;} \mathcal{D}[\![I_2]\!]$ $= \bigcup \{\mathcal{D}[\![I_1']\!] \mathbin{;} \mathcal{D}[\![I_2]\!] \mid I_1 \xrightarrow{\tau^*;\, a\,;\tau^*}_{\mathrm{Gu}} I_1'\} \subseteq \bigcup \{\mathcal{D}[\![\mathsf{strict}(I_1', I_2)]\!] \mid \mathsf{strict}(I_1, I_2) \xrightarrow{\tau^*;\, a\,;\tau^*}_{\mathrm{Gu}} \mathsf{strict}(I_1', I_2)\} \subseteq \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^*;\, a\,;\tau^*}_{\mathrm{Gu}} I'\}$, wobei im vorletzten Schritt die Regel ($\mathrm{strict}^1_{\mathrm{Gu}}$) angewendet wurde.

Z. z.: $(\mathcal{D}[\![I_1]\!] \cap \{\varepsilon\}) \mathbin{;} (\mathcal{D}[\![I_2]\!] \,/ a) \subseteq \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^*;\, a\,;\tau^*}_{\mathrm{Gu}} I'\}$

Sei o. B. d. A. $\mathcal{D}[\![I_1]\!] \cap \{\varepsilon\} \neq \emptyset$, d. h. es gilt $\varepsilon \in \mathcal{D}[\![I_1]\!]$. Aufgrund der bereits bewiesenen Teilaussage 2 von Lemma 3.5 folgt $I_1 \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{Empty}$. Mit Regel ($\mathrm{strict}^1_{\mathrm{Gu}}$) folgt $\mathsf{strict}(I_1, I_2) \xrightarrow{\tau^*}_{\mathrm{Gu}} \mathsf{strict}(\mathsf{Empty}, I_2)$. Mit dem Axiom ($\mathrm{strict}^2_{\mathrm{Gu}}$) erhält man $\mathsf{strict}(I_1, I_2) \xrightarrow{\tau^*}_{\mathrm{Gu}} I_2$.

Mit der I. V. für I_2 folgt $\mathcal{D}[\![I_2]\!] \,/ a \subseteq \bigcup\{\mathcal{D}[\![I_2']\!] \mid I_2 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2' \}$. Daraus folgt: $(\mathcal{D}[\![I_1]\!] \cap \{\varepsilon\}) \,; (\mathcal{D}[\![I_2]\!] \,/ a) = \{\varepsilon\} \,; (\mathcal{D}[\![I_2]\!] \,/ a) = \mathcal{D}[\![I_2]\!] \,/ a \subseteq \bigcup \{\mathcal{D}[\![I_2']\!] \mid I_2 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2' \}$ $\subseteq \{\mathcal{D}[\![I_2']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2' \}$.

$I = \mathsf{seq}(I_1, I_2)$: Dann gilt $\mathcal{D}[\![I]\!] \,/ a = (\mathcal{D}[\![I_1]\!] \;;_{\gtrless}\; \mathcal{D}[\![I_2]\!])\!\downarrow / \; a =$ (mit Lemma 2.3.2) $((\mathcal{D}[\![I_1]\!] \,/ a) \;;_{\gtrless}\; \mathcal{D}[\![I_2]\!])\!\downarrow \cup\, (\mathcal{D}[\![I_1]\!][\alpha(a)] \;;_{\gtrless}\; (\mathcal{D}[\![I_2]\!] \,/ a))\!\downarrow$.

Z. z.: $((\mathcal{D}[\![I_1]\!] \,/ a) \;;_{\gtrless}\; \mathcal{D}[\![I_2]\!])\!\downarrow \subseteq \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I' \}$

Dies zeigt man analog zu Fall $I = \mathsf{strict}(I_1, I_2)$, wobei die Lemmata 2.5.2 und 2.5.7 zu verwenden sind.

Z. z.: $(\mathcal{D}[\![I_1]\!][\alpha(a)] \;;_{\gtrless}\; (\mathcal{D}[\![I_2]\!] \,/ a))\!\downarrow \subseteq \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I' \}$

Sei o. B. d. A. $\mathcal{D}[\![I_1]\!][\alpha(a)] \neq \emptyset$. Nach Lemma 3.1 gilt $\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] = \mathcal{D}[\![I_1]\!][\alpha(a)]$. Aufgrund der Voraussetung $\mathcal{D}[\![I_1]\!][\alpha(a)] \neq \emptyset$ folgt $\mathrm{R}_{\alpha(a)}(I_1) \neq \mathsf{None}$. Mit der I. V. für I_2 folgt $\mathcal{D}[\![I_2]\!] \,/ a \subseteq \bigcup\{\mathcal{D}[\![I_2']\!] \mid I_2 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2'\}$. Mit Lemma 2.5.2/2.5.7 folgt $(\mathcal{D}[\![I_1]\!][\alpha(a)] \;;_{\gtrless}\; (\mathcal{D}[\![I_2]\!] \,/ a))\!\downarrow \subseteq (\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] \;;_{\gtrless}\; \bigcup\{\mathcal{D}[\![I_2']\!] \mid I_2 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2'\})\!\downarrow =$ $\bigcup\{(\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] \;;_{\gtrless}\; \mathcal{D}[\![I_2']\!])\!\downarrow \mid I_2 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2'\}$. Vermöge der Regel $(\mathrm{seq}^3_{\mathrm{Gu}})$ und $\mathrm{R}_{\alpha(a)}(I_1) \neq \mathsf{None}$ und $\mathrm{R}_{\alpha(\tau)}(I_1) = I_1 \neq \mathsf{None}$ folgt: $\bigcup\{(\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] \;;_{\gtrless}\; \mathcal{D}[\![I_2']\!])\!\downarrow \mid$ $I_2 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2'\} \subseteq \bigcup\{\mathcal{D}[\![\mathsf{seq}(\mathrm{R}_{\alpha(a)}(I_1), I_2')]\!] \mid \mathsf{seq}(I_1, I_2) \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} \mathsf{seq}(\mathrm{R}_{\alpha(a)}(I_1),$ $I_2')\} \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I'\}$.

$I = \mathsf{par}(I_1, I_2)$: Analog zu Fall $I = \mathsf{seq}(I_1, I_2)$ mit Lemmata 2.3.3, 2.5.3 und 2.5.7.

$I = \mathsf{alt}(I_1, I_2)$: Dann gilt $\mathcal{D}[\![I]\!] \,/ a = (\mathcal{D}[\![I_1]\!] \cup \mathcal{D}[\![I_2]\!]) \,/\, a = (\mathcal{D}[\![I_1]\!] \,/ a) \cup (\mathcal{D}[\![I_2]\!] \,/ a)$ (I. V. für I_1 und I_2) $\subseteq \bigcup\{\mathcal{D}[\![I_1']\!] \mid I_1 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_1'\} \cup \bigcup\{\mathcal{D}[\![I_2']\!] \mid I_2 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2'\}$.
Mit Regel $(\mathrm{alt}^1_{\mathrm{Gu}})$ folgt $\bigcup\{\mathcal{D}[\![I_1']\!] \mid I_1 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_1'\} \subseteq \bigcup\{\mathcal{D}[\![I_1']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_1'\}$.
Mit Regel $(\mathrm{alt}^2_{\mathrm{Gu}})$ folgt $\bigcup\{\mathcal{D}[\![I_2']\!] \mid I_2 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2'\} \subseteq \bigcup\{\mathcal{D}[\![I_2']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_2'\}$.

$I = \mathsf{loop}(I_1)$: Es gilt $\mathcal{D}[\![I]\!] = \bigcup_{n \in \mathbb{N}_0} \mathcal{D}^n[\![I]\!]$ mit $\mathcal{D}^n[\![I]\!] := ((\mathcal{D}[\![I_1]\!])^{n\gtrless})\!\downarrow$. Wir zeigen $\forall n \in \mathbb{N}_0 \,.\, \mathcal{D}^n[\![I]\!] \,/ a \subseteq \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I' \}$ durch vollständige Induktion nach $n \in \mathbb{N}_0$. Hieraus folgt die Behauptung mittels Lemma 2.3.5 (analog).

$n = 0$: $\mathcal{D}^0[\![I]\!] \,/ a = ((\mathcal{D}[\![I_1]\!])^{0\gtrless})\!\downarrow / \; a = \{\varepsilon\}\!\downarrow / \; a = \{\varepsilon\} \,/\, a = \emptyset \subseteq M$ für jedes M.

$n \Rightarrow n{+}1$: Es gelte $\mathcal{D}^n[\![I]\!] \,/ a \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I' \}$. Es gilt: $\mathcal{D}^{n+1}[\![I]\!] \,/ a =$ $((\mathcal{D}[\![I_1]\!])^{(n+1)\gtrless})\!\downarrow / \; a = (\mathcal{D}[\![I_1]\!] \;;_{\gtrless}\; (\mathcal{D}[\![I_1]\!])^{n\gtrless})\!\downarrow / \; a = (\mathcal{D}[\![I_1]\!] \;;_{\gtrless}\; \mathcal{D}^n[\![I]\!])\!\downarrow / \; a$, wobei im letzten Schritt das Lemma 2.2.2 verwendet wurde. Mit Lemma 2.3.2 folgt: $\mathcal{D}^{n+1}[\![I]\!] \,/ a = ((\mathcal{D}[\![I_1]\!] \,/ a) \;;_{\gtrless}\; \mathcal{D}^n[\![I]\!])\!\downarrow \cup\, (\mathcal{D}[\![I_1]\!][\alpha(a)] \;;_{\gtrless}\; (\mathcal{D}^n[\![I]\!] \,/ a))\!\downarrow$.

Z. z.: $((\mathcal{D}[\![I_1]\!] \,/ a) \;;_{\gtrless}\; \mathcal{D}^n[\![I]\!])\!\downarrow \;\subseteq\; \bigcup \{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I' \}$

Mit I. V. für I_1 folgt $\mathcal{D}[\![I_1]\!] \,/ a \subseteq \bigcup\{\mathcal{D}[\![I_1']\!] \mid I_1 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_1' \}$. Mit Lemma 2.5.2 und 2.5.7 folgt $((\mathcal{D}[\![I_1]\!] \,/ a) \;;_{\gtrless}\; \mathcal{D}^n[\![I]\!])\!\downarrow \subseteq (\bigcup\{\mathcal{D}[\![I_1']\!] \mid I_1 \xrightarrow{\tau^* ; a ; \tau^*}_{\mathrm{Gu}} I_1' \} \;;_{\gtrless}\; \mathcal{D}[\![I]\!])\!\downarrow$

$= \bigcup\{(\mathcal{D}[\![I_1']\!] \,;_{\between}\, \mathcal{D}[\![I]\!])\!\downarrow \mid I_1 \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} I_1' \}$. Aufgrund von Regel ($\mathrm{seq}^1_{\mathrm{Gu}}$) ist die letztere Menge eine Teilmenge von $\bigcup\{\mathcal{D}[\![\mathsf{seq}(I_1', I)]\!] \mid \mathsf{seq}(I_1, I) \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} \mathsf{seq}(I_1', I)\} \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid \mathsf{seq}(I_1, I) \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} I'\}$. Mit Regel ($\mathrm{loop}^2_{\mathrm{Gu}}$) folgt die Behauptung.

Z. z.: $(\mathcal{D}[\![I_1]\!][\alpha(a)] \,;_{\between}\, (\mathcal{D}^n[\![I]\!]\,/a))\!\downarrow \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} I'\}$
Sei o. B. d. A. $\mathcal{D}[\![I_1]\!][\alpha(a)] \neq \emptyset$. Nach Lemma 3.1 gilt $\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] = \mathcal{D}[\![I_1]\!][\alpha(a)]$. Wegen der Voraussetung $\mathcal{D}[\![I_1]\!][\alpha(a)] \neq \emptyset$ gilt $\mathrm{R}_{\alpha(a)}(I_1) \neq \mathsf{None}$. Die I. V. für n lautet $\mathcal{D}^n[\![I]\!]\,/a \subseteq \bigcup\{\mathcal{D}[\![\tilde{I}]\!] \mid I \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} \tilde{I}\}$. Mit Lemma 2.5.2 und 2.5.7 folgt $(\mathcal{D}[\![I_1]\!][\alpha(a)] \,;_{\between}\, (\mathcal{D}^n[\![I]\!]\,/a))\!\downarrow \;\subseteq\; (\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] \,;_{\between}\, \bigcup\{\mathcal{D}[\![\tilde{I}]\!] \mid I \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} \tilde{I}\})\!\downarrow = \bigcup\{(\mathcal{D}[\![\mathrm{R}_{\alpha(a)}(I_1)]\!] \,;_{\between}\, \mathcal{D}[\![\tilde{I}]\!])\!\downarrow \mid I \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} \tilde{I}\}$. Aufgrund von Regel ($\mathrm{seq}^3_{\mathrm{Gu}}$) und wegen $\mathrm{R}_{\alpha(a)}(I_1) \neq \mathsf{None}$ und $\mathrm{R}_{\alpha(\tau)}(I_1) = I_1 \neq \mathsf{None}$ ist die letztere Menge eine Teilmenge von $\bigcup\{\mathcal{D}[\![\mathsf{seq}(\mathrm{R}_{\alpha(a)}(I_1), \tilde{I})]\!] \mid \mathsf{seq}(I_1, I) \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} \mathsf{seq}(\mathrm{R}_{\alpha(a)}(I_1), \tilde{I})\} \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid \mathsf{seq}(I_1, I) \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} I'\}$. Mit Regel ($\mathrm{loop}^2_{\mathrm{Gu}}$) folgt die Behauptung. □

Satz 3.6 Sei $I \in \mathrm{IFrag}$ und $t \in \mathbb{T}$ und t endlich. Falls $t \in \mathcal{D}[\![I]\!]$ ist, dann gibt es ein $\bar{t} \in \mathbb{T}_\tau$ mit $t = \lfloor \bar{t} \rfloor$ und $I \xrightarrow{\bar{t}}_{\mathrm{Gu}} \mathsf{Empty}$.

Bemerkung: 1. In Abschnitt 2.4 haben wir die Semantik des loop-Konstrukts so definiert, dass immer irgendwann ein Schleifenabbruch eintritt. Folglich sind alle Traces $t \in \mathcal{D}[\![I]\!]$ für alle $I \in \mathrm{IFrag}$ endlich.
2. Der Satz 3.6 besagt, dass das Regelsystem Gu der globalen operationalen Semantik bezügl. der denotationellen Semantik vollständig ist. Es gilt $\mathcal{D}[\![I]\!] \subseteq \mathcal{G}[\![I]\!]$ für jedes $I \in \mathrm{IFrag}$.

Beweis des Satzes. Wir führen den Beweis durch vollständige Induktion nach der Länge $n = |t|$ der Trace t.

$n = 0$: Dann ist $t = \varepsilon$ und somit $\varepsilon \in \mathcal{D}[\![I]\!]$. Nach Lemma 3.5.2 gilt $I \xrightarrow{\tau^k}_{\mathrm{Gu}} \mathsf{Empty}$ mit einem $k \in \mathbb{N}_0$ mit . Wir setzen $\bar{t} := \tau^k$.

$n \Rightarrow n+1$: Seien $I \in \mathrm{IFrag}$ und $t \in \mathcal{D}[\![I]\!]$ und $|t| = n+1$. Wegen $|t| = n+1$ gilt $t = a\,;t'$ mit $a \in \mathbb{A}$ und $t' \in \mathbb{T}$ und $|t'| = n$. Weil $a\,;t' = t \in \mathcal{D}[\![I]\!]$ ist, gilt $t' \in \mathcal{D}[\![I]\!]\,/a$. Nach Lemma 3.5.1 gilt $\mathcal{D}[\![I]\!]\,/a = \bigcup\{\mathcal{D}[\![I']\!] \mid I \xrightarrow{\tau^*;a;\tau^*}_{\mathrm{Gu}} I'\}$. Folglich gibt es $I' \in \mathrm{IFrag}$ und $k, l \in \mathbb{N}_0$ mit $I \xrightarrow{\tau^k;a;\tau^l}_{\mathrm{Gu}} I'$ und $t' \in \mathcal{D}[\![I']\!]$. Weil $|t'| = n$ ist, folgt mit der I. V., dass es ein $\bar{t}' \in \mathbb{T}_\tau$ gibt, so dass gilt $t' = \lfloor \bar{t}' \rfloor$ und $I' \xrightarrow{\bar{t}'}_{\mathrm{Gu}} \mathsf{Empty}$. Setze $\bar{t} := \tau^k\,;a\,;\tau^l\,;\bar{t}'$. Dann gilt $\lfloor \bar{t} \rfloor = t$ und $I \xrightarrow{\bar{t}}_{\mathrm{Gu}} \mathsf{Empty}$. □

3.5 Die Regel (seq^3_{Gu}) unter der Lupe

Wirft man einen unbefangenen Blick auf die Regel (seq^3_{Gu}) des Regelsystems Gu in Tafel 3.1, so könnte man die Frage stellen, ob eine äquivalente globale operationale Semantik vielleicht auch ohne eine Restriktionsfunktion R_L bzw. mit der Setzung $\mathrm{R}_L := \mathrm{id}$ gefunden werden kann. Um diese Frage mit Ja oder Nein beantworten zu können, muss sie präziser formuliert werden.

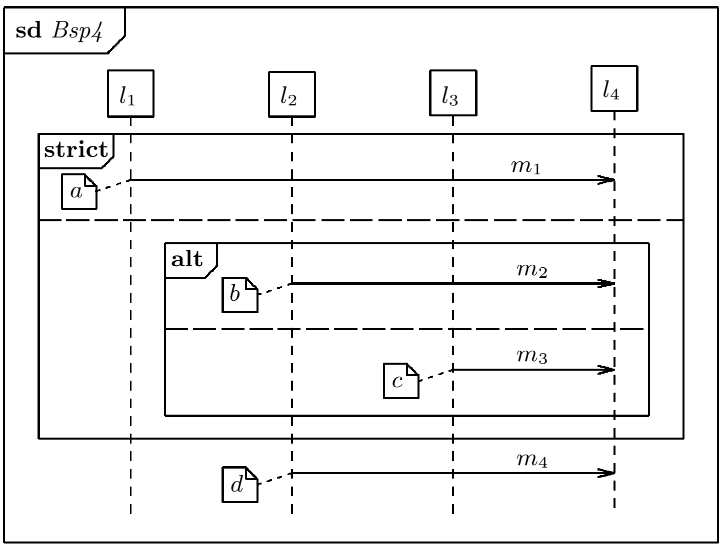
3.5.1 Ein parametrisiertes Regelsystem

Es sei ein parametrisiertes Regelsystem $\text{Gu}[B]$ definiert, das aus dem Regelschema

$$(\text{seq}^3_{\text{Gu}[B]}) \quad \frac{I_2 \xrightarrow{\bar{a}}_{\text{Gu}} I_2'}{\mathsf{seq}(I_1, I_2) \xrightarrow{\bar{a}}_{\text{Gu}} \mathsf{seq}(I_1, I_2')} \quad \text{falls } B$$

und im Übrigen aus den restlichen Axiomen- und Regelschemata von Gu bestehen möge. Der Parameter B, der der Seitenbedingung des modifizierten Regelschemas ($\text{seq}^3_{\text{Gu}[B]}$) entspricht, soll über der Menge der relevanten Metavariablen I_1, I_2, I_2' und $\bar{a}$ frei wählbar sein.

Wir reformulieren die Fragestellung wie folgt: Existiert eine Seitenbedingung B, so dass das Regelsystem $\text{Gu}[B]$ bezüglich der denotationellen Semantik korrekt und zugleich vollständig ist? Es kann leicht bewiesen werden, dass diese Frage zu verneinen ist. Hierzu setzen wir $I := \mathsf{seq}(I_1, I_2)$ mit $I_1 := \mathsf{strict}(a, \mathsf{alt}(b, c))$ und $I_2 := d$. Dabei seien $a, b, c, d \in \mathbb{A}$ paarweise verschiedene Aktionen, die bis auf $b \lessgtr d$ konfliktfrei sind. Der Term I modelliert den *Anteil der Sendeaktionen* in folgendem UML 2.0-Sequenzdiagramm:



Die positiven Erfüllungsmengen von I_1 und I berechnen sich wie folgt: $\mathcal{D}[\![I_1]\!] = \mathcal{D}[\![a]\!] \mathbin{;} \mathcal{D}[\![\mathsf{alt}(b,c)]\!] = \{a\}\mathbin{;}\{b,c\} = \{ab, ac\}$. Somit gilt $\mathcal{D}[\![I]\!] = (\mathcal{D}[\![I_1]\!] \mathbin{;_{\gg}} \mathcal{D}[\![I_2]\!])\!\downarrow = (\{ab, ac\} \mathbin{;_{\gg}} \{d\})\!\downarrow = (\{abd, ac \parallel d\})\!\downarrow = \{abd,\ acd, adc, dac\}$.

Welche Übergänge sind im Regelsystem Gu$[B]$ für I ableitbar und welche Rolle spielt dabei die Wahl von B? Zur Ableitung eines Übergangs für $I = \mathsf{seq}(I_1, I_2)$ kommen syntaktisch nur die Regeln ($\mathrm{seq}^1_{\mathrm{Gu}}$) und ($\mathrm{seq}^3_{\mathrm{Gu}[B]}$) in Frage. Ob die Regel ($\mathrm{seq}^3_{\mathrm{Gu}[B]}$) anwendbar ist, hängt dabei natürlich von der Seitenbedingung B ab. Im Rahmen einer zweiwertigen Logik, die wir für Seitenbedingungen von Regelschemata unterstellen dürfen, kann B für die Konfiguration I—unabhängig von der konkreten Definition von B—nur zu *tt* oder zu *ff* auswertet. Eine dritte Möglichkeit gibt es nicht.

Fall 1: B wertet für I zu *ff* aus. Dann ist nur ($\mathrm{seq}^1_{\mathrm{Gu}}$) anwendbar und folglich ist $I \xrightarrow{a}_{\mathrm{Gu}} \mathsf{seq}(\mathsf{strict}(\mathsf{Empty}, \mathsf{alt}(b,c)), d)$ der einzige ableitbare Übergang für I. Damit beginnen alle für I ableitbaren Traces mit der Aktion a. Wegen $dac \in \mathcal{D}[\![I]\!]$ ist das Regelsystem Gu$[B]$ nicht vollständig.

Fall 2: B wertet für I zu *tt* aus. Dann ist die Regel ($\mathrm{seq}^3_{\mathrm{Gu}[B]}$) anwendbar und somit der Übergang $I \xrightarrow{d}_{\mathrm{Gu}} \mathsf{seq}(I_1, \mathsf{Empty})$ ableitbar. Man sieht leicht, dass für I_1 der Übergang $I_1 \xrightarrow{a\tau\tau b}_{\mathrm{Gu}} \mathsf{Empty}$ ableitbar ist. Mit der Regel ($\mathrm{seq}^1_{\mathrm{Gu}}$) folgt die Ableitbarkeit von $\mathsf{seq}(I_1, \mathsf{Empty}) \xrightarrow{a\tau\tau b\tau}_{\mathrm{Gu}} \mathsf{Empty}$. Folglich ist $dab = \lfloor da\tau\tau b\tau \rfloor$ für I ableitbar, doch es gilt $dab \notin \mathcal{D}[\![I]\!]$. Das Regelsystem Gu$[B]$ ist nicht korrekt. □

Ergebnis: Das Regelsystem Gu$[B]$ ist entweder nicht korrekt oder nicht vollständig (oder beides nicht) und diese Aussage ist unabhängig von der Wahl von B. Eine entsprechende Aussage wird auch für die lokale operationale Semantik aus Kapitel 4 gelten.

3.5.2 Vorweggenommene Entscheidungsschritte

Der Beweis in Abschnitt 3.5.1 beruht auf der Prämisse, dass mit Ausnahme der Regel ($\mathrm{seq}^3_{\mathrm{Gu}}$) alle Regeln des Regelsystems Gu unverändert bleiben. Die Situation stellte sich anders dar, wenn infolge zusätzlicher Änderungen am Regelsystem Gu die Möglichkeit bestünde, *bevor* die Regel ($\mathrm{seq}^3_{\mathrm{Gu}[B]}$) auf ein $\mathsf{seq}(I_1, I_2)$-Konstrukt angewendet wird, alle im linken Teilterm I_1 enthaltenen alt- und loop-Konstrukte durch τ-Schritte zur Entscheidung zu bringen[1]. Es stellt sich heraus, dass eine solche Vorwegnahme von Entscheidungsschritten bereits realisiert werden kann, fügt man dem oben definierten parametrisierten Regelsystem Gu$[B]$ nur eine einzige weitere Regel hinzu, nämlich

$$(\mathrm{strict}^3_{\mathrm{Gu}}) \quad \frac{I_2 \xrightarrow{\tau}_{\mathrm{Gu}} I_2'}{\mathsf{strict}(I_1, I_2) \xrightarrow{\tau}_{\mathrm{Gu}} \mathsf{strict}(I_1, I_2')}$$

[1]Alexander Knapp, mündliche Äußerung.

Diese ermöglicht die Ausführung von τ-Schritten des rechten Teilterms einer strikten Komposition auch dann, wenn der linke Teilterm noch nicht vollständig abgearbeitet ($\neq \mathsf{Empty}$) ist. Rückblickend auf Lemma 3.2 erkennt man leicht, dass die zusätzliche Regel ($\mathrm{strict}^3_{\mathrm{Gu}}$) die Korrektheit des Regelsystems $\mathrm{Gu}[B]$ nicht berührt. Wählt man als Seitenbedingung $B :\Leftrightarrow \alpha(I_1) \cap \alpha(\bar{a}) = \emptyset$, so erhält man ein Regelsystem Gu' bestehend aus $\mathrm{Gu}[\alpha(I_1) \cap \alpha(\bar{a}) = \emptyset]$ und der Regel ($\mathrm{strict}^3_{\mathrm{Gu}}$), welches korrekt und zugleich vollständig ist und trotzdem ohne Restriktionsfunktion auskommt. Für den Beweis der Vollständigkeit von Gu' ist das folgende Lemma entscheidend:

Lemma 3.7 Sei $I \in \mathrm{IFrag}$ und $L \subseteq \mathbb{I}$. Dann gilt:

$$\mathcal{D}[\![I]\!][L] \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid \alpha(I') \cap L = \emptyset \wedge I \xrightarrow{\tau^*}_{\mathrm{Gu}'} I'\}.$$

Beweis. Der Beweis erfolgt durch strukturelle Induktion über I:

$I = \mathsf{Empty}$: Es gilt $\mathcal{D}[\![I]\!][L] = \{\varepsilon\}[L] = \{\varepsilon\} \subseteq \mathcal{D}[\![I]\!]$. Trivialerweise gilt $I \xrightarrow{\tau^*}_{\mathrm{Gu}'} I$ und $\alpha(I) \cap L = \emptyset$.

$I = a$: Wir unterscheiden zwei Fälle:
Fall 1: $\alpha(a) \cap L = \emptyset$. Es folgt: $\mathcal{D}[\![I]\!][L] = \{a\}[L] = \{a\} \subseteq \mathcal{D}[\![I]\!]$. Trivialerweise gilt $I \xrightarrow{\tau^*}_{\mathrm{Gu}'} I$ und $\alpha(I) \cap L = \emptyset$.
Fall 2: $\alpha(a) \cap L \neq \emptyset$. Es folgt: $\mathcal{D}[\![I]\!][L] = \{a\}[L] = \emptyset \subseteq M$ für jede Menge M.

$I = \mathsf{strict}(I_1, I_2)$: Dann gilt: $\mathcal{D}[\![I]\!][L] = (\mathcal{D}[\![I_1]\!] \,;\, \mathcal{D}[\![I_2]\!])[L] =$ (mit Lemma 2.6.2) $\mathcal{D}[\![I_1]\!][L] \,;\, \mathcal{D}[\![I_2]\!][L]$. Aus der I. V. für I_1 und I_2 und aus dem Lemma 2.5.1 folgt $\mathcal{D}[\![I_1]\!][L] \,;\, \mathcal{D}[\![I_2]\!][L] \subseteq (\bigcup\{\mathcal{D}[\![I_1']\!] \mid \alpha(I_1') \cap L = \emptyset \wedge I_1 \xrightarrow{\tau^*}_{\mathrm{Gu}'} I_1'\}) \,;\, (\bigcup\{\mathcal{D}[\![I_2']\!] \mid \alpha(I_2') \cap L = \emptyset \wedge I_2 \xrightarrow{\tau^*}_{\mathrm{Gu}'} I_2'\}) = \bigcup\{\mathcal{D}[\![I_1']\!] \,;\, \mathcal{D}[\![I_2']\!] \mid \alpha(I_1') \cap L = \emptyset \wedge \alpha(I_2') \cap L = \emptyset \wedge I_1 \xrightarrow{\tau^*}_{\mathrm{Gu}'} I_1' \wedge I_2 \xrightarrow{\tau^*}_{\mathrm{Gu}'} I_2'\} \subseteq \bigcup\{\mathcal{D}[\![\mathsf{strict}(I_1', I_2')]\!] \mid \alpha(\mathsf{strict}(I_1', I_2')) \cap L = \emptyset \wedge \mathsf{strict}(I_1, I_2) \xrightarrow{\tau^*}_{\mathrm{Gu}'} \mathsf{strict}(I_1', I_2')\} \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid \alpha(I') \cap L = \emptyset \wedge I \xrightarrow{\tau^*}_{\mathrm{Gu}'} I'\}$. Im vorletzten Schritt wurden die Regeln ($\mathrm{strict}^1_{\mathrm{Gu}}$) und ($\mathrm{strict}^3_{\mathrm{Gu}}$) angewendet.

$I = \mathsf{seq}(I_1, I_2)$: Analog zu Fall $I = \mathsf{strict}(I_1, I_2)$ mit Regeln ($\mathrm{seq}^1_{\mathrm{Gu}}$) und ($\mathrm{seq}^3_{\mathrm{Gu}[B]}$) wobei $B \Leftrightarrow \alpha(I_1) \cap \alpha(\bar{a}) = \emptyset$ und $\alpha(\tau) = \emptyset$. An die Stelle von Lemma 2.6.2 treten 2.2.4 und 2.6.3. An die Stelle von Lemma 2.5.1 treten 2.5.2 und 2.5.7.

$I = \mathsf{par}(I_1, I_2)$: Analog zu Fall $I = \mathsf{strict}(I_1, I_2)$ mit Regeln ($\mathrm{par}^1_{\mathrm{Gu}}$) und ($\mathrm{par}^2_{\mathrm{Gu}}$). An die Stelle von Lemma 2.6.2 treten 2.2.4 und 2.6.4. An die Stelle von Lemma 2.5.1 treten 2.5.3 und 2.5.7.

$I = \mathsf{loop}(I_1)$: Dann gilt $\mathcal{D}[\![I]\!] = \bigcup_{n \in \mathbb{N}_0} \mathcal{D}^n[\![I]\!]$ mit $\mathcal{D}^n[\![I]\!] := ((\mathcal{D}[\![I_1]\!])^{n\between})\!\downarrow$. Wir zeigen $\forall n \in \mathbb{N}_0 \,.\, \mathcal{D}^n[\![I]\!][L] \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid \alpha(I') \cap L = \emptyset \wedge I \xrightarrow{\tau^*}_{\mathrm{Gu}'} I'\}$ vermöge einer vollständigen Induktion nach $n \in \mathbb{N}_0$. Hieraus folgt die Behauptung mittels Lemma 2.6.5 (analog).

$n = 0$: Es gilt $\mathcal{D}^0[\![I]\!][L] = ((\mathcal{D}[\![I_1]\!])^{0\between})\!\downarrow\![L] = \{\varepsilon\}\!\downarrow\![L] = \{\varepsilon\}[L] = \{\varepsilon\} =$

$\mathcal{D}[\![\mathsf{Empty}]\!]$. Aus Axiom (loop$^1_{\text{Gu}}$) folgt $I \xrightarrow{\tau}_{\text{Gu}'} \mathsf{Empty}$. Ferner ist $\alpha(\mathsf{Empty}) \cap L = \emptyset$.

$n \Rightarrow n+1$: Es gilt $\mathcal{D}^{n+1}[\![I]\!][L] = ((\mathcal{D}[\![I_1]\!])^{(n+1)\ltimes})\downarrow[L] = (\mathcal{D}[\![I_1]\!] \,;_{\ltimes}\, (\mathcal{D}[\![I_1]\!])^{n\ltimes})\downarrow[L] = (\mathcal{D}[\![I_1]\!] \,;_{\ltimes}\, \mathcal{D}^n[\![I]\!])\downarrow[L]$, wobei im letzten Schritt das Lemma 2.2.2 verwendet wurde. Mit Lemma 2.2.4/2.6.3 folgt: $\mathcal{D}^{n+1}[\![I]\!][L] = ((\mathcal{D}[\![I_1]\!][L]) \,;_{\ltimes}\, (\mathcal{D}^n[\![I]\!][L]))\downarrow$. Aus der I. V. für I_1 folgt $\mathcal{D}[\![I_1]\!][L] \subseteq \bigcup\{\mathcal{D}[\![I_1']\!] \mid \alpha(I_1') \cap L = \emptyset \,\wedge\, I_1 \xrightarrow{\tau^*}_{\text{Gu}'} I_1' \}$. Aus der I. V. für n folgt $\mathcal{D}^n[\![I]\!][L] \subseteq \bigcup\{\mathcal{D}[\![I']\!] \mid \alpha(I') \cap L = \emptyset \,\wedge\, I \xrightarrow{\tau^*}_{\text{Gu}'} I' \}$. Aus Lemma 2.5.2 und 2.5.7 folgt $\mathcal{D}^{n+1}[\![I]\!][L] \subseteq \bigcup\{(\mathcal{D}[\![I_1']\!] \,;_{\ltimes}\, \mathcal{D}[\![I']\!])\downarrow \mid \alpha(I_1') \cap L = \emptyset \wedge \alpha(I') \cap L = \emptyset \,\wedge\, I_1 \xrightarrow{\tau^*}_{\text{Gu}'} I_1' \,\wedge\, I \xrightarrow{\tau^*}_{\text{Gu}'} I' \}$. Aufgrund der Regeln (seq$^1_{\text{Gu}}$) und (seq$^3_{\text{Gu}[B]}$) mit $B \Leftrightarrow \alpha(I_1) \cap \alpha(\bar{a}) = \emptyset$ und $\alpha(\tau) = \emptyset$ ist die letztere Menge eine Teilmenge von $\bigcup\{\mathcal{D}[\![\mathsf{seq}(I_1', I')]\!] \mid \alpha(I_1') \cap L = \emptyset \wedge \alpha(I') \cap L = \emptyset \wedge \mathsf{seq}(I_1, I) \xrightarrow{\tau^*}_{\text{Gu}'} \mathsf{seq}(I_1', I') \}$. Mit Regel (loop$^2_{\text{Gu}}$) folgt die Behauptung.

$I = \mathsf{alt}(I_1, I_2)$: Dann gilt $\mathcal{D}[\![I]\!][L] = (\mathcal{D}[\![I_1]\!] \cup \mathcal{D}[\![I_2]\!])[L] = \mathcal{D}[\![I_1]\!][L] \cup \mathcal{D}[\![I_2]\!][L]$. Mit I. V. für I_1 und I_2 folgt $\mathcal{D}[\![I]\!][L] \subseteq \bigcup\{\mathcal{D}[\![I_1']\!] \mid \alpha(I_1') \cap L = \emptyset \wedge I_1 \xrightarrow{\tau^*}_{\text{Gu}'} I_1' \} \cup \bigcup\{\mathcal{D}[\![I_2']\!] \mid \alpha(I_2') \cap L = \emptyset \wedge I_2 \xrightarrow{\tau^*}_{\text{Gu}'} I_2' \}$. Mit (alt$^1_{\text{Gu}}$), (alt$^2_{\text{Gu}}$) folgt Behauptung. $\square$

Das Lemma 3.5 und der Satz 3.6 sind auch für das Regelsystem Gu$'$ gültig. Im Beweis von Lemma 3.5 wende man unter Ad. 1 im Fall $I = \mathsf{seq}(I_1, I_2)$ (siehe S. 25) die Beziehung $\mathcal{D}[\![I_1]\!][\alpha(a)] \subseteq \bigcup\{\mathcal{D}[\![I_1']\!] \mid \alpha(I_1') \cap \alpha(a) = \emptyset \wedge I_1 \xrightarrow{\tau^*}_{\text{Gu}'} I_1' \}$ an, welche sich direkt aus Lemma 3.7 ergibt. Die verbleibenden Details sollten dem Leser keine Mühe bereiten.

Gegen das Regelschema (strict$^3_{\text{Gu}}$) könnte eingewendet werden, dass es nicht der intuitiven Idee des strict-Konstuktes entspreche, mit der Bearbeitung des rechten Teilterms zu beginnen, bevor der linke Teilterm vollständig abgearbeitet ist. Weil es sich aber nur um die Ausführung von τ-Schritten handelt, über die in den Ausführungstraces abstrahiert wird, vermag dieses Argument nicht sonderlich zu überzeugen.

In der vorliegenden Arbeit wird die Idee des Regelschemas (strict$^3_{\text{Gu}}$) nicht weiterverfolgt werden. Stattdessen arbeiten wir mit einer Restriktionsfunktion $\mathrm{R}_L \neq \mathrm{id}$.

3.5.3 Varianten einer Restriktionsfunktion

Eine Restriktionsfunktion R_L mit der Eigenschaft $\mathcal{D}[\![\mathrm{R}_L(I)]\!] = \mathcal{D}[\![I]\!][L]$ kann auf verschiedene Weise definiert werden:

1. Versähe man die Interaktionssprache IFrag mit einem Kapselungskonstrukt $\mathsf{restr}(L, I)$ und definierte man die denotationelle Semantik dieses Konstrukts gemäß $\mathcal{D}[\![\mathsf{restr}(L, I)]\!] := \mathcal{D}[\![I]\!][L]$, dann könnte—wie in Abschnitt 3.2 bereits erwähnt—eine Restriktionsfunktion in trivialer Weise durch $\mathrm{R}_L(I) := \mathsf{restr}(L, I)$ definiert werden.

2. Ein anderer Weg zur Definition von R_L, der in der vorliegenden Arbeit bei der lokalen Semantik eingeschlagen wird, ist es, die Sprache IFrag durch ein None-Konstrukt mit leerer Erfüllungsmenge zu erweitern und die Funktion R_L so zu definieren, dass sie alle in I enthaltenen Aktionen, die auf einer der Lifelines aus L liegen, durch None ersetzt. Diese Variante hat den Vorteil, dass die Interaktionsterme nur auf der Blattebene modifiziert werden und die innere Termstruktur unverändert bleibt.

3. Weil in unserer Interaktionssprache IFrag weder ein Kapselungskonstrukt $\mathsf{restr}(L, I)$ noch ein None-Konstrukt vorhanden ist, hatten wir zur Definition von R_L eine Variante gewählt, die ohne Spracherweiterung auskommt. Die Definition findet sich in Abschnitt 3.2 und soll hier nicht wiederholt werden.

Kapitel 4

Verteilte operationale Semantik

Die Grundidee der verteilten oder lokalen operationalen Semantik ist es, die globale Konfigurationsinformation auf die verschiedenen beteiligten Lifelines zu verteilen. Für jede Lifeline wird ein eigener lokaler Konfigurationsterm verwaltet, wobei die verschiedenen Lifelines nebenläufig ausgeführt werden (Interleaving-Modell). Die Bedeutung bestimmter Interaktionsfragmente (strict, loop, alt) kann in einer solchen verteilten Semantik allerdings nur dann korrekt ausgedrückt werden, wenn in der Semantik ein impliziter Mechanismus zur Synchronisation und Kommunikation zwischen den Lifelines vorhanden ist. Dies macht die Verwaltung einer gewissen Menge an gemeinsam genutzter Konfigurationsinformation unvermeidlich—eine Art "shared memory", welches wir mit h bezeichnen werden. Das wesentliche Designziel einer verteilten operationalen Semantik ist—neben Korrektheit und Vollständigkeit—die Minimierung dieser Menge h an gemeinsam genutzter Information.

4.1 Pfadannotierte Interaktionsfragmente

Das Verteilen der Konfigurationsinformation auf die verschiedenen Lifelines führt dazu, dass ein und dasselbe globale strict-, loop- oder alt-Konstruktorsymbol als Kopie in allen lokalen Konfigurationstermen auftreten kann. Der anfangs erwähnte Mechanismus zur Synchronisation und Kommunikation zwischen den Lifelines, der zur korrekten Bearbeitung von strict, loop und alt erforderlich ist, setzt ein geeignetes Nummerierungs- bzw. Annotierungsschema für diese Konstruktorsymbole voraus, welches dazu dient, dass korrespondierende Kopien von ein und demselben Konstruktorsymbol, die auf verschiedenen Lifelines liegen, (1) zueinander und (2) zu der für sie in h gespeicherten Information in Bezug gesetzt werden können.

Zur Nummerierung der Konstruktorsymbole werden *Pfade* benutzt. Ein Pfad $p \in \mathrm{Path} ::= \epsilon \,|\, p\,n$ ist eine endliche Folge von Zahlen $n \in \{1, 2\}$. Jede dieser

Zahlen n steht für eine Verzweigung in einen linken ($n = 1$) oder in einen rechten direkten Teilterm ($n = 2$). Zwei Pfade p, q können zu $p.q$ konkateniert werden, wobei der Operator '.' durch $p.\epsilon := p$ und $p.(q\,n) := (p.q)\,n$ definiert ist. Die Konkatenation '.' ist assoziativ, wir können folglich Klammerungen weglassen und Pfade z. B. in der Form $p.q.r.\ldots$ schreiben.

Die abstrakte Syntax der *pfadannotierten Interaktionen* (Interaktionsfragmente) ist in Tafel 4.1 angegeben. Aus Gründen der Einfachheit haben wir *alle* unären und binären Konstruktorsymbole mit einem Pfad p annotiert, obgleich dies eigentlich nur für die Konstruktoren strict, loop und alt erforderlich wäre. Es fällt ferner auf, dass wir unserer Interaktionssprache ein Konstrukt $\mathsf{None}(a)$ hinzugefügt haben. Dieses Konstrukt hat die übliche Bedeutung eines None-Konstrukts (leere Erfüllungsmenge). Auf die Bedeutung des Parameters a werden wir zu Beginn von Abschnitt 4.2 eingehen.

Wir führen eine Funktion top : Term $\longrightarrow$ Path $\dot\cup$ $\{\perp\}$ ein ("topmost operator path"), die uns für jeden pfadannotierten Interaktionsterm T den Pfad p liefert, mit welchem das äußerste Konstruktorsymbol annotiert ist—falls eine solche Pfadannotierung existiert. Für einen konstanten Interaktionsterm wird $\perp$ zurückgegeben. Somit ist definiert:

$$\begin{array}{lll}
\mathrm{top}(\mathit{const}) & := \perp & \text{für alle } \mathit{const} \in \{\mathsf{None}(a), \mathsf{Empty}, a\} \\
\mathrm{top}(\mathit{uop}_p(T)) & := p & \text{für alle } \mathit{uop} \in \{\mathsf{loop}\} \\
\mathrm{top}(\mathit{bop}_p(T_1, T_2)) & := p & \text{für alle } \mathit{bop} \in \{\mathsf{strict}, \mathsf{seq}, \mathsf{par}, \mathsf{alt}\}
\end{array}$$

Es versteht sich von selbst, dass die Kurznotation $\mathit{const} \in \{\mathsf{None}(a), \mathsf{Empty}, a\}$ im Sinne von $\mathit{const} \in \{\mathsf{None}(a) \mid a \in \mathbb{A}\} \,\dot\cup\, \{\mathsf{Empty}\} \,\dot\cup\, \{a \mid a \in \mathbb{A}\}$ zu verstehen ist. Dabei ist $\mathbb{A}$ der vereinbarte Bereich der Metavariable a. Ähnliche Kurznotationen werden wir förderhin benutzen, ohne sie weiter zu erläutern. Sofern es nicht ausdrücklich anders angegeben ist, haben die Metavariablen *const*, *uop* und *bop* stets die Bereiche $\{\mathsf{None}(a), \mathsf{Empty}, a\}$, $\{\mathsf{loop}\}$ bzw. $\{\mathsf{strict}, \mathsf{seq}, \mathsf{par}, \mathsf{alt}\}$.

Wir sind im Folgenden nur an *wohlnummerierten* Interaktionstermen interessiert. Die Wohlnummeriertheit ist ein einstelliges Prädikat auf Term, das induktiv definiert ist wie folgt:

1. *const* ist wohlnummeriert.
2. $\mathit{uop}_p(T)$ ist wohlnummeriert, falls T wohlnummeriert ist und ein Pfad q existiert, so dass gilt $\mathrm{top}(T) \in \{\perp, p.1.q\}$.
3. $\mathit{bop}_p(T_1, T_2)$ ist wohlnummeriert, falls T_1 und T_2 wohlnummeriert sind und Pfade q_1, q_2 existieren, so dass gilt $\mathrm{top}(T_1) \in \{\perp, p.1.q_1\}$ und $\mathrm{top}(T_2) \in \{\perp, p.2.q_2\}$.

In Abschnitt 4.3 und 4.6 werden wir zeigen, dass die Wohlnummeriertheit eines Konfigurationsterms während seiner schrittweisen Auswertung erhalten bleibt. Es sei angemerkt, dass die Konstruktorsymbole eines wohlnummerierten Interaktionsterms insbesondere auch *eindeutig* nummeriert sind.

$$
\begin{array}{rcl}
S, T \in \mathrm{Term} & ::= & \mathsf{None}(a) \\
& | & \mathsf{Empty} \\
& | & a \\
& | & \mathsf{strict}_p(T_1, T_2) \\
& | & \mathsf{seq}_p(T_1, T_2) \\
& | & \mathsf{par}_p(T_1, T_2) \\
& | & \mathsf{loop}_p(T) \\
& | & \mathsf{alt}_p(T_1, T_2)
\end{array}
$$

Tafel 4.1: Abstrakte Syntax der pfadannotierten Interaktionen

Ein Nummerierungsoperator num, ein Entnummerierungsoperator denum und ein Umnummerierungsoperator ren seien definiert wie folgt:

$$\mathrm{num} : \mathrm{IFrag} \times \mathrm{Path} \longrightarrow \mathrm{Term}$$

$$
\begin{array}{lcll}
\mathrm{num}(\mathit{const}, p) & := & \mathit{const} & \text{wobei } \mathit{const} \in \{\mathsf{Empty}, a\} \\
\mathrm{num}(\mathit{uop}(I), p) & := & \mathit{uop}_p(\mathrm{num}(I, p.1)) & \\
\mathrm{num}(\mathit{bop}(I_1, I_2), p) & := & \mathit{bop}_p(\mathrm{num}(I_1, p.1), \mathrm{num}(I_2, p.2)) &
\end{array}
$$

$$\mathrm{denum} : \mathrm{Term} \longrightarrow \mathrm{NIFrag}$$

$$
\begin{array}{lcl}
\mathrm{denum}(\mathsf{None}(a)) & := & \mathsf{None} \\
\mathrm{denum}(\mathsf{Empty}) & := & \mathsf{Empty} \\
\mathrm{denum}(a) & := & a \\
\mathrm{denum}(\mathit{uop}_p(T)) & := & \mathit{uop}(\mathrm{denum}(T)) \\
\mathrm{denum}(\mathit{bop}_p(T_1, T_2)) & := & \mathit{bop}(\mathrm{denum}(T_1), \mathrm{denum}(T_2))
\end{array}
$$

$$\mathrm{ren} : \mathrm{Path} \times \{1, 2\} \times \mathrm{Term} \longrightarrow \mathrm{Term}$$

$$
\begin{array}{lcl}
\mathrm{ren}(p, n, \mathit{const}) & := & \mathit{const} \\
\mathrm{ren}(p, n, \mathit{uop}_q(T)) & := & \begin{cases} \mathit{uop}_{p.n.q'}(\mathrm{ren}(p, n, T)) & \text{falls } q = p.q' \\ \mathit{uop}_q(T) & \text{andernfalls} \end{cases} \\
\mathrm{ren}(p, n, \mathit{bop}_q(T_1, T_2)) & := & \begin{cases} \mathit{bop}_{p.n.q'}(\mathrm{ren}(p, n, T_1), \mathrm{ren}(p, n, T_2)) & \text{falls } q = p.q' \\ \mathit{bop}_q(T_1, T_2) & \text{andernfalls} \end{cases}
\end{array}
$$

Der Zielbereich des Operators denum ist definiert wie folgt: $\bar{I} \in \mathrm{NIFrag} ::= \mathsf{None} \,|\, \mathsf{Empty} \,|\, a \,|\, \mathit{uop}(\bar{I}) \,|\, \mathit{bop}(\bar{I}_1, \bar{I}_2)$. Den umnummerierten Term $\mathrm{ren}(p, n, T)$ werden wir abkürzend als ${}_{p.n}T$ notieren.

Lemma 4.1 Sei $T \in \mathrm{Term}$ wohlnummeriert. Seien $p, q \in \mathrm{Path}$, $n \in \{1, 2\}$ und gelte $\mathrm{top}(T) = p.q$. Dann ist ${}_{p.n}T$ wohlnummeriert und es gilt $\mathrm{top}({}_{p.n}T) = p.n.q$.

Beweis. Der Beweis erfolgt durch strukturelle Induktion über T:

$T = \mathit{const}$: Nicht möglich, weil $\mathrm{top}(T) = p.q \neq \bot = \mathrm{top}(\mathit{const})$ ist.

$T = \mathit{uop}_{p.q}(T_1)$: Weil $T = \mathit{uop}_{p.q}(T_1)$ wohlnummeriert ist, ist T_1 wohlnummeriert und es gilt $\mathrm{top}(T_1) \in \{\bot, p.q.1.r_1\}$ mit einem $r_1 \in \mathrm{Path}$.

Fall 1: $\mathrm{top}(T_1) = \bot$: Dann ist $T_1 = \mathit{const}$ und ${}_{p.n}T_1 = T_1$. Somit ist ${}_{p.n}T = {}_{p.n}\mathit{uop}_{p.q}(T_1) = \mathit{uop}_{p.n.q}({}_{p.n}T_1) = \mathit{uop}_{p.n.q}(T_1)$. Also ist ${}_{p.n}T$ wohlnummeriert und es gilt $\mathrm{top}({}_{p.n}T) = p.n.q$.

Fall 2: $\mathrm{top}(T_1) = p.q.1.r_1$: Mit der I. V. für T_1 folgt: ${}_{p.n}T_1$ ist wohlnummeriert und $\mathrm{top}({}_{p.n}T_1) = p.n.q.1.r_1$. Damit ist ${}_{p.n}T = {}_{p.n}\mathit{uop}_{p.q}(T_1) = \mathit{uop}_{p.n.q}({}_{p.n}T_1)$ wohlnummeriert und $\mathrm{top}({}_{p.n}T) = p.n.q$.

$T = \mathit{bop}_{p.q}(T_1, T_2)$: Analog zu $T = \mathit{uop}_{p.q}(T_1)$ mit zwei Teiltermen T_1 und T_2. $\square$

Wir definieren schließlich die Lifelinefunktion α auf Term völlig analog zu ihrer Definition auf IFrag (siehe Abschnitt 2.1), wobei wir $\alpha(\mathsf{None}(a)) := \emptyset$ setzen. Somit ist die Lifelinefunktion α auf $\mathrm{Term} \cup \mathrm{IFrag} \cup \mathbb{D}_\tau$ definiert. Eine entsprechende Funktion $\nu : \mathrm{Term} \longrightarrow \wp(\mathbb{I})$, die wir in Ermangelung eines besseren Namens die "Nonefunktion" nennen, liefere für eine Interaktion I die Menge aller Lifelines l, so dass in I mindestens ein Konstrukt $\mathsf{None}(a)$ enthalten ist, dessen gekapselte Aktion a auf der Lifeline l liegt. Definition:

$$\nu : \mathrm{Term} \longrightarrow \wp(\mathbb{I})$$

$$\begin{array}{lcl} \nu(\mathsf{None}(a)) & := & \alpha(a) \\ \nu(\mathsf{Empty}) & := & \emptyset \\ \nu(a) & := & \emptyset \\ \nu(\mathit{uop}(I)) & := & \nu(I) \\ \nu(\mathit{bop}(I_1, I_2)) & := & \nu(I_1) \cup \nu(I_2) \end{array}$$

Für alle $L, \tilde{L} \subseteq \mathbb{I}$ setzen wir $\mathrm{Term}^{L(\tilde{L})} := \{T \in \mathrm{Term} \mid \alpha(T) \subseteq L \wedge \nu(T) \subseteq \tilde{L}\}$. Wir setzen $\mathrm{Term}^L := \mathrm{Term}^{L(L)}$ und $\mathrm{Term}^l := \mathrm{Term}^{\{l\}(\{l\})}$ für jedes $l \in \mathbb{I}$.

4.2 Syntaktische Restriktion und Projektion

In Abschnitt 3.2 ist zur Formulierung der Regel ($\mathrm{seq}^3_{\mathrm{Gu}}$) eine Variante gewählt worden, die keinerlei Spracherweiterungen erforderte. Diesen Vorteil bezahlten wir mit einer Restriktionsfunktion R_L, die im übergebenen Interaktionsterm alle Teilterme mit leerer Erfüllungsmenge komplett entfernt und somit nichttriviale Veränderungen an der Termstruktur vornimmt. Um die lokale operationale Semantik und den zugehörigen Korrektheits- und Vollständigkeitsbeweis nicht mit zusätzlichen Problemen zu befrachten, wählen wir nun eine andere Variante einer Restriktionsfunktion R^*_L, deren Eingriffe in die Termstruktur der übergebenen

Interaktionen weniger gravierend sind. Am schonendsten für die Termstruktur erscheint es, in der übergebenen Interaktion I alle Aktionen a, die auf einer der Lifelines aus L liegen, durch ein spezielles None-Konstrukt mit leerer Erfüllungsmenge zu ersetzen. Somit wirkt die syntaktische Restriktion nur auf die Blattterme der Interaktionen und läßt deren innere Struktur unverändert. Um das beweistechnische Vorgehen weiter zu vereinfachen, merken wir uns in dem None-Konstrukt überdies, welche Aktion a vor Anwendung der Restriktionsfunktion an der betreffenden Stelle des Interaktionsterms stand, d. h. wir erweitern unsere Interaktionssprache um ein Konstrukt $\mathsf{None}(a)$; siehe Tafel 4.1.

Es sei $L \subseteq \mathbb{I}$. Die Restriktionsfunktion R_L^* ist definiert wie folgt:

$$\mathrm{R}_L^* : \mathrm{Term} \longrightarrow \mathrm{Term}$$

$$\begin{array}{lcl}
\mathrm{R}_L^*(\mathsf{None}(a)) & := & \mathsf{None}(a) \\
\mathrm{R}_L^*(\mathsf{Empty}) & := & \mathsf{Empty} \\
\mathrm{R}_L^*(a) & := & \begin{cases} a & \text{falls } \alpha(a) \cap L = \emptyset \\ \mathsf{None}(a) & \text{andernfalls} \end{cases} \\
\mathrm{R}_L^*(uop_p(T)) & := & uop_p(\mathrm{R}_L^*(T)) \\
\mathrm{R}_L^*(bop_p(T_1, T_2)) & := & bop_p(\mathrm{R}_L^*(T_1), \mathrm{R}_L^*(T_2))
\end{array}$$

Mit struktureller Induktion über T sieht man, dass $\alpha(\mathrm{R}_L^*(T)) \subseteq \alpha(T) \setminus L$ ist.

Damit ein globaler Konfigurationsterm mit lokalen Konfigurationstermen in Bezug gesetzt werden kann, werden Projektionsoperatoren π_l benötigt, die einen annotierten Interaktionsterm T auf eine gewünschte Lifeline l projizieren, indem sie in T jede Aktionen a, die nicht auf der Lifeline l liegt, durch Empty ersetzen. Außerdem soll π_l jedes Konstrukt $\mathsf{None}(a)$ gegen Empty ersetzen, das eine Aktion a kapselt, die nicht auf der Lifeline l liegt. Für jede Lifeline $l \in \mathbb{I}$ definieren wir diesen Projektionsoperator π_l wie folgt:

$$\pi_l : \mathrm{Term} \longrightarrow \mathrm{Term}$$

$$\begin{array}{lcl}
\pi_l(\mathsf{None}(a)) & := & \begin{cases} \mathsf{None}(a) & \text{falls } l \in \alpha(a) \\ \mathsf{Empty} & \text{falls } l \notin \alpha(a) \end{cases} \\
\pi_l(\mathsf{Empty}) & := & \mathsf{Empty} \\
\pi_l(a) & := & \begin{cases} a & \text{falls } l \in \alpha(a) \\ \mathsf{Empty} & \text{falls } l \notin \alpha(a) \end{cases} \\
\pi_l(uop_p(T)) & := & uop_p(\pi_l(T)) \\
\pi_l(bop_p(T_1, T_2)) & := & bop_p(\pi_l(T_1), \pi_l(T_2))
\end{array}$$

Mit struktureller Induktion über T sieht man, dass $\pi_l : \mathrm{Term} \longrightarrow \mathrm{Term}^{\{l\}(\{l\})}$ ist. Es sei angemerkt, dass sowohl die Restriktion R_L^* als auch die Projektion π_l "Projektionen" im Sinne der Mathematik sind, d. h. die Abbildungen sind idempotent, d. h. es gilt $\mathrm{R}_L^* \circ \mathrm{R}_L^* = \mathrm{R}_L^*$ und $\pi_l \circ \pi_l = \pi_l$.

Lemma 4.2 Sei $L \subseteq \mathbb{I}$ und $l \in \mathbb{I}$ beliebig. Dann gilt: Die Restriktion R^*_L und die Projektion π_l kommutieren, d. h. es gilt $\mathrm{R}^*_L \circ \pi_l = \pi_l \circ \mathrm{R}^*_L$.

Beweis. Wir zeigen $\forall\, T \in \text{Term}.\ \mathrm{R}^*_L(\pi_l(T)) = \pi_l(\mathrm{R}^*_L(T))$ mit Hilfe einer strukturellen Induktion über T:

$T = \mathsf{None}(a)$: Dann gilt: $\mathrm{R}^*_L(\pi_l(\mathsf{None}(a))) = \mathrm{R}^*_L(\begin{cases} \mathsf{None}(a) & \text{falls } l \in \alpha(a) \\ \mathsf{Empty} & \text{falls } l \notin \alpha(a) \end{cases})$
$= \begin{cases} \mathsf{None}(a) & \text{falls } l \in \alpha(a) \\ \mathsf{Empty} & \text{falls } l \notin \alpha(a) \end{cases} = \pi_l(\mathsf{None}(a)) = \pi_l(\mathrm{R}^*_L(\mathsf{None}(a)))$.

$T = \mathsf{Empty}$: Dann gilt: $\mathrm{R}^*_L(\pi_l(\mathsf{Empty})) = \mathsf{Empty} = \pi_l(\mathrm{R}^*_L(\mathsf{Empty}))$.

$T = a$: Dann gilt: $\mathrm{R}^*_L(\pi_l(a)) =$

$$= \mathrm{R}^*_L(\begin{cases} a & \text{falls } l \in \alpha(a) \\ \mathsf{Empty} & \text{falls } l \notin \alpha(a) \end{cases}) = \begin{cases} a & \text{falls } l \in \alpha(a) \wedge \alpha(a) \cap L = \emptyset \\ \mathsf{None}(a) & \text{falls } l \in \alpha(a) \wedge \alpha(a) \cap L \neq \emptyset \\ \mathsf{Empty} & \text{falls } l \notin \alpha(a) \end{cases} =$$

$$= \begin{cases} a & \text{falls } \alpha(a) \cap L = \emptyset \wedge l \in \alpha(a) \\ \mathsf{Empty} & \text{falls } \alpha(a) \cap L = \emptyset \wedge l \notin \alpha(a) \\ \mathsf{None}(a) & \text{falls } \alpha(a) \cap L \neq \emptyset \wedge l \in \alpha(a) \\ \mathsf{Empty} & \text{falls } \alpha(a) \cap L \neq \emptyset \wedge l \notin \alpha(a) \end{cases} =$$

$$= \pi_l(\begin{cases} a & \text{falls } \alpha(a) \cap L = \emptyset \\ \mathsf{None}(a) & \text{falls } \alpha(a) \cap L \neq \emptyset \end{cases} = \pi_l(\mathrm{R}^*_L(a)).$$

$T = uop_p(T)$: Straightforward mit I. V. für T.

$T = bop_p(T_1, T_2)$: Straightforward mit I. V. für T_1, T_2. $\square$

4.3 Pfadannotierte globale Semantik

Bevor wir uns dem Regelsystem der lokalen operationalen Semantik zuwenden, geben wir in diesem Abschnitt eine pfadannotierte Fassung der globalen operationalen Semantik aus Abschnitt 3.3 an. Die Konfigurationen dieser modifizierten Semantik sind pfadannotierte Interaktionsterme $T \in$ Term. Die einzige Endkonfiguration ist Empty. Transitionen haben die allgemeine Form $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$ mit $T, T' \in$ Term, $T \neq \mathsf{Empty}$ und $\bar{a} \in \mathbb{A}_\tau$.

Das Regelsystem G der pfadannotierten Fassung der globalen operationalen Semantik ist in Tafel 4.2 angegeben. Die Metavariablen der Regelschemata von G haben die folgenden Bereiche: $T \in$ Term, $a \in \mathbb{A}$ und $\bar{a} \in \mathbb{A}_\tau$. Ist $I \in$ IFrag die Anfangskonfiguration der globalen operationalen Semantik aus Abschnitt 3.3, dann ist $\mathrm{num}(I, 1)$ die korrespondierende Anfangskonfiguration der pfadannotierten Fassung. Es sei warnend darauf hingewiesen, dass in den Regelsystemem G und Gu zwei unterschiedlich definierte Restriktionsfunktionen benutzt werden.

$$
\begin{array}{llll}
(\text{basic}_{\mathrm{G}}) & a \xrightarrow{a}_{\mathrm{G}} \mathsf{Empty} & & \\
(\text{strict}^1_{\mathrm{G}}) & \dfrac{T_1 \xrightarrow{\bar{a}}_{\mathrm{G}} T_1'}{\mathsf{strict}_p(T_1, T_2) \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{strict}_p(T_1', T_2)} & (\text{strict}^2_{\mathrm{G}}) & \mathsf{strict}_p(\mathsf{Empty}, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2 \\
(\text{seq}^1_{\mathrm{G}}) & \dfrac{T_1 \xrightarrow{\bar{a}}_{\mathrm{G}} T_1'}{\mathsf{seq}_p(T_1, T_2) \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{seq}_p(T_1', T_2)} & (\text{seq}^2_{\mathrm{G}}) & \mathsf{seq}_p(\mathsf{Empty}, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2 \\
(\text{seq}^3_{\mathrm{G}}) & \dfrac{T_2 \xrightarrow{\bar{a}}_{\mathrm{G}} T_2'}{\mathsf{seq}_p(T_1, T_2) \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar{a})}(T_1), T_2')} & & \\
(\text{par}^1_{\mathrm{G}}) & \dfrac{T_1 \xrightarrow{\bar{a}}_{\mathrm{G}} T_1'}{\mathsf{par}_p(T_1, T_2) \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{par}_p(T_1', T_2)} & (\text{par}^2_{\mathrm{G}}) & \dfrac{T_2 \xrightarrow{\bar{a}}_{\mathrm{G}} T_2'}{\mathsf{par}_p(T_1, T_2) \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{par}_p(T_1, T_2')} \\
(\text{par}^3_{\mathrm{G}}) & \mathsf{par}_p(\mathsf{Empty}, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2 & (\text{par}^4_{\mathrm{G}}) & \mathsf{par}_p(T_1, \mathsf{Empty}) \xrightarrow{\tau}_{\mathrm{G}} T_1 \\
(\text{loop}^1_{\mathrm{G}}) & \mathsf{loop}_p(T) \xrightarrow{\tau}_{\mathrm{G}} \mathsf{Empty} & (\text{loop}^2_{\mathrm{G}}) & \mathsf{loop}_p(T) \xrightarrow{\tau}_{\mathrm{G}} \mathsf{seq}_p(T, \mathsf{loop}_{p.2}({}_{p.2}T)) \\
(\text{alt}^1_{\mathrm{G}}) & \mathsf{alt}_p(T_1, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_1 & (\text{alt}^2_{\mathrm{G}}) & \mathsf{alt}_p(T_1, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2
\end{array}
$$

Tafel 4.2: Pfadannotierte Fassung der globalen operationalen Semantik

Satz 4.3 Die Wohlnummeriertheit des Konfigurationsterms T ist eine Invariante der pfadannotierten Fassung der globalen operationalen Semantik G. Genauer gesagt gelten die folgenden zwei Aussagen:

1. Sei $I \in \mathrm{IFrag}$ und $p \in \mathrm{Path}$. Dann ist $\mathrm{num}(I, p)$ wohlnummeriert.
2. Seien $T, T' \in \mathrm{Term}$, T wohlnummeriert und $\bar{a} \in \mathbb{A}_\tau$. Es gelte $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$. Dann ist auch T' wohlnummeriert und es gilt:
   (a) $\mathrm{top}(T) = \bot \implies \mathrm{top}(T') = \bot$
   (b) $\mathrm{top}(T) = p \implies \exists\, q \in \mathrm{Path}.\ \mathrm{top}(T') \in \{\bot, p.q\}$.

Beweis. Zu (1). Der Beweis erfolgt durch strukturelle Induktion über I:

$I = \mathit{const} \in \{\mathsf{Empty}, a\}$: Dann ist $\mathrm{num}(I, p) = \mathit{const}$ wohlnummeriert.

$I = \mathit{uop}(I_1)$: Dann ist $\mathrm{num}(I, p) = \mathrm{num}(\mathit{uop}(I_1), p) = \mathit{uop}_p(\mathrm{num}(I_1, p.1))$.
Falls $I_1 = \mathit{const}$ ist, dann ist $\mathrm{num}(I_1, p.1) = \mathit{const}$. Also ist $\mathrm{num}(I_1, p.1)$ wohlnummeriert und $\mathrm{top}(\mathrm{num}(I_1, p.1)) = \bot$. Somit ist $\mathrm{num}(I, p)$ wohlnummeriert.
Falls $I_1 \neq \mathit{const}$ ist, dann ist $\mathrm{top}(\mathrm{num}(I_1, p.1)) = p.1$ (wegen Def. von num). Nach I. V. ist $\mathrm{num}(I_1, p.1)$ wohlnummeriert. Also ist $\mathrm{num}(I, p)$ wohlnummeriert.

$I = \mathit{bop}(I_1, I_2)$: Analog zu $I = \mathit{uop}(I_1)$ mit zwei Teiltermen I_1 und I_2.

Zu (2). Der Beweis erfolgt durch Ableitungsinduktion für Judgements $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$:

$(\text{basic}_{\mathrm{G}})$ Dann ist $T = a$ und $T' = \mathsf{Empty}$. Also ist T' wohlnummeriert und es gilt $\mathrm{top}(T) = \mathrm{top}(T') = \bot$.

(strict$^1_{\mathrm{G}}$) Dann ist $T = \mathsf{strict}_p(T_1, T_2)$, $T' = \mathsf{strict}_p(T_1', T_2)$ und es gilt $T_1 \xrightarrow{\bar{a}}_{\mathrm{G}} T_1'$. Weil nach Voraussetzung T wohlnummeriert ist, sind T_1 und T_2 wohlnummeriert und es gilt $\mathrm{top}(T_1) \in \{\perp, p.1.q_1\}$, $\mathrm{top}(T_2) \in \{\perp, p.2.q_2\}$ mit $q_1, q_2 \in \mathrm{Path}$. Mit der I. V. für den Teilableitungsbaum des Judgements $T_1 \xrightarrow{\bar{a}}_{\mathrm{G}} T_1'$ folgt: T_1' ist wohlnummeriert und es gibt ein $q \in \mathrm{Path}$, so dass gilt $\mathrm{top}(T_1') \in \{\perp, p.1.q_1.q\}$. Somit ist $T' = \mathsf{strict}_p(T_1', T_2)$ wohlnummeriert.

(strict$^2_{\mathrm{G}}$) Dann ist $T = \mathsf{strict}_p(\mathsf{Empty}, T_2)$ und $T' = T_2$. Weil T wohlnummeriert ist, ist $T' = T_2$ wohlnummeriert und es gilt $\mathrm{top}(T') \in \{\perp, p.2.q_2\}$ mit $q_2 \in \mathrm{Path}$.

Die Fälle (seq$^2_{\mathrm{G}}$), (par$^3_{\mathrm{G}}$), (par$^4_{\mathrm{G}}$), (alt$^1_{\mathrm{G}}$), (alt$^2_{\mathrm{G}}$) sind analog zu Fall (strict$^2_{\mathrm{G}}$). Die Fälle (seq$^1_{\mathrm{G}}$), (seq$^3_{\mathrm{G}}$), (par$^1_{\mathrm{G}}$), (par$^2_{\mathrm{G}}$) sind analog zu Fall (strict$^1_{\mathrm{G}}$). Im Fall (seq$^3_{\mathrm{G}}$) ist zu berücksichtigen, dass mit T auch $\mathrm{R}^*_L(T)$ wohlnummeriert ist (trivial!) und $\mathrm{top}(\mathrm{R}^*_L(T)) = \mathrm{top}(T)$ gilt.

(loop$^1_{\mathrm{G}}$) Dann ist $T = \mathsf{loop}_p(T_1)$ und $T' = \mathsf{Empty}$. Also ist T' wohlnummeriert und es gilt $\mathrm{top}(T) = p \neq \perp$ und $\mathrm{top}(T') = \perp$.

(loop$^2_{\mathrm{G}}$) Dann ist $T = \mathsf{loop}_p(T_1)$ und $T' = \mathsf{seq}_p(T_1, \mathsf{loop}_{p.2}({}_{p.2}T_1))$. Weil T wohlnummeriert ist, ist T_1 wohlnummeriert und $\mathrm{top}(T_1) \in \{\perp, p.1.q\}$ mit $q \in \mathrm{Path}$.

Fall 1: $\mathrm{top}(T_1) = \perp$: Dann ist $T_1 = \mathit{const}$ und folglich auch ${}_{p.2}T_1 = \mathit{const}$. Somit ist $\mathsf{loop}_{p.2}({}_{p.2}T_1)$ wohlnummeriert mit $\mathrm{top} = p.2$. Es folgt, dass T' wohlnummeriert ist mit $\mathrm{top}(T') = p$.

Fall 2: $\mathrm{top}(T_1) = p.1.q$: Weil T_1 wohlnummeriert ist und $\mathrm{top}(T_1) = p.1.q$ gilt, folgt mit Lemma 4.1, dass ${}_{p.2}T_1$ wohlnummeriert ist mit $\mathrm{top}({}_{p.2}T_1) = p.2.1.q$. Somit ist $\mathsf{loop}_{p.2}({}_{p.2}T_1)$ wohlnummeriert mit $\mathrm{top} = p.2$. Es folgt, dass T' wohlnummeriert ist mit $\mathrm{top}(T') = p$. □

4.4 Konfigurationen und Transitionen

Wir geben uns eine endliche Lifelinemenge $L \subseteq \mathbb{I}$ fest vor. Eine leere Lifelinemenge ist erlaubt. Es sei $L = \{l_1, \ldots, l_n\}$ mit $l_1, \ldots, l_n$ paarweise verschieden; es gilt $n \geq 0$. Wir setzen $\mathrm{LConf}^L := \{\lambda : L \longrightarrow \mathrm{Term} \mid \forall\, l \in L.\, \lambda(l) \in \mathrm{Term}^{\{l\}(\{l\})}\}$. Als Bereich der gemeinsam genutzten Information h (siehe Einführungstext zu Kapitel 4) vereinbaren wir $\mathrm{H}^L := \mathrm{Path} \longrightarrow \wp(L \cup \{1, 2\})$, wobei wir $\mathbb{I} \cap \{1, 2\} = \emptyset$ voraussetzen. Eine Konfiguration γ unserer lokalen operationalen Semantik (zur Lifelinemenge L) ist ein Element aus $\mathrm{Conf}^L := \mathrm{LConf}^L \times \mathrm{H}^L$. Wir schreiben eine Konfiguration $\gamma = (\lambda, h)$ in der Form $[l_1 : T_{l_1}, \ldots, l_n : T_{l_n}, h]$ mit $T_{l_1} = \lambda(l_1)$, …, $T_{l_n} = \lambda(l_n)$. Wir vereinbaren $\mathrm{TConf}^L := \{\, [l_1 : \mathsf{Empty}, \ldots, l_n : \mathsf{Empty}, h] \mid h \in \mathrm{H}^L\}$ als Menge der Endkonfigurationen. Falls die Lifelinemenge L leer ist, dann ist *jede* Konfiguration eine Endkonfiguration. Eine Transition der lokalen operationalen Semantik hat die allgemeine Form $\gamma \xrightarrow{\bar{a}}_{\mathrm{L}} \gamma'$ mit $\gamma, \gamma' \in \mathrm{Conf}^L$, $\gamma \notin \mathrm{TConf}^L$ und $\bar{a} \in \mathbb{A}_\tau$.

Das Regelsystem L der lokalen operationalen Semantik für UML 2.0-Interaktionen (über der Lifelinemenge L) ist in Tafel 4.3 angegeben. Wir werden das Regelsystem L in Abschnitt 4.5 näher diskutieren.

Ist $T \in \mathrm{Term}$ eine Anfangskonfiguration der pfadannotierten Fassung der globalen operationalen Semantik (siehe Abschnitt 4.3), dann wählen wir als Lifelinemenge für die lokale operationale Semantik $L := \alpha(T)$. Die zu T korrespondierende Anfangskonfiguration der lokalen operationalen Semantik ist $[l_1 : \pi_{l_1}(T), \ldots, l_n : \pi_{l_n}(T), \emptyset]$. Hierbei ist $\emptyset \in \mathrm{H}^L$ und $\emptyset(p) := \emptyset$ für alle $p \in \mathrm{Path}$.

4.5 Lokale operationale Semantik

Das Regelsystem L der lokalen operationalen Semantik für UML 2.0-Interaktionen (über der Lifelinemenge L) ist in Tafel 4.3 angegeben. Zur bündigen Darstellung wurden übliche Shorthand Notations benutzt.

Die Ableitungsbäume des Regelsystems L sind linear gebaut, d. h. sie haben jeweils nur einen Blattknoten B, der einer Axiomeninstanz $l : T, h \xrightarrow{\bar{a}}_{\mathrm{L}} l : T', h'$ entspricht. Je nachdem, welches Axiomenschema für diesen Blattknoten B instantiiert wurde, ist die Art des abgeleiteten semantischen Schrittes zu unterscheiden:

1. Wurde B mit ($\mathrm{basic}_{\mathrm{L}}$) instantiiert, so sprechen wir von der Ausführung einer *echten Aktion* $\bar{a} = a$ durch die Lifeline l.
2. Wurde B mit ($\mathrm{alt}^1_{\mathrm{L}}$), ($\mathrm{alt}^2_{\mathrm{L}}$), ($\mathrm{loop}^1_{\mathrm{L}}$) oder ($\mathrm{loop}^2_{\mathrm{L}}$) instantiiert und ist $h(p) = \emptyset$ für $p = \mathrm{top}(T)$, dann sprechen wir von der erstmaligen Ausführung eines *Entscheidungsschrittes* durch die Lifeline l. Die Entscheidung der Lifeline l wird im gemeinsam benutzten Kommunikationsbereich h bekanntgegeben. Das heißt es wird $h(p)$ auf $\{1\}$ gesetzt, falls mit ($\mathrm{alt}^1_{\mathrm{L}}$) ein linker Teilterm ausgewählt oder mit ($\mathrm{loop}^1_{\mathrm{L}}$) eine Schleife abgebrochen wird. Entsprechend wird $h(p)$ auf $\{2\}$ gesetzt, falls mit ($\mathrm{alt}^2_{\mathrm{L}}$) ein rechter Teilterm ausgewählt oder mit ($\mathrm{loop}^2_{\mathrm{L}}$) eine Schleife einfach ausgerollt wird (Ausrollschritt).
3. Wurde B mit ($\mathrm{strict}^3_{\mathrm{L}}$) instantiiert, so sprechen wir von einem *Synchronisationsschritt* der Lifeline l. Die Lifeline l meldet im gemeinsam benutzten Kommunikationsbereich h, dass sie auf einen Teilterm $T = \mathsf{strict}_p(\mathsf{Empty}, T_2)$ gestoßen ist, das heißt die Lifeline ist nunmehr mit der Bearbeitung des linken Teilterms dieses strict_p-Konstrukts fertig. Der lokale Konfigurationsterm der Lifeline l bleibt hierbei unverändert. Erst wenn *alle* Lifelines mit der Bearbeitung des linken Teilterms dieses strict_p-Konstrukts fertig sind und dies mittels ($\mathrm{strict}^3_{\mathrm{L}}$) in h gemeldet haben ($h(p) = L$), erhalten alle Lifelines die Erlaubnis zur Anwendung von ($\mathrm{strict}^2_{\mathrm{L}}$) und somit zur Bearbeitung des rechten Teilterms des strict_p-Konstrukts. Das Axiomenschema ($\mathrm{strict}^3_{\mathrm{L}}$) dient folglich einer globalen Synchronisation der Termbearbeitung über alle Lifelines hinweg.

$$
\begin{array}{rl}
(\mathrm{basic_L}) & a \xrightarrow{a}_{\mathrm{L}} \mathsf{Empty} \\[1ex]
(\mathrm{strict}^1_{\mathrm{L}}) & \dfrac{T_1 \xrightarrow{\bar a}_{\mathrm{L}} T_1'}{\mathsf{strict}_p(T_1,T_2) \xrightarrow{\bar a}_{\mathrm{L}} \mathsf{strict}_p(T_1',T_2)} \\[1ex]
(\mathrm{strict}^2_{\mathrm{L}}) & \mathsf{strict}_p(\mathsf{Empty},T_2),h \xrightarrow{\tau}_{\mathrm{L}} T_2,h \quad \text{falls } h(p)=L \\[1ex]
(\mathrm{strict}^3_{\mathrm{L}}) & l:\mathsf{strict}_p(\mathsf{Empty},T_2),h \xrightarrow{\tau}_{\mathrm{L}} l:\mathsf{strict}_p(\mathsf{Empty},T_2),h[p\mapsto h(p)\cup\{l\}] \quad \text{falls } l\notin h(p) \\[1ex]
(\mathrm{seq}^1_{\mathrm{L}}) & \dfrac{T_1 \xrightarrow{\bar a}_{\mathrm{L}} T_1'}{\mathsf{seq}_p(T_1,T_2) \xrightarrow{\bar a}_{\mathrm{L}} \mathsf{seq}_p(T_1',T_2)} \qquad (\mathrm{seq}^2_{\mathrm{L}}) \;\; \mathsf{seq}_p(\mathsf{Empty},T_2) \xrightarrow{\tau}_{\mathrm{L}} T_2 \\[1ex]
(\mathrm{seq}^3_{\mathrm{L}}) & \dfrac{T_2 \xrightarrow{\bar a}_{\mathrm{L}} T_2'}{\mathsf{seq}_p(T_1,T_2) \xrightarrow{\bar a}_{\mathrm{L}} \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar a)}(T_1),T_2')} \\[1ex]
(\mathrm{par}^1_{\mathrm{L}}) & \dfrac{T_1 \xrightarrow{\bar a}_{\mathrm{L}} T_1'}{\mathsf{par}_p(T_1,T_2) \xrightarrow{\bar a}_{\mathrm{L}} \mathsf{par}_p(T_1',T_2)} \qquad (\mathrm{par}^2_{\mathrm{L}}) \;\; \dfrac{T_2 \xrightarrow{\bar a}_{\mathrm{L}} T_2'}{\mathsf{par}_p(T_1,T_2) \xrightarrow{\bar a}_{\mathrm{L}} \mathsf{par}_p(T_1,T_2')} \\[1ex]
(\mathrm{par}^3_{\mathrm{L}}) & \mathsf{par}_p(\mathsf{Empty},T_2) \xrightarrow{\tau}_{\mathrm{L}} T_2 \qquad (\mathrm{par}^4_{\mathrm{L}}) \;\; \mathsf{par}_p(T_1,\mathsf{Empty}) \xrightarrow{\tau}_{\mathrm{L}} T_1 \\[1ex]
(\mathrm{loop}^1_{\mathrm{L}}) & \mathsf{loop}_p(T),h \xrightarrow{\tau}_{\mathrm{L}} \mathsf{Empty},h[p\mapsto\{1\}] \quad \text{falls } h(p)\cap\{2\}=\emptyset \\[1ex]
(\mathrm{loop}^2_{\mathrm{L}}) & \mathsf{loop}_p(T),h \xrightarrow{\tau}_{\mathrm{L}} \mathsf{seq}_p(T,\mathsf{loop}_{p.2}({}_{p.2}T)),h[p\mapsto\{2\}] \quad \text{falls } h(p)\cap\{1\}=\emptyset \\[1ex]
(\mathrm{alt}^1_{\mathrm{L}}) & \mathsf{alt}_p(T_1,T_2),h \xrightarrow{\tau}_{\mathrm{L}} T_1,h[p\mapsto\{1\}] \quad \text{falls } h(p)\cap\{2\}=\emptyset \\[1ex]
(\mathrm{alt}^2_{\mathrm{L}}) & \mathsf{alt}_p(T_1,T_2),h \xrightarrow{\tau}_{\mathrm{L}} T_2,h[p\mapsto\{2\}] \quad \text{falls } h(p)\cap\{1\}=\emptyset
\end{array}
$$

Tafel 4.3: Lokale operationale Semantik

4. Wurde B mit $(\mathrm{alt}^1_{\mathrm{L}})$, $(\mathrm{alt}^2_{\mathrm{L}})$, $(\mathrm{loop}^1_{\mathrm{L}})$ oder $(\mathrm{loop}^2_{\mathrm{L}})$ instantiiert und ist $h(p) \in \{\{1\},\{2\}\}$ für $p = \mathrm{top}(T)$, dann sprechen wir von einem *Nachführschritt*. Das heißt die Lifeline l führt selbst keine neue Entscheidung aus, sondern führt lediglich die bereits getroffene und in h bekanntgegebene Entscheidung einer anderen Lifeline nach. Wurde B mit $(\mathrm{strict}^2_{\mathrm{L}})$ instantiiert, dann sprechen wir ebenfalls von einem Nachführschritt—und zwar aus Gründen, die später noch klar werden.
5. Wurde B mit $(\mathrm{seq}^2_{\mathrm{L}})$, $(\mathrm{par}^3_{\mathrm{L}})$ oder $(\mathrm{par}^4_{\mathrm{L}})$ instantiiert, dann sprechen wir von einem *Aufräumschritt*.

4.6 Invarianten

Wir geben eine informelle Übersicht über alle relevanten Invarianzeigenschaften von Konfigurationen $[l_1\!:\!T_{l_1},\ldots,l_n\!:\!T_{l_n},h]$ der lokalen operationalen Semantik:

(I1) T_l ist wohlnummeriert für jedes $l \in L$.

(I2) $\mathrm{Dom}(h) := \{p \in \mathrm{Path} \mid h(p) \neq \emptyset\}$ ist endlich.

(I3) $\mathrm{Ran}(h) := \{h(p) \mid p \in \mathrm{Path}\} \subseteq \{\{1\},\{2\}\} \mathbin{\dot\cup} \wp(L)$.

(I4) Für jeden Schritt $(\lambda, h) \xrightarrow{\bar{a}}_{\mathrm{L}} (\lambda', h')$ und jedes $p \in \mathrm{Path}$ ist $h(p) \leq h'(p)$. Dabei ist $\leq := (\{(\emptyset,\{1\}),(\emptyset,\{2\})\} \cup \subseteq)^*$ und $\subseteq$ die Teilmengenordnung auf $\wp(L)$.

(I5) Falls $\mathsf{strict}_p(T_1, T_2)$ als Teilterm von T_l auftritt (für mindestens ein $l \in L$) und $h(p) \neq L$ ist, dann ist $h(p.2.r) = \emptyset$ für alle $r \in \mathrm{Path}$.

(I6) Falls $\mathsf{strict}_p(T_1, T_2)$ als Teilterm von T_l auftritt und $h(p) = L$ ist, dann ist $T_1 = \mathsf{Empty}$.

(I7) Falls $\mathsf{loop}_p(T)$ als Teilterm von T_l auftritt (für mindestens ein $l \in L$) und $h(p) = \emptyset$, dann ist $h(p.r) = \emptyset$ für alle $r \in \mathrm{Path}$.

(I8) Falls $\mathsf{alt}_p(T_1, T_2)$ als Teilterm von T_l auftritt (für mindestens ein $l \in L$) und $h(p) = \emptyset$, dann ist $h(p.r) = \emptyset$ für alle $r \in \mathrm{Path}$.

(I9) Falls $\mathsf{loop}_p(T)$ oder $\mathsf{alt}_p(T_1, T_2)$ als Teilterm von T_l auftritt (für mindestens ein $l \in L$) und $h(p) \notin \{\{1\},\{2\}\}$, dann ist $h(p) = \emptyset$ (und nicht etwa eine nichtleere Teilmenge von L).

4.7 Konstruktion der Beobachtungsäquivalenz

Wir setzen $\mathrm{H}^L_{\mathrm{fin}} := \{h \in \mathrm{H}^L \mid \mathrm{Dom}(h) \text{ ist endlich}\}$. Zwischen den Konfigurationen der pfadannotierten Fassung der globalen operationalen Semantik und den Konfigurationen der lokalen operationalen Semantik definieren wir eine Relation $\beta \subseteq \mathrm{Term}^{L(L)} \times \mathrm{Conf}^L$ wie folgt:

$$\beta(T, [l_1 \colon T_{l_1}, \ldots, l_n \colon T_{l_n}, h]) :\Longleftrightarrow \forall\, i \in \{1, \ldots, n\}.\, \pi_{l_i}(T) \Rightarrow^* \mathrm{sla}(T_{l_i}, h)$$

Die Abstraktionsfunktion sla wird hierbei in Abschnitt 4.7.1 definiert. Das Symbol $\Rightarrow^*$ ist die reflexiv-transitive Hülle einer Reduktionsrelation $\Rightarrow$, die in Abschnitt 4.7.2 definiert wird.

4.7.1 Die Abstraktionsfunktion sla

Die Funktion sla ("sla" steht für "strict-loop-alt"), die in der Konstruktion der Relation β auftritt, ist eine Abstraktionsfunktion, welche rekursiv definiert ist wie folgt:

$$
\begin{array}{lll}
\mathrm{sla} : \mathrm{Term}^{L(L)} \times \mathrm{H}^{L}_{\mathrm{fin}} \longrightarrow \mathrm{Term}^{L(L)} & & \\
\mathrm{sla}(\mathsf{None}(a), h) & := & \mathsf{None}(a) \\
\mathrm{sla}(\mathsf{Empty}, h) & := & \mathsf{Empty} \\
\mathrm{sla}(a, h) & := & a \\
\mathrm{sla}(\mathsf{strict}_p(T_1, T_2), h) & := & \begin{cases} \mathrm{sla}(T_2, h) & \text{f. } h(p) = L \\ \mathsf{strict}_p(\mathrm{sla}(T_1, h), T_2) & \text{f. } h(p) \neq L \end{cases} \\
\mathrm{sla}(\mathsf{seq}_p(T_1, T_2), h) & := & \mathsf{seq}_p(\mathrm{sla}(T_1, h), \mathrm{sla}(T_2, h)) \\
\mathrm{sla}(\mathsf{par}_p(T_1, T_2), h) & := & \mathsf{par}_p(\mathrm{sla}(T_1, h), \mathrm{sla}(T_2, h)) \\
\mathrm{sla}(\mathsf{loop}_p(T), h) & := & \begin{cases} \mathsf{Empty} & \text{f. } h(p) = \{1\} \\ \mathsf{seq}_p(\mathrm{sla}(T, h), \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T), h)) & \text{f. } h(p) = \{2\} \\ \mathsf{loop}_p(T) & \text{f. } h(p) \notin \{\{1\}, \{2\}\} \end{cases} \\
\mathrm{sla}(\mathsf{alt}_p(T_1, T_2), h) & := & \begin{cases} \mathrm{sla}(T_1, h) & \text{f. } h(p) = \{1\} \\ \mathrm{sla}(T_2, h) & \text{f. } h(p) = \{2\} \\ \mathsf{alt}_p(T_1, T_2) & \text{f. } h(p) \notin \{\{1\}, \{2\}\} \end{cases}
\end{array}
$$

Bevor wir formal beweisen, dass die Funktion sla wohldefiniert ist (man beachte die nichtkompositionale Klausel für loop_p im Falle $h(p) = \{2\}$), wollen wir zunächst die Motivation für diese Abstraktionsfunktion klären. Die Funktion sla hat mit Entscheidungs-, Synchronisations- und Nachführschritten zu tun (siehe Abschnitt 4.5). Wenn eine Lifeline l einen Entscheidungsschritt für ein loop_p- oder alt_p-Konstrukt erstmalig ausführt, dann wird—je nach getroffener Entscheidung—der Wert $h(p)$ auf $\{1\}$ bzw. $\{2\}$ gesetzt und somit die sla-Abstraktionsfunktion für dieses p "eingeschaltet". Das heißt, die Funktion sla "faked" innerhalb der Verträglichkeitsbeziehung β für alle Lifelines, die den Entscheidungsschritt zu diesem loop_p- oder alt_p-Konstrukt noch nicht nachgeführt haben, eben diesen Nachführschritt. Sobald also die erste Lifeline ein bestimmtes loop_p oder alt_p Konstrukt bearbeitet hat, tun wir so, als ob bereits alle Lifelines dieses loop_p- bzw. alt_p-Konstrukt bearbeitet hätten. Ähnliches gilt für Nachführschritte, die sich auf Synchronisationsschritte beziehen. Wenn die letzte Lifeline l den Synchronisationsschritt für ein Konstrukt $\mathsf{strict}_p(\mathsf{Empty}, T_2)$ ausgeführt, wechselt der Wert $h(p)$ von $L \setminus \{l\}$ auf L und die sla-Abstraktionsfunktion wird für dieses p "eingeschaltet". Das heißt, die Funktion sla "faked" für alle Lifelines, die das Konstrukt $\mathsf{strict}_p(\mathsf{Empty}, T_2)$ noch nicht durch T_2 ersetzt haben, eben diese Ersetzung—welche wir ebenfalls einen "Nachführschritt" genannt haben. Es läßt sich zusammenfassend sagen, dass die Funktion sla alle Konfigurationen der lokalen Semantik, die sich nur durch Nachführschritte unterscheiden, zu einer Äquivalenzklasse zusammenfasst. Kurz: Die Funktion sla abstrahiert über Nachführschritte.

Zum Beweis der Wohldefiniertheit der Funktion sla definieren wir für jedes gegebene $h \in \mathrm{H}^L_{\mathrm{fin}}$ eine wohlfundierte binäre Relation $\prec^h$ auf $\mathrm{Term}^{L(L)}$ wie folgt:

$$\begin{aligned} T' \prec^h T \;:\Longleftrightarrow\; & T' \text{ ist direkter Teilterm von } T \text{ oder} \\ & \exists\, \widetilde{T} \in \mathrm{Term}^{L(L)}, p \in \mathrm{Path}\,.\, \big[\, h(p) = \{2\} \;\wedge \\ & \qquad T' = \mathsf{loop}_{p.2}({}_{p.2}\widetilde{T}) \;\wedge\; T = \mathsf{loop}_p(\widetilde{T}) \,\big] \end{aligned}$$

Lemma 4.4 Es sei $h \in \mathrm{H}^L_{\mathrm{fin}}$. Dann ist die Relation $\prec^h$ wohlfundiert.

Beweis. Angenommen die Relation $\prec^h$ wäre nicht wohlfundiert. Dann gäbe es eine unendliche absteigende Kette $T_0 \succ^h T_1 \succ^h T_2 \succ^h \ldots$ in $\mathrm{Term}^{L(L)}$. Für jedes $n \in \mathbb{N}_0$ setzen wir $\bar{I}_n := \mathrm{denum}(T_n)$. Weil der Entnummerierungsoperator denum offenkundig monoton ist bezüglich der direkten Teiltermrelationen $\sqsubset_{\mathrm{Term}}$ und $\sqsubset_{\mathrm{NIFrag}}$ und weil $\mathrm{denum}(\mathsf{loop}_{p.2}({}_{p.2}\widetilde{T})) = \mathrm{denum}(\mathsf{loop}_p(\widetilde{T}))$ ist, folgt $\bar{I}_0 \sqsupseteq_{\mathrm{NIFrag}} \bar{I}_1 \sqsupseteq_{\mathrm{NIFrag}} \bar{I}_2 \sqsupseteq_{\mathrm{NIFrag}} \ldots$. Dabei gilt für jedes $n \in \mathbb{N}_0$ die Gleichheit $\bar{I}_n = \bar{I}_{n+1}$ genau dann, wenn es $\widetilde{T} \in \mathrm{Term}^{L(L)}$ und $p \in \mathrm{Path}$ gibt, so dass $h(p) = \{2\}$ und $T_{n+1} = \mathsf{loop}_{p.2}({}_{p.2}\widetilde{T})$ und $T_n = \mathsf{loop}_p(\widetilde{T})$ ist. Weil die direkte Teiltermrelation $\sqsubset_{\mathrm{NIFrag}}$ wohlfundiert ist, muss ein $n \in \mathbb{N}_0$ existieren, so dass für alle $m \geq n$ gilt: $\bar{I}_m = \bar{I}_{m+1}$. Für alle $m \geq n$ gibt es folglich $\widetilde{T} \in \mathrm{Term}^{L(L)}$ und $p \in \mathrm{Path}$, so dass $h(p) = \{2\}$ und $T_{m+1} = \mathsf{loop}_{p.2}({}_{p.2}\widetilde{T})$ und $T_m = \mathsf{loop}_p(\widetilde{T})$ ist. Daraus folgt die Existenz von unendlich vielen $p \in \mathrm{Path}$ mit $h(p) = \{2\}$. Dies ist ein Widerspruch zur vorausgesetzen Endlichkeit von $\mathrm{Dom}(h)$. □

Korollar 4.5 Die Abstraktionsfunktion sla ist wohldefiniert.

Beweis. Siehe Lemma 4.4 und der Satz über wohlfundierte Rekursion. □

Lemma 4.6 Es sei $h \in \mathrm{H}^L_{\mathrm{fin}}$. Dann ist $\mathrm{sla}(-, h) : \mathrm{Term}^{L(L)} \longrightarrow \mathrm{Term}^{L(L)}$ eine Projektion, d. h. für alle $T \in \mathrm{Term}^{L(L)}$ gilt $\mathrm{sla}(\mathrm{sla}(T, h), h) = \mathrm{sla}(T, h)$.

Beweis. Sei $h \in \mathrm{H}^L_{\mathrm{fin}}$. Wir beweisen $\forall\, T \in \mathrm{Term}^{L(L)}.\ \mathrm{sla}(\mathrm{sla}(T, h), h) = \mathrm{sla}(T, h)$ durch eine noethersche Induktion entlang der wohlfundierten Relation $\prec^h$. Wir führen nur die Fälle zu den Konstrukten strict_p und loop_p aus:

$T = \mathsf{strict}_p(T_1, T_2)$: Wir unterscheiden zwei Fälle:

Fall 1: $h(p) = L$: Dann ist $\mathrm{sla}(T, h) = \mathrm{sla}(T_2, h)$. Wegen $T_2 \prec^h T$ und I. V. gilt $\mathrm{sla}(\mathrm{sla}(T_2, h), h) = \mathrm{sla}(T_2, h)$. Daraus folgt: $\mathrm{sla}(\mathrm{sla}(T, h), h) = \mathrm{sla}(\mathrm{sla}(T_2, h), h) = \mathrm{sla}(T_2, h) = \mathrm{sla}(T, h)$.

Fall 2: $h(p) \neq L$: Dann ist $\mathrm{sla}(T, h) = \mathsf{strict}_p(\mathrm{sla}(T_1, h), T_2)$. Wegen $T_1 \prec^h T$ und wegen der I. V. gilt $\mathrm{sla}(\mathrm{sla}(T_1, h), h) = \mathrm{sla}(T_1, h)$. Daraus folgt:
$\mathrm{sla}(\mathrm{sla}(T, h), h) = \mathrm{sla}(\mathsf{strict}_p(\mathrm{sla}(T_1, h), T_2), h) =$ [wegen $h(p) \neq L$]
$\mathsf{strict}_p(\mathrm{sla}(\mathrm{sla}(T_1, h), h), T_2) = \mathsf{strict}_p(\mathrm{sla}(T_1, h), T_2) = \mathrm{sla}(T, h)$.

$T = \mathsf{loop}_p(T_1)$: Wir unterscheiden drei Fälle:

Fall 1: $h(p) \notin \{\{1\}, \{2\}\}$: Dann ist $\mathrm{sla}(T, h) = \mathsf{loop}_p(T_1) = T$.
Es folgt: $\mathrm{sla}(\mathrm{sla}(T, h), h) = \mathrm{sla}(T, h)$.

Fall 2: $h(p) = \{1\}$: Dann ist $\mathrm{sla}(T, h) = \mathsf{Empty}$.
Es folgt: $\mathrm{sla}(\mathrm{sla}(T, h), h) = \mathrm{sla}(\mathsf{Empty}, h) = \mathsf{Empty} = \mathrm{sla}(T, h)$.

Fall 3: $h(p) = \{2\}$: Dann ist $\mathrm{sla}(T, h) = \mathsf{seq}_p(\mathrm{sla}(T_1, h), \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h))$. Es gilt $T_1 \prec^h T$ und $\mathsf{loop}_{p.2}({}_{p.2}T_1) \prec^h T$. Mit der I. V. folgt: $\mathrm{sla}(\mathrm{sla}(T_1, h), h) = \mathrm{sla}(T_1, h)$ und $\mathrm{sla}(\mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h), h) = \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h)$. Daraus folgt:
$\mathrm{sla}(\mathrm{sla}(T, h), h) = \mathrm{sla}(\mathsf{seq}_p(\mathrm{sla}(T_1, h), \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h)), h) =$
$\mathsf{seq}_p(\mathrm{sla}(\mathrm{sla}(T_1, h), h), \mathrm{sla}(\mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h), h)) =$
$\mathsf{seq}_p(\mathrm{sla}(T_1, h), \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h)) = \mathrm{sla}(T, h)$. □

Bemerkungen: (1) $\alpha(\mathrm{sla}(T, h)) \subseteq \alpha(T)$
(2) $\nu(\mathrm{sla}(T, h)) \subseteq \nu(T)$

Diese Bemerkungen können leicht durch eine noethersche Induktion entlang der Relation $\prec^h$ bewiesen werden.

4.7.2 Die Reduktionsrelation $\Rightarrow$

Die Reduktionsrelation $\Rightarrow \subseteq (\mathrm{Term}^{L(L)})^2$, die in der Konstruktion der Relation β auftritt, ist definiert durch folgendes Termersetzungssystem T:

[T1]	$C[\mathsf{seq}_p(\mathsf{Empty}, T_2)] \Rightarrow C[T_2]$	mit $C \in \text{Context} ::= \bullet \mid \mathsf{strict}_{p'}(C_1, T_2)$
[T2]	$C[\mathsf{par}_p(\mathsf{Empty}, T_2)] \Rightarrow C[T_2]$	$\mid \mathsf{seq}_{p'}(C_1, T_2) \mid \mathsf{seq}_{p'}(T_1, C_2)$
[T3]	$C[\mathsf{par}_p(T_1, \mathsf{Empty})] \Rightarrow C[T_1]$	$\mid \mathsf{par}_{p'}(C_1, T_2) \mid \mathsf{par}_{p'}(T_1, C_2)$

wobei hier stets $T_1, T_2 \in \mathrm{Term}^{L(L)}$ ist.

Die Reduktionsrelation $\Rightarrow$ hat mit Aufräumschritten zu tun. Wenn die globale operationale Semantik einen Aufräumschritt ausführt, dann können wir dies in der lokalen Semantik problemlos nachbilden, indem *jede* Lifeline den besagten (τ-) Aufräumschritt ausführt. In der Gegenrichtung ist die Situation etwas schwieriger: Einen Aufräumschritt der lokalen Semantik, der von einer *einzelnen* Lifeline l ausgeführt wird, werden wir in der globalen Semantik nur trivial nachbilden (d. h. die globale Konfiguration bleibt unverändert), weil der Aufräumschritt von den restlichen Lifelines $L \setminus \{l\}$ u. U. noch nicht ausgeführt wurde. Damit die unveränderte globale Konfiguration mit der durch den Aufräumschritt veränderten lokalen Konfiguration verträglich bleibt, abstrahieren wir in der Definition der Verträglichkeitsbeziehung mittels der Relation $\Rightarrow^*$ über Aufräumschritte.

Die Regeln [T1], [T2] und [T3] des Termersetzungssystems T entsprechen genau den Axiomen der operationalen Semantiken, die für Aufräumschritte verantwortlich sind (siehe Abschnitte 3.3 und 4.5). Dabei müssen wir nur solche Kontexte

$C \in$ Context berücksichtigen, innerhalb der durch die operationalen Semantiken ein Aufräumschritt ausgeführt werden kann. Weil die Regelsysteme der operationalen Semantiken keine Regeln vorsehen, mit deren Hilfe in die Teilterme eines loop_p- oder alt_p-Konstruktes "hineingegriffen" werden könnte, können wir auf loop_p- und alt_p-Konstrukte im Kontext C verzichten. Entsprechendes gilt für den rechten Teilterm eines strict_p-Konstruktes.

Die Einsetzung eines Terms $T \in \mathrm{Term}^{L(L)}$ in einen Kontext C liefert einen Term $C[T] \in \mathrm{Term}^{L(L)}$ und ist definiert wie folgt:

$$\begin{array}{lcl}
\bullet[T] & := & T \\
\mathsf{strict}_{p'}(C_1, T_2)[T] & := & \mathsf{strict}_{p'}(C_1[T], T_2) \\
\mathsf{seq}_{p'}(C_1, T_2)[T] & := & \mathsf{seq}_{p'}(C_1[T], T_2) \\
\mathsf{seq}_{p'}(T_1, C_2)[T] & := & \mathsf{seq}_{p'}(T_1, C_2[T]) \\
\mathsf{par}_{p'}(C_1, T_2)[T] & := & \mathsf{par}_{p'}(C_1[T], T_2) \\
\mathsf{par}_{p'}(T_1, C_2)[T] & := & \mathsf{par}_{p'}(T_1, C_2[T])
\end{array}$$

Analog zum Einsetzen eines Terms in einen Kontext definieren wir die Einsetzung eines Kontexts C in einen anderen Kontext C'. Sind $C, C' \in$ Context so ist $C'[C] \in$ Context. Es gilt folgendes Lemma [vgl. mit Berregeb et al. 1997 S. 6]:

Lemma 4.8 Es gilt: $C'[C[T]] = (C'[C])[T]$

Beweis. Der Beweis erfolgt durch strukturelle Induktion über den syntaktischen Aufbau des äußeren Kontextes C' ("Kontextinduktion"):

$C' = \bullet$: Es gilt: $\bullet[C[T]] = C[T] = (\bullet[C])[T]$.

$C' = \mathsf{strict}_{p'}(C_1', T_2')$: Es gilt: $\mathsf{strict}_{p'}(C_1', T_2')[C[T]] = \mathsf{strict}_{p'}(C_1'[C[T]], T_2') \overset{\text{I.V.}}{=}$ $\mathsf{strict}_{p'}((C_1'[C])[T], T_2') = \mathsf{strict}_{p'}(C_1'[C], T_2')[T] = (\mathsf{strict}_{p'}(C_1', T_2')[C])[T]$

Alle verbleibenden Fälle sind analog. □

Lemma 4.9 Seien $T, T' \in \mathrm{Term}^{L(L)}$, $C' \in$ Context und $m \in \mathbb{N}_0$. Dann gilt: Aus $T \Rightarrow^m T'$ folgt $C'[T] \Rightarrow^m C'[T']$.

Beweis. Wir zeigen die Aussage nur für $m = 1$. Die Verallgemeinerung auf $m \in \mathbb{N}_0$ folgt durch eine vollständige Induktion nach m. Es gelte $T \Rightarrow T'$. Dieses Judgement kann nur mit einem der Axiome [T1], [T2], [T3] abgeleitet worden sein. Falls es mit [T1] abgeleitet wurde, dann gibt es ein $C \in$ Context, so dass $T = C[\mathsf{seq}_p(\mathsf{Empty}, T_2)]$ und $T' = C[T_2]$. Mit Lemma 4.8 folgt $C'[T] = C'[C[\mathsf{seq}_p(\mathsf{Empty}, T_2)]] = (C'[C])[\mathsf{seq}_p(\mathsf{Empty}, T_2)]$ und $C'[T'] = C'[C[T_2]] = (C'[C])[T_2]$. Weil $C'[C] \in$ Context ist, folgt mit Axiom [T1], dass gilt $C'[T] \Rightarrow C'[T']$. Die Fälle [T2] und [T3] sind analog. □

Wir definieren sowohl auf Termen $T \in \mathrm{Term}^{L(L)}$ als auch auf Kontexten C eine size-Funktion rekursiv wie folgt:

$$\mathrm{size} : \mathrm{Term}^{L(L)} \,\dot{\cup}\, \mathrm{Context} \longrightarrow \mathbb{N}_0$$

$$\begin{array}{lcl}
\mathrm{size}(\mathit{const}) & := & 0 \\
\mathrm{size}(\mathit{uop}_p(T_1)) & := & 1 + \mathrm{size}(T_1) \\
\mathrm{size}(\mathit{bop}_p(T_1, T_2)) & := & 1 + \mathrm{size}(T_1) + \mathrm{size}(T_2) \\
\mathrm{size}(\bullet) & := & 0 \\
\mathrm{size}(\mathsf{strict}_{p'}(C_1, T_2)) & := & 1 + \mathrm{size}(C_1) + \mathrm{size}(T_2) \\
\mathrm{size}(\mathsf{seq}_{p'}(C_1, T_2)) & := & 1 + \mathrm{size}(C_1) + \mathrm{size}(T_2) \\
\mathrm{size}(\mathsf{seq}_{p'}(T_1, C_2)) & := & 1 + \mathrm{size}(T_1) + \mathrm{size}(C_2) \\
\mathrm{size}(\mathsf{par}_{p'}(C_1, T_2)) & := & 1 + \mathrm{size}(C_1) + \mathrm{size}(T_2) \\
\mathrm{size}(\mathsf{par}_{p'}(T_1, C_2)) & := & 1 + \mathrm{size}(T_1) + \mathrm{size}(T_2)
\end{array}$$

Bemerkungen: (1) $\mathrm{size}(C[T]) = \mathrm{size}(C) + \mathrm{size}(T)$

(2) Aus $(T \Rightarrow^k T') \implies \mathrm{size}(T) = \mathrm{size}(T') + k$.

(3) $(T \Rightarrow^* T') \iff (T \Rightarrow^{(\mathrm{size}(T) - \mathrm{size}(T'))} T')$

(4) $(T \Rightarrow T') \implies \alpha(T) = \alpha(T') \wedge \nu(T) = \nu(T')$

Die Bemerkung (1) folgt mit einer Kontextinduktion über C. Für $k = 1$ ist die Bemerkung (2) eine unmittelbare Folgerung aus (1) und der Definition des Termersetzungssystems T. Die Aussage für $k \in \mathbb{N}_0$ folgt durch vollständige Induktion nach k. Die Bemerkung (3) ist unmittelbare Folgerung aus (2). Die Bemerkung (4) zeigt man mit einer geeigneten Kontextinduktion.

Aus Bemerkung (2) folgt, dass das Termersetzungssystem T terminiert, d. h. es gibt keine unendlichen "absteigenden" Ketten $T_0 \Rightarrow T_1 \Rightarrow T_2 \Rightarrow \ldots$. Anders ausgedrückt ist $(\Rightarrow)^{-1}$ eine wohlfundierte Relation auf $\mathrm{Term}^{L(L)}$.

Wir stellen nun einige Lemmata für das Termersetzungssystem T bereit, welche wir für die weiteren Beweisführungen benötigen. Wir interessieren uns insbesondere dafür, wie Interaktionsterme S syntaktisch gebaut sind, die sich in endlich vielen Rewritingschritten zu einem gegebenen Term $\mathrm{sla}(T, h)$ reduzieren lassen. Zunächst geben wir zwei neue Definitionen an:

$$\begin{array}{lcl}
ET \in \mathrm{ETerm} & ::= & \mathsf{Empty} \\
& | & \mathsf{seq}_p(ET_1, ET_2) \\
& | & \mathsf{par}_p(ET_1, ET_2)
\end{array}
\qquad
\begin{array}{lcl}
RC \in \mathrm{RContext} & ::= & \bullet \\
& | & \mathsf{seq}_{p'}(ET_1, RC_2) \\
& | & \mathsf{par}_{p'}(ET_1, RC_2) \\
& | & \mathsf{par}_{p'}(RC_1, ET_2)
\end{array}$$

Für jedes $T \in \mathrm{Term}^{L(L)}$ setzen wir $\mathrm{RTerm}[T] := \{RC[T] \mid RC \in \mathrm{RContext}\}$. Offenkundig gilt $\mathrm{RTerm}[\mathsf{Empty}] = \mathrm{ETerm}$.

Die Menge ETerm ist gegenüber rückwärts ausgeführten Rewritingschritten abgeschlossen. Diese Aussage kann wie in folgendem Lemma formalisiert werden:

Lemma 4.11 Es gilt:

$\forall\, T \in \mathrm{Term}^{L(L)}, T' \in \mathrm{ETerm}.\,[\,(T \Rightarrow T')$ folgt $T \in \mathrm{ETerm}\,]$

Beweis. Für alle $C \in$ Context, $T_1, T_2 \in \mathrm{Term}^{L(L)}$ und $p \in$ Path zeigen wir

(i) $C[T_2] \in \mathrm{ETerm} \implies C[\mathsf{seq}_p(\mathsf{Empty}, T_2)] \in \mathrm{ETerm}$ und
(ii) $C[T_2] \in \mathrm{ETerm} \implies C[\mathsf{par}_p(\mathsf{Empty}, T_2)] \in \mathrm{ETerm}$ und
(iii) $C[T_1] \in \mathrm{ETerm} \implies C[\mathsf{par}_p(T_1, \mathsf{Empty})] \in \mathrm{ETerm}$

durch stukturelle Induktion über den syntaktischen Aufbau des Kontextes C.

$C = \bullet$: Es gelte $C[T_1], C[T_2] \in \mathrm{ETerm}$. Wegen $C[T_2] = T_2$ folgt $T_2 \in \mathrm{ETerm}$. Somit gilt $C[\mathsf{seq}_p(\mathsf{Empty}, T_2)] = \mathsf{seq}_p(\mathsf{Empty}, T_2) \in \mathrm{ETerm}$, d. h. es gilt (i). Die Aussagen (ii) und (iii) zeigt man analog.

$C = \mathsf{strict}_{p'}(C_1, \widetilde{T})$: Dann sind $C[T_1], C[T_2] \notin \mathrm{ETerm}$, weil es keinen Term in ETerm gibt, dessen Toplevelfunktionssymbol $\mathsf{strict}_{p'}$ ist.

$C = \mathsf{seq}_{p'}(C_1, \widetilde{T})$: Es gelte $C[T_1], C[T_2] \in \mathrm{ETerm}$. Somit ist $\mathsf{seq}_{p'}(C_1[T_2], \widetilde{T}) = C[T_2] \in \mathrm{ETerm}$. Es folgt: $C_1[T_2] \in \mathrm{ETerm}$ und $\widetilde{T} \in \mathrm{ETerm}$. Aus $C_1[T_2] \in \mathrm{ETerm}$ und der I. V. folgt $C_1[\mathsf{seq}_p(\mathsf{Empty}, T_2)] \in \mathrm{ETerm}$. Zusammen mit $\widetilde{T} \in \mathrm{ETerm}$ folgt: $C[\mathsf{seq}_p(\mathsf{Empty}, T_2)] = \mathsf{seq}_{p'}(C_1[\mathsf{seq}_p(\mathsf{Empty}, T_2)], \widetilde{T}) \in \mathrm{ETerm}$. Also gilt (i). Die Aussagen (ii) und (iii) zeigt man analog.

$C = \mathit{seq}_{p'}(\widetilde{T}, C_2)$ oder $\mathit{par}_{p'}(C_1, \widetilde{T})$ oder $\mathit{par}_{p'}(\widetilde{T}, C_2)$: Analog zu $\mathsf{seq}_{p'}(C_1, \widetilde{T})$. □

Lemma 4.12 Sei $T \in \mathrm{Term}^{L(L)}$. Dann gilt: $T \in \mathrm{ETerm} \iff (T \Rightarrow^* \mathsf{Empty})$

Beweis. "$\Rightarrow$" Durch strukturelle Induktion über $T \in \mathrm{ETerm}$.

"$\Leftarrow$" Mittels Lemma 4.11 und einer vollständigen Induktion über die Anzahl m der Rewritingschritte $T \Rightarrow^m \mathsf{Empty}$. □

Aus Lemma 4.12 folgt, dass ETerm die Menge aller Interaktionsterme T ist, die die Eigenschaft haben, dass ihre (eindeutig bestimmte) Normalform Empty ist. Aufbauend auf Lemma 4.12 zeigt man durch Kontextinduktion über RC, dass für alle $RC \in$ RContext und alle $T \in \mathrm{Term}^{L(L)}$ gilt $RC[T] \Rightarrow^* T$.

Lemma 4.13 Seien $T, T', T_1', T_2' \in \mathrm{Term}^{L(L)}$, $p \in$ Path und es gelte $T \Rightarrow^* T'$. Dann gilt:

1. Falls das (entnummerierte) Toplevelfunktionssymbol von T' nicht strict, seq oder par ist, dann gilt $T \in \mathrm{RTerm}[T']$.
2. Falls $T' = \mathsf{strict}_p(T_1', T_2')$ ist, dann gibt es einen Term $T_1 \in \mathrm{Term}^{L(L)}$, so dass gilt $T_1 \Rightarrow^* T_1'$ und $T \in \mathrm{RTerm}[\mathsf{strict}_p(T_1, T_2')]$.

3. Falls $T' = bop_p(T'_1, T'_2)$ mit $bop \in \{\mathsf{seq}, \mathsf{par}\}$ ist, dann gibt es Terme $T_1, T_2 \in \mathrm{Term}^{L(L)}$, so dass gilt $T_1 \Rightarrow^* T'_1$ und $T_2 \Rightarrow^* T'_2$ und $T \in \mathrm{RTerm}[bop_p(T_1, T_2)]$.

Beweis. Der Beweis erfolgt durch vollständige Induktion über die Anzahl m der Rewritingschritte $T \Rightarrow^m T'$. Der Induktionsanfang für $m = 0$ ist trivial. Für jeden der drei Fälle beweisen wir eine Hilfsaussage, aus welcher der jeweilige Induktionsschluss trivial folgt.

Ad 1. Sei T' wie gefordert. Als Hilfsaussage zeigen wir, dass $\mathrm{RTerm}[T']$ gegenüber rückwärts ausgeführten Rewritingschritten abgeschlossen ist; vgl. mit Lemma 4.11. Zu zeigen: Für alle $C \in \mathrm{Context}$, $S_1, S_2 \in \mathrm{Term}^{L(L)}$, $p \in \mathrm{Path}$ gilt:

(i) $C[S_2] \in \mathrm{RTerm}[T'] \implies C[\mathsf{seq}_p(\mathsf{Empty}, S_2)] \in \mathrm{RTerm}[T']$ und
(ii) $C[S_2] \in \mathrm{RTerm}[T'] \implies C[\mathsf{par}_p(\mathsf{Empty}, S_2)] \in \mathrm{RTerm}[T']$ und
(iii) $C[S_1] \in \mathrm{RTerm}[T'] \implies C[\mathsf{par}_p(S_1, \mathsf{Empty})] \in \mathrm{RTerm}[T']$

Wir begnügen uns mit dem Beweis der Aussage (i). Die Aussagen (ii) und (iii) zeigt man völlig analog. Der Beweis erfolgt durch strukturelle Induktion über den syntaktischen Aufbau des Kontextes C.

$C = \bullet$: Es gelte $C[S_2] \in \mathrm{RTerm}[T']$ also ist $S_2 \in \mathrm{RTerm}[T']$. Somit gibt es ein $RC_2 \in \mathrm{RContext}$ mit $S_2 = RC_2[T']$. Es folgt $C[\mathsf{seq}_p(\mathsf{Empty}, S_2)] = \mathsf{seq}_p(\mathsf{Empty}, RC_2[T']) = \mathsf{seq}_p(\mathsf{Empty}, RC_2)[T']$. Weil $\mathsf{Empty} \in \mathrm{ETerm}$ und $RC_2 \in \mathrm{RContext}$ ist, folgt $\mathsf{seq}_p(\mathsf{Empty}, RC_2) \in \mathrm{RContext}$. Also ist $C[\mathsf{seq}_p(\mathsf{Empty}, S_2)] \in \mathrm{RTerm}[T']$.

$C = \mathsf{strict}_{p'}(C_1, \widetilde{T})$: Dann ist $C[S_2] \notin \mathrm{RTerm}[T']$, weil einerseits das (entnummerierte) Toplevelfunktionssymbol von T' nicht strict ist (nach Voraussetzung) und andererseits auch kein nichtleerer Kontext $RC \in \mathrm{RContext}$ existiert, dessen Toplevelfunktionssymbol strict ist.

$C = \mathsf{seq}_{p'}(C_1, \widetilde{T})$: Es gelte $C[S_2] \in \mathrm{RTerm}[T']$. Es gilt $\mathsf{seq}_{p'}(C_1[S_2], \widetilde{T}) = C[S_2] \in \mathrm{RTerm}[T']$. Es gibt ein $RC \in \mathrm{RContext}$, so dass gilt: $\mathsf{seq}_{p'}(C_1[S_2], \widetilde{T}) = RC[T']$. Weil das Toplevelfunktionssymbol von T' nicht seq ist (nach Voraussetzung), muss $RC \neq \bullet$ gelten. Somit kommt nur $RC = \mathsf{seq}_{p'}(ET_1, RC_2)$ in Frage. Mithin gilt (1) $C_1[S_2] = ET_1 \in \mathrm{ETerm}$ und (2) $\widetilde{T} = RC_2[T']$. Aus (1) und Lemma 4.11 folgt: $C_1[\mathsf{seq}_p(\mathsf{Empty}, S_2)] \in \mathrm{ETerm}$. Hieraus und mit (2) folgt: $C[\mathsf{seq}_p(\mathsf{Empty}, S_2)] = \mathsf{seq}_{p'}(C_1[\mathsf{seq}_p(\mathsf{Empty}, S_2)], \widetilde{T}) = \mathsf{seq}_{p'}(C_1[\mathsf{seq}_p(\mathsf{Empty}, S_2)], RC_2)[T'] \in \mathrm{RTerm}[T']$.

$C = \mathsf{seq}_{p'}(\widetilde{T}, C_2)$: Es gelte $C[S_2] \in \mathrm{RTerm}[T']$. Es gilt $\mathsf{seq}_{p'}(\widetilde{T}, C_2[S_2]) = C[S_2] = RC[T']$ mit einem $RC \in \mathrm{RContext}$. Weil das Toplevelfunktionssymbol von T' nicht seq ist, muss $RC \neq \bullet$ gelten. Somit kommt nur $RC = \mathsf{seq}_{p'}(ET_1, RC_2)$ in Frage. Mithin gilt (1) $\widetilde{T} = ET_1 \in \mathrm{ETerm}$ und (2) $C_2[S_2] = RC_2[T'] \in \mathrm{RTerm}[T']$. Aus (2) und der I. V. für C_2 folgt: $C_2[\mathsf{seq}_p(\mathsf{Empty}, S_2)] \in \mathrm{RTerm}[T']$, d. h. es gibt ein $RC'_2 \in \mathrm{RContext}$ mit $C_2[\mathsf{seq}_p(\mathsf{Empty}, S_2)] = RC'_2[T']$.

Hieraus und mit (1) folgt: $C[\mathsf{seq}_p(\mathsf{Empty}, S_2)] = \mathsf{seq}_{p'}(\widetilde{T}, C_2[\mathsf{seq}_p(\mathsf{Empty}, S_2)]) = \mathsf{seq}_{p'}(ET_1, RC'_2)[T'] \in \mathrm{RTerm}[T']$.

$C = \mathsf{par}_{p'}(C_1, \widetilde{T})$: Es gelte $C[S_2] \in \mathrm{RTerm}[T']$. Es gilt $\mathsf{par}_{p'}(C_1[S_2], \widetilde{T}) = C[S_2] = RC[T']$ mit einem $RC \in \mathrm{RContext}$. Weil das Toplevelfunktionssymbol von T' nicht par ist, muss $RC \neq \bullet$ gelten. Es kommen nur die Fälle (a) $RC = \mathsf{par}_{p'}(ET_1, RC_2)$ oder (b) $RC = \mathsf{par}_{p'}(RC_1, ET_2)$ in Frage. Im Fall (a) argumentiert man analog zu $C = \mathsf{seq}_{p'}(C_1, \widetilde{T})$, im Fall (b) analog zu $C = \mathsf{seq}_{p'}(\widetilde{T}, C_2)$.

$C = \mathsf{par}_{p'}(\widetilde{T}, C_2)$: Analog.

Ad 2. Es gelte $T' = \mathsf{strict}_p(T'_1, T'_2)$. Als Hilfsaussage zeigen wir, dass für alle $C \in \mathrm{Context}$, $S_1, S_2 \in \mathrm{Term}^{L(L)}$ und $q \in \mathrm{Path}$ gilt:

$$\begin{aligned}
&\text{(i)} \quad C[S_2] \in \mathrm{RTerm}[T'] \implies \exists\, T_1 \in \mathrm{Term}^{L(L)}.\big((T_1 = T'_1 \lor T_1 \Rightarrow T'_1) \land \\
&\qquad\qquad C[\mathsf{seq}_q(\mathsf{Empty}, S_2)] \in \mathrm{RTerm}[\mathsf{strict}_p(T_1, T'_2)]\big) \\
&\text{(ii)} \quad C[S_2] \in \mathrm{ETerm}[T'] \implies \exists\, T_1 \in \mathrm{Term}^{L(L)}.\big((T_1 = T'_1 \lor T_1 \Rightarrow T'_1) \land \\
&\qquad\qquad C[\mathsf{par}_q(\mathsf{Empty}, S_2)] \in \mathrm{RTerm}[\mathsf{strict}_p(T_1, T'_2)]\big) \\
&\text{(iii)} \quad C[S_1] \in \mathrm{ETerm}[T'] \implies \exists\, T_1 \in \mathrm{Term}^{L(L)}.\big((T_1 = T'_1 \lor T_1 \Rightarrow T'_1) \land \\
&\qquad\qquad C[\mathsf{par}_q(S_1, \mathsf{Empty})] \in \mathrm{RTerm}[\mathsf{strict}_p(T_1, T'_2)]\big)
\end{aligned}$$

Wir begnügen uns mit dem Beweis der Aussage (i). Die Aussagen (ii) und (iii) zeigt man analog. Der Beweis erfolgt durch stukturelle Induktion über den syntaktischen Aufbau des Kontextes C.

$C = \bullet$: Analog zum Fall $C = \bullet$ aus Ad 1. mit $T_1 := T'_1$.

$C = \mathsf{strict}_{q'}(C_1, \widetilde{T})$: Es gelte $C[S_2] \in \mathrm{RTerm}[T']$. Es gibt $RC \in \mathrm{RContext}$ mit $\mathsf{strict}_{q'}(C_1[S_2], \widetilde{T}) = C[S_2] = RC[T'] = RC[\mathsf{strict}_p(T'_1, T'_2)]$. Weil kein Kontext RC existiert, dessen Toplevelfunktionssymbol strict ist, muss $RC = \bullet$ sein. Folglich ist $q' = p$ und $C_1[S_2] = T'_1$ und $\widetilde{T} = T'_2$. Setze $T_1 := C_1[\mathsf{seq}_q(\mathsf{Empty}, S_2)]$. Es ist $T_1 \in \mathrm{Term}^{L(L)}$. Mit Axiom [T1] folgt $T_1 \Rightarrow C_1[S_2] = T'_1$. Außerdem gilt $C[\mathsf{seq}_q(\mathsf{Empty}, S_2)] = \mathsf{strict}_{q'}(C_1[\mathsf{seq}_q(\mathsf{Empty}, S_2)], \widetilde{T}) = \mathsf{strict}_p(T_1, T'_2) = \bullet[\mathsf{strict}_p(T_1, T'_2)] \in \mathrm{RTerm}[\mathsf{strict}_p(T_1, T'_2)]$.

$C = \mathsf{seq}_{q'}(C_1, \widetilde{T})$: Analog zum Fall $C = \mathsf{seq}_{q'}(C_1, \widetilde{T})$ aus Ad 1. mit $T_1 := T'_1$.

$C = \mathsf{seq}_{q'}(\widetilde{T}, C_2)$: Es gelte $C[S_2] \in \mathrm{RTerm}[T']$. Es gibt $RC \in \mathrm{RContext}$ mit $\mathsf{seq}_{q'}(\widetilde{T}, C_2[S_2]) = C[S_2] = RC[T'] = RC[\mathsf{strict}_p(T'_1, T'_2)]$. Offenkundig ist $RC \neq \bullet$. Es kommt nur $RC = \mathsf{seq}_{q'}(ET_1, RC_2)$ in Frage. Mithin ist (1) $\widetilde{T} = ET_1 \in \mathrm{ETerm}$ und (2) $C_2[S_2] = RC_2[T'] \in \mathrm{RTerm}[T']$. Aus (2) und der I. V. für C_2 folgt: Es gibt $T_1 \in \mathrm{Term}^{L(L)}$ und $RC'_2 \in \mathrm{RContext}$, so dass gilt ($T_1 = T'_1$ oder $T_1 \Rightarrow T'_1$) und $C_2[\mathsf{seq}_q(\mathsf{Empty}, S_2)] = RC'_2[\mathsf{strict}_p(T_1, T'_2)]$. Es folgt: $C[\mathsf{seq}_q(\mathsf{Empty}, S_2)] = \mathsf{seq}_{q'}(\widetilde{T}, C_2[\mathsf{seq}_q(\mathsf{Empty}, S_2)]) = \mathsf{seq}_{q'}(ET_1, RC'_2)[\mathsf{strict}_p(T_1, T'_2)] \in \mathrm{RTerm}[\ldots]$.

$C = \mathsf{par}_{q'}(C_1, \widetilde{T})$ oder $\mathsf{par}_{q'}(\widetilde{T}, C_2)$: Analog.

Ad 3. Dieser Fall ist methodisch so ähnlich zu den Fällen 1. und 2., dass wir den Beweis dem geneigten Leser überlassen. □

4.8 Beobachtungsäquivalenz als Regelsystem

Ausgehend von der Konstruktion der Relation β aus Abschnitt 4.7 könnte man versuchen zu beweisen, dass β eine Beobachtungsäquivalenz zwischen der pfad-annotierten globalen operationalen Semantik und der lokalen operationalen Semantik ist. Aufgrund der etwas technisch geratenen Definiton von β, die sowohl auf die Funktion sla als auch auf die Relation $\Rightarrow^*$ zurückgreift, erweist sich ein solches direktes Vorgehen als überraschend schwierig. Es stellt sich daher die Frage, ob es möglich ist, die Definition von β in eine Gestalt zu bringen, die uns während der Ausführung der Koinduktion hilft, die verschiedenen kombinatorischen Fallgestaltungen überschaubar zu halten. Zu diesem Zweck isolieren wir zunächst den eigentlich problematischen Teil der Definition von β, nämlich die Kombination der Abstraktionsfunktion sla und der reflexiv-transitiven Hülle $\Rightarrow^*$ der Reduktionsrelation. Sei $l \in L$ und $h \in \mathrm{H}^L_{\mathrm{fin}}$. Die Relation $\rho^h_l \subseteq (\mathrm{Term}^{L(L)})^2$ sei definiert wie folgt:

$$\rho^h_l(S,T) \;:\Longleftrightarrow\; S \Rightarrow^* \mathrm{sla}(T,h) \text{ und } (S,T) \in (\mathrm{Term}^{\{l\}(\{l\})})^2$$

Dann gilt $\beta(T,[l_1 : T_{l_1},\dots,l_n : T_{l_n},h]) \iff \forall\, i \in \{1,\dots,n\}.\ \rho^h_{l_i}(\pi_{l_i}(T),T_{l_i})$. Es erscheint uns sinnvoll, die Relationen ρ^h_l mit Hilfe von Regelsystemen R^h_l so zu axiomatisieren, dass $\rho^h_l(S,T) \iff \Vdash_{\mathrm{R}^h_l} (S,T)$ gilt. Die weiteren Designziele für diese Regelsysteme R^h_l sind zunächst vage. Es versteht sich von selbst, dass wir möglichst wenige und möglichst einfache Regelschemata wünschen. Bei naivem Vorgehen könnte man z. B. versuchen, ein Regelsystem $\mathrm{R}_{\mathrm{sla}}$ für die Abstraktionsfunktion und ein zweites Regelsystem $\mathrm{R}_{\Rightarrow}$ für die Reduktionsrelation aufzustellen. Anschließend erweitert man das Regelsystem $\mathrm{R}_{\Rightarrow}$ um zwei Regelschemata

$$[\mathrm{refl}]\quad (T,T) \qquad\qquad [\mathrm{trans}]\quad \frac{(T,T') \quad (T',T'')}{(T,T'')}$$

und bildet dadurch die reflexiv-transitive Hülle. Schließlich fügt man das Regelsystem $\mathrm{R}_{\mathrm{sla}}$ und das um [refl] und [trans] erweiterte Regelsystem $\mathrm{R}_{\Rightarrow}$ mit Hilfe eines weiteren Regelschemas [combine] zum Regelsystem R^h_l zusammen. Das Problem an dieser einfachen Methode ist, dass die Regelschemata [trans] und [combine] nicht syntax-directed sind. Überdies enthält [trans] einen frei wählbaren Parameter T'. Somit würde dieses naive Regelsystem R^h_l zu genau denselben kombinatorischen Problemen führen, denen wir durch Einführung des Regelsystems eigentlich ausweichen wollten. Ein Designziel für R^h_l ist somit der Verzicht auf ein Regelschema der Form [trans]. Außerdem wäre es misslich, zwei getrennte Regelsysteme aufzustellen, die mit Hilfe eines Regelschemas [combine] zusammengefügt werden müssen. Stattdessen wollen wir ein gemeinsames verschmolzenes Regelsystem finden. Ein Indiz dafür, dass ein solches Vorhaben nicht chancenlos ist, findet sich in folgender Beobachtung, welche wir im Übrigen nicht beweisen und auch nicht weiter verwenden werden:

Beobachtung: Sei $h \in \mathrm{H}^L_{\mathrm{fin}}$. Seien $S, T \in \mathrm{Term}^{L(L)}$ und der Term S erfülle bezüglich h die Invariante (I6). Dann gilt: $(S \Rightarrow T) \implies (\mathrm{sla}(S,h) \Rightarrow \mathrm{sla}(T,h))$

Demnach wirkt die Abstraktionsfunktion $\mathrm{sla}(-,h)$ für festes h wie ein Kontext des Termersetzungssystems T (vgl. mit Lemma 4.9). Dies unterstreicht die Tatsache, dass die sla-Funktion und die Reduktionsrelation $\Rightarrow$ nicht viel miteinander zu tun haben. Sie interferieren nicht miteinander. Dies wiederum nährt die Hoffnung, dass man die Regelschemata für $\mathrm{R}_{\mathrm{sla}}$ und $\mathrm{R}_{\Rightarrow}$ einfach miteinander "mischen" kann.

Sei $l \in L$ und $h \in \mathrm{H}^L_{\mathrm{fin}}$. Das Regelsystem R^h_l über Judgements der Form $(S,T) \in (\mathrm{Term}^{L(L)})^2$ ist in Tafel 4.4 angegeben. Der allergrößte Teil der Regelschemata—nämlich alle Regelschemata außer [seq^2], [par^2] und [par^3]—sind nur eine Straightforward-Übersetzung der $\prec^h$-rekursiven Definition der sla-Funktion. Das Geheimnis der hier angewandten Methode liegt aber in den drei unscheinbaren zusätzlichen Regeln [seq^2], [par^2] und [par^3]. Der gesamte Effekt der reflexiv-transitiven Hülle $\Rightarrow^*$ der Reduktionsrelation $\Rightarrow$, die in der Definition von ρ^h_l der sla-Funktion vorgeschaltet ist, äußert sich in diesen hinzugemischten drei Regeln. Es liegt auf der Hand, dass dies eine drastische Vereinfachung der Definition von ρ^h_l und somit auch der Definition von β bedeutet. Alle technischen Probleme der ursprünglichen Definition von β werden durch die Regelsysteme R^h_l und durch den zugehörigen Korrektheits- und Vollständigkeitsbeweis gekapselt und somit vor der eigentlichen Koinduktion, die sich nur noch auf die Regelschemata in den Tafeln 4.2, 4.3 und 4.4 stützt, verschattet. Wir betrachten dieses Vorgehen als methodisch schön.

[None]	$(\mathsf{None}(a), \mathsf{None}(a))$	falls $l \in \alpha(a)$
[Empty]	$(\mathsf{Empty}, \mathsf{Empty})$	
[Action]	(a, a)	falls $l \in \alpha(a)$
[strict^1]	$\dfrac{(S, T_2)}{(S, \mathsf{strict}_p(T_1, T_2))}$	falls $h(p) = L \,\wedge\, T_1 \in \mathrm{Term}^{\{l\}(\{l\})}$
[strict^2]	$\dfrac{(S_1, T_1)}{(\mathsf{strict}_p(S_1, T_2), \mathsf{strict}_p(T_1, T_2))}$	falls $h(p) \neq L \,\wedge\, T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$
[seq^1]	$\dfrac{(S_1, T_1) \quad (S_2, T_2)}{(\mathsf{seq}_p(S_1, S_2), \mathsf{seq}_p(T_1, T_2))}$	
[seq^2]	$\dfrac{(S_1, \mathsf{Empty}) \quad (S_2, T_2)}{(\mathsf{seq}_p(S_1, S_2), T_2)}$	-- entspricht [T1]
[par^1]	$\dfrac{(S_1, T_1) \quad (S_2, T_2)}{(\mathsf{par}_p(S_1, S_2), \mathsf{par}_p(T_1, T_2))}$	
[par^2]	$\dfrac{(S_1, \mathsf{Empty}) \quad (S_2, T_2)}{(\mathsf{par}_p(S_1, S_2), T_2)}$	-- entspricht [T2]
[par^3]	$\dfrac{(S_1, T_1) \quad (S_2, \mathsf{Empty})}{(\mathsf{par}_p(S_1, S_2), T_1)}$	-- entspricht [T3]
[loop^1]	$\dfrac{(S, \mathsf{Empty})}{(S, \mathsf{loop}_p(T_1))}$	falls $h(p) = \{1\} \,\wedge\, T_1 \in \mathrm{Term}^{\{l\}(\{l\})}$
[loop^2]	$\dfrac{(S_1, T_1) \quad (S_2, \mathsf{loop}_{p.2}({}_{p.2}T_1))}{(\mathsf{seq}_p(S_1, S_2), \mathsf{loop}_p(T_1))}$	falls $h(p) = \{2\}$
[loop^3]	$(\mathsf{loop}_p(T_1), \mathsf{loop}_p(T_1))$	falls $h(p) \notin \{\{1\}, \{2\}\} \wedge T_1 \in \mathrm{Term}^{\{l\}(\{l\})}$
[alt^1]	$\dfrac{(S_1, T_1)}{(S_1, \mathsf{alt}_p(T_1, T_2))}$	falls $h(p) = \{1\} \,\wedge\, T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$
[alt^2]	$\dfrac{(S_2, T_2)}{(S_2, \mathsf{alt}_p(T_1, T_2))}$	falls $h(p) = \{2\} \,\wedge\, T_1 \in \mathrm{Term}^{\{l\}(\{l\})}$
[alt^3]	$(\mathsf{alt}_p(T_1, T_2), \mathsf{alt}_p(T_1, T_2))$	falls $h(p) \notin \{\{1\}, \{2\}\} \wedge T_1, T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$

Tafel 4.4: Axiomatisierung der Relation ρ_l^h (= Regelsystem R_l^h)

4.8.1 Korrektheit

In diesem Abschnitt wird gezeigt, dass das Regelsystem R_l^h aus Tafel 4.4 eine korrekte Axiomatisierung der Relation ρ_l^h darstellt. Anschließend zeigen wir in Abschnitt 4.8.2 die Vollständigkeit des Regelsystems R_l^h.

Satz 4.14 (Korrektheit von $\mathbf{R}_l^h$) Sei $l \in L$ und $h \in \mathrm{H}_{\mathrm{fin}}^L$. Dann gilt:
$\forall\,(S,T) \in (\mathrm{Term}^{L(L)})^2.\ \Vdash_{\mathrm{R}_l^h} (S,T) \implies \rho_l^h(S,T)$.

Beweis. Der Beweis erfolgt durch Ableitungsinduktion für Judgements (S,T). Wir erinnern daran, dass $\mathrm{Term}^l := \mathrm{Term}^{\{l\}(\{l\})}$ gesetzt wurde.

[None], [Empty], [Action] Dann ist $S = T = \mathit{const}$ und $(\mathit{const}, \mathit{const}) \in (\mathrm{Term}^l)^2$. Außerdem gilt $\mathit{const} \Rightarrow^* \mathit{const} = \mathrm{sla}(\mathit{const}, h)$.

[strict¹] Dann ist $T = \mathsf{strict}_p(T_1,T_2)$ und $h(p) = L$ und $T_1 \in \mathrm{Term}^l$ und $\Vdash_{\mathrm{R}_l^h} (S,T_2)$. Mit der I. V. folgt $\rho_l^h(S,T_2)$. Es folgt $S \Rightarrow^* \mathrm{sla}(T_2,h)$ und $(S,T_2) \in (\mathrm{Term}^l)^2$. Wegen $h(p) = L$ gilt $\mathrm{sla}(T_2,h) = \mathrm{sla}(\mathsf{strict}_p(T_1,T_2),h) = \mathrm{sla}(T,h)$. Also gilt $S \Rightarrow^* \mathrm{sla}(T,h)$. Außerdem ist $(S,T) \in (\mathrm{Term}^l)^2$. Es folgt: $\rho_l^h(S,T)$.

[strict²] Dann ist $S = \mathsf{strict}_p(S_1,T_2)$ und $T = \mathsf{strict}_p(T_1,T_2)$ und $h(p) \neq L$ und $T_2 \in \mathrm{Term}^l$ und $\Vdash_{\mathrm{R}_l^h} (S_1,T_1)$. Mit I. V. folgt $\rho_l^h(S_1,T_1)$. Es folgt $S_1 \Rightarrow^* \mathrm{sla}(T_1,h)$ und $(S_1,T_1) \in (\mathrm{Term}^l)^2$. Setze $C := \mathsf{strict}_p(\bullet,T_2) \in \mathrm{Context}$. Mit Lemma 4.9 folgt: $C[S_1] \Rightarrow^* C[\mathrm{sla}(T_1,h)]$. Somit gilt $\mathsf{strict}_p(S_1,T_2) \Rightarrow^* \mathsf{strict}_p(\mathrm{sla}(T_1,h),T_2)$. Und weil $h(p) \neq L$ ist, folgt $\mathsf{strict}_p(\mathrm{sla}(T_1,h),T_2) = \mathrm{sla}(\mathsf{strict}_p(T_1,T_2),h)$. Somit gilt $S \Rightarrow^* \mathrm{sla}(T,h)$ und $(S,T) \in \mathrm{Term}^l$, d. h. es gilt $\rho_l^h(S,T)$.

Die restlichen Fälle seien in einem kürzeren Stil behandelt.

[seq¹] Es gelte $S_1 \Rightarrow^* \mathrm{sla}(T_1,h)$ und $S_2 \Rightarrow^* \mathrm{sla}(T_2,h)$. Aus $S_1 \Rightarrow^* \mathrm{sla}(T_1,h)$ und Lemma 4.9 mit $C := \mathsf{seq}_p(\bullet,S_2)$ folgt $\mathsf{seq}_p(S_1,S_2) \Rightarrow^* \mathsf{seq}_p(\mathrm{sla}(T_1,h),S_2)$. In entsprechender Weise folgt aus $S_2 \Rightarrow^* \mathrm{sla}(T_2,h)$ und Lemma 4.9 mit $C := \mathsf{seq}_p(\mathrm{sla}(T_1,h),\bullet)$, dass gilt $\mathsf{seq}_p(\mathrm{sla}(T_1,h),S_2) \Rightarrow^* \mathsf{seq}_p(\mathrm{sla}(T_1,h),\mathrm{sla}(T_2,h))$. Insgesammt folgt: $\mathsf{seq}_p(S_1,S_2) \Rightarrow^* \mathsf{seq}_p(\mathrm{sla}(T_1,h),\mathrm{sla}(T_2,h)) = \mathrm{sla}(\mathsf{seq}_p(T_1,T_2),h)$.

[seq²] Es gelte $S_1 \Rightarrow^* \mathrm{sla}(\mathsf{Empty},h)$ und $S_2 \Rightarrow^* \mathrm{sla}(T_2,h)$. Aus $S_1 \Rightarrow^* \mathrm{sla}(\mathsf{Empty},h)$ und $\mathrm{sla}(\mathsf{Empty},h) = \mathsf{Empty}$ und Lemma 4.9 mit Kontext $C := \mathsf{seq}_p(\bullet,S_2)$ folgt $\mathsf{seq}_p(S_1,S_2) \Rightarrow^* \mathsf{seq}_p(\mathsf{Empty},S_2)$. Mit Axiom [T1] folgt $\mathsf{seq}_p(\mathsf{Empty},S_2) \Rightarrow S_2$. Insgesammt gilt $\mathsf{seq}_p(S_1,S_2) \Rightarrow^* \mathsf{seq}_p(\mathsf{Empty},S_2) \Rightarrow S_2 \Rightarrow^* \mathrm{sla}(T_2,h)$ und somit $\mathsf{seq}_p(S_1,S_2) \Rightarrow^* \mathrm{sla}(T_2,h)$.

[par¹] Analog zu [seq¹].

[par²] Analog zu [seq²] mit Axiom [T2].

[par³] Analog zu [seq²] mit Axiom [T3].

[loop¹] Es gelte $h(p) = \{1\}$ und $S \Rightarrow^* \mathrm{sla}(\mathsf{Empty},h)$. Weil $h(p) = \{1\}$ ist, gilt $\mathrm{sla}(\mathsf{Empty},h) = \mathsf{Empty} = \mathrm{sla}(\mathsf{loop}_p(T_1),h)$. Also gilt $S \Rightarrow^* \mathrm{sla}(\mathsf{loop}_p(T_1),h)$.

[loop^2] Es gelte $h(p) = \{2\}$ und $S_1 \Rightarrow^* \mathrm{sla}(T_1, h)$ und $S_2 \Rightarrow^* \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h)$. Durch zweimalige Anwendung von Lemma 4.9 erhält man: $\mathsf{seq}_p(S_1, S_2) \Rightarrow^* \mathsf{seq}_p(\mathrm{sla}(T_1, h), S_2) \Rightarrow^* \mathsf{seq}_p(\mathrm{sla}(T_1, h), \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1)) = \mathrm{sla}(\mathsf{loop}_p(T_1), h)$, wobei das Gleichheitszeichen wegen $h(p) = \{2\}$ gilt.

[loop^3] Es gelte $h(p) \notin \{\{1\}, \{2\}\}$. Es folgt $\mathsf{loop}_p(T_1) = \mathrm{sla}(\mathsf{loop}_p(T_1), h)$. Also gilt $\mathsf{loop}_p(T_1) \Rightarrow^* \mathrm{sla}(\mathsf{loop}_p(T_1), h)$.

[alt^1] Es gelte $h(p) = \{1\}$ und $S_1 \Rightarrow^* \mathrm{sla}(T_1, h)$. Weil $h(p) = \{1\}$ ist, folgt $\mathrm{sla}(T_1, h) = \mathrm{sla}(\mathsf{alt}_p(T_1, T_2), h)$. Somit gilt $S_1 \Rightarrow^* \mathrm{sla}(\mathsf{alt}_p(T_1, T_2), h)$.

[alt^2] Analog zu [alt^1].

[alt^3] Es gelte $h(p) \notin \{\{1\}, \{2\}\}$. Es folgt $\mathsf{alt}_p(T_1, T_2) = \mathrm{sla}(\mathsf{alt}_p(T_1, T_2), h)$. Also gilt $\mathsf{alt}_p(T_1, T_2) \Rightarrow^* \mathrm{sla}(\mathsf{alt}_p(T_1, T_2), h)$. □

4.8.2 Vollständigkeit

In diesem Abschnitt wird gezeigt, dass das Regelsystem R_l^h aus Tafel 4.4 eine vollständige Axiomatisierung der Relation ρ_l^h darstellt.

Lemma 4.15 Sei $ET \in \mathrm{ETerm}$ beliebig. Dann gilt: $\Vdash_{\mathrm{R}_l^h} (ET, \mathsf{Empty})$.

Beweis. Der Beweis erfolgt durch strukturelle Induktion über ET:

$ET = \mathsf{Empty}$: Mit Axiom [Empty] folgt $\Vdash_{\mathrm{R}_l^h} (\mathsf{Empty}, \mathsf{Empty})$.

$ET = \mathsf{seq}_p(ET_1, ET_2)$: Nach I. V. gilt: $\Vdash_{\mathrm{R}_l^h} (ET_1, \mathsf{Empty})$ und $\Vdash_{\mathrm{R}_l^h} (ET_2, \mathsf{Empty})$. Mit Regel [seq^2] folgt $\Vdash_{\mathrm{R}_l^h} (\mathsf{seq}_p(ET_1, ET_2), \mathsf{Empty})$.

$ET = \mathsf{par}_p$: Analog zum Fall $ET = \mathsf{seq}_p(ET_1, ET_2)$ mit Regel [par^2]. □

Lemma 4.16 Es gelte $\widetilde{S} \in \mathrm{RTerm}[S]$ und $\Vdash_{\mathrm{R}_l^h} (S, T)$. Dann gilt: $\Vdash_{\mathrm{R}_l^h} (\widetilde{S}, T)$.

Beweis. Wir zeigen $\forall\, RC \in \mathrm{RContext}.\ \Vdash_{\mathrm{R}_l^h} (RC[S], T)$ durch strukturelle Induktion über den syntaktischen Aufbau des Kontextes RC:

$RC = \bullet$: Trivial.

$RC = \mathsf{seq}_p(ET_1, RC_2)$: Aus Lemma 4.15 folgt $\Vdash_{\mathrm{R}_l^h} (ET_1, \mathsf{Empty})$. Aus der I. V. folgt $\Vdash_{\mathrm{R}_l^h} (RC_2[S], T)$. Mit Regel [seq^2] folgt: $\Vdash_{\mathrm{R}_l^h} (\mathsf{seq}_p(ET_1, RC_2[S]), T)$, d. h. es gilt $\Vdash_{\mathrm{R}_l^h} (RC[S], T)$.

$RC = \mathsf{par}_p(ET_1, RC_2)$: Analog zum Fall $RC = \mathsf{seq}_p(ET_1, RC_2)$ mit Regel [par^2].

$RC = \mathsf{par}_p(RC_1, ET_2)$: Analog zum Fall $RC = \mathsf{seq}_p(ET_1, RC_2)$ mit Regel [par^3].

□

Satz 4.17 (Vollständigkeit von $\mathbf{R}_l^h$) Sei $l \in L$ und $h \in \mathrm{H}_{\mathrm{fin}}^L$. Dann gilt:

$$\forall (S,T) \in (\mathrm{Term}^{L(L)})^2.\, \rho_l^h(S,T) \implies \Vdash_{\mathrm{R}_l^h} (S,T).$$

Beweis. Seien $l \in L$ und $h \in \mathrm{H}_{\mathrm{fin}}^L$ vorgegeben. Wir setzen

$$A(T) \;:\Longleftrightarrow\; \forall S \in \mathrm{Term}^{\{l\}(\{l\})}.\big((S \Rightarrow^* \mathrm{sla}(T,h)) \implies \Vdash_{\mathrm{R}_l^h} (S,T)\big)$$

für jedes $T \in \mathrm{Term}^l$. Wir zeigen $\forall\, T \in \mathrm{Term}^l.\, A(T)$ durch noethersche Induktion entlang der Relation $\prec^h$ aus Abschnitt 4.7.1:

Sei $T \in \mathrm{Term}^l$ und gelte $\forall T' \prec^h T.\, A(T')$. Zu zeigen ist $A(T)$. Sei $S \in \mathrm{Term}^l$ und gelte $S \Rightarrow^* \mathrm{sla}(T,h)$. Zu zeigen ist $\Vdash_{\mathrm{R}_l^h} (S,T)\big)$. Wir führen eine Fallunterscheidung nach der syntaktischen Form von T aus:

$T = \mathsf{None}(a)$: Dann ist $l \in \alpha(a)$ (weil $T \in \mathrm{Term}^l$ ist) und $\mathrm{sla}(T,h) = \mathsf{None}(a)$. Damit gilt $S \Rightarrow^* \mathsf{None}(a)$ und mit Lemma 4.13.1 folgt $S \in \mathrm{RTerm}[\mathsf{None}(a)]$. Mit Axiom [None] und $l \in \alpha(a)$ folgt $\Vdash_{\mathrm{R}_l^h} (\mathsf{None}(a), \mathsf{None}(a))$. Mit Lemma 4.16 folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{None}(a))$.

$T = \mathsf{Empty}$: Dann ist $\mathrm{sla}(T,h) = \mathsf{Empty}$. Somit gilt $S \Rightarrow^* \mathsf{Empty}$ und mit Lemma 4.13.1 folgt $S \in \mathrm{RTerm}[\mathsf{Empty}]$. Mit Axiom [Empty] folgt $\Vdash_{\mathrm{R}_l^h} (\mathsf{Empty}, \mathsf{Empty})$. Mit Lemma 4.16 folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{Empty})$.

$T = a$: Analog zum Fall $T = \mathsf{None}(a)$ mit Axiom [Action].

$T = \mathsf{strict}_p(T_1, T_2)$: Wir unterscheiden zwei Fälle:

Fall 1: $h(p) = L$: Dann ist $\mathrm{sla}(T,h) = \mathrm{sla}(T_2,h)$. Somit gilt $S \Rightarrow^* \mathrm{sla}(T_2,h)$. Wegen $T_2 \prec^h T$ gilt $A(T_2)$. Es folgt $\Vdash_{\mathrm{R}_l^h} (S,T_2)$. Wegen $h(p) = L$ und $T_1 \in \mathrm{Term}^l$ und Regel [strict^1] folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{strict}_p(T_1,T_2))$.

Fall 2: $h(p) \neq L$: Dann ist $\mathrm{sla}(T,h) = \mathsf{strict}_p(\mathrm{sla}(T_1,h), T_2)$. Somit gilt $S \Rightarrow^* \mathsf{strict}_p(\mathrm{sla}(T_1,h), T_2)$. Nach Lemma 4.13.2 gibt es ein $T_1' \in \mathrm{Term}^{L(L)}$, so dass gilt (1) $T_1' \Rightarrow^* \mathrm{sla}(T_1,h)$ und (2) $S \in \mathrm{RTerm}[\mathsf{strict}_p(T_1', T_2)]$. Wegen $T_1 \prec^h T$ gilt $A(T_1)$. Bevor wir $A(T_1)$ auf (1) anwenden können, müssen wir noch zeigen, dass $T_1' \in \mathrm{Term}^l$ gilt: Weil $T \in \mathrm{Term}^l$ ist, folgt $\alpha(T), \nu(T) \subseteq \{l\}$. Somit gilt $\alpha(T_1') = \alpha(\mathrm{sla}(T_1,h)) \subseteq \alpha(T_1) \subseteq \alpha(T) \subseteq \{l\}$ und $\nu(T_1') = \nu(\mathrm{sla}(T_1,h)) \subseteq \nu(T_1) \subseteq \nu(T) \subseteq \{l\}$. Folglich ist $T_1' \in \mathrm{Term}^l$. Aus $A(T_1)$ und $T_1' \in \mathrm{Term}^l$ und (1) folgt $\Vdash_{\mathrm{R}_l^h} (T_1', T_1)$. Wegen $h(p) \neq L$ und $T_2 \in \mathrm{Term}^l$ und Regel [strict^2] folgt (3) $\Vdash_{\mathrm{R}_l^h} (\mathsf{strict}_p(T_1', T_2), \mathsf{strict}_p(T_1, T_2))$. Mit (2) und (3) und Lemma 4.16 folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{strict}_p(T_1, T_2))$.

$T = \mathsf{seq}_p(T_1, T_2)$: Dann ist $\mathrm{sla}(T,h) = \mathsf{seq}_p(\mathrm{sla}(T_1,h), \mathrm{sla}(T_2,h))$. Folglich gilt $S \Rightarrow^* \mathsf{seq}_p(\mathrm{sla}(T_1,h), \mathrm{sla}(T_2,h))$. Nach Lemma 4.13.3 gibt es $T_1', T_2' \in \mathrm{Term}^{L(L)}$ (wegen $T \in \mathrm{Term}^l$ folgt $T_1', T_2' \in \mathrm{Term}^l$), so dass gilt (1) $T_1' \Rightarrow^* \mathrm{sla}(T_1,h)$ und (2) $T_2' \Rightarrow^* \mathrm{sla}(T_2,h)$ und (3) $S \in \mathrm{RTerm}[\mathsf{seq}_p(T_1', T_2')]$. Wegen $T_1, T_2 \prec^h T$ gilt $A(T_1)$ und $A(T_2)$. Aus $A(T_1)$ und $T_1' \in \mathrm{Term}^l$ und (1) folgt $\Vdash_{\mathrm{R}_l^h} (T_1', T_1)$. Aus $A(T_2)$ und $T_2' \in \mathrm{Term}^l$ und (2) folgt $\Vdash_{\mathrm{R}_l^h} (T_2', T_2)$. Mittels der Regel [$\mathrm{seq}^1$]

folgt (4) $\Vdash_{\mathrm{R}_l^h} (\mathsf{seq}_p(T_1', T_2'), \mathsf{seq}_p(T_1, T_2))$. Mit (3) und (4) und Lemma 4.16 folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{seq}_p(T_1, T_2))$.

$T = \mathsf{par}_p(T_1, T_2)$: Analog zum Fall $T = \mathsf{seq}_p(T_1, T_2)$ mit Regel [par^1].

$T = \mathsf{loop}_p(T_1)$: Wir unterscheiden drei Fälle:

Fall 1: $h(p) = \{1\}$: Dann ist $\mathrm{sla}(T, h) = \mathsf{Empty}$ und es gilt $S \Rightarrow^* \mathsf{Empty}$. Mit Lemma 4.13.1 folgt $S \in \mathrm{RTerm}[\mathsf{Empty}]$. Mit Lemma 4.16 und Axiom [Empty] folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{Empty})$. Wegen $h(p) = \{1\}$ und $T_1 \in \mathrm{Term}^l$ und Regel [loop^1] folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{loop}_p(T_1))$.

Fall 2: $h(p) = \{2\}$: Dann ist $\mathrm{sla}(T, h) = \mathsf{seq}_p(\mathrm{sla}(T_1, h), \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h))$. Somit gilt $S \Rightarrow^* \mathsf{seq}_p(\mathrm{sla}(T_1, h), \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h))$. Aufgrund von Lemma 4.13.3 (und weil $T \in \mathrm{Term}^l$ ist) gibt es $T_1', T_2' \in \mathrm{Term}^l$, so dass gilt (1) $T_1' \Rightarrow^* \mathrm{sla}(T_1, h)$ und (2) $T_2' \Rightarrow^* \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h)$ und (3) $S \in \mathrm{RTerm}[\mathsf{seq}_p(T_1', T_2')]$. Es gilt $T_1 \prec^h T$. Weil $h(p) = \{2\}$ ist, gilt ferner $\mathsf{loop}_{p.2}({}_{p.2}T_1) \prec^h \mathsf{loop}_p(T_1) = T$. Somit gilt $A(T_1)$ und $A(\mathsf{loop}_{p.2}({}_{p.2}T_1))$. Aus $A(T_1)$ und $T_1' \in \mathrm{Term}^l$ und (1) folgt $\Vdash_{\mathrm{R}_l^h} (T_1', T_1)$. Aus $A(\mathsf{loop}_{p.2}({}_{p.2}T_1))$ und $T_2' \in \mathrm{Term}^l$ und (2) folgt $\Vdash_{\mathrm{R}_l^h} (T_2', \mathsf{loop}_{p.2}({}_{p.2}T_1))$. Mit $h(p) = \{2\}$ und Regel [loop^2] folgt (4) $\Vdash_{\mathrm{R}_l^h} (\mathsf{seq}_p(T_1', T_2'), \mathsf{loop}_p(T_1))$. Mit (3) und (4) und Lemma 4.16 folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{loop}_p(T_1))$.

Fall 3: $h(p) \notin \{\{1\}, \{2\}\}$: Dann ist $\mathrm{sla}(T, h) = \mathsf{loop}_p(T_1)$. Somit gilt $S \Rightarrow^* \mathsf{loop}_p(T_1)$. Mit Lemma 4.13.1 folgt $S \in \mathrm{RTerm}[\mathsf{loop}_p(T_1)]$. Mit Lemma 4.16 und Axiom [loop^3] und $h(p) \notin \{\{1\}, \{2\}\}$ und $T_1 \in \mathrm{Term}^l$ folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{loop}_p(T_1))$.

$T = \mathsf{alt}_p(T_1, T_2)$: Wir unterscheiden drei Fälle:

Fall 1: $h(p) = \{1\}$: Dann ist $\mathrm{sla}(T, h) = \mathrm{sla}(T_1, h)$ und es gilt $S \Rightarrow^* \mathrm{sla}(T_1, h)$. Wegen $T_1 \prec^h T$ gilt $A(T_1)$. Aus $A(T_1)$ und $S \in \mathrm{Term}^l$ und $S \Rightarrow^* \mathrm{sla}(T_1, h)$ folgt $\Vdash_{\mathrm{R}_l^h} (S, T_1)$. Weil $h(p) = \{1\}$ und $T_2 \in \mathrm{Term}^l$ gilt, folgt mit Regel [alt^1] dass $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{alt}_p(T_1, T_2))$.

Fall 2: $h(p) = \{2\}$: Analog zu Fall 1 mit Regel [alt^2].

Fall 3: $h(p) \notin \{\{1\}, \{2\}\}$: Dann ist $\mathrm{sla}(T, h) = \mathsf{alt}_p(T_1, T_2)$. Somit gilt $S \Rightarrow^* \mathsf{alt}_p(T_1, T_2)$. Mit Lemma 4.13.1 folgt $S \in \mathrm{RTerm}[\mathsf{alt}_p(T_1, T_2)]$. Mit Lemma 4.16 und Axiom [alt^3] und $h(p) \notin \{\{1\}, \{2\}\}$ und $T_1, T_2 \in \mathrm{Term}^l$ folgt $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{alt}_p(T_1, T_2))$. $\square$

4.9 Beweis der Beobachtungsäquivalenz

4.9.1 Simulation globaler semantischer Schritte

Wirft man einen unbefangenen Blick auf das Regelsystem R_l^h aus Tafel 4.4, so stellt man zunächst fest, dass die Regeln [strict^1], [loop^1], [alt^1] und [alt^2] linksseitig nicht syntax-directed sind, während die Regeln [seq^2], [par^2] und [par^3] rechtsseitig nicht syntax-directed sind. Für den Beweis, welchen wir in diesem Abschnitt anstreben, dass jeder Schritt der globalen operationalen Semantik in der lokalen operationalen Semantik simuliert werden kann, werden wir eine strukturelle Induktion über die syntaktische Form des globalen Konfigurationsterms ausführen. Weil die linke Seite der Judgements (S, T) des Regelsystems R_l^h dem (projizierten) globalen Konfigurationsterm entspricht, ist vorhersehbar, dass uns innerhalb der Fallunterscheidung der strukturellen Induktion diejenigen Regeln aus R_l^h stören werden, die linksseitig nicht syntax-directed sind. Insofern liegt der Gedanke nahe, vor Ausführung der Fallunterscheidung für die Ableitungsbäume der Verträglichkeitsbeziehungen quasi eine Art von Beweisbaumnormalisierung auszuführen, die zu einer Beseitigung der Regeln [strict^1], [loop^1], [alt^1], [alt^2] in der Baumstruktur führen. Es stellt sich heraus, dass mit dieser Methode auch die Regel [loop^2] beseitigt werden kann. Man erhält das folgende Lemma:

Lemma 4.18 Sei $l \in L$ und $h \in \mathrm{H}_{\mathrm{fin}}^L$. Es gelte $\Vdash_{\mathrm{R}_l^h} (S, T)$. Der Term T erfülle bezüglich h die Invariante (I6). Dann gibt es einen Term $T' \in \mathrm{Term}^{\{l\}(\{l\})}$, so dass gilt:

1. $l : T, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T', h$ $\qquad$ (h unverändert!)
2. $\Vdash_{\underline{\mathrm{R}}_l^h} (S, T')$

Das Symbol $\underline{\mathrm{R}}_l^h$ bezeichne das Regelsystem aus Tafel 4.4 ohne die Regeln [strict^1], [loop^1], [loop^2], [alt^1] und [alt^2]—also ohne diejenigen Regeln, die den Fall betreffen, dass die Abstraktionsfunktion sla "eingeschaltet" ist. Das Symbol $\underline{\mathrm{L}}$ bezeichne das Regelsystem der lokalen operationalen Semantik aus Tafel 4.3 ohne die Axiome ($\mathrm{seq}_{\mathrm{L}}^2$), ($\mathrm{par}_{\mathrm{L}}^3$) und ($\mathrm{par}_{\mathrm{L}}^4$)—also ohne die Axiome zur Ausführung von Aufräumschritten.

Beweis. Der Beweis erfolgt durch Ableitungsinduktion für Judgements (S, T).

(I) Regeln, die linksseitig nicht syntax-directed sind:

[strict^1] Dann ist $T = \mathsf{strict}_p(T_1, T_2)$ und $h(p) = L$ und $\Vdash_{\mathrm{R}_l^h} (S, T_2)$. Weil T, h die Invariante (I6) erfüllen, gilt dies ebenso für T_2 und h. Nach I. V. gibt es $T' \in \mathrm{Term}^{\{l\}(\{l\})}$, so dass gilt: $l : T_2, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T', h$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (S, T')$. Weil T, h die Invariante (I6) erfüllen und $h(p) = L$ ist, folgt $T_1 = \mathsf{Empty}$. Somit gilt $T = \mathsf{strict}_p(\mathsf{Empty}, T_2)$. Mit $h(p) = L$ und Axiom ($\mathrm{strict}_{\mathrm{L}}^2$) folgt $l : T, h \xrightarrow{\tau}_{\underline{\mathrm{L}}} l : T_2, h$. Es folgt $l : T, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T', h$.

[loop^1] Dann ist $T = \mathsf{loop}_p(T_1)$ und $h(p) = \{1\}$ und $\Vdash_{\mathrm{R}_l^h} (S, \mathsf{Empty})$. Trivialerweise erfüllen Empty, h die Invariante (I6). Weil es für Empty keine ableitbaren Übergänge gibt, folgt aus der I. V. dass $\Vdash_{\underline{\mathrm{R}}_l^h} (S, \mathsf{Empty})$. Weil $h(p) = \{1\}$ ist (also $h(p) \cap \{2\} = \emptyset$), folgt mit Axiom ($\mathrm{loop}_{\mathrm{L}}^1$) dass gilt: $l : T, h \xrightarrow{\tau}_{\underline{\mathrm{L}}} l : \mathsf{Empty}, h$.

[alt^1] Dann ist $T = \mathsf{alt}_p(T_1, T_2)$ und $h(p) = \{1\}$ und $\Vdash_{\mathrm{R}_l^h} (S, T_1)$. Weil T, h die Invariante (I6) erfüllen, gilt dies ebenso für T_1 und h. Nach I. V. gibt es $T' \in \mathrm{Term}^{\{l\}(\{l\})}$, so dass gilt: $l : T_1, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T', h$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (S, T')$. Weil $h(p) = \{1\}$ ist (also $h(p) \cap \{2\} = \emptyset$), folgt mit Axiom ($\mathrm{alt}^1_{\mathrm{L}}$) dass gilt: $l : T, h \xrightarrow{\tau}_{\underline{\mathrm{L}}} l : T_1, h$. Es folgt $l : T, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T', h$.

[alt^2] Analog zu Fall [alt^1] mit Axiom ($\mathrm{alt}^2_{\mathrm{L}}$).

(II) Regeln, die linksseitig syntax-directed sind:

[None], [Empty], [Action], [loop^3], [alt^3] Setze $T' := T$. Fertig.

[strict^2] Dann ist $S = \mathsf{strict}_p(S_1, T_2)$ und $T = \mathsf{strict}_p(T_1, T_2)$ und $h(p) \neq L$ und $T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$ und $\Vdash_{\mathrm{R}_l^h} (S_1, T_1)$. Es erfüllen T_1, h die Invariante (I6). Nach I. V. gibt es $T_1' \in \mathrm{Term}^{\{l\}(\{l\})}$, so dass gilt: $l : T_1, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T_1', h$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (S_1, T_1')$. Mit Regel ($\mathrm{strict}^1_{\mathrm{L}}$) folgt: $l : T, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T', h$ mit $T' := \mathsf{strict}_p(T_1', T_2)$. Mit $h(p) \neq L$ und $T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$ und Regel [strict^2] folgt $\Vdash_{\underline{\mathrm{R}}_l^h} (S, T')$.

[seq^1] Methodisch ähnlich zu Fall [strict^2], wobei die Regeln ($\mathrm{seq}^1_{\mathrm{L}}$) und ($\mathrm{seq}^3_{\mathrm{L}}$) zu verwenden sind. Man beachte $\alpha(\tau) = \emptyset$ und $\mathrm{R}^*_\emptyset = \mathrm{id}$.

[seq^2] Methodisch ähnlich zu den Fällen [strict^2] und [loop^1], wobei man ausnutzt, dass es für Empty keine ableitbaren Übergänge gibt.

[par^1], [par^2], [par^3] Analog.

[loop^2] Dann ist $S = \mathsf{seq}_p(S_1, S_2)$ und $T = \mathsf{loop}_p(T_1)$ und $h(p) = \{2\}$ und $\Vdash_{\mathrm{R}_l^h} (S_1, T_1)$ und $\Vdash_{\mathrm{R}_l^h} (S_2, \mathsf{loop}_{p.2}({}_{p.2}T_1))$. Weil T, h die Invariante (I6) erfüllen, gilt dies ebenso für T_1 und h. Mit $h(p) = \{2\}$ und Regel ($\mathrm{loop}^2_{\mathrm{L}}$) folgt die Ableitbarkeit von (1) $l : T, h \xrightarrow{\tau}_{\underline{\mathrm{L}}} l : \mathsf{seq}_p(T, \mathsf{loop}_{p.2}({}_{p.2}T_1)), h$. Weil die Eigenschaft (I6) invariant bleibt, wird sie folglich auch von $\mathsf{loop}_{p.2}({}_{p.2}T_1)$ und h erfüllt. Nach I. V. gibt es $T_1', T_2' \in \mathrm{Term}^{\{l\}(\{l\})}$, so dass gilt (2) $l : T_1, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T_1', h$ und (3) $\Vdash_{\underline{\mathrm{R}}_l^h} (S_1, T_1')$ und (4) $l : \mathsf{loop}_{p.2}({}_{p.2}T_1), h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : T_2', h$ und (5) $\Vdash_{\underline{\mathrm{R}}_l^h} (S_2, T_2')$. Aus (2) und (4) und ($\mathrm{seq}^1_{\mathrm{L}}$) und ($\mathrm{seq}^3_{\mathrm{L}}$) folgt: $l : \mathsf{seq}_p(T, \mathsf{loop}_{p.2}({}_{p.2}T_1)), h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : \mathsf{seq}_p(T_1', T_2'), h$. Hieraus und mit (1) folgt $l : T, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} T', h$ mit $T' := \mathsf{seq}_p(T_1', T_2')$. Aus (3) und (5) und [seq^1] folgt $\Vdash_{\underline{\mathrm{R}}_l^h} (S, T')$. □

Das Lemma 4.18 bezieht sich auf eine Situation, in welcher ein bestimmter Entscheidungsschritt (loop_p oder alt_p) von mindestens einer Lifeline oder ein Synchronisationsschritt (strict_p) von allen Lifelines ausgeführt wurde. Dann ist also $h(p) \in \{\{1\}, \{2\}\}$ bzw. $h(p) = L$ und die sla-Funktion ist für dieses p "eingeschaltet". Die sla-Funktion täuscht nunmehr innerhalb der Verträglichkeitsbeziehung β für alle Lifelines, die den Entscheidungsschritt zu p noch nicht nachgeführt haben, eben diesen Nachführschritt vor. Für alle Lifelines, die ein Konstrukt $\mathsf{strict}_p(\mathsf{Empty}, T_2)$ noch nicht durch T_2 ersetzt haben, täuscht die sla-Funktion diese Ersetzung vor—welche wir ebenfalls Nachführschritt genannt haben. Kurz gesagt: Die sla-Funktion "faket" Nachführschritte. Bevor nun die globale Semantik ihren nächsten Schritt macht, gewährleistet es Lemma 4.18, dass alle diese gefaketen τ-Nachführschritte tatsächlich ausgeführt werden können.

Lemma 4.19 Sei $l \in L$ und $L' \subseteq L$ und $h \in \mathrm{H}^L_{\mathrm{fin}}$. Dann gilt:

1. $\Vdash_{\mathrm{R}^h_l} (S, T) \implies \Vdash_{\mathrm{R}^h_l} (\mathrm{R}^*_{L'}(S), \mathrm{R}^*_L(T))$
2. $\Vdash_{\underline{\mathrm{R}}^h_l} (S, T) \implies \Vdash_{\underline{\mathrm{R}}^h_l} (\mathrm{R}^*_{L'}(S), \mathrm{R}^*_L(T))$

Beweis. Der Beweis erfolgt durch Ableitungsinduktion für Judgements (S, T). Wir führen nur einige Fälle exemplarisch aus.

[None] Dann ist $S = \mathsf{None}(a) = T$ und $l \in \alpha(a)$. Somit ist $\mathrm{R}^*_{L'}(S) = S$ und $\mathrm{R}^*_{L'}(T) = T$. Daraus folgt die Behauptung.

[Empty] Analog zu Fall [None].

[Action] Dann ist $S = a = T$ und $l \in \alpha(a)$. Falls $\alpha(a) \cap L' = \emptyset$ ist, dann gilt $\mathrm{R}^*_{L'}(S) = S$.und $\mathrm{R}^*_{L'}(T) = T$. Falls $\alpha(a) \cap L' \neq \emptyset$ ist, dann gilt $\mathrm{R}^*_{L'}(S) = \mathsf{None}(a) = \mathrm{R}^*_{L'}(T)$. Daraus folgt die Behauptung.

[strict[1]] Dann ist $T = \mathsf{strict}_p(T_1, T_2)$ und $h(p) = L$ und $T_1 \in \mathrm{Term}^{\{l\}(\{l\})}$ und $\Vdash_{\mathrm{R}^h_l} (S, T_2)$. Nach I. V. gilt $\Vdash_{\mathrm{R}^h_l} (\mathrm{R}^*_{L'}(S), \mathrm{R}^*_{L'}(T_2))$. Weil $T_1 \in \mathrm{Term}^{\{l\}(\{l\})}$ ist, gilt auch $\mathrm{R}^*_{L'}(T_1) \in \mathrm{Term}^{\{l\}(\{l\})}$. Mittels $h(p) = L$ und Regel [strict[1]] folgt $\Vdash_{\mathrm{R}^h_l} (\mathrm{R}^*_{L'}(S), \mathsf{strict}_p(\mathrm{R}^*_{L'}(T_1), \mathrm{R}^*_{L'}(T_2)))$. Mit $\mathrm{R}^*_{L'}(T) = \mathsf{strict}_p(\mathrm{R}^*_{L'}(T_1), \mathrm{R}^*_{L'}(T_2))$ folgt die Behauptung.

Alle weiteren Induktionsfälle sind ähnlich zum Fall [strict[1]]. Im Fall [loop[2]] ist $\mathrm{R}^*_{L'}({}_{p.n}T) = {}_{p.n}\mathrm{R}^*_{L'}(T)$ zu verwenden. ☐

Lemma 4.20 Sei $l \in L$ und $h \in \mathrm{H}^L_{\mathrm{fin}}$. Sei $T \in \mathrm{Term}^{\{l\}(\{l\})}$, T wohlnummeriert und $\mathrm{top}(T) \in \{\bot, p\}$ mit $p \in \mathrm{Path}$. Es gelte $h(p.r) = \emptyset$ für alle $r \in \mathrm{Path}$. Dann folgt: $\Vdash_{\mathrm{R}^h_l} (T, T)$.

Beweis. Mit Hilfe einer trivialen strukturellen Induktion über T zeige man, dass $\mathrm{sla}(T, h) = T$ gilt. Die Behauptung folgt dann mit Satz 4.17. ☐

Wir setzen $\mathbb{A}^L := \{a \in \mathbb{A} \mid \alpha(a) \subseteq L\}$ und $\mathbb{A}^L_\tau := \mathbb{A}^L \mathbin{\dot\cup} \{\tau\}$. Für jedes $p \in \mathrm{Path}$ setzen wir $\mathrm{PathSpace}(p) := \{q \in \mathrm{Path} \mid \exists\, r \in \mathrm{Path}.\, q = p.r\}$.

Lemma 4.21 Seien $l \in L$, $h, h' \in \mathrm{H}^L_{\mathrm{fin}}$, $T, T' \in \mathrm{Term}^{\{l\}(\{l\})}$ und $\bar{a} \in \mathbb{A}^L_\tau$.
Es gelte $l : T, h \xrightarrow{\bar{a}}_{\mathrm{L}} l : T', h'$. Dann gilt:

1. Falls $\bar{a} = a \in \mathbb{A}^L$ ist, dann gilt $h' = h$ und $a \in \mathbb{A}^{\{l\}}$.
2. Falls T wohlnummeriert und $\mathrm{top}(T) = p \in \mathrm{Path}$ ist,
   dann gilt $h'(q) = h(q)$ für alle $q \in \mathrm{Path} \setminus \mathrm{PathSpace}(p)$.

Beweis. Auch dieser Beweis ist trivial. Er erfolgt durch Ableitungsinduktion für Judgements $l : T, h \xrightarrow{\bar{a}}_{\mathrm{L}} l : T', h'$. $\square$

Lemma 4.22 Seien $l \in L$, $h, h' \in \mathrm{H}^L_{\mathrm{fin}}$ und es gelte $\Vdash_{\mathrm{R}^h_l} (S, T)$. Dann gilt:

1. Falls $\mathrm{top}(T) = \bot$ ist, dann folgt $\Vdash_{\mathrm{R}^{h'}_l} (S, T)$.
2. Falls $\mathrm{top}(T) \neq \bot$ und T wohlnummeriert ist und überdies $h(q) = h'(q)$ für alle $q \in \mathrm{PathSpace}(\mathrm{top}(T))$ gilt, dann folgt $\Vdash_{\mathrm{R}^{h'}_l} (S, T)$.

Anmerkung: In beiden Fällen lässt sich sogar zeigen, dass der gegebene Ableitungsbaum $d \in \mathrm{D}_{\mathrm{R}^h_l}$ mit $d \Vdash_{\mathrm{R}^h_l} (S, T)$ ein Element aus $\mathrm{D}_{\mathrm{R}^{h'}_l}$ ist.

Beweis. Mittels einer noetherschen Induktion entlang der Relation $\prec^h$ zeigen wir $\mathrm{sla}(T, h) = \mathrm{sla}(T, h')$, wobei wir nur exemplarisch die Fälle $T = \mathsf{strict}_p(T_1, T_2)$ und $T = \mathsf{loop}_p(T_1)$ ausführen werden. Die Behauptung folgt dann mit den Sätzen 4.14 und 4.17.

$T = \mathsf{strict}_p(T_1, T_2)$: Es ist $\mathrm{top}(T) = p \neq \bot$. Es sei T wohlnummeriert und es gelte $h \restriction \mathrm{PathSpace}(p) = h' \restriction \mathrm{PathSpace}(p)$. Somit sind T_1 und T_2 wohlnummeriert und es gibt $q_1, q_2 \in \mathrm{Path}$ mit $\mathrm{top}(T_1) \in \{\bot, p.1.q_1\}$ und $\mathrm{top}(T_2) \in \{\bot, p.2.q_2\}$. Weil $\mathrm{PathSpace}(p.1.q_1), \mathrm{PathSpace}(p.2.q_2) \subseteq \mathrm{PathSpace}(p)$ und $T_1, T_2 \prec^h T$ ist, folgt mit der I. V. dass gilt: $\mathrm{sla}(T_1, h) = \mathrm{sla}(T_1, h')$ und $\mathrm{sla}(T_2, h) = \mathrm{sla}(T_2, h')$. Falls $h(p) = h'(p) = L$ ist, dann gilt $\mathrm{sla}(T, h) = \mathrm{sla}(T_2, h) = \mathrm{sla}(T_2, h') = \mathrm{sla}(T, h')$. Falls $h(p) = h'(p) \neq L$ ist, dann gilt $\mathrm{sla}(T, h) = \mathsf{strict}_p(\mathrm{sla}(T_1, h), T_2) = \mathsf{strict}_p(\mathrm{sla}(T_1, h'), T_2) = \mathrm{sla}(T, h')$.

$T = \mathsf{loop}_p(T_1)$: Es ist $\mathrm{top}(T) = p \neq \bot$. Es sei T wohlnummeriert und es gelte $h \restriction \mathrm{PathSpace}(p) = h' \restriction \mathrm{PathSpace}(p)$. Somit ist T_1 wohlnummeriert und es gibt $q_1 \in \mathrm{Path}$ mit $\mathrm{top}(T_1) \in \{\bot, p.1.q_1\}$. Analog zu Fall $T = \mathsf{strict}_p(T_1, T_2)$ folgt $\mathrm{sla}(T_1, h) = \mathrm{sla}(T_1, h')$. Wegen Lemma 4.1 ist ${}_{p.2}T_1$ wohlnummeriert und es gilt $\mathrm{top}({}_{p.2}T_1) \in \{\bot, p.2.1.q_1\}$. Somit ist $\mathsf{loop}_{p.2}({}_{p.2}T_1)$ wohlnummeriert. Falls $h(p) = h'(p) = \{1\}$ ist, dann gilt $\mathrm{sla}(T, h) = \mathsf{Empty} = \mathrm{sla}(T, h')$. Falls $h(p) = h'(p) = \{2\}$ ist, dann gilt $\mathsf{loop}_{p.2}({}_{p.2}T_1) \prec^h T$. Weil $\mathsf{loop}_{p.2}({}_{p.2}T_1)$ wohlnummeriert und $\mathrm{PathSpace}(p.2) \subseteq \mathrm{PathSpace}(p)$ ist, folgt mit der I. V. dass gilt: $\mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h) = \mathrm{sla}(\mathsf{loop}_{p.2}({}_{p.2}T_1), h')$. Zusammen mit $\mathrm{sla}(T_1, h) = \mathrm{sla}(T_1, h')$ folgt $\mathrm{sla}(T, h) = \mathrm{sla}(T, h')$. Falls $h(p) = h'(p) \notin \{\{1\}, \{2\}\}$ ist, dann gilt $\mathrm{sla}(T, h) = T = \mathrm{sla}(T, h')$. $\square$

Lemma 4.23 Seien $l \in L$, $h \in \mathrm{H}^L_{\mathrm{fin}}$ und $S, T \in \mathrm{Term}^{\{l\}(\{l\})}$.
Es sei S wohlnummeriert und es gelte $\Vdash_{\underline{\mathrm{R}}^h_l} (S, T)$. Dann gilt:

1. $\mathrm{top}(S) = \bot \implies \mathrm{top}(T) = \bot \quad$ (genauer: $S = T$)
2. $\mathrm{top}(S) = p \implies \mathrm{top}(T) \in \{\bot, p.r\}$ mit $r \in \mathrm{Path}$.

Beweis. Der Beweis erfolgt durch Ableitungsinduktion für Judgements (S, T). Für alle Regeln außer [seq^2], [par^2] und [par^3] ist die Aussage trivial wahr.

[seq²] Dann ist $S = \mathsf{seq}_p(S_1, S_2)$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (S_1, \mathsf{Empty})$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (S_2, T)$. Es gilt $\mathrm{top}(S) = p \neq \bot$. Weil S wohlnummeriert ist, ist auch S_2 wohlnummeriert und es gilt $\mathrm{top}(S_2) \in \{\bot, p.2.q_2\}$, $q_2 \in \mathrm{Path}$. Mit der I. V. folgt: Falls $\mathrm{top}(S_2) = \bot$ ist, dann ist $\mathrm{top}(T) = \bot$. Falls $\mathrm{top}(S_2) = p.2.q_2$ ist, dann ist $\mathrm{top}(T) \in \{\bot, p.2.q_2.r\}$, $r \in \mathrm{Path}$.

[par²] und [par³] Analog. □

Satz 4.24 Sei $L = \{l_1, \dots, l_n\} \subseteq \mathbb{I}$ mit $l_1, \dots, l_n$ paarw. verschieden und $n \geq 0$. Seien $T \in \mathrm{Term}^{L(L)}$, $\gamma = [l_1 \!:\! T_{l_1}, \dots, l_n \!:\! T_{l_n}, h] \in \mathrm{Conf}^L$, $\bar{a} \in \mathbb{A}_\tau^L$ und es gelte

T	ist wohlnummeriert und
$T_{l_1}, \dots, T_{l_n}$	sind wohlnummeriert und
$h \in \mathrm{H}_{\mathrm{fin}}^L$	d. h. h hat endliche Domain und
$T_{l_1}, \dots, T_{l_n}, h$	erfüllen die Invarianten (I3), (I5), (I6), (I7), (I8), (I9)

und $\beta(T, \gamma)$. Dann gilt:

1. Für alle $T' \in \mathrm{Term}^{L(L)}$ mit $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$ gibt es $\gamma' \in \mathrm{Conf}^L$ mit $\gamma \overset{\bar{a}}{\Rightarrow}_{\mathrm{L}} \gamma'$ und $\beta(T', \gamma')$.
2. Falls $T \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{Empty}$, dann $\gamma \overset{\bar{a}}{\Rightarrow}_{\mathrm{L}} \gamma'$ mit $\gamma' \in \mathrm{TConf}^L$.

Hierbei ist definiert $\gamma \overset{a}{\Rightarrow}_{\mathrm{L}} \gamma' :\Leftrightarrow \gamma \xrightarrow{\tau^* a}_{\mathrm{L}} \gamma'$ und $\gamma \overset{\tau}{\Rightarrow}_{\mathrm{L}} \gamma' :\Leftrightarrow \gamma \xrightarrow{\tau^*}_{\mathrm{L}} \gamma'$.

Beweis. Ad 1. Der Beweis erfolgt durch strukturelle Induktion über die Form des globalen Konfigurationsterms T. Für alle $T \in \mathrm{Term}^L$ setzen wir

$$\begin{aligned}
A(T) \;:\Longleftrightarrow\; & \forall\, T' \in \mathrm{Term}^L,\ (T_{l_1}, \dots, T_{l_n}) \in \mathrm{Term}^{l_1} \times \dots \times \mathrm{Term}^{l_n} \\
& \forall\, h \in \mathrm{H}_{\mathrm{fin}}^L,\ \bar{a} \in \mathbb{A}_\tau^L . \big(\\
& \quad \big[\ T \text{ ist wohlnummeriert und} \\
& \quad\ \ T_{l_1}, \dots, T_{l_n} \text{ sind wohlnummeriert und} \\
& \quad\ \ T_{l_1}, \dots, T_{l_n}, h \text{ erfüllen (I3), (I5), (I6), (I7), (I8), (I9) und} \\
& \quad\ \ \forall\, i \in \{1, \dots, n\}.\ \Vdash_{\mathrm{R}_{l_i}^h} (\pi_{l_i}(T), T_{l_i}) \text{ und } T \xrightarrow{\bar{a}}_{\mathrm{G}} T'\ \big] \implies \\
& \quad \exists\, (T'_{l_1}, \dots, T'_{l_n}) \in \mathrm{Term}^{l_1} \times \dots \times \mathrm{Term}^{l_n},\ h' \in \mathrm{H}_{\mathrm{fin}}^L . \big[\\
& \quad\ \ [l_1 \!:\! T_{l_1}, \dots, l_n \!:\! T_{l_n}, h] \overset{\bar{a}}{\Rightarrow}_{\mathrm{L}} [l_1 \!:\! T'_{l_1}, \dots, l_n \!:\! T'_{l_n}, h'] \text{ und} \\
& \quad\ \ \forall\, i \in \{1, \dots, n\}.\ \Vdash_{\mathrm{R}_{l_i}^{h'}} (\pi_{l_i}(T'), T'_{l_i}) \big] \big)
\end{aligned}$$

Sei $T \in \mathrm{Term}^L$ und T wohlnummeriert und es gelte $A(\tilde{T})$ für jeden direkten Teilterm $\tilde{T}$ von T. Seien T', $T_{l_1}, \dots, T_{l_n}$, h und $\bar{a}$ wie gefordert. Bevor wir in die Fallunterscheidung eintreten, wenden wir das Lemma 4.18 auf jede Lifeline an, d. h. wir führen alle von der sla-Funktion gefaketen τ-Nachführschritte aus. Dadurch finden wir ein $(\bar{T}_{l_1}, \dots, \bar{T}_{l_n}) \in \mathrm{Term}^{l_1} \times \dots \times \mathrm{Term}^{l_n}$, so dass gilt

$[l_1 : T_{l_1}, \ldots, l_n : T_{l_n}, h] \xrightarrow{\tau^*}_{\mathrm{L}} [l_1 : \bar{T}_{l_1}, \ldots, l_n : \bar{T}_{l_n}, h]$ und
$\forall\, i \in \{1, \ldots, n\}.\ \Vdash_{\underline{\mathrm{R}}^h_{l_i}} (\pi_{l_i}(T), \bar{T}_{l_i})$.

Auch die Terme $\bar{T}_{l_1}, \ldots, \bar{T}_{l_n}$ sind wohlnummeriert und $\bar{T}_{l_1}, \ldots, \bar{T}_{l_n}, h$ erfüllen die Invarianten (I3), (I5), (I6), (I7), (I8) und (I9). Für den restlichen Beweis lassen wir die Oberstriche wieder weg, d. h. wir schreiben $T_{l_1}, \ldots, T_{l_n}$ an Stelle von $\bar{T}_{l_1}, \ldots, \bar{T}_{l_n}$.

Wir führen eine Fallunterscheidung nach der syntaktischen Form von T aus:

$T = \mathsf{None}(a)$ oder Empty: Nicht möglich, weil $\mathsf{None}(a) \overset{\bar{a}}{\not\to}_{\mathrm{G}} T'$ und $\mathsf{Empty} \overset{\bar{a}}{\not\to}_{\mathrm{G}} T'$.

$T = a$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$ kommt nur (basic$_{\mathrm{G}}$) in Frage. Folglich ist $T = a = \bar{a} \in \mathbb{A}^L$ und $T' = \mathsf{Empty}$. Wir setzen $l_0 \in L$ so dass $l_0 \in \alpha(a)$. Es ist $\pi_{l_0}(T) = a$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}^h_{l_0}} (\pi_{l_0}(T), T_{l_0})$ kommt nur [Action] in Frage. Somit ist $T_{l_0} = a$ und mit (basic$_{\mathrm{L}}$) folgt: $l_0 : T_{l_0}, h \xrightarrow{a}_{\mathrm{L}} l_0 : \mathsf{Empty}, h$. Mit [Empty] folgt: $\Vdash_{\mathrm{R}^h_{l_0}} (\pi_{l_0}(T'), \mathsf{Empty})$. Außerdem ist $\pi_l(T) = \mathsf{Empty} = \pi_l(T')$ für alle $l \in L \setminus \{l_0\}$, d. h. die restlichen Lifelines bemerken von der Aktion a nichts.

$T = \mathsf{alt}_p(T_1, T_2)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$ kommen nur (alt$^1_{\mathrm{G}}$) oder (alt$^2_{\mathrm{G}}$) in Frage. Entsprechend unterscheiden wir zwei Fälle:

Fall (alt$^1_{\mathrm{G}}$): Dann ist $\bar{a} = \tau$ und $T' = T_1$. Für jedes $l \in L$ kommt zur Ableitung von $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T), T_l)$ aufgrund von $\pi_l(T) = \mathsf{alt}_p(\pi_l(T_1), \pi_l(T_2))$ nur das Axiom [alt^3] in Frage. Somit ist $T_l = \mathsf{alt}_p(\pi_l(T_1), \pi_l(T_2))$ für alle $l \in L$ und es gilt $h(p) \notin \{\{1\}, \{2\}\}$. Mit Invariante (I9) folgt $h(p) = \emptyset$. Wir wählen eine beliebige Lifeline $l_0 \in L$. Bemerkung: Dies ist die Lifeline, die "mutig voranschreitet" und das alt_p-Konstrukt als Erste bearbeitet (Entscheidungsschritt). Mit Axiom (alt$^1_{\mathrm{L}}$) und $h(p) = \emptyset$ folgt: $l_0 : T_{l_0}, h \xrightarrow{\tau}_{\mathrm{L}} l_0 : T'_{l_0}, h'$ wobei $T'_{l_0} = \pi_{l_0}(T_1)$ und $h' = h[p \mapsto \{1\}]$. Zu zeigen bleibt:

(i) $\Vdash_{\mathrm{R}^{h'}_{l_0}} (\pi_{l_0}(T'), T'_{l_0}) = (\pi_{l_0}(T_1), \pi_{l_0}(T_1))$
(ii) $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T'), T_l) = (\pi_l(T_1), \mathsf{alt}_p(\pi_l(T_1), \pi_l(T_2)))$ für alle $l \in L \setminus \{l_0\}$.

Weil $T_l = \mathsf{alt}_p(\pi_l(T_1), \pi_l(T_2))$ wohlnummeriert ist (für beliebiges $l \in L$), ist auch $\pi_l(T_1)$ wohlnummeriert und $\mathrm{top}(\pi_l(T_1)) \in \{\perp, p.1.q_1\}$, $q_1 \in \mathrm{Path}$. Aus Invariante (I8) folgt $h(p.r) = \emptyset$ für alle $r \in \mathrm{Path}$. Somit gilt $h'(p.1.q_1.r) = h(p.1.q_1.r) = \emptyset$ für alle $r \in \mathrm{Path}$. Mit Lemma 4.20 folgt $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T_1), \pi_l(T_1))$. Für $l = l_0$ folgt (i). Für $l \in L \setminus \{l_0\}$ folgt (ii) mit Regel [alt^1] und $h'(p) = \{1\}$.

Fall (alt$^2_{\mathrm{G}}$): Analog zu Fall (alt$^1_{\mathrm{G}}$) mit Regel [alt^2] und Axiom (alt$^2_{\mathrm{L}}$).

$T = \mathsf{loop}_p(T_1)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$ kommen nur (loop$^1_{\mathrm{G}}$) oder (loop$^2_{\mathrm{G}}$) in Frage. Für jedes $l \in L$ kommt zur Ableitung von $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T), T_l)$ aufgrund von $\pi_l(T) = \mathsf{loop}_p(\pi_l(T_1))$ nur das Axiom [loop^3] in Frage. Somit ist $T_l = \mathsf{loop}_p(\pi_l(T_1))$ für alle $l \in L$ und es gilt $h(p) \notin \{\{1\}, \{2\}\}$. Mit Invariante (I9) folgt $h(p) = \emptyset$. Mit Invariante (I7) folgt $h(p.r) = \emptyset$ für alle $r \in \mathrm{Path}$.

Fall (loop$^1_{\mathrm{G}}$): Dann ist $\bar{a} = \tau$ und $T' = \mathsf{Empty}$. Wir wählen eine Lifeline $l_0 \in L$.

Mit ($\mathrm{loop}^1_{\mathrm{L}}$) und $h(p) = \emptyset$ folgt $l_0 : T_{l_0}, h \xrightarrow{\tau}_{\mathrm{L}} l_0 : \mathsf{Empty}, h'$ mit $h' = h[p \mapsto \{1\}]$. Mit [Empty] folgt $\Vdash_{\mathrm{R}^{h'}_{l_0}} (\pi_{l_0}(T'), \mathsf{Empty})$. Mit [Empty] und [$\mathrm{loop}^1$] und $h'(p) = \{1\}$ folgt $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_l(T'), T_l)$ für alle $l \in L \setminus \{l_0\}$.

Fall ($\mathrm{loop}^2_{\mathrm{G}}$): Dann ist $\bar{a} = \tau$ und $T' = \mathsf{seq}_p(T_1, \mathsf{loop}_{p.2}({}_{p.2}T_1))$. Wähle ein $l_0 \in L$. Mit ($\mathrm{loop}^2_{\mathrm{L}}$) und $h(p) = \emptyset$ folgt $l_0 : T_{l_0}, h \xrightarrow{\tau}_{\mathrm{L}} l_0 : T'_{l_0}, h'$ mit $h' = h[p \mapsto \{2\}]$ und $T'_{l_0} = \mathsf{seq}_p(\pi_{l_0}(T_1), \mathsf{loop}_{p.2}({}_{p.2}\pi_{l_0}(T_1)))$. Zu zeigen bleibt

(i) $\Vdash_{\mathrm{R}^{h'}_{l_0}} (\pi_{l_0}(T'), T'_{l_0}) =$
$$(\mathsf{seq}_p(\pi_{l_0}(T_1), \mathsf{loop}_{p.2}({}_{p.2}\pi_{l_0}(T_1))), \mathsf{seq}_p(\pi_{l_0}(T_1), \mathsf{loop}_{p.2}({}_{p.2}\pi_{l_0}(T_1))))$$

(ii) $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_l(T'), T_l) =$
$$(\mathsf{seq}_p(\pi_l(T_1), \mathsf{loop}_{p.2}({}_{p.2}\pi_l(T_1))), \mathsf{loop}_p(\pi_l(T_1))) \quad \text{für alle } l \in L \setminus \{l_0\}$$

wobei wir $\pi_l({}_{p.2}T_1) = {}_{p.2}\pi_l(T_1)$ verwendet haben. Weil $T_l = \mathsf{loop}_p(\pi_l(T_1))$ wohlnummeriert ist (für beliebiges $l \in L$), ist auch $\pi_l(T_1)$ wohlnummeriert und es gilt $\mathrm{top}(\pi_l(T_1)) \in \{\bot, p.1.q_1\}$, $q_1 \in \mathrm{Path}$. Es gilt $h'(p.1.q_1.r) = h(p.1.q_1.r) = \emptyset$ für alle $r \in \mathrm{Path}$. Mit Lemma 4.20 folgt (iii) $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_l(T_1), \pi_l(T_1))$ für alle $l \in L$. Weil $h'(p.2) = h(p.2) = \emptyset \notin \{\{1\}, \{2\}\}$ ist, folgt mit Axiom [loop^3] dass gilt: (iv) $\Vdash_{\mathrm{R}^{h'}_{l}} (\mathsf{loop}_{p.2}({}_{p.2}\pi_l(T_1)), \mathsf{loop}_{p.2}({}_{p.2}\pi_l(T_1)))$ für alle $l \in L$. Für $l = l_0$ folgt (i) aus (iii), (iv) und Regel [seq^1]. Für $l \neq l_0$ folgt (ii) aus (iii), (iv) und Regel [loop^2] und $h'(p) = \{2\}$.

$T = \mathsf{seq}_p(T_1, T_2)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$ kommen nur ($\mathrm{seq}^1_{\mathrm{G}}$), ($\mathrm{seq}^2_{\mathrm{G}}$) oder ($\mathrm{seq}^3_{\mathrm{G}}$) in Frage. Entsprechend unterscheiden wir drei Fälle:

Fall ($\mathrm{seq}^2_{\mathrm{G}}$): Dann gilt $T_1 = \mathsf{Empty}$, $T' = T_2$ und $\bar{a} = \tau$. Sei $l \in L$ beliebig. Es ist $\pi_l(T) = \mathsf{seq}_p(\mathsf{Empty}, \pi_l(T_2))$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}^{h}_{l}} (\pi_l(T), T_l)$ kommen nur [seq^1] oder [seq^2] in Frage.

Unterfall [seq^1]: Dann ist $T_l = \mathsf{seq}_p(T^1_l, T^2_l)$ und es gilt $\Vdash_{\underline{\mathrm{R}}^{h}_{l}} (\mathsf{Empty}, T^1_l)$ und $\Vdash_{\underline{\mathrm{R}}^{h}_{l}} (\pi_l(T_2), T^2_l) = (\pi_l(T'), T^2_l)$. Aus $\Vdash_{\underline{\mathrm{R}}^{h}_{l}} (\mathsf{Empty}, T^1_l)$ folgt $T^1_l = \mathsf{Empty}$. Somit ist $T_l = \mathsf{seq}_p(\mathsf{Empty}, T^2_l)$ und mit Axiom ($\mathrm{seq}^2_{\mathrm{L}}$) folgt $l : T_l, h \xrightarrow{\tau}_{\mathrm{L}} l : T^2_l, h$.

Unterfall [seq^2]: Dann ist $\Vdash_{\underline{\mathrm{R}}^{h}_{l}} (\pi_l(T_2), T_l) = (\pi_l(T'), T_l)$.

Bemerkung: Der globale Aufräumschritt für $\mathsf{seq}_p(\mathsf{Empty}, T_2)$ wird simuliert, indem jede Lifeline, die den lokalen Aufräumschritt für das korrespondierende Konstrukt $\mathsf{seq}_p(\mathsf{Empty}, T^2_l)$ bisher noch nicht ausgeführt hat, eben diesen Aufräumschritt ausführt; siehe Unterfall [seq^1]. Demgegenüber können Lifelines, die den Aufräumschritt für $\mathsf{seq}_p(\mathsf{Empty}, T^2_l)$ bereits ausgeführt haben, ihre Konfiguration beibehalten; siehe Unterfall [seq^2].

Fall ($\mathrm{seq}^1_{\mathrm{G}}$): Dann ist $T' = \mathsf{seq}_p(T'_1, T_2)$ und $T_1 \xrightarrow{\bar{a}}_{\mathrm{G}} T'_1$. Für jede Lifeline $l \in L$ existiert (mindestens) ein Ableitungsbaum $d_l \in \mathrm{D}_{\underline{\mathrm{R}}^{h}_{l}}$ mit $d_l \Vdash_{\underline{\mathrm{R}}^{h}_{l}} (\pi_l(T), T_l)$. Es sei mit d_l einer dieser existenten Ableitungsbäume ausgewählt und fest vorgegeben. Aufgrund von $\pi_l(T) = \mathsf{seq}_p(\pi_l(T_1), \pi_l(T_2))$ kann im letzten Schritt des induktiven Aufbaus von d_l nur die Regel [seq^1] oder [seq^2] angewendet worden sein. Je nachdem welche dieser Regeln angewendet wurde, “verpacken” wir die Lifeline l

in einer der beiden disjunkten Mengen $L_{[\mathrm{seq}^1]}$ oder $L_{[\mathrm{seq}^2]}$:

Fall [seq¹]: Dann sei $l \in L_{[\mathrm{seq}^1]}$. Es gilt $T_l = \mathsf{seq}_p(T_l^1, T_l^2)$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T_1), T_l^1)$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T_2), T_l^2)$.

Fall [seq²]: Dann sei $l \in L_{[\mathrm{seq}^2]}$. Es gilt $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T_1), \mathsf{Empty})$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T_2), T_l)$.

Es gilt $L = L_{[\mathrm{seq}^1]} \cup L_{[\mathrm{seq}^2]}$ und $L_{[\mathrm{seq}^1]} \cap L_{[\mathrm{seq}^2]} = \emptyset$. Um die einfache Beweisidee nicht mit einer formalistischen Vielzahl von Indizierungen zuzuschütten, erlauben wir uns o. B. d. A. anzunehmen, dass gilt: $L_{[\mathrm{seq}^1]} = \{l_1, \ldots, l_k\}$ und $L_{[\mathrm{seq}^2]} = \{l_{k+1}, \ldots, l_n\}$ mit $0 \leq k \leq n$.

Wir beabsichtigen, die Induktionsvoraussetzung für T_1 anzuwenden. Hierzu definieren wir eine Konfiguration $\tilde{\gamma} = [l_1 : \tilde{T}_{l_1}, \ldots, l_n : \tilde{T}_{l_n}, h] \in \mathrm{Conf}^L$, so dass gilt: $\Vdash_{\mathrm{R}_{l_i}^h} (\pi_{l_i}(T_1), \tilde{T}_{l_i})$ für jedes $i \in \{1, \ldots, n\}$. Offenkundig leistet

$$\tilde{T}_l := \begin{cases} T_l^1 & \text{falls } l \in L_{[\mathrm{seq}^1]} \\ \mathsf{Empty} & \text{falls } l \in L_{[\mathrm{seq}^2]} \end{cases}$$

das Gewünschte. Weil T wohlnummeriert ist, ist auch T_1 wohlnummeriert. Weil T_l wohlnummeriert ist, ist auch T_l^1 wohlnummeriert für alle $l \in L_{[\mathrm{seq}^1]}$. Empty ist sowieso wohlnummeriert. Weil $T_{l_1}, \ldots, T_{l_n}, h$ die Invarianten (I3), (I5), (I6), (I7), (I8), (I9) erfüllen, gilt dies auch für $\tilde{T}_{l_1}, \ldots, \tilde{T}_{l_n}, h$—wie man bei Betrachtung der Invarianten leicht erkennt. Aus der Induktionsvoraussetzung $A(T_1)$ folgt:

$$\begin{aligned} \tilde{\gamma} = {} & [l_1 : T_{l_1}^1, \ldots, l_k : T_{l_k}^1, l_{k+1} : \mathsf{Empty}, \ldots, l_n : \mathsf{Empty},\ h] \stackrel{\bar{a}}{\Longrightarrow}_{\mathrm{L}} \\ & [l_1 : T_{l_1}^{1\prime}, \ldots, l_k : T_{l_k}^{1\prime}, l_{k+1} : \mathsf{Empty}, \ldots, l_n : \mathsf{Empty}, h'] \end{aligned}$$

$$\begin{aligned} &\text{mit (a)} \ \Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_1'), T_l^{1\prime}) \ \text{ für alle } l \in L_{[\mathrm{seq}^1]} \\ &\text{und (b)} \ \Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_1'), \mathsf{Empty}) \ \text{ für alle } l \in L_{[\mathrm{seq}^2]} \end{aligned}$$

Durch sukzessive Anwendung von Regel ($\mathrm{seq}_{\mathrm{L}}^1$) folgt:

$$\begin{aligned} & [l_1 : \mathsf{seq}_p(T_{l_1}^1, T_{l_1}^2), \ldots, l_k : \mathsf{seq}_p(T_{l_k}^1, T_{l_k}^2), l_{k+1} : \mathsf{Empty}, \ldots, l_n : \mathsf{Empty},\ h] \stackrel{\bar{a}}{\Longrightarrow}_{\mathrm{L}} \\ & [l_1 : \mathsf{seq}_p(T_{l_1}^{1\prime}, T_{l_1}^2), \ldots, l_k : \mathsf{seq}_p(T_{l_k}^{1\prime}, T_{l_k}^2), l_{k+1} : \mathsf{Empty}, \ldots, l_n : \mathsf{Empty}, h'] \end{aligned}$$

Die Lifelines $l \in L_{[\mathrm{seq}^2]} = \{l_{k+1}, \ldots, l_n\}$ nehmen an den obigen lokalen Übergängen nicht teil. Welche Konfigurationsterme diese Lifelines haben, ist für die Ableitbarkeit der semantischen Schritte irrelevant. Es gilt also auch das Folgende:

$$\begin{aligned} & [l_1 : \mathsf{seq}_p(T_{l_1}^1, T_{l_1}^2), \ldots, l_k : \mathsf{seq}_p(T_{l_k}^1, T_{l_k}^2), l_{k+1} : T_{l_{k+1}}, \ldots, l_n : T_{l_n},\ h] \stackrel{\bar{a}}{\Longrightarrow}_{\mathrm{L}} \\ & [l_1 : \mathsf{seq}_p(T_{l_1}^{1\prime}, T_{l_1}^2), \ldots, l_k : \mathsf{seq}_p(T_{l_k}^{1\prime}, T_{l_k}^2), l_{k+1} : T_{l_{k+1}}, \ldots, l_n : T_{l_n}, h'] \end{aligned}$$

Zu zeigen bleibt: (i) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'), \mathsf{seq}_p(T_l^{1\prime}, T_l^2))$ für alle $l \in L_{[\mathrm{seq}^1]}$

(ii) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'), T_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$

Weil der globale Konfigurationsterm $T = \mathsf{seq}_p(T_1, T_2)$ wohlnummeriert ist, sind auch T_1 und T_2 wohlnummeriert und es gilt: $\mathrm{top}(T_1) \in \{\bot, p.1.q_1\}$, $q_1 \in \mathrm{Path}$ und $\mathrm{top}(T_2) \in \{\bot, p.2.q_2\}$, $q_2 \in \mathrm{Path}$. Mit Lemma 4.23 folgt:

$$\begin{aligned}&\mathrm{top}(T_l^1) \in \{\bot, p.1.q_1.r_l^1\},\ r_l^1 \in \mathrm{Path} \text{ für alle } l \in L_{[\mathrm{seq}^1]} \text{ und}\\&\mathrm{top}(T_l^2) \in \{\bot, p.2.q_2.r_l^2\},\ r_l^2 \in \mathrm{Path} \text{ für alle } l \in L_{[\mathrm{seq}^1]} \text{ und}\\&\mathrm{top}(T_l) \in \{\bot, p.2.q_2.r_l\},\ r_l \in \mathrm{Path} \text{ für alle } l \in L_{[\mathrm{seq}^2]}.\end{aligned}$$

Wegen $\mathrm{top}(T_l^1) \in \{\bot, p.1.q_1.r_l^1\}$ für alle $l \in L_{[\mathrm{seq}^1]}$ und wegen Satz 4.3.2 (b) in analoger Anwendung auf T_l^1 (d. h. wenn der top most operator path von T_l^1 in PathSpace(p) liegt, so bleibt er in diesem Raum oder wechselt nach $\bot$) und wegen Lemma 4.21 folgt: Für alle $q \in \mathrm{Path} \setminus \mathrm{PathSpace}(p.1.q_1)$ ist $h(q) = h'(q)$.

Mit Lemma 4.22 folgt: (c) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_2), T_l^2)$ für alle $l \in L_{[\mathrm{seq}^1]}$

(d) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_2), T_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$

Mit (a) und (c) und [seq^1] folgt (i). Mit (b) und (d) und [seq^2] folgt (ii).

Fall ($\mathrm{seq}_\mathrm{G}^3$): Dann ist $T' = \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar a)}(T_1), T_2')$ und $T_2 \xrightarrow{\bar a}_\mathrm{G} T_2'$. Die Mengen $L_{[\mathrm{seq}^1]}$ und $L_{[\mathrm{seq}^2]}$ seien genauso wie im Fall ($\mathrm{seq}_\mathrm{G}^1$) definiert. Wir beabsichtigen, die Induktionsvoraussetzung für T_2 anzuwenden. Hierzu definieren wir eine Konfiguration $\tilde\gamma = [l_1 : \tilde T_{l_1}, \ldots, l_n : \tilde T_{l_n}, h] \in \mathrm{Conf}^L$, so dass gilt: $\Vdash_{\mathrm{R}_{l_i}^h} (\pi_{l_i}(T_2), \tilde T_{l_i})$ für jedes $i \in \{1, \ldots, n\}$. Offenkundig leistet

$$\tilde T_l := \begin{cases} T_l^2 & \text{falls } l \in L_{[\mathrm{seq}^1]} = \{l_1, \ldots, l_k\}\\ T_l & \text{falls } l \in L_{[\mathrm{seq}^2]} = \{l_{k+1}, \ldots, l_n\}\end{cases}$$

das Gewünschte. Aus der Induktionsvoraussetzung $A(T_2)$ folgt:

$$\begin{aligned}\tilde\gamma = [l_1 : T_{l_1}^2, \ldots, l_k : T_{l_k}^2, l_{k+1} : T_{l_{k+1}}, \ldots, l_n : T_{l_n},\ h] &\overset{\bar a}{\Longrightarrow}_\mathrm{L}\\ [l_1 : T_{l_1}^{2\prime}, \ldots, l_k : T_{l_k}^{2\prime}, l_{k+1} : T'_{l_{k+1}}, \ldots, l_n : T'_{l_n}, h']&\end{aligned}$$

mit (a) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_2'), T_l^{2\prime})$ für alle $l \in L_{[\mathrm{seq}^1]}$

und (b) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_2'), T_l')$ für alle $l \in L_{[\mathrm{seq}^2]}$

Mit einer sukzessiven Anwendung von Regel ($\mathrm{seq}_\mathrm{L}^3$) und mit $\mathrm{R}^*_{\alpha(\tau)} = \mathrm{R}^*_\emptyset = \mathrm{id}$ und $\mathrm{R}^*_{l'}(S) = \mathrm{id}(S)$ für jedes $S \in \mathrm{Term}^l$ mit $l \neq l'$ erhält man:

$$\begin{aligned}&[l_1 : \mathsf{seq}_p(\qquad T_{l_1}^1, T_{l_1}^2), \ldots, l_k : \mathsf{seq}_p(\qquad T_{l_k}^1, T_{l_k}^2), l_{k+1} : T_{l_{k+1}}, \ldots, l_n : T_{l_n},\ h] \overset{\bar a}{\Longrightarrow}_\mathrm{L}\\&[l_1 : \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar a)}(T_{l_1}^1), T_{l_1}^{2\prime}), \ldots, l_k : \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar a)}(T_{l_k}^1), T_{l_k}^{2\prime}), l_{k+1} : T_{l_{k+1}}, \ldots, l_n : T_{l_n}, h']\end{aligned}$$

Zu zeigen bleibt: (i) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'), \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar a)}(T_l^1), T_l^{2\prime}))$ für alle $l \in L_{[\mathrm{seq}^1]}$

(ii) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'), T_l')$ für alle $l \in L_{[\mathrm{seq}^2]}$

Sei $l \in L_{[\mathrm{seq}^1]}$ beliebig. Es gilt $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T_1), T_l^1)$. Mit Hilfe von Lemma 4.19 folgt $\Vdash_{\underline{\mathrm{R}}_l^h} (\mathrm{R}^*_{\alpha(\bar a)}(\pi_l(T_1)), \mathrm{R}^*_{\alpha(\bar a)}(T_l^1))$. Aufgrund von Lemma 4.2 können Projektion und Restriktion vertauscht werden. Es gilt $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(\mathrm{R}^*_{\alpha(\bar a)}(T_1)), \mathrm{R}^*_{\alpha(\bar a)}(T_l^1))$. Diese Aussage lässt sich wie in Fall ($\mathrm{seq}_\mathrm{G}^1$) von h nach h' übertragen, d. h. es gilt $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(\mathrm{R}^*_{\alpha(\bar a)}(T_1)), \mathrm{R}^*_{\alpha(\bar a)}(T_l^1))$. Hieraus und aus (a) und [seq^1] folgt (i).

Sei $l \in L_{[\mathrm{seq}^2]}$ beliebig. Es gilt $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T_1), \mathsf{Empty})$. Mit Hilfe von Lemma 4.19 und Lemma 4.2 folgt $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(\mathrm{R}^*_{\alpha(\bar a)}(T_1)), \mathsf{Empty})$. Mit Lemma 4.22.1 folgt $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(\mathrm{R}^*_{\alpha(\bar a)}(T_1)), \mathsf{Empty})$. Hieraus und aus (b) und [seq^2] folgt (ii).

$T = \mathsf{strict}_p(T_1, T_2)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$ kommen $(\mathrm{strict}^1_{\mathrm{G}})$ oder $(\mathrm{strict}^2_{\mathrm{G}})$ in Frage. Aufgrund von $\pi_l(T) = \mathsf{strict}_p(\pi_l(T_1), \pi_l(T_2))$ kommt zur Ableitung von $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T), T_l)$ nur die Regel $[\mathrm{strict}^2]$ in Frage (und zwar für jedes $l \in L$). Es folgt: $T_l = \mathsf{strict}_p(T^1_l, \pi_l(T_2))$ und $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T_1), T^1_l)$ und $h(p) \neq L$.

Fall $(\mathrm{strict}^2_{\mathrm{G}})$: Dann ist $T_1 = \mathsf{Empty}$ und $T' = T_2$ und $\bar{a} = \tau$. Wegen $T_1 = \mathsf{Empty}$ gilt $\Vdash_{\underline{\mathrm{R}}^h_l} (\mathsf{Empty}, T^1_l)$ und somit ist $T^1_l = \mathsf{Empty}$ für alle $l \in L$. Folglich gilt: $T_l = \mathsf{strict}_p(\mathsf{Empty}, \pi_l(T_2))$ für alle $l \in L$. Mit Axiom $(\mathrm{strict}^3_{\mathrm{L}})$ folgt:

$$[l_1 : T_{l_1}, \ldots, l_n : T_{l_n}, h] \xrightarrow{\tau^*}_{\mathrm{L}} [l_1 : T_{l_1}, \ldots, l_n : T_{l_n}, h'] \quad \text{wobei } h' = h[p \mapsto L]$$

Bemerkung: Die letzte Lifeline meldet, dass sie das Konstrukt $\mathsf{strict}_p(\mathsf{Empty}, \ldots)$ angetroffen hat. Zu zeigen bleibt:

$$\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T'), T_l) = \\ (\pi_l(T_2), \mathsf{strict}_p(\mathsf{Empty}, \pi_l(T_2))) \quad \text{für alle } l \in L.$$

Dies folgt mit Regel $[\mathrm{strict}^1]$ und $h'(p) = L$, sofern man noch zeigt:

$$\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T_2), \pi_l(T_2)) \text{ für alle } l \in L.$$

Es ist $\mathrm{top}(\pi_l(T_2)) \in \{\bot, p.2.q_2\}$, $q_2 \in \mathrm{Path}$. Mit $h(p) \neq L$ und Invariante (I5) folgt $h(p.2.r) = \emptyset$ für alle $r \in \mathrm{Path}$. Somit ist $h'(p.2.q_2.r) = \emptyset$ für alle $r \in \mathrm{Path}$. Mit Lemma 4.20 folgt $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T_2), \pi_l(T_2))$.

Fall $(\mathrm{strict}^1_{\mathrm{G}})$: Dann ist $T' = \mathsf{strict}_p(T'_1, T_2)$ und $T_1 \xrightarrow{\bar{a}}_{\mathrm{G}} T'_1$. Aus $A(T_1)$ folgt:

$$[l_1 : T^1_{l_1}, \ldots, l_n : T^1_{l_n}, h] \overset{\bar{a}}{\Rightarrow}_{\mathrm{L}} [l_1 : T^{1\prime}_{l_1}, \ldots, l_n : T^{1\prime}_{l_n}, h']$$

mit (*) $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T'_1), T^{1\prime}_l)$ für alle $l \in L$. Mit Regel $(\mathrm{strict}^1_{\mathrm{L}})$ folgt:

$$[l_1 : \mathsf{strict}_p(T^1_{l_1}, \pi_{l_1}(T_2)), \ldots, l_n : \mathsf{strict}_p(T^1_{l_n}, \pi_{l_n}(T_2)), h] \overset{\bar{a}}{\Rightarrow}_{\mathrm{L}} \\ [l_1 : \mathsf{strict}_p(T^{1\prime}_{l_1}, \pi_{l_1}(T_2)), \ldots, l_n : \mathsf{strict}_p(T^{1\prime}_{l_n}, \pi_{l_n}(T_2)), h']$$

Zu zeigen bleibt:

$$\Vdash_{\mathrm{R}^{h'}_l} (\mathsf{strict}_p(\pi_l(T'_1), \pi_l(T_2)), \mathsf{strict}_p(T^{1\prime}_l, \pi_l(T_2))) \quad \text{für alle } l \in L.$$

Dies folgt mit (*) und Regel $[\mathrm{strict}^2]$, sofern wir noch zeigen, dass $h'(p) \neq L$ ist. Es ist $h(p) \neq L$. Somit genügt es $h'(p) = h(p)$ zu zeigen. Es ist $\mathrm{top}(T^1_l) \in \{\bot, p.1.q_l\}$ mit $q_l \in \mathrm{Path}$ für jedes $l \in L$. Mit Satz 4.3.2 und Lemma 4.21 folgt: Für alle $q \in \mathrm{Path} \setminus \mathrm{PathSpace}(p.1)$ ist $h(q) = h'(q)$. Insbesondere ist $h(p) = h'(p)$.

$T = \mathsf{par}_p(T_1, T_2)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} T'$ kommen nur die Regeln $(\mathrm{par}^1_{\mathrm{G}})$, $(\mathrm{par}^2_{\mathrm{G}})$, $(\mathrm{par}^3_{\mathrm{G}})$ oder $(\mathrm{par}^4_{\mathrm{G}})$ in Frage. Entsprechend sind vier Fälle zu unterscheiden. Für jedes $l \in L$ kommen zur Ableitung von $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T), T_l)$ wegen $\pi_l(T) = \mathsf{par}_p(\pi_l(T_1), \pi_l(T_2))$ nur die Regeln $[\mathrm{par}^1]$, $[\mathrm{par}^2]$ oder $[\mathrm{par}^3]$ in Frage. Entsprechend ist die Lifelinemenge zu zerlegen: $L = L_{[\mathrm{par}^1]} \mathbin{\dot\cup} L_{[\mathrm{par}^2]} \mathbin{\dot\cup} L_{[\mathrm{par}^3]}$. Mit den Methoden aus dem Fall $T = \mathsf{seq}_p(T_1, T_2)$ sollte die weitere Argumentation kein Problem darstellen.

Ad 2. Der Beweis erfolgt durch einfache Fallunterscheidung nach der Form von T. Wie im Beweis von 1. wenden wir das Lemma 4.18 auf jede Lifeline an.

$T = \mathsf{None}(a)$ oder Empty: Nicht möglich, weil $\mathsf{None}(a) \overset{\bar{a}}{\not\to}_{\mathrm{G}} T'$ und $\mathsf{Empty} \overset{\bar{a}}{\not\to}_{\mathrm{G}} T'$.

$T = a$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{Empty}$ kommt nur $(\mathrm{basic_G})$ in Frage. Folglich ist $a = \bar{a} \in \mathbb{A}^L$. Sei $l \in L$. Falls $l \notin \alpha(a)$ ist, dann folgt $\pi_l(T) = \pi_l(a) = \mathsf{Empty}$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T), T_l)$ kommt nur [Empty] in Frage, d. h. es ist $T_l = \mathsf{Empty}$. Falls $l \in \alpha(a)$ ist, dann folgt $\pi_l(T) = \pi_l(a) = a$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T), T_l)$ kommt nur [Action] in Frage, d. h. es ist $T_l = a$. Mit Axiom $(\mathrm{basic_L})$ folgt die Behauptung.

$T = \mathsf{strict}_p(T_1, T_2)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{Empty}$ kommt nur $(\mathrm{strict}^2_{\mathrm{G}})$ in Frage. Es ist $T_1 = T_2 = \mathsf{Empty}$ und $\bar{a} = \tau$. Für jedes $l \in L$ gilt $\pi_l(T) = \mathsf{strict}_p(\mathsf{Empty}, \mathsf{Empty})$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T), T_l)$ kommt nur $[\mathrm{strict}^2]$ in Frage. Es folgt $h(p) \neq L$ und $T_l = \mathsf{strict}_p(\mathsf{Empty}, \mathsf{Empty})$ für alle $l \in L$. Mit den Axiomen $(\mathrm{strict}^3_{\mathrm{L}})$ und $(\mathrm{strict}^2_{\mathrm{L}})$ folgt die Behauptung.

$T = \mathsf{seq}_p(T_1, T_2)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{Empty}$ kommt nur $(\mathrm{seq}^2_{\mathrm{G}})$ in Frage. Es ist $T_1 = T_2 = \mathsf{Empty}$ und $\bar{a} = \tau$. Für jedes $l \in L$ gilt $\pi_l(T) = \mathsf{seq}_p(\mathsf{Empty}, \mathsf{Empty})$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T), T_l)$ kommen nur $[\mathrm{seq}^1]$ oder $[\mathrm{seq}^2]$ in Frage. Im Fall $[\mathrm{seq}^1]$ ist $T_l = \mathsf{seq}_p(\mathsf{Empty}, \mathsf{Empty})$. Mit Axiom $(\mathrm{seq}^2_{\mathrm{L}})$ folgt $l : T_l, h \xrightarrow{\tau}_{\mathrm{L}} l : \mathsf{Empty}, h$. Im Fall $[\mathrm{seq}^2]$ ist $T_l = \mathsf{Empty}$. Daraus folgt die Behauptung.

$T = \mathsf{par}_p(T_1, T_2)$: Ähnlich zu Fall $T = \mathsf{seq}_p(T_1, T_2)$.

$T = \mathsf{loop}_p(T_1)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{Empty}$ kommt nur $(\mathrm{loop}^1_{\mathrm{G}})$ in Frage. Es folgt $\bar{a} = \tau$. Für jedes $l \in L$ gilt $\pi_l(T) = \mathsf{loop}_p(\pi_l(T_1))$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T), T_l)$ kommt nur $[\mathrm{loop}^3]$ in Frage. Es folgt $h(p) \notin \{\{1\}, \{2\}\}$ und $T_l = \mathsf{loop}_p(\pi_l(T_1))$ für alle $l \in L$. Mit Invariante (I9) folgt $h(p) = \emptyset$. Mit Axiom $(\mathrm{loop}^1_{\mathrm{L}})$ folgt die Behauptung.

$T = \mathsf{alt}_p(T_1, T_2)$: Zur Ableitung von $T \xrightarrow{\bar{a}}_{\mathrm{G}} \mathsf{Empty}$ kommen nur $(\mathrm{alt}^1_{\mathrm{G}})$ oder $(\mathrm{alt}^2_{\mathrm{G}})$ in Frage. Es liege o. B. d. A. der Fall $(\mathrm{alt}^1_{\mathrm{G}})$ vor. Es ist $T_1 = cEmpty$ und $\bar{a} = \tau$. Für jedes $l \in L$ gilt $\pi_l(T) = \mathsf{alt}_p(\mathsf{Empty}, \pi_l(T_2))$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T), T_l)$ kommt nur $[\mathrm{alt}^3]$ in Frage. Es folgt $h(p) \notin \{\{1\}, \{2\}\}$ und $T_l = \mathsf{alt}_p(\mathsf{Empty}, \pi_l(T_2))$ für alle $l \in L$. Mit Invariante (I9) folgt $h(p) = \emptyset$. Mit Axiom $(\mathrm{alt}^1_{\mathrm{L}})$ folgt die Behauptung. □

4.9.2 Simulation lokaler semantischer Schritte

Das folgende Lemma kann als Umkehrung von Lemma 4.16 aufgefasst werden.

Lemma 4.25 Sei $l \in L$ und $h \in \mathrm{H}^L_{\mathrm{fin}}$. Es gelte $\Vdash_{\mathrm{R}_l^h} (S, T)$. Dann existiert ein $\tilde{S} \in \mathrm{Term}^{\{l\}(\{l\})}$, so dass gilt $S \in \mathrm{RTerm}[\tilde{S}]$ und $\Vdash_{\mathrm{R}_l^h} (\tilde{S}, T)$, wobei im letzten Schritt der Ableitung keine der Regeln $[\mathrm{seq}^2]$, $[\mathrm{par}^2]$, $[\mathrm{par}^3]$ benutzt wurde (symbolisch: $\Vdash_{\underline{\mathrm{R}}_l^h} (\tilde{S}, T)$).

Beweis. Der Beweis erfolgt durch Ableitungsinduktion für Judgements (S,T). Für alle Regeln außer [seq^2], [par^2] und [par^3] ist die Behauptung mit $\tilde{S} := S$ trivial wahr.

[seq^2] Dann ist $S = \mathsf{seq}_p(S_1, S_2)$ und $\Vdash_{\mathrm{R}_l^h} (S_1, \mathsf{Empty})$ und $\Vdash_{\mathrm{R}_l^h} (S_2, T)$. Aus $\Vdash_{\mathrm{R}_l^h} (S_1, \mathsf{Empty})$ und Satz 4.14 und sla$(\mathsf{Empty}, h) = \mathsf{Empty}$ und Lemma 4.12 folgt $S_1 = ET_1 \in$ ETerm. Aus $\Vdash_{\mathrm{R}_l^h} (S_2, T)$ und der I. V. folgt: Es gibt ein $\tilde{S}_2 \in \mathrm{Term}^{\{l\}(\{l\})}$ und ein $RC_2 \in$ RContext, so dass gilt: $S_2 = RC_2[\tilde{S}_2]$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (\tilde{S}_2, T)$. Es gilt: $S = \mathsf{seq}_p(S_1, S_2) = \mathsf{seq}_p(ET_1, RC_2)[\tilde{S}_2] \in \mathrm{RTerm}[\tilde{S}_2]$.

[par^2] oder [par^3]: Analog. □

Das nächste Lemma ist—quasi—eine Umkehrung von Lemma 4.18. Innerhalb der Koinduktion dient das Lemma 4.18 dazu, alle τ-Nachführschritte, die von der sla-Funktion für eine Lifeline l gefaked werden, tatsächlich auszuführen. Hierdurch werden die lokalen Konfigurationsterme T_l gleichsam zum Repräsentanten ihrer sla-Äquivalenzklasse "vorgespult". Das Lemma 4.26 dient dazu, dieses "Vorspulen" rückgängig machen zu können. Es sei hier daran erinnert, dass das Symbol $\underline{\mathrm{L}}$ das Regelsystem der lokalen operationalen Semantik aus Tafel 4.3 ohne die Axiome (seq$^2_{\mathrm{L}}$), (par$^3_{\mathrm{L}}$) und (par$^4_{\mathrm{L}}$) bezeichnet.

Lemma 4.26 Sei $l \in L$ und $h \in \mathrm{H}^L_{\mathrm{fin}}$. Es gelte $\Vdash_{\mathrm{R}_l^h} (S, T')$ und $l : T, h \xrightarrow{\tau}_{\underline{\mathrm{L}}} l : T', h$. Ferner erfüllen T, h die Invariante (I6). Dann gilt: $\Vdash_{\mathrm{R}_l^h} (S, T)$.

Beweis. Wir führen diesen Beweis durch eine Ableitungsinduktion für Judgements $l : T, h \xrightarrow{\tau}_{\underline{\mathrm{L}}} l : T', h$.

Axiome von $\underline{\mathrm{L}}$:

(basic$_{\mathrm{L}}$) Nicht möglich, weil es sich nach Voraussetz. um einen τ-Schritt handelt.

(strict$^2_{\mathrm{L}}$) Dann ist $T = \mathsf{strict}_p(\mathsf{Empty}, T_2)$ und $T_2 = T'$ und $h(p) = L$. Es gilt $\Vdash_{\mathrm{R}_l^h} (S, T') = (S, T_2)$. Mit Regel [strict^1] und $h(p) = L$ folgt $\Vdash_{\mathrm{R}_l^h} (S, T)$.

(strict$^3_{\mathrm{L}}$) Nicht möglich, weil h nach Voraussetzung unverändert bleibt.

(seq$^2_{\mathrm{L}}$), (par$^3_{\mathrm{L}}$), (par$^4_{\mathrm{L}}$) gehören nicht zu $\underline{\mathrm{L}}$.

(loop$^1_{\mathrm{L}}$) Dann ist $T = \mathsf{loop}_p(T_1)$ und $T' = \mathsf{Empty}$ und $T_1 \in \mathrm{Term}^{\{l\}(\{l\})}$ und $h = h[p \mapsto \{1\}]$, d. h. es ist $h(p) = \{1\}$. Es gilt $\Vdash_{\mathrm{R}_l^h} (S, T') = (S, \mathsf{Empty})$. Mit Regel [loop^1] und $h(p) = \{1\}$ und $T_1 \in \mathrm{Term}^{\{l\}(\{l\})}$ folgt $\Vdash_{\mathrm{R}_l^h} (S, T)$.

(loop$^2_{\mathrm{L}}$) Dann ist $T = \mathsf{loop}_p(T_1)$, $T' = \mathsf{seq}_p(T_1, \mathsf{loop}_{p.2}({}_{p.2}T_1))$ und $h = h[p \mapsto \{2\}]$, d. h. es ist $h(p) = \{2\}$. Es gilt $\Vdash_{\mathrm{R}_l^h} (S, T')$. Nach Lemma 4.25 gibt es ein $\tilde{S} \in \mathrm{Term}^{\{l\}(\{l\})}$ mit $S \in \mathrm{RTerm}[\tilde{S}]$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (\tilde{S}, T')$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\tilde{S}, T')$ kommt wegen $T' = \mathsf{seq}_p(T_1, \mathsf{loop}_{p.2}({}_{p.2}T_1))$ nur die Regel [seq^1] in Frage. Es folgt: $\tilde{S} = \mathsf{seq}_p(\tilde{S}_1, \tilde{S}_2)$ und $\Vdash_{\mathrm{R}_l^h} (\tilde{S}_1, T_1)$ und $\Vdash_{\mathrm{R}_l^h} (\tilde{S}_2, \mathsf{loop}_{p.2}({}_{p.2}T_1))$. Mit Regel [loop^2]

und $h(p) = \{2\}$ folgt $\Vdash_{\mathrm{R}_l^h} (\mathsf{seq}_p(\tilde{S}_1, \tilde{S}_2), \mathsf{loop}_p(T_1)) = (\tilde{S}, T)$. Mit Lemma 4.16 und $S \in \mathrm{RTerm}[\tilde{S}]$ folgt $\Vdash_{\mathrm{R}_l^h} (S, T)$.

($\mathrm{alt}^1_{\mathrm{L}}$) Dann ist $T = \mathsf{alt}_p(T_1, T_2)$ und $T_1 = T'$ und $T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$ und $h = h[p \mapsto \{1\}]$, d. h. es ist $h(p) = \{1\}$. Es gilt $\Vdash_{\mathrm{R}_l^h} (S, T') = (S, T_1)$. Mit Regel [$\mathrm{alt}^1$] und $h(p) = \{1\}$ und $T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$ folgt $\Vdash_{\mathrm{R}_l^h} (S, T)$.

($\mathrm{alt}^2_{\mathrm{L}}$) Analog zu Fall ($\mathrm{alt}^1_{\mathrm{L}}$) mit Regel [$\mathrm{alt}^2$].

Echte Regeln von $\underline{\mathrm{L}}$:

($\mathrm{strict}^1_{\mathrm{L}}$) Dann ist $T = \mathsf{strict}_p(T_1, T_2)$, $T' = \mathsf{strict}_p(T_1', T_2)$ und $T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$ und $l : T_1, h \xrightarrow{\tau}_{\underline{\mathrm{L}}} l : T_1', h$. Folglich ist $T_1 \neq \mathsf{Empty}$, weil es andernfalls für die Lifeline l keinen ableitbaren Übergang gäbe. Weil T, h die Invariante (I6) erfüllen und $T_1 \neq \mathsf{Empty}$ ist, muss $h(p) \neq L$ sein. Es gilt $\Vdash_{\mathrm{R}_l^h} (S, T')$. Nach Lemma 4.25 gibt es ein $\tilde{S} \in \mathrm{Term}^{\{l\}(\{l\})}$ mit $S \in \mathrm{RTerm}[\tilde{S}]$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (\tilde{S}, T')$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\tilde{S}, T')$ kommt wegen $T' = \mathsf{strict}_p(T_1', T_2)$ und $h(p) \neq L$ nur die Regel [strict^2] in Frage. Es folgt: $\tilde{S} = \mathsf{strict}_p(\tilde{S}_1, T_2)$ und $\Vdash_{\mathrm{R}_l^h} (\tilde{S}_1, T_1')$. Mit der I. V. folgt $\Vdash_{\mathrm{R}_l^h} (\tilde{S}_1, T_1)$. Mit [$\mathrm{strict}^2$] und $h(p) \neq L$ und $T_2 \in \mathrm{Term}^{\{l\}(\{l\})}$ folgt $\Vdash_{\mathrm{R}_l^h} (\mathsf{strict}_p(\tilde{S}_1, T_2), \mathsf{strict}_p(T_1, T_2)) = (\tilde{S}, T)$. Mit Lemma 4.16 und $S \in \mathrm{RTerm}[\tilde{S}]$ folgt $\Vdash_{\mathrm{R}_l^h} (S, T)$.

($\mathrm{seq}^1_{\mathrm{L}}$) Analog zum Fall ($\mathrm{seq}^3_{\mathrm{L}}$).

($\mathrm{seq}^3_{\mathrm{L}}$) Dann ist $T = \mathsf{seq}_p(T_1, T_2)$ und $T' = \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\tau)}(T_1), T_2') = \mathsf{seq}_p(T_1, T_2')$ und $l : T_2, h \xrightarrow{\tau}_{\underline{\mathrm{L}}} l : T_2', h$. Es gilt $\Vdash_{\mathrm{R}_l^h} (S, T')$. Nach Lemma 4.25 gibt es ein $\tilde{S} \in \mathrm{Term}^{\{l\}(\{l\})}$ mit $S \in \mathrm{RTerm}[\tilde{S}]$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (\tilde{S}, T')$. Zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\tilde{S}, T')$ kommt wegen $T' = \mathsf{seq}_p(T_1, T_2')$ nur die Regel [seq^1] in Frage. Es folgt: $\tilde{S} = \mathsf{seq}_p(\tilde{S}_1, \tilde{S}_2)$ und (1) $\Vdash_{\mathrm{R}_l^h} (\tilde{S}_1, T_1)$ und (2) $\Vdash_{\mathrm{R}_l^h} (\tilde{S}_2, T_2')$. Aus (2) und der I. V. folgt (3) $\Vdash_{\mathrm{R}_l^h} (\tilde{S}_2, T_2)$. Aus (1) und (3) und [$\mathrm{seq}^1$] folgt $\Vdash_{\mathrm{R}_l^h} (\tilde{S}, T)$. Mit Lemma 4.16 und $S \in \mathrm{RTerm}[\tilde{S}]$ folgt $\Vdash_{\mathrm{R}_l^h} (S, T)$.

($\mathrm{par}^1_{\mathrm{L}}$) oder ($\mathrm{par}^2_{\mathrm{L}}$) Analog. □

In der Situation $\mathrm{top}(S) = p$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (S, T)$ und $l : T, h \xrightarrow{\bar{a}}_{\mathrm{L}} l : T', h'$ (mit wohlnummerierten Termen S, T) haben wir im Beweis von Satz 4.24 häufig mit Lemma 4.23 argumentiert, dass ein $r \in \mathrm{Path}$ existiert, so dass gilt $\mathrm{top}(T) \in \{\bot, p.r\}$. Dies ermöglichte uns, mittels Lemma 4.21 zu schließen, dass gilt $h(q) = h'(q)$ für alle $q \in \mathrm{Path} \setminus \mathrm{PathSpace}(p)$. Tatsächlich kann man diesen Schluss auch dann ziehen, wenn man lediglich $\Vdash_{\mathrm{R}_l^h} (S, T)$ im *nicht* eingeschränkten Regelsystem R_l^h gegeben hat. Denn die Regeln [strict^1], [loop^1], [alt^1] und [alt^2], die einzig dafür verantwortlich sind, dass im rechten Argument einer Verträglichkeitsbeziehung $\Vdash_{\mathrm{R}_l^h} (S, T)$ ein T mit $\mathrm{top}(T) = q \notin \mathrm{PathSpace}(p)$ auftreten kann, entsprechen durch ihre Seitenbedingungen einer für q "eingeschalteten" sla-Funktion mit einem bezüglich $\leq$ maximalen Wert für $h(q)$. Dabei ist $\leq$ diejenige Ordnung, die in der Formulierung der Invariante (I4) auftritt. Folglich können sich diese maximalen Werte $h(q)$ durch den Übergang $l : T, h \xrightarrow{\bar{a}}_{\mathrm{L}} l : T', h'$ nicht ändern.

Lemma 4.27 Seien $l \in L$ und $h, h' \in \mathrm{H}^L_{\mathrm{fin}}$ und $S, T, T' \in \mathrm{Term}^{\{l\}(\{l\})}$. Es seien S und T wohlnummeriert. Es erfüllen T, h die Invariante (I6). Es gelte ferner $\Vdash_{\mathrm{R}^h_l} (S, T)$ und $l : T, h \xrightarrow{\bar{a}}_{\mathrm{L}} l : T', h'$. Dann gilt:

1. $\mathrm{top}(S) = \bot \implies h' = h'$
2. $\mathrm{top}(S) = p \implies \forall\, q \in \mathrm{Path} \setminus \mathrm{PathSpace}(p) \,.\, \big[h'(q) = h(q)\big]$

Beweis. Der Beweis erfolgt durch Ableitungsinduktion für Judgements (S, T).

[None] oder [Empty] Nicht möglich, weil z. B. $l : \mathsf{None}(a), h \overset{\bar{a}}{\not\to}_{\mathrm{L}} l : T', h$.

[Action] Dann ist $S = T = a$. Zur Ableitung von $l : T, h \xrightarrow{\bar{a}}_{\mathrm{L}} l : T', h'$ kommt nur die Regel ($\mathrm{basic}_{\mathrm{L}}$) in Frage. Es folgt $h' = h$.

[strict^1] Dann ist $T = \mathsf{strict}_{\tilde{p}}(T_1, T_2)$ und $h(\tilde{p}) = L$. Weil T, h die Invariante (I6) erfüllen, folgt $T_1 = \mathsf{Empty}$. Weil $T = \mathsf{strict}_{\tilde{p}}(\mathsf{Empty}, T_2)$ und $h(\tilde{p}) = L$ ist und weil es für Empty keine Übergänge gibt, kommt zur Ableitung von $l : T, h \xrightarrow{\bar{a}}_{\mathrm{L}} l : T', h'$ nur die Regel ($\mathrm{strict}^2_{\mathrm{L}}$) in Frage. Es folgt $h' = h$.

[strict^2], [seq^1], [par^1], [loop^2], [loop^3] oder [alt^3] Es ist $\mathrm{top}(S) = p = \mathrm{top}(T)$ und die Behauptung folgt mit Lemma 4.21.

[seq^2] Dann ist $S = \mathsf{seq}_p(S_1, S_2)$ und $\Vdash_{\mathrm{R}^h_l} (S_2, T)$. Weil S wohlnummeriert ist, ist auch S_2 wohlnummeriert und es gilt $\mathrm{top}(S_2) \in \{\bot, p.2.r_2\}$ mit $r_2 \in \mathrm{Path}$. Mit der I. V. folgt: $h(q) = h'(q)$ für alle $p \in \mathrm{Path} \setminus \mathrm{PathSpace}(p.2.r_2)$. Die Behauptung folgt mit $\mathrm{PathSpace}(p.2.r_2) \subseteq \mathrm{PathSpace} p$.

[loop^1] Dann ist $T = \mathsf{loop}_{\tilde{p}}(T_1)$ und $h(\tilde{p}) = \{1\}$. Die einzige matchende Regel ist ($\mathrm{loop}^1_{\mathrm{L}}$). Weil $h(\tilde{p}) = \{1\}$ ist, folgt $h' = h[\tilde{p} \mapsto \{1\}] = h$.

[alt^1] oder [alt^2] Analog zu Fall [loop^1] mit Axiom ($\mathrm{alt}^1_{\mathrm{L}}$) oder ($\mathrm{alt}^2_{\mathrm{L}}$) □

Satz 4.28 Sei $L = \{l_1, \ldots, l_n\} \subseteq \mathbb{I}$ mit $l_1, \ldots, l_n$ paarw. verschieden und $n \geq 0$. Seien $T \in \mathrm{Term}^{L(L)}$, $\gamma = [l_1 : T_{l_1}, \ldots, l_n : T_{l_n}, h] \in \mathrm{Conf}^L$, $\bar{a} \in \mathbb{A}^L_\tau$ und es gelte

T	ist wohlnummeriert und
$T_{l_1}, \ldots, T_{l_n}$	sind wohlnummeriert und
$h \in \mathrm{H}^L_{\mathrm{fin}}$	d. h. h hat endliche Domain und
$T_{l_1}, \ldots, T_{l_n}, h$	erfüllen die Invarianten (I3), (I4), (I5), (I6), (I7), (I8), (I9)

und $\beta(T, \gamma)$. Dann gilt:

1. Für alle $\gamma' \in \mathrm{Conf}^L$ mit $\gamma \xrightarrow{\bar{a}}_{\mathrm{L}} \gamma'$
   gibt es $T' \in \mathrm{Term}^{L(L)}$ mit $T \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T'$ und $\beta(T', \gamma')$.
2. Falls $\gamma \xrightarrow{\bar{a}}_{\mathrm{L}} \gamma'$ mit $\gamma' \in \mathrm{TConf}^L$, dann $T \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} \mathsf{Empty}$.

Hierbei ist definiert $T \overset{a}{\Rightarrow}_{\mathrm{G}} T' :\Leftrightarrow T \xrightarrow{\tau^* a}_{\mathrm{G}} T'$ und $T \overset{\tau}{\Rightarrow}_{\mathrm{G}} T' :\Leftrightarrow T \xrightarrow{\tau^*}_{\mathrm{G}} T'$.

Beweis. Ad 1. Der Beweis erfolgt durch Ableitungsinduktion für das Verträglichkeitsjudgement $\Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T), T_{l_j})$ zwischen dem globalen Term T und dem lokalen Term T_{l_j} derjenigen Lifeline j, welche für den Schritt $\gamma \xrightarrow{\bar{a}}_{\mathrm{L}} \gamma'$ aktiv ist. Sei $j \in \{1, \ldots, n\}$ und $h \in \mathrm{H}^L_{\mathrm{fin}}$ beliebig. Es bezeichne $\mathrm{D}_{\mathrm{R}^h_{l_j}}$ die Menge aller $\mathrm{R}^h_{l_j}$-Ableitungsbäume und $\prec$ die echte Teilbaumrelation auf $\mathrm{D}_{\mathrm{R}^h_{l_j}}$. Für alle $d \in \mathrm{D}_{\mathrm{R}^h_{l_j}}$ setzen wir

$$
\begin{aligned}
A(d) \ :\Longleftrightarrow\ & \forall\, T \in \mathrm{Term}^L,\ (T_{l_1}, \ldots, T_{l_n}) \in \mathrm{Term}^{l_1} \times \ldots \times \mathrm{Term}^{l_n} \\
& \forall\, T'_{l_j} \in \mathrm{Term}^{l_j}, h' \in \mathrm{H}^L_{\mathrm{fin}},\ \bar{a} \in \mathbb{A}^{l_j}_{\tau} .\big(\\
& \quad \big[\ T \text{ ist wohlnummeriert und} \\
& \quad\ \ T_{l_1}, \ldots, T_{l_n} \text{ sind wohlnummeriert und} \\
& \quad\ \ T_{l_1}, \ldots, T_{l_n}, h \text{ erfüllen (I3), (I4), (I5), (I6), (I7), (I8), (I9) und} \\
& \quad\ \ \forall\, i \in \{1, \ldots, n\}.\ \Vdash_{\mathrm{R}^h_{l_i}} (\pi_{l_i}(T), T_{l_i}) \text{ und} \\
& \quad\ \ d \Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T), T_{l_j}) \text{ und } l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'\ \big] \implies \\
& \quad \exists\, T' \in \mathrm{Term}^L .\big[\ T \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T' \text{ und} \\
& \qquad\qquad \forall\, i \in \{1, \ldots, n\} \setminus \{j\} .\ \Vdash_{\mathrm{R}^{h'}_{l_i}} (\pi_{l_i}(T'), T_{l_i}) \\
& \qquad\qquad \Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T'), T'_{l_j})\big]\big)
\end{aligned}
$$

Sei $d \in \mathrm{D}_{\mathrm{R}^h_{l_j}}$ und gelte $\forall\, \tilde{d} \prec d \,.\, A(\tilde{d})$. Seien T, $T_{l_1}, \ldots, T_{l_n}$, T'_{l_j}, $\bar{a}$, h' wie gefordert. Sei $l \in L \setminus \{l_j\}$. Es gilt $\Vdash_{\mathrm{R}^h_{l}} (\pi_l(T), T_l)$ und T_l, h erfüllen die Invariante (I6). Nach Lemma 4.18 gibt es $\bar{T}_l \in \mathrm{Term}^l$ mit $l : T_l, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : \bar{T}_l, h$ und $\Vdash_{\underline{\mathrm{R}}^h_{l}} (\pi_l(T), \bar{T}_l)$.

Wir führen eine Fallunterscheidung nach der letzten zum induktiven Aufbau von d angewandten Regel aus:

[None] Nicht möglich, weil $l_j : \mathsf{None}(a), h \overset{\bar{a}}{\not\to}_{\mathrm{L}} l : T'_{l_j}, h'$.

[Empty] Nicht möglich, weil $l_j : \mathsf{Empty}, h \overset{\bar{a}}{\not\to}_{\mathrm{L}} l : T'_{l_j}, h'$.

[Action] Dann ist $\pi_{l_j}(T) = a$ und $T_{l_j} = a$ und $l_j \in \alpha(a)$. Zur Ableitung von $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ kommt nur (basic$_{\mathrm{L}}$) in Frage. Es ist $\bar{a} = a$ und $T'_{l_j} = \mathsf{Empty}$ und $h' = h$. Aus $\pi_{l_j}(T) = a$ folgt $T = a$. Wir setzen $T' := \mathsf{Empty}$. Mit Axiom (basic$_{\mathrm{G}}$) folgt $T \xrightarrow{a}_{\mathrm{G}} T'$.

Zu zeigen bleibt:
(i) $\Vdash_{\mathrm{R}^h_{l_i}} (\pi_{l_i}(T'), T_{l_i})$ für alle $i \in \{1, \ldots, n\} \setminus \{j\}$
(ii) $\Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T'), T'_{l_j})$

Mit $T' = \mathsf{Empty}$ und $T'_{l_j} = \mathsf{Empty}$ und [Empty] folgt (ii). Für jedes $i \in \{1, \ldots, n\}$ gilt $\Vdash_{\mathrm{R}^h_{l_i}} (\pi_{l_i}(T), T_{l_i})$ nach Voraussetzung. Falls $i \neq j$ ist, dann gilt $\pi_{l_i}(T) =$

$\pi_{l_i}(a) = \mathsf{Empty} = \pi_{l_i}(\mathsf{Empty}) = \pi_{l_i}(T')$. Somit gilt (i). Deutung: Das Nachbilden der echten Aktion a in der globalen Semantik wird von den restlichen Lifelines $i \neq j$ nicht bemerkt, weil sich die Projektionen des globalen Terms auf die restlichen Lifelines nicht ändern.

[seq^1] Dann ist $\pi_{l_j}(T) = \mathsf{seq}_p(S_1, S_2)$ [also $T = \mathsf{seq}_p(T_1, T_2)$ und $S_1 = \pi_{l_j}(T_1)$ und $S_2 = \pi_{l_j}(T_2)$] und $T_{l_j} = \mathsf{seq}_p(T^1_{l_j}, T^2_{l_j})$. Es gibt $d_1, d_2 \in \mathrm{D}_{\mathrm{R}^h_{l_j}}$ mit $d_1, d_2 \prec d$ und $d_1 \Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T_1), T^1_{l_j})$ und $d_2 \Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T_2), T^2_{l_j})$.

Sei $l \in L \setminus \{l_j\}$. Wegen $\pi_l(T) = \mathsf{seq}_p(\pi_l(T_1), \pi_l(T_2))$ kommen zur Ableitung von $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T), \bar{T}_l)$ nur die Regeln [seq^1] oder [seq^2] in Frage:

Fall [seq^1]: Dann sei $l \in L_{[\mathrm{seq}^1]}$. Es gilt $\bar{T}_l = \mathsf{seq}_p(\bar{T}^1_l, \bar{T}^2_l)$ und $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T_1), \bar{T}^1_l)$ und $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T_2), \bar{T}^2_l)$.

Fall [seq^2]: Dann sei $l \in L_{[\mathrm{seq}^2]}$. Es gilt $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T_1), \mathsf{Empty})$ und $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T_2), \bar{T}_l)$.

Wegen $T_{l_j} = \mathsf{seq}_p(T^1_{l_j}, T^2_{l_j})$ kommen zur Ableitung von $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ nur (seq$^1_{\mathrm{L}}$), (seq$^2_{\mathrm{L}}$) oder (seq$^3_{\mathrm{L}}$) in Frage:

Fall (seq$^2_{\mathrm{L}}$): Dann ist $T^1_{l_j} = \mathsf{Empty}$ und $T'_{l_j} = T^2_{l_j}$ und $\bar{a} = \tau$ und $h' = h$. Es genügt zu zeigen: $\Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T), T'_{l_j})$. Es gilt $\Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T_1), T^1_{l_j}) = (\pi_{l_j}(T_1), \mathsf{Empty})$ und $\Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T_2), T^2_{l_j}) = (\pi_{l_j}(T_2), T'_{l_j})$. Mit Regel [seq^2] folgt die Behauptung. Deutung: Ein lokaler τ-Schritt zur Beseitigung eines Konstruktes $\mathsf{seq}_p(\mathsf{Empty}, T_2)$ wird in der globalen Semantik trivial nachgebildet. Die globale Konfiguration bleibt unverändert.

Fall (seq$^1_{\mathrm{L}}$): Dann ist $T'_{l_j} = \mathsf{seq}_p(T^{1\prime}_{l_j}, T^2_{l_j})$ und $l_j : T^1_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T^{1\prime}_{l_j}, h'$. Unser Ziel ist die Anwendung der I. V. für d_1. O. B. d. A. sei $j = 1$ und $L_{[\mathrm{seq}^1]} = \{l_2, \ldots, l_k\}$ und $L_{[\mathrm{seq}^2]} = \{l_{k+1}, \ldots, l_n\}$ mit $1 \leq k \leq n$. Wir betrachten die Konfiguration

$$[l_j : T^1_{l_j}, l_2 : \bar{T}^1_{l_2}, \ldots, l_k : \bar{T}^1_{l_k}, l_{k+1} : \mathsf{Empty}, \ldots, l_n : \mathsf{Empty}, h].$$

Die Terme T_1 und $T^1_{l_j}$ sind wohlnummeriert. Die Terme $\bar{T}^1_l$ für $l \in L_{[\mathrm{seq}^1]}$ sind ebenfalls wohlnummeriert. Ferner ist Empty wohlnummeriert. Man überlege sich, dass $T^1_{l_j}, \bar{T}^1_{l_2}, \ldots, \bar{T}^1_{l_k}, \mathsf{Empty}, \ldots, \mathsf{Empty}, h$ die Invarianten (I3), (I4), (I5), (I6), (I7), (I8), (I9) erfüllen. Aus $A(d_1)$ folgt: $T_1 \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T'_1$ mit $T'_1 \in \mathrm{Term}^L$ und

(i) $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T'_1), T^{1\prime}_{l_j})$

(ii) $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T'_1), \bar{T}^1_l)$ für alle $l \in L_{[\mathrm{seq}^1]}$

(iii) $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T'_1), \mathsf{Empty})$ für alle $l \in L_{[\mathrm{seq}^2]}$.

Aus $T_1 \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T'_1$ und Regel (seq$^1_{\mathrm{G}}$) folgt: $T = \mathsf{seq}_p(T_1, T_2) \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} \mathsf{seq}_p(T'_1, T_2)$. Wir setzen $T' := \mathsf{seq}_p(T'_1, T_2)$. Es gilt $\mathrm{top}(\pi_{l_j}(T_1)) = \mathrm{top}(T_1) \in \{\perp, p.1.q_1\}$ mit $q_1 \in \mathrm{Path}$. Mit Lemma 4.27 folgt $h(q) = h'(q)$ für alle $q \in \mathrm{Path} \setminus \mathrm{PathSpace}(p.1)$. Mit Lemma 4.22 folgt:

(iv) $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T_2), T^2_{l_j})$

Mit Lemma 4.23 und Lemma 4.22 folgt:

(v) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_2), \bar{T}_l^2)$ für alle $l \in L_{[\mathrm{seq}^1]}$

(vi) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_2), \bar{T}_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$.

Aus (ii) und (v) und Regel [seq^1] folgt $\Vdash_{\mathrm{R}_l^{h'}} (\mathsf{seq}_p(\pi_l(T_1'), \pi_l(T_2)), \mathsf{seq}_p(\bar{T}_l^1, \bar{T}_l^2)) = (\pi_l(T'), \bar{T}_l)$ für alle $l \in L_{[\mathrm{seq}^1]}$. Mit $l : T_l, h \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : \bar{T}_l, h$ und Invariante (I4) folgt $l : T_l, h' \xrightarrow{\tau^*}_{\underline{\mathrm{L}}} l : \bar{T}_l, h'$. Aus Lemma 4.26 folgt $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'), T_l)$ für alle $l \in L_{[\mathrm{seq}^1]}$. Aus (iii) und (vi) und Regel [seq^2] folgt $\Vdash_{\mathrm{R}_l^{h'}} (\mathsf{seq}_p(\pi_l(T_1'), \pi_l(T_2)), \bar{T}_l) = (\pi_l(T'), \bar{T}_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$. Mit Lemma 4.26 folgt $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'), T_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$. Aus (i) und (iv) und Regel [seq^1] folgt $\Vdash_{\mathrm{R}_{l_j}^{h'}} (\mathsf{seq}_p(\pi_{l_j}(T_1'), \pi_{l_j}(T_2)), \mathsf{seq}_p(T_{l_j}^{1\prime}, T_{l_j}^2)) = (\pi_{l_j}(T'), T_{l_j}')$.

Fall (seq$^3_{\mathrm{L}}$): Dann ist $T_{l_j}' = \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar{a})}(T_{l_j}^1), T_{l_j}^{2\prime})$ und $l_j : T_{l_j}^2, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T_{l_j}^{2\prime}, h'$. Unser Ziel ist die Anwendung der I. V. für d_2. Wir betrachten die Konfiguration

$$[l_j : T_{l_j}^2, l_2 : \bar{T}_{l_2}^2, \ldots, l_k : \bar{T}_{l_k}^2, l_{k+1} : \bar{T}_{l_{k+1}}, \ldots, l_n : \bar{T}_{l_n}, h].$$

Aus $A(d_2)$ folgt: $T_2 \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T_2'$ mit $T_2' \in \mathrm{Term}^L$ und

(i) $\Vdash_{\mathrm{R}_{l_j}^{h'}} (\pi_{l_j}(T_2'), T_{l_j}^{2\prime})$

(ii) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_2'), \bar{T}_l^2)$ für alle $l \in L_{[\mathrm{seq}^1]}$

(iii) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_2'), \bar{T}_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$.

Aus $T_2 \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T_2'$ und Regel (seq$^3_{\mathrm{G}}$) folgt: $T = \mathsf{seq}_p(T_1, T_2) \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar{a})}(T_1), T_2')$. Wir setzen $T' := \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar{a})}(T_1), T_2')$. Analog zur Argumentation in Fall (seq$^1_{\mathrm{L}}$) erhält man:

(iv) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_1), \bar{T}_l^1)$ für alle $l \in L_{[\mathrm{seq}^1]}$

(v) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T_1), \mathsf{Empty})$ für alle $l \in L_{[\mathrm{seq}^2]}$.

(vi) $\Vdash_{\mathrm{R}_{l_j}^{h'}} (\pi_{l_j}(T_1), T_{l_j}^1)$

Mit (iv) und $l \notin \alpha(\bar{a})$ und (ii) und Regel [seq^1] folgt $\Vdash_{\mathrm{R}_l^{h'}} (\mathsf{seq}_p(\pi_l(\mathrm{R}^*_{\alpha(\bar{a})}(T_1)), \pi_l(T_2')), \mathsf{seq}_p(\bar{T}_l^1, \bar{T}_l^2)) = (\pi_l(T'), \bar{T}_l)$ für alle $l \in L_{[\mathrm{seq}^1]}$. Mit Lemma 4.26 folgt $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'), T_l)$ für alle $l \in L_{[\mathrm{seq}^1]}$. Aus (v) und $l \notin \alpha(\bar{a})$ und (iii) und Regel [seq^2] folgt $\Vdash_{\mathrm{R}_l^{h'}} (\mathsf{seq}_p(\pi_l(\mathrm{R}^*_{\alpha(\bar{a})}(T_1)), \pi_l(T_2')), \bar{T}_l) = (\pi_l(T'), \bar{T}_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$. Mit Lemma 4.26 folgt $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'), T_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$. Mit (vi) und Lemma 4.19 und Lemma 4.2 und (i) und Regel [seq^1] erhält man $\Vdash_{\mathrm{R}_{l_j}^{h'}} (\mathsf{seq}_p(\pi_{l_j}(\mathrm{R}^*_{\alpha(\bar{a})}(T_1)), \pi_{l_j}(T_2')), \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar{a})}(T_{l_j}^1), T_{l_j}^{2\prime})) = (\pi_{l_j}(T'), T_{l_j}')$.

[seq^2] Dann ist $\pi_{l_j}(T) = \mathsf{seq}_p(S_1, S_2)$ [also $T = \mathsf{seq}_p(T_1, T_2)$ und $S_1 = \pi_{l_j}(T_1)$ und $S_2 = \pi_{l_j}(T_2)$] und es gibt $d_1, d_2 \in \mathrm{D}_{\mathrm{R}_{l_j}^h}$ mit $d_1, d_2 \prec d$ und $d_1 \Vdash_{\mathrm{R}_{l_j}^h} (\pi_{l_j}(T_1), \mathsf{Empty})$ und $d_2 \Vdash_{\mathrm{R}_{l_j}^h} (\pi_{l_j}(T_2), T_{l_j})$. Wir setzen $L_{[\mathrm{seq}^1]}$ und $L_{[\mathrm{seq}^2]}$ wir im Fall [seq^1]. Unser Ziel ist die Anwendung der I. V. für d_2. Wir betrachten die Konfiguration

$$[l_j : T_{l_j}, l_2 : \bar{T}_{l_2}^2, \ldots, l_k : \bar{T}_{l_k}^2, l_{k+1} : \bar{T}_{l_{k+1}}, \ldots, l_n : \bar{T}_{l_n}, h].$$

Aus $A(d_2)$ folgt: $T_2 \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T_2'$ mit $T_2' \in \mathrm{Term}^L$ und

(i) $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T'_2), T'_{l_j})$

(ii) $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_{l}(T'_2), \bar{T}^2_l)$ für alle $l \in L_{[\mathrm{seq}^1]}$

(iii) $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_{l}(T'_2), \bar{T}_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$.

Aus $T_2 \stackrel{\bar{a}}{\Rightarrow}_{\mathrm{G}} T'_2$ und Regel ($\mathrm{seq}^3_{\mathrm{G}}$) folgt: $T = \mathsf{seq}_p(T_1, T_2) \stackrel{\bar{a}}{\Rightarrow}_{\mathrm{G}} \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar{a})}(T_1), T'_2)$. Wir setzen $T' := \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar{a})}(T_1), T'_2)$. Analog zur Argumentation in Fall ($\mathrm{seq}^1_{\mathrm{L}}$) erhält man:

(iv) $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_{l}(T_1), \bar{T}^1_l)$ für alle $l \in L_{[\mathrm{seq}^1]}$

(v) $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_{l}(T_1), \mathsf{Empty})$ für alle $l \in L_{[\mathrm{seq}^2]}$.

(vi) $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T_1), \mathsf{Empty})$

Aus (iv) und (ii) folgt wie in Fall [seq^1] Unterfall ($\mathrm{seq}^3_{\mathrm{L}}$) dass gilt $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_l(T'), T_l)$ für alle $l \in L_{[\mathrm{seq}^1]}$. Aus (v) und (iii) folgt wie in Fall [seq^1] Unterfall ($\mathrm{seq}^3_{\mathrm{L}}$) dass gilt $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_l(T'), T_l)$ für alle $l \in L_{[\mathrm{seq}^2]}$. Mit (vi) und Lemma 4.19 und Lemma 4.2 und (i) und Regel [seq^2] folgt: $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\mathsf{seq}_p(\pi_{l_j}(\mathrm{R}^*_{\alpha(\bar{a})}(T_1)), \pi_{l_j}(T'_2)), T'_{l_j}) = (\pi_{l_j}(T'), T'_{l_j})$.

[alt^1] Dann ist $T_{l_j} = \mathsf{alt}_p(T^1_{l_j}, T^2_{l_j})$ und $h(p) = \{1\}$ und $\Vdash_{\mathrm{R}^{h}_{l_j}} (\pi_{l_j}(T), T^1_{l_j})$. Zur Ableitung von $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ kommt nur das Axiom ($\mathrm{alt}^1_{\mathrm{L}}$) in Frage. Es folgt: $T'_{l_j} = T^1_{l_j}$ und $\bar{a} = \tau$ und $h' = h$. Somit gilt $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T), T'_{l_j})$ und mehr ist nicht zu zeigen. Deutung: Ein τ-Nachführschritt wird in der globalen Semantik trivial nachgebildet. Die globale Konfiguration bleibt unverändert.

[alt^2] Analog zu Fall [alt^1].

[alt^3] Dann ist $\pi_{l_j}(T) = \mathsf{alt}_p(T^1_{l_j}, T^2_{l_j}) = T_{l_j}$ [also $T = \mathsf{alt}_p(T_1, T_2)$ und $T^1_{l_j} = \pi_{l_j}(T_1)$ und $T^2_{l_j} = \pi_{l_j}(T_2)$] und $h(p) \notin \{\{1\}, \{2\}\}$ und $T^1_{l_j}, T^2_{l_j} \in \mathrm{Term}^{l_j}$. Mit Invariante (I9) folgt $h(p) = \emptyset$. Mit Invariante (I8) folgt $h(p.r) = \emptyset$ für alle $r \in \mathrm{Path}$.

Sei $l \in L \setminus \{l_j\}$. Wegen $\pi_l(T) = \mathsf{alt}_p(\pi_l(T_1), \pi_l(T_2))$ kommt zur Ableitung von $\Vdash_{\underline{\mathrm{R}}^{h}_{l}} (\pi_l(T), \bar{T}_l)$ nur das Axiom [alt^3] in Frage. Es folgt: $\bar{T}_l = \mathsf{alt}_p(\pi_l(T_1), \pi_l(T_2))$.

Weil $T_{l_j} = \mathsf{alt}_p(T^1_{l_j}, T^2_{l_j})$ und $h(p) = \emptyset$ ist, kann $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ entweder mit ($\mathrm{alt}^1_{\mathrm{L}}$) oder mit ($\mathrm{alt}^2_{\mathrm{L}}$) abgeleitet worden sein.

Fall ($\mathrm{alt}^1_{\mathrm{L}}$): Dann ist $T'_{l_j} = T^1_{l_j}$ und $h' = h[p \mapsto \{1\}]$ und $\bar{a} = \tau$. Mit Axiom ($\mathrm{alt}^1_{\mathrm{G}}$) folgt $T = \mathsf{alt}_p(T_1, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_1$. Wir setzen $T' := T_1$. Weil $T^1_{l_j}$ wohlnummeriert ist und $\mathrm{top}(T^1_{l_j}) \in \{\bot, p.1.q_1\}$ mit $q_1 \in \mathrm{Path}$ gilt und weil für alle $r \in \mathrm{Path}$ gilt $h'(p.1.q_1.r) = h(p.1.q_1.r) = \emptyset$, folgt mit Lemma 4.20 dass gilt $\Vdash_{\mathrm{R}^{h'}_{l_j}} (T^1_{l_j}, T^1_{l_j}) = (\pi_{l_j}(T_1), T'_{l_j}) = (\pi_{l_j}(T'), T'_{l_j})$. Mit der Wohlnummeriertheit von $\pi_l(T)$ und einer Argumentation wie eben folgt: $\Vdash_{\mathrm{R}^{h'}_{l}} (\pi_l(T_1), \pi_l(T_1))$ für alle $l \in L \setminus \{l_j\}$. Mit $h'(p) = \{1\}$ und Regel [alt^1] folgt: $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_l(T_1), \mathsf{alt}_p(\pi_l(T_1), \pi_l(T_2))) = (\pi_l(T'), \bar{T}_l)$. Mit Lemma 4.26 folgt das Gewünschte. Deutung: Ein lokaler τ-Entscheidungsschritt wird in der globalen Semantik durch den entsprechenden globalen Entscheidungsschritt nachgebildet.

Fall ($\mathrm{alt}^2_{\mathrm{L}}$): Analog zu Fall ($\mathrm{alt}^1_{\mathrm{L}}$) mit ($\mathrm{alt}^2_{\mathrm{G}}$) und [$\mathrm{alt}^2$].

[loop^1] Dann ist $T_{l_j} = \mathsf{loop}_p(T^1_{l_j})$ und $h(p) = \{1\}$ und $\Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T), \mathsf{Empty})$. Zur Ableitung von $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ kommt nur das Axiom ($\mathrm{loop}^1_{\mathrm{L}}$) in Frage. Es folgt: $T'_{l_j} = \mathsf{Empty}$ und $\bar{a} = \tau$ und $h' = h$. Somit gilt $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T), T'_{l_j})$ und mehr ist nicht zu zeigen. Ein τ-Nachführschritt wird in der globalen Semantik trivial nachgebildet. Die globale Konfiguration bleibt unverändert.

[loop^2] Dann ist $\pi_{l_j}(T) = \mathsf{seq}_p(S_1, S_2)$ [also $T = \mathsf{seq}_p(T_1, T_2)$ und $S_1 = \pi_{l_j}(T_1)$ und $S_2 = \pi_{l_j}(T_2)$] und $T_{l_j} = \mathsf{loop}_p(T^1_{l_j})$ und $h(p) = \{2\}$ und

(1) $\Vdash_{\mathrm{R}^h_{l_j}} (S_1, T^1_{l_j})$

(2) $\Vdash_{\mathrm{R}^h_{l_j}} (S_2, \mathsf{loop}_{p.2}({}_{p.2}T^1_{l_j}))$.

Zur Ableitung von $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ kommt nur das Axiom ($\mathrm{loop}^2_{\mathrm{L}}$) in Frage. Es folgt: $T'_{l_j} = \mathsf{seq}_p(T^1_{l_j}, \mathsf{loop}_{p.2}({}_{p.2}T^1_{l_j}))$ und $\bar{a} = \tau$ und $h' = h$. Mit (1) und (2) und Regel [seq^1] folgt $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\mathsf{seq}_p(S_1, S_2), \mathsf{seq}_p(T^1_{l_j}, \mathsf{loop}_{p.2}({}_{p.2}T^1_{l_j}))) = (\pi_{l_j}(T), T'_{l_j})$ und mehr ist nicht zu zeigen. Deutung: Identisch zu Fall [loop^1].

[loop^3] Dann ist $\pi_{l_j}(T) = \mathsf{loop}_p(T^1_{l_j}) = T_{l_j}$ [also $T = \mathsf{loop}_p(T_1)$ und $\pi_{l_j}(T_1) = T^1_{l_j}$ und $h(p) \notin \{\{1\}, \{2\}\}$ und $T^1_{l_j} \in \mathrm{Term}^{l_j}$. Mit Invariante (I9) folgt $h(p) = \emptyset$. Mit Invariante (I7) folgt $h(p.r) = \emptyset$ für alle $r \in \mathrm{Path}$.

Sei $l \in L \setminus \{l_j\}$. Weil $\pi_l(T) = \mathsf{loop}_p(\pi_l(T_1))$ ist, kommt zur Ableitung von $\Vdash_{\mathrm{R}^h_l} (\pi_l(T), \bar{T}_l)$ nur das Axiom [loop^3] in Frage. Es folgt: $\bar{T}_l = \mathsf{loop}_p(\pi_l(T_1))$.

Weil $T_{l_j} = \mathsf{loop}_p(T^1_{l_j})$ und $h(p) = \emptyset$ ist, kann $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ entweder mit ($\mathrm{loop}^1_{\mathrm{L}}$) oder mit ($\mathrm{loop}^2_{\mathrm{L}}$) abgeleitet worden sein.

Fall ($\mathrm{loop}^1_{\mathrm{L}}$): Dann ist $T'_{l_j} = \mathsf{Empty}$ und $h' = h[p \mapsto \{1\}]$ und $\bar{a} = \tau$. Mit Axiom ($\mathrm{loop}^1_{\mathrm{G}}$) folgt $T = \mathsf{loop}_p(T_1) \xrightarrow{\tau}_{\mathrm{G}} \mathsf{Empty}$. Wir setzen $T' := \mathsf{Empty}$. Mit Axiom [Empty] folgt $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(\mathsf{Empty}), T'_{l_j})$. Mit [Empty] und [loop^1] und $h'(p) = \{1\}$ folgt $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(\mathsf{Empty}), \bar{T}_l)$ für alle $l \in L \setminus \{l_j\}$.

Fall ($\mathrm{loop}^2_{\mathrm{L}}$): Dann ist $T'_{l_j} = \mathsf{seq}_p(T^1_{l_j}, \mathsf{loop}_{p.2}({}_{p.2}T^1_{l_j}))$ und $h' = h[p \mapsto \{2\}]$ und $\bar{a} = \tau$. Mit Axiom ($\mathrm{loop}^2_{\mathrm{G}}$) folgt $T = \mathsf{loop}_p(T_1) \xrightarrow{\tau}_{\mathrm{G}} \mathsf{seq}_p(T_1, \mathsf{loop}_{p.2}({}_{p.2}T_1))$. Wir setzen $T' := \mathsf{seq}_p(T_1, \mathsf{loop}_{p.2}({}_{p.2}T_1))$.

Es ist T_{l_j} wohlnummeriert. Somit ist $T^1_{l_j}$ wohlnum. und $\mathrm{top}(T^1_{l_j}) \in \{\bot, p.1.q_1\}$ mit $q_1 \in \mathrm{Path}$. Für alle $r \in \mathrm{Path}$ gilt $h'(p.1.q_1.r) = h(p.1.q_1.r) = \emptyset$. Mit Lemma 4.20 folgt (1) $\Vdash_{\mathrm{R}^{h'}_{l_j}} (T^1_{l_j}, T^1_{l_j}) = (\pi_{l_j}(T_1), T^1_{l_j})$. Andererseits gilt $h'(p.2) = h(p.2) = \emptyset$ und mit [loop^3] folgt (2) $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\mathsf{loop}_{p.2}({}_{p.2}\pi_{l_j}(T_1)), \mathsf{loop}_{p.2}({}_{p.2}T^1_{l_j}))$. Mit (1) und (2) und [seq^1] folgt: $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\mathsf{seq}_p(\pi_{l_j}(T_1), \mathsf{loop}_{p.2}({}_{p.2}\pi_{l_j}(T_1))), \mathsf{seq}_p(T^1_{l_j}, \mathsf{loop}_{p.2}({}_{p.2}T^1_{l_j}))) = (\pi_{l_j}(T'), T'_{l_j})$.

Mit der Wohlnummeriertheit von $\pi_l(T)$ und einer Argumentation wie eben folgt (1) $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T_1), \pi_l(T_1))$ für jedes $l \in L \setminus \{l_j\}$. Mit [loop^3] und einer Argumentation wie eben folgt (2) $\Vdash_{\mathrm{R}^{h'}_l} (\mathsf{loop}_{p.2}({}_{p.2}\pi_l(T_1)), \mathsf{loop}_{p.2}({}_{p.2}\pi_l(T_1)))$. Mit (1) und (2) und [loop^2] und $h'(p) = \{2\}$ folgt: $\Vdash_{\mathrm{R}_l^{h'}} (\mathsf{seq}_p(\pi_l(T_1), \mathsf{loop}_{p.2}(_{p.2}\pi_l(T_1))), \mathsf{loop}_p(\pi_l(T_1))) = (\pi_l(T'), \bar{T}_l)$. Mit Lemma 4.26 folgt das Gewünschte. Deutung: Ein lokaler τ-Entscheidungsschritt wird in der globalen Semantik durch den entsprechenden globalen Entscheidungsschritt nachgebildet.

[strict^1] Dann ist $T_{l_j} = \mathsf{strict}_p(T^1_{l_j}, T^2_{l_j})$ und $h(p) = L$ und $\Vdash_{\mathrm{R}_{l_j}^h} (\pi_{l_j}(T), T^2_{l_j})$. Weil T_{l_j}, h die Invariante (I6) erfüllen, ist $T^1_{l_j} = \mathsf{Empty}$. Weil $T_{l_j} \doteq \mathsf{strict}_p(\mathsf{Empty}, T^2_{l_j})$ und $h(p) = L$ ist und weil es für den Teilterm Empty keine ableitbaren Übergänge gibt, kommt zur Ableitung von $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ nur das Axiom (strict$^2_{\mathrm{L}}$) in Frage. Es folgt: $T'_{l_j} = T^2_{l_j}$ und $h' = h$ und $\bar{a} = \tau$. Wir wissen also, dass $\Vdash_{\mathrm{R}_{l_j}^{h'}} (\pi_{l_j}(T), T'_{l_j})$ und mehr ist nicht zu zeigen. Deutung: Ein τ-Nachführschritt wird in der globalen Semantik trivial nachgebildet. Die globale Konfiguration bleibt unverändert.

[strict^2] Dann ist $\pi_{l_j}(T) = \mathsf{strict}_p(S_1, T^2_{l_j})$ [also $T = \mathsf{strict}_p(T_1, T_2)$ und $\pi_{l_j}(T_1) = S_1$ und $\pi_{l_j}(T_2) = T^2_{l_j}$] und $T_{l_j} = \mathsf{strict}_p(T^1_{l_j}, T^2_{l_j})$ und $h(p) \neq L$ und es gibt $d_1 \in \mathrm{D}_{\mathrm{R}_{l_j}^h}$ mit $d_1 \prec d$ und $d_1 \Vdash_{\mathrm{R}_{l_j}^h} (\pi_{l_j}(T_1), T^1_{l_j})$. Weil T_{l_j}, h die Invariante (I5) erfüllen, gilt $h(p.2.r) = \emptyset$ für alle $r \in \mathrm{Path}$.

Sei $l \in L \backslash \{l_j\}$. Aufgrund von $\pi_l(T) = \mathsf{strict}_p(\pi_l(T_1), \pi_l(T_2))$ kommt zur Ableitung von $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T), \bar{T}_l)$ nur die Regel [strict^2] in Frage. Es gilt $\bar{T}_l = \mathsf{strict}_p(\bar{T}^1_l, \pi_l(T_2))$ und $\Vdash_{\underline{\mathrm{R}}_l^h} (\pi_l(T_1), \bar{T}^1_l)$.

Weil $T_{l_j} = \mathsf{strict}_p(T^1_{l_j}, T^2_{l_j})$ und $h(p) \neq L$ ist, kommt zur Ableitung des Judgements $l_j : T_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T'_{l_j}, h'$ nur (strict$^1_{\mathrm{L}}$) oder (strict$^3_{\mathrm{L}}$) in Frage:

Fall (strict$^1_{\mathrm{L}}$): Dann ist $T'_{l_j} = \mathsf{strict}_p(T^{1\prime}_{l_j}, T^2_{l_j})$ und $l_j : T^1_{l_j}, h \xrightarrow{\bar{a}}_{\mathrm{L}} l_j : T^{1\prime}_{l_j}, h'$. Mit T_{l_j} ist auch $T^1_{l_j}$ wohlnummeriert und es gilt $\mathrm{top}(T^1_{l_j}) \in \{\perp, p.1.q_1\}$ mit $q_1 \in \mathrm{Path}$. Mit Lemma 4.21 folgt $h'(q) = h(q)$ für alle $q \in \mathrm{Path} \setminus \mathrm{PathSpace}(p.1)$. Insbesondere gilt $h'(p) = h(p) \neq L$.

O. B. d. A. sei $j = 1$. Die Anwendung der I. V. für d_1 auf die Konfiguration $[l_j : T^1_{l_j}, l_2 : \bar{T}^1_{l_2}, \ldots, l_n : \bar{T}^1_{l_n}, h]$ liefert $T_1 \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T'_1$ mit $T'_1 \in \mathrm{Term}^L$ und

(i) $\Vdash_{\mathrm{R}_{l_j}^{h'}} (\pi_{l_j}(T'_1), T^{1\prime}_{l_j})$

(ii) $\Vdash_{\mathrm{R}_l^{h'}} (\pi_l(T'_1), \bar{T}^1_l)$ für alle $l \in L \setminus \{l_j\}$.

Mit $T_1 \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} T'_1$ und Regel (strict$^1_{\mathrm{G}}$) folgt: $T = \mathsf{strict}_p(T_1, T_2) \overset{\bar{a}}{\Rightarrow}_{\mathrm{G}} \mathsf{strict}_p(T'_1, T_2)$. Wir setzen $T' := \mathsf{strict}_p(T'_1, T_2)$. Mit (i) und Regel [strict^2] und $h'(p) \neq L$ folgt: $\Vdash_{\mathrm{R}_{l_j}^{h'}} (\mathsf{strict}_p(\pi_{l_j}(T'_1), \pi_{l_j}(T_2)), \mathsf{strict}_p(T^{1\prime}_{l_j}, \pi_{l_j}(T_2))) = (\pi_{l_j}(T'), \mathsf{strict}_p(T^{1\prime}_{l_j}, T^2_{l_j})) = (\pi_{l_j}(T'), T'_{l_j})$. Mit (ii) und Regel [strict^2] und $h'(p) \neq L$ folgt: $\Vdash_{\mathrm{R}_l^{h'}} (\mathsf{strict}_p(\pi_l(T'_1), \pi_l(T_2)), \mathsf{strict}_p(\bar{T}^1_l, \pi_l(T_2))) = (\pi_l(T'), \bar{T}_l)$ für alle $l \in L \setminus \{l_j\}$. Mit Lemma 4.26 folgt das Gewünschte.

Fall ($\text{strict}^3_{\text{L}}$): Dann ist $T_{l_j} = \mathsf{strict}_p(T^1_{l_j}, T^2_{l_j}) = T'_{l_j}$ und $T^1_{l_j} = \mathsf{Empty}$ und $\bar{a} = \tau$ und $h' = h[p \mapsto h(p) \cup \{l_j\}]$ und $l_j \notin h(p)$.

Fall 1: $h'(p) \neq L$: Mit $T^1_{l_j} = \mathsf{Empty}$ und Lemma 4.22.1 folgt $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T_1), T^1_{l_j})$. Mit Regel [strict^2] und $h'(p) \neq L$ folgt: $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\mathsf{strict}_p(\pi_{l_j}(T_1), \pi_{l_j}(T_2)), \mathsf{strict}_p(T^1_{l_j}, \pi_{l_j}(T_2))) = (\pi_{l_j}(T), T_{l_j}) = (\pi_{l_j}(T), T'_{l_j})$.

Für $l \in L \setminus \{l_j\}$ gilt $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T_1), \bar{T}^1_l)$. Mit Lemma 4.23 folgt $\mathrm{top}(\bar{T}^1_l) \in \{\perp, p.1.r\}$ mit $r \in \mathrm{Path}$. Es gilt $h'(q) = h(q)$ für jedes $q \in \mathrm{PathSpace}(p.1.r)$. Mit Lemma 4.22 folgt $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T), \bar{T}_l)$. Mit Lemma 4.26 folgt das Gewünschte.

Fall 2: $h'(p) = L$: Es gilt $T^1_{l_j} = \mathsf{Empty}$ und somit $\Vdash_{\mathrm{R}^h_{l_j}} (\pi_{l_j}(T_1), \mathsf{Empty})$. Aus Satz 4.14 folgt $\pi_{l_j}(T_1) \Rightarrow^* \mathsf{Empty}$. Mit Hilfe von Lemma 4.13.1 folgt $\pi_{l_j}(T_1) \in \mathrm{RTerm}[\mathsf{Empty}]$.

O. B. d. A. sei $j = 1$. Betrachte die Konfiguration $[l_j : T'_{l_j}, l_2 : \bar{T}_{l_2}, \ldots, l_n : \bar{T}_{l_n}, h']$. Diese Konfig. ist in endlich vielen τ-Schritten aus $[l_j : T_{l_j}, l_2 : T_{l_2}, \ldots, l_n : T_{l_n}, h]$ hervorgegangen und erfüllt daher ebenfalls die Invariante (I6). Weil $h'(p) = L$ ist, folgt $\bar{T}^1_{l_2} = \ldots = \bar{T}^1_{l_n} = \mathsf{Empty}$. Es gilt demnach $\Vdash_{\underline{\mathrm{R}}^h_l} (\pi_l(T_1), \mathsf{Empty})$ für alle $l \in L \setminus \{l_j\}$. Folglich gilt $\pi_l(T_1) \in \mathrm{RTerm}[\mathsf{Empty}]$ für alle $l \in L$.

Weil $\pi_l(T_1) \in \mathrm{RTerm}[\mathsf{Empty}]$ für *alle* Lifelines l gilt, kann T_1 als Blattterme nur Empty enthalten. Daraus folgt $T_1 \in \mathrm{RTerm}[\mathsf{Empty}]$. Es folgt $T_1 \xrightarrow{\tau^*}_{\mathrm{G}} \mathsf{Empty}$. Mit ($\text{strict}^1_{\text{G}}$) folgt $T = \mathsf{strict}_p(T_1, T_2) \xrightarrow{\tau^*}_{\mathrm{G}} \mathsf{strict}_p(\mathsf{Empty}, T_2)$. Wegen ($\text{strict}^2_{\text{G}}$) gilt $\mathsf{strict}_p(\mathsf{Empty}, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2$. Wir setzen $T' := T_2$. Es bleibt zu zeigen:

(i) $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T'), T'_{l_j})$

(ii) $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T'), \bar{T}_l)$ für alle $l \in L \setminus \{l_j\}$.

Mit der Wohlnummeriertheit von T folgt, dass T_2 wohlnummeriert ist und $\mathrm{top}(T_2) \in \{\perp, p.2.q_2\}$ mit $q_2 \in \mathrm{Path}$. Weil $h'(p.2.q_2.r) = h(p.2.q_2.r) = \emptyset$ für alle $r \in \mathrm{Path}$, folgt mit Lemma 4.20, dass gilt $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T_2), \pi_l(T_2))$ für alle $l \in L$.

Ad (i) Aus $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T_2), \pi_{l_j}(T_2))$ und [strict^1] und $h'(p) = L$ folgt: $\Vdash_{\mathrm{R}^{h'}_{l_j}} (\pi_{l_j}(T_2), \mathsf{strict}_p(\mathsf{Empty}, \pi_{l_j}(T_2))) = (\pi_{l_j}(T_2), \mathsf{strict}_p(\mathsf{Empty}, T^2_{l_j})) = (\pi_{l_j}(T'), T'_{l_j})$.

Ad (ii) Aus $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T_2), \pi_l(T_2))$ und [strict^1] und $h'(p) = L$ folgt: $\Vdash_{\mathrm{R}^{h'}_l} (\pi_l(T_2), \mathsf{strict}_p(\mathsf{Empty}, \pi_l(T_2))) = (\pi_l(T'), \bar{T}_l)$ für alle $l \in L \setminus \{l_j\}$.

Deutung: Ein lokaler τ-Synchronisationsschritt einer Lifeline l_j für ein Konstrukt $\mathsf{strict}_p(\mathsf{Empty}, T^2_{l_j})$ wird—solange l_j nicht die letzte Lifeline ist, die für dieses p den Synchronisationsschritt ausführt—in der globalen Semantik trivial nachgebildet (= Fall 1). Die globale Konfiguration bleibt unverändert. Erst wenn die letzte Lifeline den Synchronisationsschritt für p ausführt (= Fall 2), bilden wir diesen durch einen globalen Synchronisationsschritt nach. Die hierzu erforderliche Regel ($\text{strict}^2_{\text{G}}$) setzt allerdings voraus, dass der linke Teilterm des globalen Konfigurationsterms gleich Empty ist. Aufgrund der Tatsache, dass wir lokale Aufräumschritte in der globalen Semantik trivial nachbilden (müssen), steht im linken Teilterm des globalen Konfigurationsterms i. A. nicht Empty sondern irgendein zu Empty reduzierbarer "Müll", den wir zunächst durch globale τ-Aufräumschritte beseitigen müssen, bevor wir den globalen τ-Synchronisationsschritt ausführen können.

Ad 2. Dieser Beweisteil wird aus Zeitgründen nicht ausgeführt. Methodische Schwierigkeiten sind allerdings keine zu erwarten. □

4.10 Die Regel ($\mathrm{seq}^3_{\mathrm{L}}$) unter der Lupe

Aus Zeitgründen sei hier nur angemerkt, dass sich das Unmöglichkeitsresultat aus Abschnitt 3.5.1 problemlos auf die verteilte operationale Semantik übertragen läßt. Das Problem ist genau das gleiche.

Kapitel 5

Kanalidentifikatoren

Die Semantiken, die in den Kapiteln 2, 3 und 4 angegeben wurden, interpretierten Sende- und Empfangsaktionen in nahezu identischer Weise. Ein Unterschied bestand nur insoweit, als für Sendeaktionen $\mathsf{snd}(s, r, m)$ die Instanz s (= Senderinstanz) und für Empfangsaktionen $\mathsf{rcv}(s, r, m)$ die Instanz r (= Empfängerinstanz) aktiv war. Die in Sendeaktionen enthaltenen Empfängerangaben und die in Empfängeraktionen enthaltenen Senderangaben wurden vollständig ignoriert. Die Modellierung der zeitlichen Abhängigkeit korrespondierender Sende- und Empfangsaktionen geschah explizit durch Verwendung von strict-Konstrukten. Diese zweckentfremdende, exzessive Verwendung von strict-Konstrukten hatte in der lokalen Semantik den Nachteil, dass ständig alle Lifelines synchronisiert werden mussten, was eine hohe Anzahl von Lese- und Schreibzugriffen auf das Shared Memory h zur Folge hatte. Aus diesen Gründen wird in Kapitel 5 schrittweise eine lokale Semantik entwickelt, die den asynchronen Nachrichtenversand mittels gerichteter Kanäle nachbildet und somit einen impliziten Mechanismus zur Erzwingung der korrekten zeitlichen Reihenfolge korrespondierender Sende- und Empfangsaktionen anbietet. Das geordnete Paar (s, r) aus einer Sender- und einer Empfängerinstanz wird als Kanalidentifikator interpretiert. Um dieses Kanalkonzept besser nutzen zu können und um eine natürlichere Modellierung der Interaktionen zu ermöglichen, wird die Termsprache der positiven UML 2.0-Interaktionen überdies um allgemeine, durch Pomsets beschriebene Basic Interactions B erweitert.

5.1 Allgemeine elementare Interaktionen

Die abstrakte Syntax der um allgemeine elementare Interaktionen erweiterten UML 2.0-Interaktionssprache ist in Tafel 5.1 angegeben. Dabei ist B Metavariable mit Bereich Basic $:= \mathbb{P}_{\text{fin}} := \{p \in \mathbb{D} \,|\, p \text{ ist lokal linear und endlich}\}$. Als

$$
\begin{array}{rcl}
I \in \mathrm{IFrag} & ::= & B \\
& | & \mathsf{strict}(I_1, I_2) \\
& | & \mathsf{seq}(I_1, I_2) \\
& | & \mathsf{par}(I_1, I_2) \\
& | & \mathsf{loop}(I) \\
& | & \mathsf{alt}(I_1, I_2)
\end{array}
$$

Tafel 5.1: Erweiterte abstrakte Syntax der Interaktionen

syntaktischer Zucker sei definiert $\mathsf{Empty} := \varepsilon \in \mathrm{Basic}$. Anmerkung: Wir erlauben uns in diesem Kapitel Symbole wie z. B. IFrag neu zu definieren.

5.2 Denotationelle und globale operationale Semantik

Für die erweiterte Interaktionssprache IFrag aus Tafel 5.1 gelte die denotationelle Semantik aus Abschnitt 2.4, wobei die definierenden Klauseln für Empty und a durch die Klausel $\mathcal{D}[\![B]\!] := B{\downarrow}$ zu ersetzen sind. Offensichtlich gilt $\mathcal{D}[\![\mathsf{Empty}]\!] = \mathcal{D}[\![\varepsilon]\!] = \varepsilon{\downarrow} = \{\varepsilon\}$ und $\mathcal{D}[\![a]\!] = a{\downarrow} = \{a\}$ für jedes $a \in \mathbb{A}$.

Die Konfigurationen und die Transitionen der (unnummerierten) globalen operationalen Semantik seien genauso wie in Abschnitt 3.1 definiert. Dabei ist das redefinierte Symbol IFrag im Sinne von Tafel 5.1 zu verstehen. Die Restriktionsfunktion R_L werde wie in Abschnitt 3.2 auf S. 18 definiert, wobei die definierenden Klauseln für Empty und a durch folgende Klausel zu ersetzen sind:

$$
\mathrm{R}_L(B) := \begin{cases} B & \text{falls } \alpha(B) \cap L = \emptyset \\ \mathsf{None} & \text{andernfalls} \end{cases}
$$

Lemma 3.1 bleibt gültig. Die erforderlichen Änderungen am Beweis sind trivial.

Sei $p \in \mathbb{P}$ und $(E, \leq, \lambda)$ ein Repräsentant von p. Die Menge aller Minima von E bezüglich $\leq$ sei als $\mathrm{Min}_{\leq}(E)$ notiert. Wir definieren $\mathrm{Min}(p) := \lambda(\mathrm{Min}_{\leq}(E)) \subseteq \mathbb{A}$. Für jedes $a \in \mathrm{Min}(p)$ existiert ein $e_0 \in \mathrm{Min}_{\leq}(E)$ mit $\lambda(e_0) = a$. Aufgrund der lokalen Linearität von p ist dieses e_0 eindeutig bestimmt (siehe Satz 2.1). Wir definieren $p \setminus \{a\} := [(E', \leq', \lambda')]$ mit $E' := E \setminus \{e_0\}$ und $\leq' := \leq \cap (E' \times E')$ und $\lambda' := \lambda \upharpoonright E'$.

Lemma 5.1 Sei $p, p' \in \mathbb{P}$, $a \in \mathrm{Min}(p)$ und gelte $p' = p \setminus \{a\}$. Dann gilt:

$$\forall\, t' \in \mathbb{T} \,.\, (\, t' \in p'{\downarrow} \iff a\,; t' \in p{\downarrow}\,)$$

Beweis. "$\Rightarrow$": Es sei $p = [(E, \leq, \lambda)] \in \mathbb{P}$ und $a \in \mathrm{Min}(p) = \lambda(\mathrm{Min}_{\leq}(E))$ und es gelte $p' = p \setminus \{a\}$. Es gibt ein eindeutig bestimmtes $e_0 \in \mathrm{Min}_{\leq}(E)$ mit $\lambda(e_0) = a$

(siehe Satz 2.1). Nach Definition gilt $p \setminus \{a\} = [(E', \leq', \lambda')]$ mit $E' := E \setminus \{e_0\}$ und $\leq' := \leq \cap (E' \times E')$ und $\lambda' := \lambda \restriction E'$. Sei $t \in p'{\downarrow}$ beliebig. Es gibt eine totale Ordnung $\leq'_{\mathrm{t}}$ auf E' mit $\leq' \subseteq \leq'_{\mathrm{t}}$ und $t = [(E', \leq'_{\mathrm{t}}, \lambda')]$. Es gilt $a\,; t' = [(E, \leq_{\mathrm{t}}, \lambda)]$ mit $\leq_{\mathrm{t}} := \{(e_0, e_0)\} \cup \leq'_{\mathrm{t}} \cup \{e_0\} \times E'$ und $\leq_{\mathrm{t}}$ ist eine totale Ordnung auf E. Zu zeigen ist $\leq \subseteq \leq_{\mathrm{t}}$. Weil $e_0 \in \mathrm{Min}_{\leq}(E)$ ist, gilt $\leq \cap (E' \times \{e_0\}) = \emptyset$. Daraus folgt: $\leq = (\leq \cap \{e_0\} \times \{e_0\}) \cup (\leq \cap \{e_0\} \times E') \cup (\leq \cap E' \times \{e_0\}) \cup (\leq \cap E' \times E') \subseteq \{(e_0, e_0)\} \cup \{e_0\} \times E' \cup \emptyset \cup \leq' \subseteq \{(e_0, e_0)\} \cup \leq'_{\mathrm{t}} \cup \{e_0\} \times E' = \leq_{\mathrm{t}}$.

“$\Leftarrow$”: Wir überlassen den Beweis der Gegenrichtung dem Leser. $\square$

Die (unnummerierte) globale operationale Semantik der erweiterten Interaktionssprache IFrag werde wie in Abschnitt 3.3 definiert, wobei in Tafel 3.1 das Regelschema (basic$_{\mathrm{Gu}}$) durch folgendes Regelschema zu ersetzen ist:

$$(\mathrm{basic}_{\mathrm{Gu}}) \quad B \xrightarrow{a}_{\mathrm{Gu}} B \setminus \{a\} \qquad \text{falls } a \in \mathrm{Min}(B)$$

Lemma 3.2, Satz 3.3, Lemma 3.4, Lemma 3.5 und Satz 3.6 bleiben gültig. In dem Beweis von Lemma 3.2 ist der Fall (basic$_{\mathrm{Gu}}$) in folgender Weise anzupassen:

(basic$_{\mathrm{Gu}}$) Dann gilt $I = B \in$ Basic und $a \in \mathrm{Min}(B)$ und $I' = B \setminus \{a\}$. Zu zeigen ist $\mathcal{D}[\![B \setminus \{a\}]\!] \subseteq \mathcal{D}[\![B]\!] \,/ a$. D. h. es ist zu zeigen $(B \setminus \{a\}){\downarrow} \subseteq B{\downarrow} \,/\, a$. Sei $t' \in (B \setminus \{a\}){\downarrow}$ beliebig. Aus Lemma 5.1 folgt $a\,; t' \in B{\downarrow}$.

In dem Beweis von Lemma 3.5 sind unter Ad. 1 die Fälle $I = \mathsf{Empty}$ und $I = a'$ durch folgenden untergliederten Fall $I = B$ zu ersetzen:

$I = B$: Wir unterscheiden zwei Fälle:

Fall 1: $a \in \mathrm{Min}(B)$: Aus Lemma 5.1 folgt $\mathcal{D}[\![B]\!] \,/ a = \mathcal{D}[\![B \setminus \{a\}]\!]$. Mit Axiom (basic$_{\mathrm{Gu}}$) folgt $I = B \xrightarrow{a}_{\mathrm{Gu}} B \setminus \{a\}$.

Fall 2: $a \notin \mathrm{Min}(B)$: Dann ist $\mathcal{D}[\![B]\!] \,/ a = B{\downarrow} \,/\, a = \emptyset$.

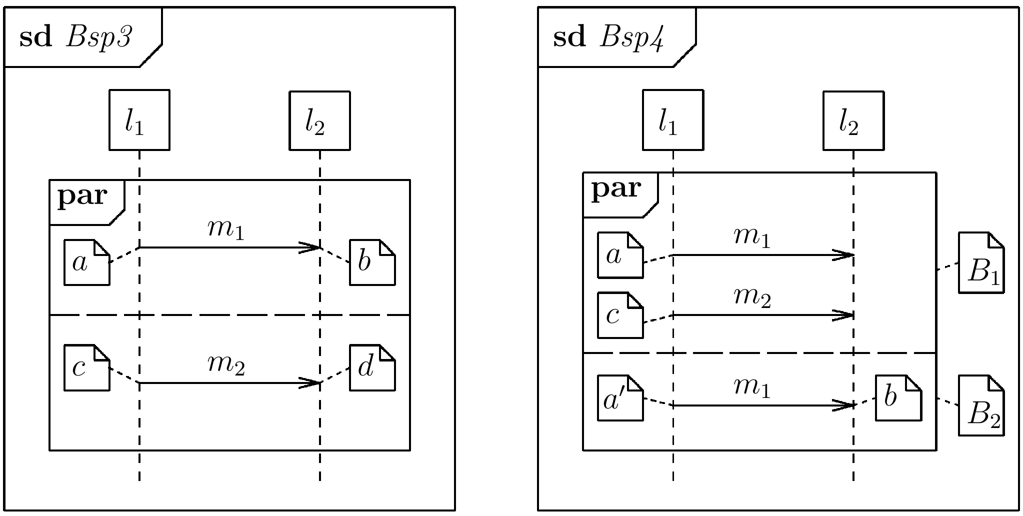
5.3 Globale operationale Semantik mit Kanälen

Als Zwischenschritt auf unserem Weg zu einer lokalen operationalen Semantik mit Kanälen wird in diesem Abschnitt eine *globale* operationale Semantik mit Kanälen eingeführt. Es wird dabei die Frage untersucht werden, wie die Kanäle zu modellieren sind und welche Wohlgeformtheitseigenschaft die Interaktionsterme haben müssen, damit korrespondierende Startterme in der globalen operationalen Semantik ohne Kanäle (siehe Abschnitt 5.2) und in der globalen operationalen Semantik mit Kanälen *bisimilar* sind.

5.3.1 Kanalmodelle

Grundsätzlich kommt eine Modellierung der Kanäle als Queues und als Pools in Frage. Komplexere Kanalmodelle wie z. B. Prioritätswarteschlangen werden nicht betrachtet. Aus Gründen der Einfachheit sei überdies eine unendliche Kanalkapazität vorausgesetzt.

Zwischen jedem Paar $(s, r) \in \mathbb{I} \times \mathbb{I}$ von Lifelines sei ein gerichteter Kanal $C(s, r)$ vorhanden, der eine beliebig hohe endliche Zahl von Nachrichten $m \in \mathbb{M}$ puffern kann. Angenommen diese Kanäle seien als Queues modelliert und wir führten das Sequenzdiagramm *Bsp3* aus (siehe Abb. unten links). Aufgrund der Reihenfolgeerhaltung der Nachrichten in der Queue $C(l_1, l_2)$ sind die Abläufe *acdb* und *cabd* nicht möglich, obgleich diese Traces zur Erfüllungsmenge $\mathcal{D}[\![Bsp3]\!]$ gehören.



Mithin würde eine Modellierung als Queues zu einer operationalen Semantik führen, die nicht äquivalent zur denotationellen Semantik der Interaktionsdiagramme ist. Als richtige Wahl eines Kanalmodells stellt sich der Nachrichtenpool heraus, der mathematisch als endliche Multimenge über $\mathbb{M}$ zu beschreiben ist[1].

Leider tritt bei der Kanalmodellierung eine weitere Schwierigkeit auf, die ihre Ursache darin findet, dass ein und dieselbe Nachricht zwischen denselben Lifelines in verschiedenen Basic Interactions versendet werden kann. Man betrachte hierzu das Sequenzdiagramm *Bsp4* in der Abbildung oben rechts. Die beiden Operanden des Combinded Fragments par seien die Basic Interactions B_1 und B_2. Nach Ausführung der Aktion a enthalte der Kanal $C(l_1, l_2)$ die Nachricht m_1. In dieser Konfiguration sei der Blick auf die Basic Interaction B_2 gerichtet. Die Version von B_2, welche der operationalen Semantik *mit* Kanälen vorgelegt wird, lautet $a' \parallel b$ und nicht $a' ; b$. Denn es ist ja gerade der Sinn der operationalen Semantik

[1]María Victoria Cengarle, mündliche Äußerung.

mit Kanälen die explizite Modellierung der Ordnungsbeziehungen zwischen korrespondierenden Sende- und Empfangsaktionen entbehrlich zu machen. Mithin hält die operationale Semantik mit Kanälen die Aktion b für ein Minimum der Basic Interaction B_2. Weil die Aktion b eine Empfangsaktion $\mathsf{rcv}(l_1, l_2, m_1)$ ist, muss für die Ausführbarkeit von b eine passende Nachricht m_1 im Kanal $C(l_1, l_2)$ vorhanden sein. Eine solche passende Nachricht *ist* vorhanden und folglich kann die Aktion b ausgeführt werden. Anschließend könnte es zur Ausführung der Aktion c kommen, so dass sich—bis zu diesem Zeitpunkt—der Teilablauf abc ergeben würde. Dieser Ablauf verletzt aber die denotationalle Semantik des Diagramms, weil der Ausführung der Teiltrace bc korrekterweise *zwei* Sendeaktionen $a = a' = \mathsf{snd}(l_1, l_2, m_1)$ vorangegangen sein müssen.

Um das Problem, das am Diagramm *Bsp4* beschrieben wurde, besser zu verstehen, deuten wir jede Basic Interaction B_i als einen verteilt realisierten Prozess und das zugehörige $i \in \{1, 2\}$ als einen Prozessidentifikator. Wenn man so weit geht, die Lifelines l_1 und l_2 als Quell- und Ziel-IP-Adressen zu interpretieren, so könnte der Prozessidentifikator i mutig als eine Portnummer gedeutet werden.

Es ist nun klar, wie das Kanalmodell für das Diagramm *Bsp4* zu verfeinern ist: Die Kanäle müssen zusätzlich nach dem Prozessidentifikator $i \in \{1, 2\}$ unterschieden werden. Im allgemeinen Fall benutzen wir (statt i) eindeutige Pfadannotierungen $p \in \text{Path}$, welche wir im nächsten Unterabschnitt bis auf die Blattebene der Interaktionsterme führen werden. Somit gelangen wir zu folgender Definition des Kanalzustandes:

$$C \in \text{Channels} := \mathbb{I} \times \mathbb{I} \times \text{Path} \times \mathbb{M} \longrightarrow \mathbb{N}_0$$

Aus technischen Gründen wurden die endlichen Multimengen über $\mathbb{M}$ als Abbildungen $\mathbb{M} \longrightarrow \mathbb{N}_0$ dargestellt. Der leere Kanalzustand $\emptyset \in \text{Channels}$ sei definiert durch $(s, r, p, m) \mapsto 0$.

5.3.2 Pfadannotierte elementare Interaktionen

Die abstrakte Syntax der pfadannotierten Interaktionsterme mit pfadannotierten elementaren Interaktionen ist in Tafel 5.2 angegeben. Dabei ist B Metavariable mit Bereich Basic und p Metavariable mit Bereich Path. Die Funktion top werde redefiniert wir folgt:

$$\text{top} : \text{Term} \longrightarrow \text{Path}$$

$$\begin{array}{lll} \text{top}(const_p) & := p & \text{für alle } const \in \{\mathsf{None}(B), B\} \\ \text{top}(uop_p(T)) & := p & \text{für alle } uop \in \{\mathsf{loop}\} \\ \text{top}(bop_p(T_1, T_2)) & := p & \text{für alle } bop \in \{\mathsf{strict}, \mathsf{seq}, \mathsf{par}, \mathsf{alt}\} \end{array}$$

Weil die redefinierte Fassung von top nicht über den Funktionswert $\perp$ verfügt, sei als Ersatz ein unäres Prädikat isConst $\subseteq$ Term eingeführt, das definiert sei wie

$$
\begin{array}{rcl}
S, T \in \text{Term} & ::= & \mathsf{None}(B)_p \\
& | & B_p \\
& | & \mathsf{strict}_p(T_1, T_2) \\
& | & \mathsf{seq}_p(T_1, T_2) \\
& | & \mathsf{par}_p(T_1, T_2) \\
& | & \mathsf{loop}_p(T) \\
& | & \mathsf{alt}_p(T_1, T_2)
\end{array}
$$

Tafel 5.2: Abstrakte Syntax der erweiterten pfadannotierten Interaktionsterme

folgt: $\text{isConst}(T) :\Longleftrightarrow \exists B \in \text{Basic}, p \in \text{Path}\,.\,(T = B_p \vee T = \mathsf{None}(B)_p)$. Die induktive Definition der Wohlnummeriertheit von Interaktionstermen $T \in \text{Term}$ ist wie folgt anzupassen:

1. $const_p$ ist wohlnummeriert.
2. $uop_p(T)$ ist wohlnummeriert, falls T wohlnummeriert ist und ein Pfad q existiert, so dass gilt $\text{top}(T) = p.1.q$.
3. $bop_p(T_1, T_2)$ ist wohlnummeriert, falls T_1 und T_2 wohlnummeriert sind und Pfade q_1, q_2 existieren, so dass gilt $\text{top}(T_1) = p.1.q_1$ und $\text{top}(T_2) = p.2.q_2$.

Innerhalb der Definitionen des Nummerierungsoperators num und des Umnummerierungsoperators ren sind die Basisfälle wie folgt anzupassen:

$$\text{num}(B, p) := B_p$$

$$\text{ren}(p, n, const_q) := \begin{cases} const_{p.n.q'} & \text{falls } q = p.q' \\ const_q & \text{andernfalls} \end{cases}$$

Das Lemma 4.1 bleibt gültig. Die Basisfälle der Definition der Restriktionsfunktion R^*_L (siehe S. 37) müssen ebenfalls angepasst werden:

$$
\begin{array}{lcl}
\mathrm{R}^*_L(\mathsf{None}(B)_p) & := & \mathsf{None}(B)_p \\
\mathrm{R}^*_L(B_p) & := & \begin{cases} B_p & \text{falls } \alpha(B) \cap L = \emptyset \\ \mathsf{None}(B)_p & \text{andernfalls} \end{cases}
\end{array}
$$

5.3.3 Globale Semantiken ohne Kanäle und mit Kanälen

In diesem Unterabschnitt werden zwei pfadannotierte Fassungen der globalen operationalen Semantik—eine ohne Kanäle und eine mit Kanälen—spezifiziert und gegenübergestellt. In den Unterabschnitten 5.3.4 und 5.3.5 wird untersucht werden, unter welchen Voraussetzungen korrespondierende Startterme in diesen beiden Semantiken (stark) bisimilar sind. In Abschnitt 5.4 wird schließlich die

$$
\begin{array}{llll}
(\mathrm{basic_G}) & B_p \xrightarrow{a}_{\mathrm{G}} (B \setminus \{a\})_p \quad \text{falls } a \in \mathrm{Min}(B) & & \\[1ex]
(\mathrm{strict}^1_{\mathrm{G}}) & \dfrac{T_1 \xrightarrow{\bar a}_{\mathrm{G}} T_1'}{\mathsf{strict}_p(T_1, T_2) \xrightarrow{\bar a}_{\mathrm{G}} \mathsf{strict}_p(T_1', T_2)} & (\mathrm{strict}^2_{\mathrm{G}}) & \mathsf{strict}_p(\mathsf{Empty}_q, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2 \\[2ex]
(\mathrm{seq}^1_{\mathrm{G}}) & \dfrac{T_1 \xrightarrow{\bar a}_{\mathrm{G}} T_1'}{\mathsf{seq}_p(T_1, T_2) \xrightarrow{\bar a}_{\mathrm{G}} \mathsf{seq}_p(T_1', T_2)} & (\mathrm{seq}^2_{\mathrm{G}}) & \mathsf{seq}_p(\mathsf{Empty}_q, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2 \\[2ex]
(\mathrm{seq}^3_{\mathrm{G}}) & \dfrac{T_2 \xrightarrow{\bar a}_{\mathrm{G}} T_2'}{\mathsf{seq}_p(T_1, T_2) \xrightarrow{\bar a}_{\mathrm{G}} \mathsf{seq}_p(\mathrm{R}^*_{\alpha(\bar a)}(T_1), T_2')} & & \\[2ex]
(\mathrm{par}^1_{\mathrm{G}}) & \dfrac{T_1 \xrightarrow{\bar a}_{\mathrm{G}} T_1'}{\mathsf{par}_p(T_1, T_2) \xrightarrow{\bar a}_{\mathrm{G}} \mathsf{par}_p(T_1', T_2)} & (\mathrm{par}^2_{\mathrm{G}}) & \dfrac{T_2 \xrightarrow{\bar a}_{\mathrm{G}} T_2'}{\mathsf{par}_p(T_1, T_2) \xrightarrow{\bar a}_{\mathrm{G}} \mathsf{par}_p(T_1, T_2')} \\[2ex]
(\mathrm{par}^3_{\mathrm{G}}) & \mathsf{par}_p(\mathsf{Empty}_q, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2 & (\mathrm{par}^4_{\mathrm{G}}) & \mathsf{par}_p(T_1, \mathsf{Empty}_q) \xrightarrow{\tau}_{\mathrm{G}} T_1 \\[1ex]
(\mathrm{loop}^1_{\mathrm{G}}) & \mathsf{loop}_p(T) \xrightarrow{\tau}_{\mathrm{G}} \mathsf{Empty}_p & (\mathrm{loop}^2_{\mathrm{G}}) & \mathsf{loop}_p(T) \xrightarrow{\tau}_{\mathrm{G}} \mathsf{seq}_p(T, \mathsf{loop}_{p.2}({}_{p.2}T)) \\[1ex]
(\mathrm{alt}^1_{\mathrm{G}}) & \mathsf{alt}_p(T_1, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_1 & (\mathrm{alt}^2_{\mathrm{G}}) & \mathsf{alt}_p(T_1, T_2) \xrightarrow{\tau}_{\mathrm{G}} T_2
\end{array}
$$

Tafel 5.3: Pfadannotierte globale operationalen Semantik ohne Kanäle

Beobachtungsäquivalenz der globalen operationalen Semantik mit Kanälen und einer *lokalen* operationalen Semantik mit Kanälen gezeigt werden.

a) Pfadannotierte globale operationale Semantik ohne Kanäle

Die Konfigurationen der pfadannotierten globalen operationalen Semantik ohne Kanäle sind die Interaktionsterme $T \in \mathrm{Term}$ aus Tafel 5.2. Die Endkonfigurationen haben die allgemeine Form $\mathsf{Empty}_p := \varepsilon_p$. Transitionen haben die allgemeine Form $T \xrightarrow{\bar a}_{\mathrm{G}} T'$ mit $T, T' \in \mathrm{Term}$, $T \neq \mathsf{Empty}_p$ und $\bar a \in \mathbb{A}_\tau$ (vgl. mit Abschnitt 4.3). Das redefinierte Regelsystem G der pfadannotierten Fassung der globalen operationalen Semantik ohne Kanäle ist in Tafel 5.3 angegeben. Die Metavariablen der Regelschemata von G haben die folgenden Bereiche: $T \in \mathrm{Term}$, $B \in \mathrm{Basic}$, $a \in \mathbb{A}$, $\bar a \in \mathbb{A}_\tau$ und $p, q \in \mathrm{Path}$.

Die Wohlnummeriertheit der Konfigurationsterme T ist eine Invariante der pfadannotierten Fassung der globalen operationalen Semantik ohne Kanäle. Die geringfügigen Änderungen, die an der Formulierung von Satz 4.3 und dem zugehörigen Beweis erforderlich sind, seien dem Leser überlassen.

Ist $I \in \mathrm{IFrag}$ eine Anfangskonfiguration der (unnummerierten) globalen operationalen Semantik aus Abschnitt 5.2, dann ist $\mathrm{num}(I, 1)$ die korrespondierende Anfangskonfiguration der pfadannotierten Fassung.

b) Pfadannotierte globale operationale Semantik mit Kanälen

Ein Pomset $p \in \mathbb{D}$ heiße *parallel*, falls ein Repräsentant $(E, \leq, \lambda)$ von p existiert, der die Eigenschaft $\forall\, e_1, e_2 \in E. (\lambda(e_1) \asymp \lambda(e_2) \iff e_1 \leq e_2 \vee e_2 \leq e_1)$ hat. Ein paralleles Pomset kann mit einer Menge von Traces identifiziert werden, die auf paarweise verschiedenen Lifelines liegen. Es sei definiert $\mathrm{Basic}_{\|} := \{p \in \mathbb{D} \mid p \text{ ist parallel und endlich}\}$. Offenkundig gilt $\mathrm{Basic}_{\|} \subseteq \mathrm{Basic}$. Eine Projektion $\mathrm{shrink} : \mathrm{Basic} \to \mathrm{Basic}_{\|}$ sei definiert durch $\mathrm{shrink}([(E, \leq, \lambda)]) := [(E, \leq', \lambda)]$ mit $\leq' := \leq \cap \{(e_1, e_2) \in E \times E \mid \lambda(e_1) \asymp \lambda(e_2)\}$. Die Kategorie $\mathrm{Term}_{\|}$ enthalte alle diejenigen Terme T aus Tafel 5.2, die nur Blattterme der Form B_p oder $\mathsf{None}(B)_p$ mit $B \in \mathrm{Basic}_{\|}$ enthalten. Das Symbol shrink sei überladen und bezeichne auch eine Funktion $\mathrm{Term} \to \mathrm{Term}_{\|}$, die definiert ist wie folgt:

$$
\begin{aligned}
&\mathrm{shrink}(B_p) &&:= (\mathrm{shrink}(B))_p \\
&\mathrm{shrink}(\mathsf{None}(B)_p) &&:= \mathsf{None}(\mathrm{shrink}(B))_p \\
&\mathrm{shrink}(uop_p(T_1)) &&:= uop_p(\mathrm{shrink}(T_1)) \\
&\mathrm{shrink}(bop_p(T_1, T_2)) &&:= bop_p(\mathrm{shrink}(T_1), \mathrm{shrink}(T_2))
\end{aligned}
$$

Die Konfigurationen der pfadannotierten globalen operationalen Semantik mit Kanälen sind geordnete Paare $\gamma = (T, C) \in \mathrm{Term}_{\|} \times \mathrm{Channels}$. Die Endkonfigurationen haben die allgemeine Form $(\mathsf{Empty}_p\,, \emptyset)$. Die Transitionen haben die allgemeine Form $\gamma \xrightarrow{\bar{a}}_{\mathrm{Gc}} \gamma'$ mit $\gamma, \gamma' \in \mathrm{Term}_{\|} \times \mathrm{Channels}$, $\gamma \neq (\mathsf{Empty}_p\,, \emptyset)$ und $\bar{a} \in \mathbb{A}_\tau$. Das Regelsystem Gc der pfadannotierten globalen operationalen Semantik mit Kanälen entspricht dem Regelsystem aus Tafel 5.3, sofern man die üblichen Shorthand Notations benutzt, die Annotierung G durch Gc ersetzt und das Axiomenschema ($\mathrm{basic}_{\mathrm{G}}$) durch folgende zwei Axiomenschemata ersetzt:

$$
\begin{aligned}
&(\mathrm{basic}^1_{\mathrm{Gc}}) \quad B_p\,, C \xrightarrow{a}_{\mathrm{Gc}} (B \setminus \{a\})_p\,, C[(s, r, p, m) \mapsto C(s, r, p, m) + 1] \\
&\qquad\qquad \text{falls } a \in \mathrm{Min}(B) \\
&\qquad\qquad \text{und } a = \mathsf{snd}(s, r, m).
\end{aligned}
$$

$$
\begin{aligned}
&(\mathrm{basic}^2_{\mathrm{Gc}}) \quad B_p\,, C \xrightarrow{a}_{\mathrm{Gc}} (B \setminus \{a\})_p\,, C[(s, r, p, m) \mapsto C(s, r, p, m) - 1] \\
&\qquad\qquad \text{falls } a \in \mathrm{Min}(B) \\
&\qquad\qquad \text{und } a = \mathsf{rcv}(s, r, m) \\
&\qquad\qquad \text{und } C(s, r, p, m) \geq 1.
\end{aligned}
$$

Ist $T \in \mathrm{Term}$ eine Startkonfiguration der pfadannotierten globalen operationalen Semantik ohne Kanäle, dann ist $(\mathrm{shrink}(T), \emptyset)$ die korrespondierende Startkonfiguration der pfadannotierten globalen operationalen Semantik mit Kanälen.

5.3.4 Wohlgeformte elementare Interaktionen

Seien p, q zwei Pomsets. Es heiße p ein *Teilpomset von* q (symbolisch: $p \subseteq q$), falls es Repräsentanten $(E, \leq, \lambda)$ von p und $(E', \leq', \lambda')$ von q gibt, so dass gilt $E = E'$ und $\leq\ \subseteq\ \leq'$ und $\lambda = \lambda'$. Anmerkung: Die Teilpomsetbeziehung $\leq$ ist auf der Klasse der Pomsets reflexiv und transitiv. Auf endlichen Pomsets ist die Teilpomsetbeziehung auch antisymmetrisch.

Sei $p = [(E, \leq, \lambda)] \in \mathbb{D}$. Es heiße $[(E, \bar{\lesssim}, \lambda)]$ eine *lokale Linearisierung von* p, falls $\leq\ \subseteq\ \bar{\lesssim}$ gilt und $[(E, \bar{\lesssim}, \lambda)]$ lokal linear ist. Es sei $p{\gtrless}{\downarrow}$ definiert als die Menge aller lokalen Linearisierungen von p. Ein Pomset $q \in \mathbb{D}$ heiße eine *minimale lokale Linearisierung von* p, falls q eine lokale Linearisierung von p ist und kein echtes Teilpomset von q existiert, das lokale Linearisierung von p ist. Es sei $p{\gtrless}{\downarrow}_{\min}$ definiert als die Menge aller minimalen lokalen Linearisierungen von p. Die Abbildungen ${\gtrless}{\downarrow}$ und ${\gtrless}{\downarrow}_{\min}$ werden in kanonischer Weise auf Prozesse $P \subseteq \mathbb{D}$ angehoben.

Sei $P \subseteq \mathbb{P}$ ein Prozess (aus lokal linearen Pomsets) und $a \in \mathbb{A}$. Es sei definiert $P \setminus \{a\} := \{p \setminus \{a\} \mid p \in P \wedge a \in \mathrm{Min}(p)\}$, wobei die Notation $p \setminus \{a\}$ auf S. 82 erklärt ist. Anmerkung: Man beachte die starke Analogie zur Definition des linken Quotienten $P \,/\, a$ von S. 11. Für Prozesse $P \subseteq \mathbb{T}$ gilt $P \setminus \{a\} = P \,/\, a$.

Lemma 5.2 Seien $P_1, P_2 \subseteq \mathbb{P}$ Prozesse und $a \in \mathbb{A}$. Es gilt:

$$((P_1 \parallel P_2){\gtrless}{\downarrow}_{\min}) \setminus \{a\} = ((P_1 \setminus \{a\}) \parallel P_2){\gtrless}{\downarrow}_{\min} \cup (P_1 \parallel (P_2 \setminus \{a\})){\gtrless}{\downarrow}_{\min}$$

Anmerkung: Das Lemma 5.2 ist eine Verallgemeinerung von Lemma 2.3.3. Der Gedanke liegt nahe, dass sich *alle* Teilaussagen von Lemmas 2.3 in der hier beschriebenen Art verallgemeinern lassen.

Beweis. Seien $p_1, p_2 \in \mathbb{P}$ und $a \in \mathbb{A}$. Es genügt zu zeigen:

Z. z. $((p_1 \parallel p_2){\gtrless}{\downarrow}_{\min}) \setminus \{a\} = ((\{p_1\} \setminus \{a\}) \parallel \{p_2\}){\gtrless}{\downarrow}_{\min} \cup (\{p_1\} \parallel (\{p_2\} \setminus \{a\})){\gtrless}{\downarrow}_{\min}$

“$\subseteq$”: Sei $q \in ((p_1 \parallel p_2){\gtrless}{\downarrow}_{\min}) \setminus \{a\}$ beliebig. Es gibt ein $p \in (p_1 \parallel p_2){\gtrless}{\downarrow}_{\min}$ mit $a \in \mathrm{Min}(p)$ und $q = p \setminus \{a\}$. Wir wählen Repräsentanten von p_1 und p_2, so dass gilt: $p_1 = [(E_1, \leq_1, \lambda_1)]$ und $p_2 = [(E_2, \leq_2, \lambda_2)]$ und $E_1 \cap E_2 = \emptyset$. Dann gilt $p_1 \parallel p_2 = [(E, \leq, \lambda)]$ mit $E = E_1 \mathbin{\dot\cup} E_2$ und $\leq\ =\ \leq_1 \cup \leq_2$ und $\lambda = \lambda_1 \cup \lambda_2$. Wegen $p \in (p_1 \parallel p_2){\gtrless}{\downarrow}$ gibt es eine partielle Ordnung $\bar{\lesssim}$ auf E mit $\leq\ \subseteq\ \bar{\lesssim}$ und $[(E, \bar{\lesssim}, \lambda)] = p$ ist lokal linear. Wegen $p \in (p_1 \parallel p_2){\gtrless}{\downarrow}_{\min}$ gilt für jede partielle Ordnung $\tilde{\lesssim}$ auf E mit $\tilde{\lesssim}\ \subseteq\ \bar{\lesssim}$ und $\tilde{\lesssim}\ \neq\ \bar{\lesssim}$, dass $\leq\ \not\subseteq\ \tilde{\lesssim}$ oder $[(E, \tilde{\lesssim}, \lambda)]$ ist nicht lokal linear. Außerdem gibt es ein (wg. p lokal linear) eindeutig bestimmtes $e_0 \in E$ mit $\lambda(e_0) = a$ und $\forall e \in E \,.\, (e \mathrel{\bar{\lesssim}} e_0 \Rightarrow e = e_0)$. Schließlich ist $q = p \setminus \{a\} = [(E', \bar{\lesssim}', \lambda')]$ wobei $E' := E \setminus \{e_0\}$ und $\bar{\lesssim}' := \bar{\lesssim} \cap (E' \times E')$ und $\lambda' := \lambda \restriction E'$.

Fall 1: $e_0 \in E_1$. Weil $\leq_1\ \subseteq\ \leq\ \subseteq\ \bar{\lesssim}$ gilt, ist e_0 auch bezüglich $\leq_1$ minimal in E_1. Also ist $a \in \mathrm{Min}(p_1)$ und $p_1 \setminus \{a\} = [(E_1', \leq_1', \lambda_1')]$ mit $E_1' := E_1 \setminus \{e_0\}$

und $\leq_1' := \leq_1 \cap (E_1' \times E_1')$ und $\lambda_1' = \lambda_1 \restriction E_1'$. Dann gilt: $(p_1 \setminus \{a\}) \,\|\, p_2 = [(E_1' \cup E_2, \leq_1' \cup \leq_2, \lambda_1' \cup \lambda_2)]$.

Zu zeigen: $q \in ((p_1 \setminus \{a\}) \,\|\, p_2){\asymp}{\downarrow}$

Es gilt $E_1' \,\dot\cup\, E_2 = (E_1 \setminus \{e_0\}) \,\dot\cup\, E_2 = (E_1 \,\dot\cup\, E_2) \setminus \{e_0\} = E \setminus \{e_0\} = E'$. Es gilt $\leq_1' \cup \leq_2 = (\leq_1 \cap (E_1' \times E_1')) \cup \leq_2 = (\leq_1 \cup \leq_2) \cap ((E_1' \cup E_2) \times (E_1' \cup E_2)) = \leq \cap (E' \times E') \subseteq \bar{\leq} \cap (E' \times E') = \bar{\leq}'$. Es gilt ferner $\lambda_1' \cup \lambda_2 = \lambda_1 \restriction E_1' \cup \lambda_2 = (\lambda_1 \cup \lambda_2) \restriction (E_1' \cup E_2) = \lambda \restriction E' = \lambda'$. Mit p ist auch q lokal linear.

Zu zeigen: $q \in ((p_1 \setminus \{a\}) \,\|\, p_2){\asymp}{\downarrow}_{\min}$

Angenommen es gäbe eine partielle Ordnung $\tilde{\leq}'$ auf $E' = E_1' \,\dot\cup\, E_2$ mit $\tilde{\leq}' \subseteq \bar{\leq}'$ und $\tilde{\leq}' \neq \bar{\leq}'$ und $\leq_1' \cup \leq_2 \subseteq \tilde{\leq}'$ und $[(E', \tilde{\leq}', \lambda')]$ ist lokal linear. Wir setzen: $\tilde{\leq} := (\bar{\leq} \cap (\{e_0\} \times E)) \,\dot\cup\, \tilde{\leq}'$.

(1) Z. z. $\tilde{\leq}$ ist partielle Ordnung auf E:

Die Reflexivität von $\tilde{\leq}$ auf E ist klar. Zum Beweis der Transitivität von $\tilde{\leq}$ seien $e_1, e_2, e_3 \in E$ mit $e_1 \,\tilde{\leq}\, e_2$ und $e_2 \,\tilde{\leq}\, e_3$ vorgegeben. Falls $e_1 \neq e_0$ ist, dann sind $e_1, e_2, e_3 \in E'$ und $e_1 \,\tilde{\leq}'\, e_2$ und $e_2 \,\tilde{\leq}'\, e_3$. Mit der Transitivität von $\tilde{\leq}'$ folgt $e_1 \,\tilde{\leq}'\, e_3$ und somit auch $e_1 \,\tilde{\leq}\, e_3$. Falls $e_1 = e_0$ und o. B. d. A. $e_2 \neq e_0$ ist, dann gilt $e_2, e_3 \in E'$ und $e_1 \,\bar{\leq}\, e_2$ und $e_2 \,\tilde{\leq}'\, e_3$. Weil $\tilde{\leq}' \subseteq \bar{\leq}' \subseteq \bar{\leq}$ gilt und $\bar{\leq}$ transitiv ist, folgt $e_1 \,\bar{\leq}\, e_3$ und somit auch $e_1 \,\tilde{\leq}\, e_3$. Zum Beweis der Antisymmetrie von $\tilde{\leq}$ seien $e_1, e_2 \in E$ mit $e_1 \,\tilde{\leq}\, e_2$ und $e_2 \,\tilde{\leq}\, e_1$ vorgegeben. Angenommen es wäre $e_1 \neq e_2$. Der Fall $e_1 = e_0$ ist wegen $e_2 \,\tilde{\leq}\, e_1$ nicht möglich. Also gilt $e_1 \neq e_0$. Dann sind $e_1, e_2 \in E'$ und $e_1 \,\tilde{\leq}'\, e_2$ und $e_2 \,\tilde{\leq}'\, e_1$. Mit der Antisymmetrie von $\tilde{\leq}'$ folgt $e_1 = e_2$.

(2) Z. z. $\tilde{\leq} \subseteq \bar{\leq}$:

Das ist klar, weil $\tilde{\leq}' \subseteq \bar{\leq}' \subseteq \bar{\leq}$.

(3) Z. z. $\tilde{\leq} \neq \bar{\leq}$:

Angenommen es wäre $\bar{\leq} = \tilde{\leq}$. Dann gilt $\bar{\leq} = (\bar{\leq} \cap (\{e_0\} \times E)) \,\dot\cup\, \tilde{\leq}'$ und folglich $\bar{\leq}' = \bar{\leq} \cap (E' \times E') = \tilde{\leq}'$. Widerspruch.

(4) Z. z. $\leq \subseteq \tilde{\leq}$:

Es gilt $\leq_1' \,\dot\cup\, \leq_2 \subseteq \tilde{\leq}'$. Es folgt: $\leq = \leq_1 \cup \leq_2 = (\leq_1 \cap (\{e_0\} \times E_1)) \,\dot\cup\, (\leq_1 \cap (E_1' \times E_1')) \,\dot\cup\, \leq_2 \subseteq (\bar{\leq} \cap (\{e_0\} \times E)) \,\dot\cup\, \leq_1' \,\dot\cup\, \leq_2 \subseteq (\bar{\leq} \cap (\{e_0\} \times E)) \,\dot\cup\, \tilde{\leq}' = \tilde{\leq}$.

(5) Z. z. $[(E, \tilde{\leq}, \lambda)]$ ist lokal linear:

Seien $e_1, e_2 \in E$ mit $\lambda(e_1) \asymp \lambda(e_2)$. Weil $[(E, \bar{\leq}, \lambda)]$ lokal linear ist, gilt $e_1 \,\bar{\leq}\, e_2$ oder $e_2 \,\bar{\leq}\, e_1$. O. B. d. A. sei $e_1 \,\bar{\leq}\, e_2$ und $e_1 \neq e_2$. Falls $e_1 = e_0$ ist, dann gilt $e_1 \,\tilde{\leq}\, e_2$. Falls $e_1 \neq e_0$ ist, dann sind $e_1, e_2 \in E'$ und $\lambda'(e_1) \asymp \lambda'(e_2)$. Weil $[(E', \tilde{\leq}', \lambda')]$ lokal linear ist, folgt $e_1 \,\tilde{\leq}'\, e_2$ oder $e_2 \,\tilde{\leq}'\, e_1$. Mit $\tilde{\leq}' \subseteq \tilde{\leq}$ folgt die Behauptung.

Die Aussagen (1) bis (5) sind ein Widerspruch zur obigen Voraussetzung der Nichtexistenz einer solchen partiellen Ordnung $\tilde{\leq}$.

Fall 2: $e_0 \in E_2$: Analog.

"$\supseteq$": Der Beweis in Gegenrichtung sei dem Leser übelassen. $\square$

Sei $p \in \mathbb{P}$. Wir schreiben $p^{n\between} \in \mathbb{P}$ für die n-fache $\between$-Konkatenation von p mit sich selbst, welche rekursiv definiert ist wie folgt: $p^{0\between} := \varepsilon$ und $p^{(n+1)\between} := p \,;_{\between} p^{n\between}$ für jedes $n \in \mathbb{N}_0$. Es sei darauf hingewiesen, dass die Hebung dieser n-fachen $\between$-Konkatenation *nicht* mit der in Abschnitt 2.2 eingeführten n-ten Potenz eines Prozesses übereinstimmt, weil bei der n-te Potenz des Prozesses P in jedem Schritt ein anderes Pomset p aus dem Prozess P zur $\between$-Konkatenation ausgewählt werden kann. Seien $s, r \in \mathbb{I}$, $m \in \mathbb{M}$ und $\mu, \nu \in \mathbb{N}_0$. Definition:

$$B(s, r, m, \mu, \nu) := (\mathsf{rcv}(s, r, m))^{\mu\between} ;_{\between} (\mathsf{snd}(s, r, m) \,;\, \mathsf{rcv}(s, r, m))^{\nu\between}$$

Lemma 5.3 Seien $s, r \in \mathbb{I}$, $m \in \mathbb{M}$, $\mu, \nu \in \mathbb{N}_0$ und $a \in \mathrm{Min}(B(s, r, m, \mu, \nu))$. Dann ist ($a = \mathsf{snd}(s, r, m)$ oder $a = \mathsf{rcv}(s, r, m)$) und es gilt:

1. $a = \mathsf{snd}(s, r, m) \implies \nu \geq 1 \wedge B(s, r, m, \mu, \nu) \setminus \{a\} = B(s, r, m, \mu + 1, \nu - 1)$
2. $a = \mathsf{rcv}(s, r, m) \implies \mu \geq 1 \wedge B(s, r, m, \mu, \nu) \setminus \{a\} = B(s, r, m, \mu - 1, \nu)$

Der Beweis der anschaulichen Aussage von Lemma 5.3 sei dem Leser überlassen.

Definition: Eine Basic Interaction $B \in \mathrm{Basic} = \mathbb{P}_{\mathrm{fin}}$ heiße *wohlgeformt*, falls ein $n \in \mathbb{N}_0$ und paarweise verschiedene Tripel $(s_1, r_1, m_1), \ldots, (s_n, r_n, m_n) \in \mathbb{I} \times \mathbb{I} \times \mathbb{M}$ und $\mu_1, \ldots, \mu_n \in \mathbb{N}_0$ und $\nu_1, \ldots, \nu_n \in \mathbb{N}_0$ existieren, so dass gilt:

$$B \in (B(s_1, r_1, m_1, \mu_1, \nu_1) \parallel \ldots \parallel B(s_n, r_n, m_n, \mu_n, \nu_n)) \between\!\downarrow_{\min}$$

Satz 5.4 Sei $B \in \mathrm{Basic}$ wohlgeformt und $a \in \mathrm{Min}(B)$. Dann ist $B \setminus \{a\}$ wohlgeformt.

Anmerkung: Das Lemma 5.4 besagt, dass die Eigenschaft der Wohlgeformtheit der elementaren Blattterme eines Interaktionsterms eine Invariante der globalen operationalen Semantiken ohne Kanäle ist.

Beweis. Aus $B \in (B(s_1, r_1, m_1, \mu_1, \nu_1) \parallel \ldots \parallel B(s_n, r_n, m_n, \mu_n, \nu_n)) \between\!\downarrow_{\min}$ und $a \in \mathrm{Min}(B)$ folgt $B \setminus \{a\} \in (B(s_1, r_1, m_1, \mu_1, \nu_1) \parallel \ldots \parallel B(s_n, r_n, m_n, \mu_n, \nu_n)) \between\!\downarrow_{\min} \setminus \{a\}$ $= (\{B(s_1, r_1, m_1, \mu_1, \nu_1)\} \parallel \ldots \parallel \{B(s_n, r_n, m_n, \mu_n, \nu_n)\}) \between\!\downarrow_{\min} \setminus \{a\}$.

Mit dem Lemma 5.2 (analog) und $a = \mathsf{xxx}(s_i, r_i, m_i)$ mit $\mathsf{xxx} \in \{\mathsf{snd}, \mathsf{rcv}\}$ und $i \in \{1, \ldots, n\}$ und $\{B(s_j, r_j, m_j, \mu_j, \nu_j)\} \setminus \{\mathsf{xxx}(s_i, r_i, m_i)\} = \emptyset$ für $j \neq i$ folgt:

$$\begin{aligned} B \setminus \{a\} \in (&\{B(s_1, r_1, m_1, \mu_1, \nu_1)\} \parallel \ldots \parallel \{B(s_{i-1}, r_{i-1}, m_{i-1}, \mu_{i-1}, \nu_{i-1})\} \\ &\parallel \{B(s_i, r_i, m_i, \mu_i, \nu_i)\} \setminus \{\mathsf{xxx}(s_i, r_i, m_i)\} \\ &\parallel \{B(s_{i+1}, r_{i+1}, m_{i+1}, \mu_{i+1}, \nu_{i+1})\} \parallel \ldots \parallel \{B(s_n, r_n, m_n, \mu_n, \nu_n)\}) \between\!\downarrow_{\min} \end{aligned}$$

Weil der Prozess als Ganzes nichtleer ist, sind alle Teilprozesse nichtleer; insbesondere gilt $\{B(s_i, r_i, m_i, \mu_i, \nu_i)\} \setminus \{\mathsf{xxx}(s_i, r_i, m_i)\} \neq \emptyset$. Daraus folgt $\mathsf{xxx}(s_i, r_i, m_i) \in \mathrm{Min}(B(s_i, r_i, m_i, \mu_i, \nu_i))$ und außerdem $\{B(s_i, r_i, m_i, \mu_i, \nu_i)\} \setminus \{\mathsf{xxx}(s_i, r_i, m_i)\} = \{B(s_i, r_i, m_i, \mu_i, \nu_i) \setminus \{\mathsf{xxx}(s_i, r_i, m_i)\}\}$. Mit Lemma 5.3 folgt die Behauptung. $\square$

Die Funktion pending : $\mathbb{P}_{\text{fin}} \times \mathbb{I} \times \mathbb{I} \times \mathbb{M} \longrightarrow \mathbb{Z}$ mit $\text{pending}([(E, \leq, \lambda)], s, r, m) := |\{e \in E \mid \lambda(e) = \mathsf{rcv}(s,r,m)\}| - |\{e \in E \mid \lambda(e) = \mathsf{snd}(s,r,m)\}|$ berechne die Differenz zwischen der Anzahl der in einer elementaren Interaktion enthaltenen Empfangsaktionen $\mathsf{rcv}(s,r,m)$ und der Anzahl den korrespondierenden Sendeaktionen $\mathsf{snd}(s,r,m)$. Bemerkung: Für wohlgeformte B ist $\text{pending}(B, -, -, -) \geq 0$.

Lemma 5.5 Sei $B \in \text{Basic}$ wohlgeformt und $\underline{B} := \text{shrink}(B)$ und $a \in \text{Min}(\underline{B})$ und $a = \mathsf{snd}(s,r,m)$. Dann gilt:

1. $a \in \text{Min}(B)$
2. $\underline{B} \setminus \{a\} = \text{shrink}(B \setminus \{a\})$

Lemma 5.6 Sei $B \in \text{Basic}$ wohlgeformt und $\underline{B} := \text{shrink}(B)$ und $a \in \text{Min}(\underline{B})$ und $a = \mathsf{rcv}(s,r,m)$ und $\text{pending}(B,s,r,m) \geq 1$. Dann gilt:

1. $a \in \text{Min}(B)$
2. $\underline{B} \setminus \{a\} = \text{shrink}(B \setminus \{a\})$

Beweis. Wir beweisen nur das Lemma 5.6. Der Beweis von Lemma 5.5 ist analog.

Sei $B \in \text{Basic}$ wohlgeformt. Nach Definition ist $B \in p{\not\lessgtr}{\downarrow_{\text{min}}}$ wobei eine Darstellung $p = B(s_1, r_1, m_1, \mu_1, \nu_1) \parallel \ldots \parallel B(s_n, r_n, m_n, \mu_n, \nu_n)$ gilt. Es sei $p = [(E, \leq, \lambda)]$. Dann ist $B = [(E, \bar{\leq}, \lambda)]$ lokal linear und $\leq \subseteq \bar{\leq}$ und für jede partielle Ordnung $\tilde{\leq}$ auf E mit $\tilde{\leq} \subseteq \bar{\leq}$ und $\tilde{\leq} \neq \bar{\leq}$ gilt, dass $\leq \not\subseteq \tilde{\leq}$ gilt oder $[(E, \tilde{\leq}, \lambda)]$ nicht lokal linear ist.

Es gelte $a = \mathsf{rcv}(s_i, r_i, m_i) \in \text{Min}(\text{shrink}(B))$ und $\text{pending}(B, s_i, r_i, m_i) \geq 1$. Weil $a \in \text{Min}(\text{shrink}(B))$ ist, gibt es ein (wegen B lokal linear) eindeutig bestimmtes $e_0 \in \text{Min}_{\bar{\leq}'}(E)$ mit $\lambda(e_0) = a$; dabei ist $\bar{\leq}' := \bar{\leq} \cap \{(e_1, e_2) \in E \times E \mid \lambda(e_1) \not\lessgtr \lambda(e_2)\}$. Wir richten unseren Blick auf die aktive Lifeline $r_i = \alpha(a)$. Mit den Setzungen $E^{r_i} := \lambda^{-1}(\mathbb{A}^{r_i})$ und $\bar{\leq}^{r_i} := \bar{\leq} \cap (E^{r_i} \times E^{r_i})$ gilt $e_0 \in \text{Min}_{\bar{\leq}^{r_i}}(E)$.

Zu zeigen: $e_0 \in \text{Min}_{\bar{\leq}}(E)$

Angenommen es wäre $e_0 \notin \text{Min}_{\bar{\leq}}(E)$. Dann gäbe es ein $e_{-1} \in E$ mit $e_{-1} \bar{\leq} e_0$ und $e_{-1} \neq e_0$. Weil $e_0 \in \text{Min}_{\bar{\leq}^{r_i}}(E)$ ist, muss $r_i \notin \alpha(\lambda(e_{-1}))$ sein. Es sei o. B. d. A. e_{-1} überdies in E bezüglich $\bar{\leq}$ maximal mit der Eigenschaft $e_{-1} \bar{\leq} e_0$ und $e_{-1} \neq e_0$. Wenn es uns gelingt zu zeigen, dass $e_{-1} \leq e_0$ gilt, dann haben wir einen Widerspruch zur Konstruktion von $B(s_i, r_i, m_i, \mu_i, \nu_i)$ gefunden. Denn eine Empfangsaktion $\mathsf{rcv}(s_i, r_i, m_i)$, die innerhalb von $B(s_i, r_i, m_i, \mu_i, \nu_i)$ auf der Lifeline r_i minimal ist ("zuoberst liegt") und überzählig ist (d. h. $\text{pending}(p, s_i, r_i, m_i) = \text{pending}(B, s_i, r_i, m_i) \geq 1$), hat bezüglich der partiellen Ordnung von $B(s_i, r_i, m_i, \mu_i, \nu_i)$ keine echten Vorgänger.

Zu zeigen: $e_{-1} \leq e_0$

Angenommen es wäre $e_{-1} \not\leq e_0$. Wir setzen $\tilde{\leq} := \bar{\leq} \setminus \{(e_{-1}, e_0)\}$. Dann ist $\tilde{\leq} \subseteq \bar{\leq}$ und $\tilde{\leq} \neq \bar{\leq}$. Wegen $\leq \subseteq \bar{\leq}$ und $e_{-1} \not\leq e_0$ folgt $\leq \subseteq \tilde{\leq}$. Die Reflexivität auf E und die Antisymmetrie von $\tilde{\leq}$ ergeben sich aus den entsprechenden Eigenschaften

$$
\begin{array}{rcll}
T \in \mathrm{Term}_{\mathrm{w}} & ::= & \mathsf{None}(B)_p & \text{falls } B \text{ wohlgeformt} \\
& \mid & B_p & \text{falls } B \text{ wohlgeformt} \\
& \mid & \mathsf{strict}_p(T_1, T_2) & \\
& \mid & \mathsf{seq}_p(T_1, T_2) & \\
& \mid & \mathsf{par}_p(T_1, T_2) & \\
& \mid & \mathsf{loop}_p(T) & \text{falls } T \text{ ausgewogen} \\
& \mid & \mathsf{alt}_p(T_1, T_2) & \text{falls } T_1, T_2 \text{ ausgewogen}
\end{array}
$$

Tafel 5.4: Abstrakte Syntax der wohlgeformten Interaktionsterme

von $\bar{\leq}$. Zum Beweis der Transitivität seien $e_1, e_2, e_3 \in E$ mit $e_1 \mathrel{\tilde{\leq}} e_2$ und $e_2 \mathrel{\tilde{\leq}} e_3$ vorgegeben. Dann gilt $e_1 \mathrel{\bar{\leq}} e_2$ und $e_2 \mathrel{\bar{\leq}} e_3$. Weil $\bar{\leq}$ transitiv ist, folgt $e_1 \mathrel{\bar{\leq}} e_3$. Angenommen $e_1 \mathrel{\tilde{\leq}} e_3$ wäre falsch. Dann ist $e_1 = e_{-1}$ und $e_3 = e_0$. Es folgt $e_{-1} \mathrel{\bar{\leq}} e_2 \mathrel{\bar{\leq}} e_0$ und $e_2 \neq e_{-1}$ und $e_2 \neq e_0$. Das ist ein Widerspruch zur Maximalität von e_{-1}. Also ist $\tilde{\leq}$ auch transitiv und somit eine partielle Ordnung auf E. Weil $B = [(E, \bar{\leq}, \lambda)]$ lokal linear ist und $\alpha(\lambda(e_{-1})) \cap \alpha(\lambda(e_0)) = \alpha(\lambda(e_{-1})) \cap \{r_i\} = \emptyset$ gilt, ist auch $[(E, \tilde{\leq}, \lambda)]$ lokal linear. Das ist ein Widerspruch zur Minimalität von $\bar{\leq}$. Also gilt $e_{-1} \leq e_0$. Also gilt $e_0 \in \mathrm{Min}_{\bar{\leq}}(E)$.

Zu zeigen bleibt: $\mathrm{shrink}(B) \setminus \{a\} = \mathrm{shrink}(B \setminus \{a\})$

Diese Gleichheitsaussage ist aufgrund der Definitionen trivial. $\square$

5.3.5 Skizze des Bisimilaritätsbeweises

Wir definieren eine Funktion $\mathrm{pen} : \mathrm{Term} \times \mathbb{I} \times \mathbb{I} \times \mathrm{Path} \times \mathbb{M} \longrightarrow \mathbb{Z}$ wie folgt:

$$
\begin{array}{ll}
\mathrm{pen}(B_p\,, s, r, q, m) & := \begin{cases} \mathrm{pending}(B, s, r, m) & \text{falls } p = q \\ 0 & \text{falls } p \neq q \end{cases} \\
\mathrm{pen}(\mathsf{None}(B)_p\,, s, r, q, m) & := 0 \\
\mathrm{pen}(\mathit{uop}_p(T_1), s, r, q, m) & := \mathrm{pen}(T_1, s, r, q, m) \\
\mathrm{pen}(\mathit{bop}_p(T_1, T_2), s, r, q, m) & := \mathrm{pen}(T_1, s, r, q, m) + \mathrm{pen}(T_2, s, r, q, m)
\end{array}
$$

Ein $T \in \mathrm{Term}$ heiße *ausgewogen*, falls für alle $s, r \in \mathbb{I}$, $q \in \mathrm{Path}$ und $m \in \mathbb{M}$ gilt $\mathrm{pen}(T, s, r, q, m) = 0$. Die syntaktische Kategorie $\mathrm{Term}_{\mathrm{w}}$ der *wohlgeformten* Interaktionsterme ist in Tafel 5.4 spezifiziert. Dabei ist B Metavariable mit Bereich Basic und p Metavariable mit Bereich Path. Anmerkung: Für alle $T \in \mathrm{Term}_{\mathrm{w}}$ gilt $\mathrm{pen}(T, -, -, -, -) \geq 0$.

Als Ansatz für eine (starke) Bisimulationsrelation zwischen den Transitionssystemen der beiden pfadannotierten globalen operationalen Semantiken aus Abschnitt 5.3.3 definieren wir $\beta \subseteq \mathrm{Term} \times (\mathrm{Term}_{\parallel} \times \mathrm{Channels})$ wie folgt:

$$\beta := \{(T, \mathrm{shrink}(T), \mathrm{pen}(T, -, -, -, -)) \mid T \in \mathrm{Term_w} \text{ und } T \text{ wohlnummeriert}\}$$

Innerhalb dieser Koinduktion sind nur die Basisfälle interessant. Wir skizzieren einen solchen Basisfall: Mit ($\mathrm{basic}^2_{\mathrm{Gc}}$) gelte $\underline{B}_p, C \xrightarrow{a}_{\mathrm{Gc}} (\underline{B} \setminus \{a\})_p, C'$ mit $C' = C[(s, r, p, m) \mapsto C(s, r, p, m) - 1]$ und $a \in \mathrm{Min}(\underline{B})$ und $a = \mathsf{rcv}(s, r, m)$ und $C(s, r, p, m) \geq 1$. Außerdem gelte $\beta(T, \underline{B}_p, C)$. Es gibt ein $\tilde{T} \in \mathrm{Term_w}$ mit $T = \tilde{T}$ und $\underline{B}_p = \mathrm{shrink}(\tilde{T})$ und $C = \mathrm{pen}(\tilde{T}, -, -, -, -)$. Damit ist $T = \tilde{T} = B_p$ mit $B \in \mathrm{Basic}$ wohlgeformt. Wegen $a \in \mathrm{Min}(\underline{B})$ und $a = \mathsf{rcv}(s, r, m)$ und $\mathrm{pending}(B, s, r, m) = \mathrm{pen}(B_p, s, r, p, m) = C(s, r, p, m) \geq 1$ und Lemma 5.6 folgt $a \in \mathrm{Min}(B)$ und $\underline{B} \setminus \{a\} = \mathrm{shrink}(B \setminus \{a\})$. Mit Axiom ($\mathrm{basic_G}$) folgt $T = B_p \xrightarrow{a}_{\mathrm{G}} (B \setminus \{a\})_p$. Außerdem gilt $\beta((B \setminus \{a\})_p, (\underline{B} \setminus \{a\})_p, C')$. □

5.4 Lokale operationale Semantik mit Kanälen

Wir begnügen uns damit, das Vorgehen zu skizzieren. Sei $p = [(E, \leq, \lambda)]$ ein Pomset und A eine Menge. Dann heißt $p \restriction A := [(E', \leq', \lambda')]$ mit $E' := \lambda^{-1}(A)$ und $\leq' := \leq \cap (E' \times E')$ und $\lambda' := \lambda \restriction E'$ die Einschränkung von p auf A.

Sei $l \in \mathbb{I}$. Wir definieren Projektionen $\pi_l : \mathrm{Term} \longrightarrow \mathrm{Term}$ wie folgt:

$$\begin{aligned}
\pi_l(B_p) &:= (B \restriction \mathbb{A}^l)_p \\
\pi_l(\mathsf{None}(B)_p) &:= \begin{cases} \mathsf{None}(B \restriction \mathbb{A}^l)_p & \text{falls } l \in \alpha(B) \\ \mathsf{Empty}_p & \text{andernfalls} \end{cases} \\
\pi_l(uop_p(T)) &:= uop_p(\pi_l(T)) \\
\pi_l(bop_p(T_1, T_2)) &:= bop_p(\pi_l(T_1), \pi_l(T_2))
\end{aligned}$$

Man erweitere zunächst den Konfigurationsraum aus Abschnitt 4.4 zu $\mathrm{Conf}^L := \mathrm{LConf}^L \times \mathrm{H}^L \times \mathrm{Channels}^L$. Anschließend mache man die lokale operationale Semantik ohne Kanäle aus Abschnitt 4.5 zu einer lokalen operationalen Semantik mit Kanälen, indem man an den Regelschemata die Annotierungen L durch Lc ersetzt und das Axiomenschema ($\mathrm{basic_L}$) durch zwei Axiomenschemata ($\mathrm{basic}^1_{\mathrm{Lc}}$) und ($\mathrm{basic}^2_{\mathrm{Lc}}$) ersetzt, welche mit ($\mathrm{basic}^1_{\mathrm{Gc}}$) und ($\mathrm{basic}^2_{\mathrm{Gc}}$) von S. 88 bis auf die Annotierung übereinstimmen.

Den Ansatz für die Beobachtungsäquivalenz β aus Abschnitt 4.7 auf S. 43 erweitere man um ein logisches Konjunkt, welches das genaue Übereinstimmen des Kanalzustandes auf beiden Seiten (der globalen und der lokalen Seite) fordert.

Die Beweismethode aus Kapitel 4 bleibt anwendbar.

Kapitel 6

Zusammenfassung

In den Kapiteln 2 und 3 wurden bisherige Resultate über die Semantik von UML 2.0-Interaktionsfragmente zusammengefasst, ergänzt und z. T. korrigiert. In Kapitel 4 wurde eine verteilte operationale Semantik für positive UML 2.0-Interaktionen angegeben und der Beweis der Beobachtungsäquivalenz zur globalen operationalen Semantik detailliert ausgeführt. Schließlich wurden in Kapitel 5 die Resultate der Kapitel 2, 3 und 4 auf globale und verteilte operationale Semantiken mit Kanälen übertragen.

Weiterführende interessante Fragestellungen könnten die Entwicklung einer lokalen operationalen Semantik mit Negation, die Vermeidung von τ-Regeln, das Problem des Non-Local-Choice und die Durchleuchtung der Möglichkeit zur Automatisierung der im Kapitel 4 angewandten Beweismethodik sein. Idealerweise könnte ein Algorithmus entwickelt werden, der eine [in gewissen Grenzen] frei vorgebbare globale operationale Semantik auf eine beobachtungsäquivalente lokale operationale Semantik abbilden würde.

Literaturverzeichnis

Berregeb, Narjes et al. 1997. *Observational Proofs by Implicit Context Induction.* Research Report 3151. Nancy, France: Institut national de recherche en informatique et en automatique (INRIA).

Cengarle, María Victoria/ Alexander Knapp. 2004. "UML 2.0 Interactions: Semantics and Refinement." In *Third International Workshop on Critical Systems Development with UML (CSDUML'04, Proceedings).* Hgg Jan Jürjens et al. München: Technische Universität, 2004. 85–99. [= Technical Report TUM-I0415, München: Technische Universität, Institut für Informatik.]

———. 2005. *Operational Semantics of UML 2.0 Interactions.* Technical Report TUM-I0505. München: Technische Universität, Institut für Informatik.

———. 2007. *Operational Semantics of UML 2.0 Interactions.* [Spezifikation einer verteilten operationalen Semantik mit einer kurzen Skizze eines Äquivalenzbeweises; vervielfältigt.]

Day, Robert A./ Barbara Gastel. [6]2006 [[1]1979]. *How to Write and Publish a Scientific Paper.* Cambridge: Cambridge University Press.

Knapp, Alexander/ Harald Störrle. 2006. *An Equational Calculus for Positive UML 2.0 Interactions.* [Vervielfältigt.]

Object Management Group (Hg). 2005. *Unified Modeling Language: Superstructure.* Version 2.0: formal/05-07-04. <http://www.omg.org/cgi-bin/apps/doc?formal/05-07-04.pdf> [Geprüft am 09. November 2007].

Pratt, Vaughan R. 1986. "Modelling Concurrency with Partial Orders." *International Journal of Parallel Programming* 15:1. 33–71.

Standop, Ewald/ Matthias L. G. Meyer. [17]2004 [[1]1955]. *Die Form der wissenschaftlichen Arbeit.* Wiebelsheim: Quelle & Meyer. [Korrigierte und ergänzte Auflage. Titel der ersten Auflage: *Die Form des wissenschaftlichen Manuskripts.* Bochum: Pöppinghaus.]

Störrle, Harald. 2003. "Semantics of Interactions in UML 2.0." In *Proceedings of the 2003 IEEE Symposium on Human Centric Computing Languages and Environments (HCC '03)*. Washington, D. C.: IEEE Computer Society. 129–136.

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

....

Ort, Datum   Unterschrift