Reaktive Systeme
(Formelsammlung)

SoSe 2019
— v1 —

Uwe Nestmann
Technische Universität Berlin

2. April 2019

Zusammenfassung

Das empfehlenswerte Lehrbuch *Reactive Systems* [AILS07] bildet die Grundlage eines Moduls, das seit 2013 im Bachelor Informatik an der Technischen Universität Berlin angeboten wird. Bis 2014 war es Pflichtmodul mit der Bezeichnung *Spezifikation und Semantik* (bzw. *Theoretische Grundlagen der Informatik 4*). Ab 2015 wird es als Wahlpflichtmodul unter dem Namen *Reaktive Systeme* geführt.

Das vorliegende Dokument listet die für das Modul notwendigen Definitionen und Theoreme des genannten Lehrbuchs in deutscher Sprache. Das Dokument ist bei weitem keine Übersetzung des kompletten Lehrbuchs ins Deutsche, da es auf Erklärungen, Texte und Beispiele verzichtet. Es dient eher als schnelles Nachschlagewerk und hält sich weitgehend an die Notationen des Lehrbuchs.

Soweit möglich, verwenden wir bei Übereinstimmungen mit dem Lehrbuch die exakt gleichen Referenz-Zähler. Oftmals werden Definitionen und Aussagen im Lehrbuch jedoch im Fließtext eingeführt; in diesen Fällen verwenden wir einen separaten Zähler.

Literatur

[AILS07] Luca Aceto, Anna Ingólfsdóttír, Kim Guldstrand Larsen, and Jiří Srba. *Reactive Systems — Modelling, Specification and Verification*. Cambridge University Press, 2007.

Inhaltsverzeichnis

2 Die Sprache CCS **3**

3 Verhaltensäquivalenzen **9**

4 Fixpunkttheorie **14**

5 Hennessy-Milner-Logik **19**

6 HML mit Rekursion **22**

2 Die Sprache CCS

2.1 Definition (LTS; Definition 2.1 in [AILS07])
Ein *beschriftetes Transitionssystem* (LTS),
gelegentlich auch *Transitionsgraph* genannt,
ist ein Tripel[1]

$$(\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$$

mit:

- Proc ist eine Menge von *Zuständen* (oder *Prozessen*);
- Act ist eine Menge von *Aktionen* (oder *Beschriftungen*); und
- $\mathsf{Tran} \subseteq \mathsf{Proc} \times \mathsf{Act} \times \mathsf{Proc}$ ist eine *Transitionsrelation*.

Wir schreiben auch $s \xrightarrow{\alpha}_{\mathsf{Tran}} s'$ anstelle von $(s, \alpha, s') \in \mathsf{Tran}$.
Wir schreiben auch $s \xrightarrow{\alpha} s'$ anstelle von $s \xrightarrow{\alpha}_{\mathsf{Tran}} s'$.
Wir schreiben $s \not\xrightarrow{\alpha} s'$, falls es kein s' gibt mit $s \xrightarrow{\alpha} s'$.

Ein LTS ist *endlich*, falls sowohl Proc als auch Act endlich sind;
ansonsten nennen wir es *unendlich*.

2.2 Definition (Alternative Transitionsrelation)
Die Relation $\mathsf{Tran} \subseteq \mathsf{Proc} \times \mathsf{Act} \times \mathsf{Proc}$ wird auch durch die Familie

$$\left(\xrightarrow{\alpha}\right)_{\alpha \in \mathsf{Act}}$$

mit $\xrightarrow{\alpha} \triangleq \{\, (s,t) \mid (s,\alpha,t) \in \mathsf{Tran} \,\}$ dargestellt.

2.3 Definition (Abkürzungen)
Wir schreiben:

- $s \xrightarrow{\lambda} s$, für alle $s \in \mathsf{Proc}$;
- $s \xrightarrow{\alpha w} s'$ für alle $s, s' \in \mathsf{Proc}$, $\alpha \in \mathsf{Act}$ und $w \in \mathsf{Act}^*$,
  falls es $t \in \mathsf{Proc}$ gibt mit $s \xrightarrow{\alpha} t$ und $t \xrightarrow{w} s'$;
- $s \rightarrow^* s'$ für alle $s, s' \in \mathsf{Proc}$,
  falls es $w \in \mathsf{Act}^*$ gibt mit $s \xrightarrow{w} s'$;
- $s \rightarrow s'$ für alle $s, s' \in \mathsf{Proc}$,
  falls es $\alpha \in \mathsf{Act}$ gibt mit $s \xrightarrow{\alpha} s'$;
- $s \xrightarrow{\alpha}$ für alle $s \in \mathsf{Proc}$,
  falls es $s' \in \mathsf{Proc}$ gibt mit $s \xrightarrow{\alpha} s'$;
- $s \rightarrow$ für alle $s \in \mathsf{Proc}$,
  falls es $s' \in \mathsf{Proc}$ und $\alpha \in \mathsf{Act}$ gibt mit $s \xrightarrow{\alpha} s'$.

[1] Wir bevorzugen hier die Variante aus Remark 2.2 [AILS07]

2.4 Definition (Erreichbare Zustände; Definition 2.2 in [AILS07])
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Seien $s, s' \in \mathsf{Proc}$.
Dann heißt s' erreichbar von s, falls $s \to^* s'$.
$\mathrm{Reach}(s) \triangleq \{ s' \in \mathsf{Proc} \mid s \to^* s' \}$
ist der *von* s *aus (erreichbare) Zustandsraum*.

2.5 Definition (Syntax)

- $\mathcal{A}$ ist eine abzählbar unendlich große Menge von *(Kanal-)Namen*;
- $\overline{\mathcal{A}} \triangleq \{ \bar{a} \mid a \in \mathcal{A} \}$ ist die Menge der *Co-Namen*;
- $\mathcal{L} \triangleq \mathcal{A} \cup \overline{\mathcal{A}}$ ist die Menge der *Beschriftungen*;
  wir benutzen die Metavariablen $a, b, \ldots$;
- $\mathsf{Act} \triangleq \mathcal{L} \cup \{ \tau \}$ ist die Menge der *Aktionen*.
  wir benutzen die Metavariablen $\alpha, \beta, \ldots$.

Es gilt $\bar{\bar{a}} = a$.
Wir schreiben $\overline{L} \triangleq \{ \bar{a} \mid a \in L \}$ für beliebige $L \subseteq \mathcal{L}$.
Es gibt kein $\bar{\tau}$.

- $\mathcal{K}$ ist eine abzählbar unendlich große Menge von *Prozesskonstanten*;
  wir benutzen die Metavariablen $A, \ldots, K, \ldots$
  sowie die konkreten Konstanten $\mathbf{A}, \ldots, \mathbf{K}, \ldots$

2.6 Definition (Syntax; Definition 2.3 in [AILS07])
Die Menge $\mathcal{P}$ der CCS-Ausdrücke ist gegegeben durch folgende Grammatik:

$$P \ ::= \ \mathbf{K} \ \big| \ \alpha.P \ \big| \ \sum_{i\in I} P_i \ \big| \ P \mid P \ \big| \ P[f] \ \big| \ P \setminus A$$

- $\mathbf{K} \in \mathcal{K}$ ist eine Prozesskonstante;
  für jede konkret benutzte Konstante $\mathbf{K}$ muss eine Definition der Form
  $$\mathbf{K} \stackrel{\text{def}}{=} P$$
  angegeben werden.
- $\alpha \in \mathsf{Act}$ ist eine Aktion;
- I ist eine (potenziell abzählbar unendlich große) Indexmenge;
- $f : \mathsf{Act} \to \mathsf{Act}$ ist eine *Umbenennungsfunktion* mit
  $$\begin{aligned} f(\tau) &= \tau \\ f(\overline{a}) &= \overline{f(a)} \qquad \text{für alle } a \in \mathcal{L} \end{aligned}$$
- $A \subseteq \mathcal{A}$ ist eine Menge von Namen;
- als Abkürzungen gelten:
  $$\mathbf{0} \triangleq \sum_{i\in\emptyset} P_i \qquad\qquad P_1 + P_2 \triangleq \sum_{i\in\{1,2\}} P_i$$
- es gilt Operatorenvorrang in absteigender Folge entsprechend:
  - Restriktion $P \setminus A$ und Umbenennung $P[f]$,
  - Präfix $\alpha.P$,
  - Parallelkomposition $P_1 \mid P_2$, und
  - Summe $\sum_{i\in I} P_i$.

2.7 Definition (Umbenennung) Sei $A \triangleq \{a_1, \ldots, a_n\} \subseteq \mathcal{A}$ eine Menge paarweise verschiedener Aktionsnamen. Oft benutzen wir die Schreibweise

$$P[b_1/a_1, \ldots, b_n/a_n]$$

für $n \geqslant 1$ und $\{b_1, \ldots, b_n\} \subseteq \mathcal{A}$.
Dies entspricht dem CCS-Ausdruck $P[f]$, wobei
neben der explizit angezeigten Definition $f(a_i) = b_i$
implizit (neben $f(\overline{a_i}) = \overline{b_i}$) auch $f(\alpha) = \alpha$ für alle $\alpha \in \mathsf{Act} \setminus (A \cup \overline{A})$ gilt.

2.8 Definition (Semantik)
$(\mathcal{P}, \mathsf{Act}, \mathsf{Tran})$ wird wie folgt bestimmt:

- $\mathcal{P}$ ist die Menge der CCS-Prozesse gemäß Definition 2.6.
- Act ist die Menge von Aktionen gemäß Zusatz 2.5.
- Tran ist definiert durch die folgenden acht Ableitungs-Regeln:

$$\text{ACT} \quad \frac{}{\alpha.P \xrightarrow{\alpha} P}$$

$$\text{SUM}_k \quad \frac{P_k \xrightarrow{\alpha} P'_k}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'_k} \quad \text{für } k \in I$$

$$\text{CON} \quad \frac{P \xrightarrow{\alpha} P'}{\mathbf{K} \xrightarrow{\alpha} P'} \quad \text{für } \mathbf{K} \stackrel{\text{def}}{=} P$$

$$\text{COM1} \quad \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\text{COM2} \quad \frac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'}$$

$$\text{COM3} \quad \frac{P \xrightarrow{a} P' \qquad Q \xrightarrow{\bar{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\text{RES} \quad \frac{P \xrightarrow{\alpha} P'}{P \setminus A \xrightarrow{\alpha} P' \setminus A} \quad \text{für } \{\alpha, \overline{\alpha}\} \cap A = \emptyset$$

$$\text{REL} \quad \frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$$

2.9 Definition (VP-CCS Syntax)
Die Menge $\mathcal{P}^{VP}$ der VP-CCS-Ausdrücke ist gegegeben durch:

$$P ::= \sum_{i\in I} P_i \mid P \mid P \mid P[f] \mid P \setminus A \\ \mid \mathbf{K}(e_1,\ldots,e_n) \mid \overline{a}(e).P \mid a(x).P \mid \textbf{if } \phi \textbf{ then } P \textbf{ else } P$$

- $x \in \mathcal{X}$ ist eine *Variable*;
- $e, e_1 \ldots, e_n \in \mathcal{E}$ sind *Ausdrücke* (auch über Variablen) zur Berechnung von *Datenelementen*;
- $v \in \mathcal{V} \subseteq \mathcal{E}$ ist die Menge von berechneten Datenelementen bzw. *Werten*;
- $\phi \in \mathcal{B} \subseteq \mathcal{E}$ sind *bool'sche Ausdrücke*, mit $\{ \text{true}, \text{false} \} \subseteq \mathcal{V}$;
- $\text{eval} : \mathcal{E} \to \mathcal{V}$ ist eine terminierend berechenbare *Auswertefunktion*;
- für jede konkret benutzte Konstante **K** muss eine Definition der Form
$$\mathbf{K}(x_1,\ldots,x_n) \stackrel{\text{def}}{=} P$$
angegeben werden, die alle Variablen in P auflistet.
- die Mengen $\mathcal{X}$, $\mathcal{V}$, und Act sind paarweise disjunkt;
- alle weiteren Komponenten wie in Definition 2.6.

2.10 Definition (Substitution) Sei $X \triangleq \{ x_1,\ldots,x_n \} \subseteq \mathcal{X}$ mit $n \geqslant 1$ eine Menge paarweise verschiedener Variablen. Die Schreibweise

$$P\{ v_1/x_1,\ldots,v_n/x_n \}$$

mit $\{ v_1,\ldots,v_n \} \subseteq \mathcal{V}$ entspricht einer Ersetzungsfunktion auf dem CCS-Ausdruck P — als Meta-Operation außerhalb des Kalküls[2] — die alle freien Vorkommen von x_i in P konsistent durch das jeweilige v_i ersetzt. .

2.11 Definition (VP-CCS Semantik)
$\left(\mathcal{P}^{VP}, \text{Act}^{VP}, \text{Tran}^{VP}\right)$ wird bestimmt wie folgt:

- $\mathcal{P}^{VP}$ ist die Menge der VP-CCS-Prozesse gemäß Definition 2.9.
- Act^{VP} ist die Menge von Aktionen gemäß der Form
$$\alpha ::= \tau \mid \overline{a}(v) \mid a(v)$$
mit $\overline{a(v)} = \overline{a}(v)$ und $\overline{\overline{a}(v)} = a(v)$.
- Tran^{VP} ist definiert durch die folgenden Ableitungs-Regeln:

[2] Im Gegensatz zur Umbenennungsfunktion, die Teil der Syntax des Kalküls ist.

$$\text{OUT} \quad \frac{}{\overline{a}(e).P \xrightarrow{\overline{a}(v)} P} \quad \text{für } \mathrm{eval}(e) = v$$

$$\text{INP} \quad \frac{}{a(x).P \xrightarrow{a(v)} P\{v/x\}} \quad \text{für } v \in \mathcal{V}$$

$$\text{COM} \quad \frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\overline{\alpha}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \quad \text{für } \alpha \neq \tau$$

$$\text{CON} \quad \frac{P\{v_1/x_1, \cdots, v_n/x_n\} \xrightarrow{\alpha} P'}{\mathbf{K}(e_1, \ldots, e_n) \xrightarrow{\alpha} P'} \quad \begin{array}{l}\text{für } \mathbf{K}(x_1, \ldots, x_n) \stackrel{\text{def}}{=} P \\ \text{und } v_i = \mathrm{eval}(e_i) \text{ für alle } 1 \leqslant i \leqslant n\end{array}$$

$$\text{TRUE} \quad \frac{P \xrightarrow{\alpha} P'}{\textbf{if } \phi \textbf{ then } P \textbf{ else } Q \xrightarrow{\alpha} P'} \quad \text{für } \mathrm{eval}(\phi) = \mathrm{true}$$

$$\text{FALSE} \quad \frac{Q \xrightarrow{\alpha} Q'}{\textbf{if } \phi \textbf{ then } P \textbf{ else } Q \xrightarrow{\alpha} Q'} \quad \text{für } \mathrm{eval}(\phi) = \mathrm{false}$$

$$\text{SUM}_k \quad \frac{P_k \xrightarrow{\alpha} P'_k}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'_k} \quad \text{für } k \in I$$

$$\text{COM1} \quad \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\text{COM2} \quad \frac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'}$$

$$\text{RES} \quad \frac{P \xrightarrow{\alpha} P'}{P \setminus A \xrightarrow{\alpha} P' \setminus A} \quad \text{falls der Kanalname von } \alpha \text{ nicht in } A \text{ vorkommt.}$$

$$\text{REL} \quad \frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]} \quad \text{wobei } f(\alpha) \text{ nur auf dem Kanalnamen von } \alpha \text{ operiert.}$$

3 Verhaltensäquivalenzen

3.1 Definition (Äquivalenzrelation; Definition 3.1 in [AILS07])
Sei X eine Menge.
Sei $\mathcal{R} \subseteq X \times X$ eine binäre homogene Relation über X.
Wir schreiben auch $x \mathcal{R} y$, falls $(x, y) \in \mathcal{R}$.
$\mathcal{R}$ heißt *Äquivalenz[relation]*,
falls folgende Eigenschaften erfüllt sind:

- $\mathcal{R}$ ist *reflexiv*, d.h.
  für alle $x \in X$ gilt:
  $(x, x) \in \mathcal{R}$;
- $\mathcal{R}$ ist *symmetrisch*, d.h.
  für alle $x, y \in X$ gilt:
  wenn $(x, y) \in \mathcal{R}$, dann auch $(y, x) \in \mathcal{R}$;
- $\mathcal{R}$ ist *transitiv*, d.h.
  für alle $x, y, z \in X$ gilt:
  wenn $(x, y) \in \mathcal{R}$ und $(y, z) \in \mathcal{R}$, dann auch $(x, z) \in \mathcal{R}$.

3.2 Definition (Traces)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Seien

- $k \in \mathbb{N}$
- $s_0, \ldots, s_k \in \mathsf{Proc}$
- $\alpha_1, \ldots, \alpha_k \in \mathsf{Act}$

so daß für alle $1 \leqslant j \leqslant k$ eine Transition $s_{j-1} \xrightarrow{\alpha_j} s_j$ existiert.
Dann heißt $\alpha_1 \cdot \ldots \cdot \alpha_k \in \mathsf{Act}^*$ eine *Spur* bzw. ein *Trace* in von s_0 in T.

$$\mathrm{Traces}(s) \triangleq \{\, w \in \mathsf{Act}^* \mid w \text{ ist Trace von } s \text{ in } T \,\}$$

3.3 Definition (Trace-Äquivalenz)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Seien $s_1, s_2 \in \mathsf{Proc}$.
Dann heißen s_1 und s_2 *Trace-äquivalent*, wenn:

$$\mathrm{Traces}(s_1) = \mathrm{Traces}(s_2)$$

3.4 Definition (Starke Simulation) [3]
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} \subseteq \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *(starke) Simulation*, wenn gilt:
für alle $(s_1, s_2) \in \mathcal{R}$,
für alle $\alpha \in \mathsf{Act}$,
für alle $s_1' \in \mathsf{Proc}$
wenn $s_1 \xrightarrow{\alpha} s_1'$,
dann gibt es s_2' mit $s_2 \xrightarrow{\alpha} s_2'$ mit $(s_1', s_2') \in \mathcal{R}$.
Seien $s_1, s_2 \in \mathsf{Proc}$.

- s_1 *wird von* s_2 *(stark) simuliert*, geschrieben $s_1 \lesssim s_2$,
  falls es eine (starke) Simulation $\mathcal{R}$ gibt mit $(s_1, s_2) \in \mathcal{R}$.

- s_1 *und* s_2 *simulieren sich (stark) gegenseitig*, geschrieben $s_1 \simeq s_2$,
  falls es zwei (starke) Simulationen $\mathcal{R}_1$ und $\mathcal{R}_2$ gibt mit

$$(s_1, s_2) \in \mathcal{R}_1 \text{ und } (s_2, s_1) \in \mathcal{R}_2 \,.$$

3.5 Definition (Alternative Definition für Bisimulation)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} \subseteq \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *(starke) Bisimulation*,
wenn sowohl $\mathcal{R}$ als auch $\mathcal{R}^{-1}$ eine (starke) Simulation ist.

3.6 Definition (Bisimulation; Definition 3.2 in [AILS07])
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} \subseteq \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *(starke) Bisimulation*, wenn gilt:

1. für alle $(s_1, s_2) \in \mathcal{R}$,
   für alle $\alpha \in \mathsf{Act}$,
   für alle $s_1' \in \mathsf{Proc}$
   wenn $s_1 \xrightarrow{\alpha} s_1'$,
   dann gibt es s_2' mit $s_2 \xrightarrow{\alpha} s_2'$ mit $(s_1', s_2') \in \mathcal{R}$.

2. für alle $(s_1, s_2) \in \mathcal{R}$,
   für alle $\alpha \in \mathsf{Act}$,
   für alle $s_2' \in \mathsf{Proc}$
   wenn $s_2 \xrightarrow{\alpha} s_2'$,
   dann gibt es s_1' mit $s_1 \xrightarrow{\alpha} s_1'$ mit $(s_1', s_2') \in \mathcal{R}$.

Seien $s_1, s_2 \in \mathsf{Proc}$.

- s_1 *und* s_2 *sind (stark) bisimilar*, geschrieben $s_1 \sim s_2$,
  falls es eine (starke) Bisimulation $\mathcal{R}$ gibt mit $(s_1, s_2) \in \mathcal{R}$.

Die Relation $\sim$ heißt *(starke) Bisimulationsäquivalenz* bzw. *(starke) Bisimilarität*.

[3] Siehe auch Exercise 3.17 [AILS07]

3.7 Theorem (Theorem 3.1 in [AILS07])
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$ ein beliebiges LTS.
Dann gilt für die Relation $\sim$ auf T:

1. $\sim$ ist eine Äquivalenz.
2. $\sim$ ist die größte (starke) Bisimulation.[4]
3. $\sim$ erfüllt die (starke) Bisimulationsbedingung, d.h.[5]
   $s_1 \sim s_2$ genau dann, wenn für alle $\alpha \in \mathsf{Act}$ gilt:
   - wenn $s_1 \xrightarrow{\alpha} s_1'$, dann gibt es $s_2 \xrightarrow{\alpha} s_2'$ mit $s_1' \sim s_2'$;
   - wenn $s_2 \xrightarrow{\alpha} s_2'$, dann gibt es $s_1 \xrightarrow{\alpha} s_1'$ mit $s_1' \sim s_2'$.

3.8 Theorem (Theorem 3.2 in [AILS07])
Seien P, Q, R CCS-Prozesse.
Sei $P \sim Q$.
Dann gilt

1. für alle $\alpha \in \mathsf{Act}$: $\alpha.P \sim \alpha.Q$;
2. für alle R : $P + R \sim Q + R$ und $R + P \sim R + Q$;
3. für alle R : $P \mid R \sim Q \mid R$ und $R \mid P \sim R \mid Q$;
4. für alle Umbenennungen f : $P[f] \sim Q[f]$;
5. für alle $A \subseteq \mathcal{A}$: $P \setminus A \sim Q \setminus A$.

[4] Vergleiche mit der Relation $\bigcup\{\, \mathcal{B} \mid \mathcal{B} \text{ ist Bisimulation} \,\}$.
[5] im Vergleich zu Definition 3.6 etwas verkürzt ausgedrückt

3.9 Definition (Schwache Transition; Definition 3.3 in [AILS07])
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Seien $s, s' \in \mathsf{Proc}$ und $\alpha \in \mathsf{Act}$.
Wir schreiben $s \overset{\alpha}{\Rightarrow} s'$, falls

1. $\alpha = \tau$ und $s \ (\overset{\tau}{\rightarrow})^* \ s'$, oder
2. $\alpha \neq \tau$ und $s \ (\overset{\tau}{\rightarrow})^* \ t \overset{\alpha}{\rightarrow} t' \ (\overset{\tau}{\rightarrow})^* \ s'$ für $t, t' \in \mathsf{Proc}$.

3.10 Definition (Schwache Simulation)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} \subseteq \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *schwache Simulation*, wenn gilt:
für alle $(s_1, s_2) \in \mathcal{R}$,
für alle $\alpha \in \mathsf{Act}$,
für alle $s_1' \in \mathsf{Proc}$
wenn $s_1 \overset{\alpha}{\rightarrow} s_1'$,
dann gibt es s_2' mit $s_2 \overset{\alpha}{\Rightarrow} s_2'$ mit $(s_1', s_2') \in \mathcal{R}$.

Seien $s_1, s_2 \in \mathsf{Proc}$.

- s_1 *wird von* s_2 *schwach simuliert*, geschrieben $s_1 \lessapprox s_2$,
  falls es eine schwache Simulation $\mathcal{R}$ gibt mit $(s_1, s_2) \in \mathcal{R}$.
- s_1 *und* s_2 *simulieren sich gegenseitig schwach*, geschrieben $s_1 \approxeq s_2$,
  falls es zwei schwache Simulationen $\mathcal{R}_1$ und $\mathcal{R}_2$ gibt mit

$$(s_1, s_2) \in \mathcal{R}_1 \text{ und } (s_2, s_1) \in \mathcal{R}_2 \ .$$

3.11 Definition (Alternative Definition für schwache Bisimulation)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} \subseteq \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *schwache Bisimulation*,
wenn sowohl $\mathcal{R}$ als auch $\mathcal{R}^{-1}$ eine schwache Simulation ist.

3.12 Definition (Schwache Bisimulation; Definition 3.3 in [AILS07])
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} \subseteq \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *schwache Bisimulation*, wenn gilt:

1. für alle $(s_1, s_2) \in \mathcal{R}$,
   für alle $\alpha \in \mathsf{Act}$,
   für alle $s_1' \in \mathsf{Proc}$
   wenn $s_1 \xrightarrow{\alpha} s_1'$,
   dann gibt es s_2' mit $s_2 \stackrel{\alpha}{\Rightarrow} s_2'$ mit $(s_1', s_2') \in \mathcal{R}$.

2. für alle $(s_1, s_2) \in \mathcal{R}$,
   für alle $\alpha \in \mathsf{Act}$,
   für alle $s_2' \in \mathsf{Proc}$
   wenn $s_2 \xrightarrow{\alpha} s_2'$,
   dann gibt es s_1' mit $s_1 \stackrel{\alpha}{\Rightarrow} s_1'$ mit $(s_1', s_2') \in \mathcal{R}$.

Seien $s_1, s_2 \in \mathsf{Proc}$.

- s_1 *und* s_2 *sind schwach bisimilar*, geschrieben $s_1 \approx s_2$,
  falls es eine schwache Bisimulation $\mathcal{R}$ gibt mit $(s_1, s_2) \in \mathcal{R}$.

Die Relation $\approx$ heißt *schwache Bisimulationsäquivalenz* bzw. *schwache Bisimilarität*; $\approx$ heißt auch *Beobachtungsäquivalenz*.

3.13 Theorem (Theorem 3.3 in [AILS07])
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$ ein beliebiges LTS.
Dann gilt für die Relation $\approx$ auf T:

1. $\approx$ ist eine Äquivalenz.
2. $\approx$ ist die größte schwache Bisimulation.
3. $\approx$ erfüllt die schwache Bisimulationsbedingung, d.h.[6]
   $s_1 \approx s_2$ genau dann, wenn für alle $\alpha \in \mathsf{Act}$ gilt:
   - wenn $s_1 \xrightarrow{\alpha} s_1'$, dann gibt es $s_2 \stackrel{\alpha}{\Rightarrow} s_2'$ mit $s_1' \approx s_2'$;
   - wenn $s_2 \xrightarrow{\alpha} s_2'$, dann gibt es $s_1 \stackrel{\alpha}{\Rightarrow} s_1'$ mit $s_1' \approx s_2'$.

3.14 Theorem (Theorem 3.4 in [AILS07])
Seien P, Q, R CCS-Prozesse.
Sei $P \approx Q$.
Dann gilt

1. für alle $\alpha \in \mathsf{Act}$: $\alpha.P \approx \alpha.Q$;
2. für alle R : $P \mid R \approx Q \mid R$ und $R \mid P \approx R \mid Q$;
3. für alle Umbenennungen f : $P[f] \approx Q[f]$;
4. für alle $A \subseteq \mathcal{A}$: $P \setminus A \approx Q \setminus A$.

[6] im Vergleich zu Definition 3.12 wiederum verkürzt ausgedrückt

4 Fixpunkttheorie

4.1 Definition (Fixpunkt; Teil von Definition 4.4 in [AILS07])
Sei $f : D \to D$ eine Abbildung.
Sei $d \in D$.
Wenn $f(d) = d$, dann heißt d *Fixpunkt von* f.

4.2 Bemerkung Im Rest dieses Dokuments gehen wir implizit immer davon aus, dass D eine nicht-leere Menge ist.

4.3 Definition (Eigenschaften homogener Relationen; Definition 4.1 in [AILS07])

Sei $R : D \times D$ eine beliebige homogene Relation.
Dann heißt R:

reflexiv	falls	für alle $a \in D$:	$a \mathrel{R} a$
antisymmetrisch	falls	für alle $a, b \in D$:	$a \neq b \wedge a \mathrel{R} b \to \neg(b \mathrel{R} a)$
transitiv	falls	für alle $a, b, c \in D$:	$a \mathrel{R} b \wedge b \mathrel{R} c \to a \mathrel{R} c$
linear	falls	für alle $a, b \in D$:	$a \mathrel{R} b \vee b \mathrel{R} a$

- (D, R) heißt *partielle Ordnung*
  bzw. *partiell geordnete Menge* (*poset*),
  falls R reflexiv, antisymmetrisch und transitiv ist.
- Eine partielle Ordnung (D, R) heißt *totale Ordnung*
  bzw. *total geordnete Menge*,
  falls R zudem linear ist.
- Wir benutzen typischerweise $\leqslant$ oder $\sqsubseteq$
  als Metavariable für Ordnungsrelationen.

4.4 Definition (Extremwerte)
Sei $(D, \sqsubseteq)$ eine partielle Ordnung. Sei $A \subseteq D$.

1. (a) $k \in A$ heißt *kleinstes Element von* A, falls gilt:

$$\forall a \in A \;.\; k \sqsubseteq a$$

   (b) $g \in A$ heißt *größtes Element von* A, falls gilt:

$$\forall a \in A \;.\; a \sqsubseteq g$$

   Ein kleinstes Element heißt auch *Minimum von* A,
   ein größtes Element heißt auch *Maximum von* A,
   notiert als $\mathrm{Min}_{\sqsubseteq}(A)$ bzw. $\mathrm{Max}_{\sqsubseteq}(A)$.

2. (a) $u \in A$ heißt *minimales Element in* A, falls gilt:

$$\forall a \in A \;.\; \big(a \neq u \to \neg(a \sqsubseteq u)\big)$$

   (b) $o \in A$ heißt *maximales Element in* A, falls gilt:

$$\forall a \in A \;.\; \big(a \neq o \to \neg(o \sqsubseteq a)\big)$$

4.5 Bemerkung Bezüglich einer partiellen Ordnung $(D, \sqsubseteq)$ existieren Extremwerte nicht notwendigerweise für jede Teilmenge $A \subseteq D$. Minimale und maximale Elemente sind zudem nicht immer eindeutig.

4.6 Definition (Schranken; Definition 4.2 in [AILS07])
Sei $(D, \sqsubseteq)$ eine partielle Ordnung. Sei $A \subseteq D$.

1. $u \in D$ heißt *untere Schranke* von A (bzgl. D), falls gilt:

$$\forall a \in A \ . \ u \sqsubseteq a$$

Eine *größte untere Schranke von* A *(bzgl.* D*)*, geschrieben $\bigsqcap A$,
auch als *Infimum von* A *(bzgl.* D*)* mit $\inf^D(A)$ bezeichnet,
ist definiert als ein größtes Element
in der Menge der unteren Schranken von A (bzgl. D).

2. $o \in D$ heißt *obere Schranke* von A (bzgl. D), falls gilt:

$$\forall a \in A \ . \ a \sqsubseteq o$$

Eine *kleinste obere Schranke von* A *(bzgl.* D*)*, geschrieben $\bigsqcup A$,
auch als *Supremum von* A *(bzgl.* D*)* mit $\sup^D(A)$ bezeichnet,
ist definiert als ein kleinstes Element
in der Menge der oberen Schranken von A (bzgl. D).

4.7 Bemerkung
Sei $(D, \sqsubseteq)$ eine partielle Ordnung.
Dann können sowohl $\bigsqcap$ als auch $\bigsqcup$
als partielle Funktionen des Typs $2^D \to D$ verstanden werden.

4.8 Bemerkung Bezüglich einer partiellen Ordnung $(D, \sqsubseteq)$ existieren Schranken nicht notwendigerweise für jede Teilmenge $A \subseteq D$.

4.9 Bemerkung Schranken werden aus D gewählt. Sie liegen nicht notwendigerweise innerhalb der Menge A, die sie beschränken.

4.10 Lemma
Sei $(D, \sqsubseteq)$ eine totale Ordnung.
Sei $A \subseteq D$ endliche Teilmenge (also $\#(A) \in \mathbb{N}^+$).
Dann sind $\inf^D(A) \in A$ und $\sup^D(A) \in A$.
Somit sind $\mathrm{Min}_{\sqsubseteq}(A)$ und $\mathrm{Max}_{\sqsubseteq}(A)$ definiert.

4.11 Definition (Verband)
Sei $(D, \sqsubseteq)$ eine partielle Ordnung.
Falls für alle $\{ d_1, d_2 \} \subseteq D$
sowohl $\bigsqcap\{ d_1, d_2 \}$ als auch $\bigsqcup\{ d_1, d_2 \}$ existieren,
dann heißt $(D, \sqsubseteq)$ *Verband*.

4.12 Definition (Vollständiger Verband; Definition 4.3 in [AILS07])
Sei $(D, \sqsubseteq)$ eine partielle Ordnung.
Falls für alle $A \subseteq D$
sowohl $\bigsqcap A$ als auch $\bigsqcup A$ existieren,
dann heißt $(D, \sqsubseteq)$ *vollständiger Verband*.
Dann bezeichnen $\bot \triangleq \bigsqcap D$ und $\top \triangleq \bigsqcup D$
jeweils das kleinste bzw. größte Element von D.

4.13 Lemma Jeder endliche Verband ist vollständig.

4.14 Definition (Monotonie; Teil von Definition 4.4 in [AILS07])
Sei $(D, \sqsubseteq)$ partielle Ordnung.
Sei $f : D \to D$ eine Abbildung.
Dann ist f *monoton bzgl.* $\sqsubseteq$, falls gilt:

für alle $d, d' \in D$,
wenn $d \sqsubseteq d'$, dann auch $f(d) \sqsubseteq f(d')$.

4.15 Definition (Definition 4.5 in [AILS07])
Sei D eine Menge. Sei $f : D \to D$.
Für $n \in \mathbb{N}$ ist f^n induktiv definiert durch:

$$\begin{aligned} f^0(d) &\triangleq d \\ f^{n+1}(d) &\triangleq f(f^n(d)) \end{aligned}$$

4.16 Definition („Fast"-Fixpunkte)
Sei $(D, \sqsubseteq)$ eine partielle Ordnung.
Sei $f : D \to D$ eine Abbildung.
Sei $d \in D$.
Wenn $f(d) \sqsubseteq d$, dann heißt d *Prä-Fixpunkt von* f.
Wenn $d \sqsubseteq f(d)$, dann heißt d *Post-Fixpunkt von* f.

4.17 Theorem (Tarski's Fixpunkttheorem; Theorem 4.1 in [AILS07])
Sei $(D, \sqsubseteq)$ ein vollständiger Verband.
Sei $f : D \to D$ monoton.
Dann gibt es einen größten und einen kleinsten Fixpunkt von f,
bestimmt durch:

$$z_{\max} \triangleq \bigsqcup \{ x \in D \mid x \sqsubseteq f(x) \}$$

$$z_{\min} \triangleq \bigsqcap \{ x \in D \mid f(x) \sqsubseteq x \}$$

4.18 Theorem (Theorem 4.2 in [AILS07])
Sei $(D, \sqsubseteq)$ ein endlicher vollständiger Verband.
Sei $f : D \to D$ monoton.
Dann wird der größte Fixpunkt von f berechnet durch:

$$z_{\max} \triangleq f^M(\top)$$

für ein $M \in \mathbb{N}$.
Ebenso wird der kleinste Fixpunkt von f berechnet durch:

$$z_{\min} \triangleq f^m(\bot)$$

für ein $m \in \mathbb{N}$.

4.19 Definition
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Seien $p \in \mathsf{Proc}$ und $\alpha \in \mathsf{Act}$.
Dann heißen die Elemente von

$$\mathrm{Der}(p, \alpha) \triangleq \left\{ p' \in \mathsf{Proc} \mid p \xrightarrow{\alpha} p' \right\}$$

*α-Nachfolger (*bzw. *α-Derivatives) von* p.

4.20 Definition (Alternative Definition für Bisimulation)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} : \mathsf{Proc} \times \mathsf{Proc}$.
$\mathcal{R}$ heißt *(starke) Bisimulation*, wenn gilt:

1. $\forall (p, q) \in \mathcal{R}. \forall \alpha \in \mathsf{Act}. \forall p' \in \mathrm{Der}(p, \alpha).$
   $\exists q' \in \mathrm{Der}(q, \alpha). (p', q') \in \mathcal{R}$

2. $\forall (p, q) \in \mathcal{R}. \forall \alpha \in \mathsf{Act}. \forall q' \in \mathrm{Der}(q, \alpha).$
   $\exists p' \in \mathrm{Der}(p, \alpha). (p', q') \in \mathcal{R}$

4.21 Definition (1-Schritt-Bisimulation)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{R} : \mathsf{Proc} \times \mathsf{Proc}$.

„vorwärts“ Die Relation $\overset{1}{\sim}_{\mathcal{R}} : \mathsf{Proc} \times \mathsf{Proc}$ ist gegeben durch:

- $p \overset{1}{\sim}_{\mathcal{R}} q$ gdw.
  1. $\forall \alpha \in \mathsf{Act}. \forall p' \in \mathrm{Der}(p, \alpha). \exists q' \in \mathrm{Der}(q, \alpha). (p', q') \in \mathcal{R}$
  2. $\forall \alpha \in \mathsf{Act}. \forall q' \in \mathrm{Der}(q, \alpha). \exists p' \in \mathrm{Der}(p, \alpha). (p', q') \in \mathcal{R}$

„rückwärts“ Die Funktion $\mathcal{F} : 2^{\mathsf{Proc}\times\mathsf{Proc}} \to 2^{\mathsf{Proc}\times\mathsf{Proc}}$ ist gegeben durch:

- $$\begin{aligned} \mathcal{F}(\mathcal{R}) &\triangleq \overset{1}{\sim}_{\mathcal{R}} \\ &= \left\{ (p,\ q) \in \mathsf{Proc} \times \mathsf{Proc} \mid p \overset{1}{\sim}_{\mathcal{R}} q \right\} \end{aligned}$$

4.22 Lemma
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$. Dann gilt:

1. $\mathcal{F}$ ist monoton bzgl. $\subseteq$.
2. $\mathcal{R}$ ist (starke) Bisimulation gdw. $\mathcal{R} \subseteq \mathcal{F}(\mathcal{R})$.

4.23 Bemerkung
Mit der Beobachtung, dass

$$\begin{aligned} \sim &= \bigcup \left\{ \mathcal{R} \in 2^{(\mathsf{Proc}\times\mathsf{Proc})} \mid \mathcal{R} \text{ ist (starke) Bisimulation} \right\} \\ &= \bigcup \left\{ \mathcal{R} \in 2^{(\mathsf{Proc}\times\mathsf{Proc})} \mid \mathcal{R} \subseteq \mathcal{F}(\mathcal{R}) \right\} \end{aligned}$$

folgt mit Theorem 4.17,
dass $\sim$ genau dem größten Fixpunkt von $\mathcal{F}$ entspricht.

Demzufolge kann $\sim$ für endliche LTS durch Theorem 4.18 als

$$\sim \ = \ \mathcal{F}^{M}(\mathsf{Proc} \times \mathsf{Proc})$$

im Verband $\left(2^{\mathsf{Proc}\times\mathsf{Proc}}, \subseteq\right)$ immer algorithmisch berechnet werden.
Mit $\sim_i \triangleq \mathcal{F}^i(\mathsf{Proc} \times \mathsf{Proc})$ für alle $i \in \mathbb{N}$ gilt zudem[7]:

1. $\sim_i$ ist eine Äquivalenz;
2. $\sim_{i+1} \subseteq \sim_i$.

[7] Vergleiche mit Exercise 4.14 in [AILS07].

5 Hennessy-Milner-Logik

5.1 Definition (HML Syntax; Definition 5.1 in [AILS07])
Die Menge $\mathcal{M}$ der HML-Formeln über Act ist gegegeben durch:

$$F ::= tt \mid ff \mid F \wedge F \mid F \vee F \mid \langle \alpha \rangle F \mid [\alpha] F$$

wobei $\alpha \in \mathsf{Act}$.
Wir definieren dazu, für $A = \{ \alpha_1, \ldots, \alpha_n \}$ und F beliebig:

- $\langle A \rangle F \triangleq \langle \alpha_1 \rangle F \vee \cdots \vee \langle \alpha_n \rangle F$
- $[A] F \triangleq [\alpha_1] F \wedge \cdots \wedge [\alpha_n] F$

mit $\langle \emptyset \rangle F = ff$ und $[\emptyset] F = tt$.
Es gilt Operatorenvorrang in absteigender Folge:

- Gleichberechtigt die Modaloperatoren $\langle \alpha \rangle$ und $[\alpha]$ und
- gleichberechtigt die Disjunktion $F \vee F$ und Konjunktion $F \wedge F$.

5.2 Definition (Modaltiefe; S. 126 in [AILS07])
Die *Modaltiefe* $\mathrm{md}(F)$ einer HML-Formel F bezeichnet die maximale Verschachtelungstiefe von Modaloperatoren in F, gegeben durch die Funktion $\mathrm{md} : \mathcal{M} \to \mathbb{N}$ mit:

$$\begin{aligned}
\mathrm{md}(tt) &\triangleq 0 \\
\mathrm{md}(ff) &\triangleq 0 \\
\mathrm{md}(F_1 \wedge F_2) &\triangleq \max\{ \mathrm{md}(F_1), \mathrm{md}(F_2) \} \\
\mathrm{md}(F_1 \vee F_2) &\triangleq \max\{ \mathrm{md}(F_1), \mathrm{md}(F_2) \} \\
\mathrm{md}(\langle \alpha \rangle F) &\triangleq 1 + \mathrm{md}(F) \\
\mathrm{md}([\alpha] F) &\triangleq 1 + \mathrm{md}(F)
\end{aligned}$$

5.3 Definition („Model-Checking-Semantik“ von HML; S. 96 in [AILS07])

Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}$ definiert über Act.
Die Relation $\models : (\mathsf{Proc}, \mathcal{M})$ ist induktiv definiert durch:

$$
\begin{array}{lcll}
p & \models & \mathit{tt} & \text{für alle } p \in \mathsf{Proc} \\
p & \models & \mathit{ff} & \text{für kein } p \in \mathsf{Proc} \\
p & \models & F \wedge G & \text{falls } p \models F \text{ und } p \models G \\
p & \models & F \vee G & \text{falls } p \models F \text{ oder } p \models G \\
p & \models & \langle\alpha\rangle F & \text{falls es } p' \in \mathrm{Der}(p, \alpha) \text{ gibt mit } p' \models F \\
p & \models & [\alpha] F & \text{falls für alle } p' \in \mathrm{Der}(p, \alpha) \text{ gilt } p' \models F
\end{array}
$$

5.4 Definition (Denotationelle HML-Semantik; Def. 5.2 in [AILS07])
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}$ definiert über Act.
Die Funktion $[\![\cdot]\!] : \mathcal{M} \to 2^{\mathsf{Proc}}$ ist induktiv definiert durch:

$$
\begin{array}{rcl}
[\![\mathit{tt}]\!] & \triangleq & \mathsf{Proc} \\
[\![\mathit{ff}]\!] & \triangleq & \emptyset \\
[\![F \wedge G]\!] & \triangleq & [\![F]\!] \cap [\![G]\!] \\
[\![F \vee G]\!] & \triangleq & [\![F]\!] \cup [\![G]\!] \\
[\![\langle\alpha\rangle F]\!] & \triangleq & \langle\cdot\alpha\cdot\rangle [\![F]\!] \\
[\![[\alpha] F]\!] & \triangleq & [\cdot\alpha\cdot] [\![F]\!]
\end{array}
$$

wobei die Operatoren $\langle\cdot\alpha\cdot\rangle, [\cdot\alpha\cdot] : 2^{\mathsf{Proc}} \to 2^{\mathsf{Proc}}$ gegeben sind durch:

$$
\begin{array}{rcl}
\langle\cdot\alpha\cdot\rangle S & \triangleq & \{\, p \in \mathsf{Proc} \mid \exists p' \in \mathrm{Der}(p,\ \alpha)\ .\ p' \in S \,\} \\
[\cdot\alpha\cdot] S & \triangleq & \{\, p \in \mathsf{Proc} \mid \forall p' \in \mathrm{Der}(p,\ \alpha)\ .\ p' \in S \,\}
\end{array}
$$

5.5 Lemma Es gilt:

$$p \models F \quad \text{genau dann, wenn} \quad p \in [\![F]\!]$$

5.6 Definition
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Als Abkürzungen im Kontext von Definition 5.4
benutzen wir $\langle\cdot\mathsf{Act}\cdot\rangle, [\cdot\mathsf{Act}\cdot] : 2^{\mathsf{Proc}} \to 2^{\mathsf{Proc}}$ mit:

$$
\begin{array}{rcl}
\langle\cdot\mathsf{Act}\cdot\rangle S & \triangleq & \bigcup\limits_{\alpha \in \mathsf{Act}} \langle\cdot\alpha\cdot\rangle S \\
[\cdot\mathsf{Act}\cdot] S & \triangleq & \bigcap\limits_{\alpha \in \mathsf{Act}} [\cdot\alpha\cdot] S
\end{array}
$$

5.7 Definition (Negation)
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}$ definiert über Act.
Die Funktion $(\cdot)^c : \mathcal{M} \to \mathcal{M}$ ist gegeben durch:

$$\begin{aligned}
(tt)^c &\triangleq ff \\
(ff)^c &\triangleq tt \\
(\mathsf{F} \wedge \mathsf{G})^c &\triangleq (\mathsf{F})^c \vee (\mathsf{G})^c \\
(\mathsf{F} \vee \mathsf{G})^c &\triangleq (\mathsf{F})^c \wedge (\mathsf{G})^c \\
(\langle\alpha\rangle \mathsf{F})^c &\triangleq [\alpha]\,(\mathsf{F})^c \\
([\alpha]\mathsf{F})^c &\triangleq \langle\alpha\rangle\,(\mathsf{F})^c
\end{aligned}$$

Es gilt für alle $\mathsf{F} \in \mathcal{M}$:

1. $[\![\,(\mathsf{F})^c\,]\!] = \mathsf{Proc} \setminus [\![\,\mathsf{F}\,]\!]$;
2. $\big((\mathsf{F})^c\big)^c = \mathsf{F}$.

5.8 Definition
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}$ definiert über Act.
Die Funktion $[\![\,\cdot\,]\!] : \mathsf{Proc} \to 2^{\mathcal{M}}$ ist gegeben durch[8]:

$$[\![\,p\,]\!] \triangleq \{\,\mathsf{F} \in \mathcal{M} \mid p \models \mathsf{F}\,\}$$

Die Funktion $[\![\,\cdot\,]\!]^{\leqslant i} : \mathsf{Proc} \to 2^{\mathcal{M}}$ ist gegeben durch:

$$[\![\,p\,]\!]^{\leqslant i} \triangleq \{\,\mathsf{F} \in \mathcal{M} \mid p \models \mathsf{F} \wedge \mathrm{md}(\mathsf{F}) \leqslant i\,\}$$

5.9 Definition (Bild-Endlichkeit; Definition 5.3 in [AILS07])
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.

- $p \in \mathsf{Proc}$ heißt *Bild-endlich*,
  falls für alle $\alpha \in \mathsf{Act}$: $\#(\mathrm{Der}(p, \alpha)) \in \mathbb{N}$.
- T heißt *Bild-endlich*,
  falls alle $p \in \mathsf{Proc}$ Bild-endlich sind.

5.10 Theorem (Hennessy-Milner-Theorem; Theorem 5.1 in [AILS07])
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$ Bild-endlich.
Seien $p, q \in \mathsf{Proc}$.
Dann gilt:

$p \sim q$ genau dann, wenn $[\![\,p\,]\!] = [\![\,q\,]\!]$

$p \sim_i q$ genau dann, wenn $[\![\,p\,]\!]^{\leqslant i} = [\![\,q\,]\!]^{\leqslant i}$

[8] per Overloading, siehe Def. 5.4.

6 HML mit Rekursion

6.1 Definition
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, und $\alpha \in \mathsf{Act}$.

$$\begin{aligned}
\langle\alpha\rangle^{n+1}F &\triangleq \langle\alpha\rangle\,(\langle\alpha\rangle^{n}F)\\
\langle\alpha\rangle^{0}F &\triangleq F
\end{aligned}$$

$$\begin{aligned}
[\alpha]^{n+1}F &\triangleq [\alpha]\,([\alpha]^{n}F)\\
[\alpha]^{0}F &\triangleq F
\end{aligned}$$

$$\begin{aligned}
\langle\mathsf{Act}\rangle^{n+1}F &\triangleq \langle\mathsf{Act}\rangle\,(\langle\mathsf{Act}\rangle^{n}F)\\
\langle\mathsf{Act}\rangle^{0}F &\triangleq F
\end{aligned}$$

$$\begin{aligned}
[\mathsf{Act}]^{n+1}F &\triangleq [\mathsf{Act}]\,([\mathsf{Act}]^{n}F)\\
[\mathsf{Act}]^{0}F &\triangleq F
\end{aligned}$$

6.2 Definition (1HML Syntax)
Die Menge $\mathcal{M}_{\{X\}}$ der 1HML-Formeln über Act mit einer Variablen X, ist gegegeben durch:

$$F \;::=\; X \;\big|\; tt \;\big|\; ff \;\big|\; F \wedge F \;\big|\; F \vee F \;\big|\; \langle\alpha\rangle F \;\big|\; [\alpha]F$$

wobei $\alpha \in \mathsf{Act}$.

6.3 Bemerkung (1HML Rekursion: Syntaktisch)
Rekursives Verhalten wird durch Gleichungen der Form

$$X \;=\; F_X$$

angegeben, wobei F_X eine 1HML-Formel bezeichnet, in der X vorkommt.

Unter der Annahme, dass Fixpunkte solcher Gleichungen existieren, spezifizieren wir die Art des gewünschten Fixpunktes durch

$$X \overset{\max}{=} F_X \qquad \text{und} \qquad X \overset{\min}{=} F_X.$$

6.4 Bemerkung (1HML Rekursion: Semantisch)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}_{\{X\}}$ definiert über Act.
Wenn X durch eine Gleichung der Form

$$X \;=\; F_X$$

spezifiziert ist, dann entspricht dies semantisch (in 2^{Proc}) der Gleichung

$$[\![X]\!] \;=\; \mathcal{O}_{F_X}([\![X]\!])$$

bezüglich einer Funktion

$$\mathcal{O}_{F_X} : 2^{\mathsf{Proc}} \to 2^{\mathsf{Proc}}$$

(als Instanz von $\mathcal{O} : \mathcal{M}_{\{X\}} \to (2^{\mathsf{Proc}} \to 2^{\mathsf{Proc}})$).

Wenn $\mathcal{O}_{F_X}$ monoton bzgl. $\left(2^{\mathsf{Proc}}, \subseteq\right)$ ist, dann
gibt es nach Tarski sowohl eine größte als auch eine kleinste Lösung:

$$\mathrm{FIX}\,\mathcal{O}_{F_X} \;\triangleq\; \bigcup \left\{\, S \in 2^{\mathsf{Proc}} \mid S \subseteq \mathcal{O}_{F_X}(S) \,\right\}$$

$$\mathrm{fix}\,\mathcal{O}_{F_X} \;\triangleq\; \bigcap \left\{\, S \in 2^{\mathsf{Proc}} \mid \mathcal{O}_{F_X}(S) \subseteq S \,\right\}$$

6.5 Definition (Semantik von 1HML-Rekursion; Definition 6.1 in [AILS07])
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}_{\{X\}}$ definiert über Act.
Sei $F \in \mathcal{M}_{\{X\}}$.
Die Funktion $\mathcal{O}_F : 2^{\mathsf{Proc}} \to 2^{\mathsf{Proc}}$ ist induktiv definiert durch:

$$\begin{aligned}
\mathcal{O}_X(S) &\triangleq S \\
\mathcal{O}_{tt}(S) &\triangleq \mathsf{Proc} \\
\mathcal{O}_{ff}(S) &\triangleq \emptyset \\
\mathcal{O}_{F \wedge G}(S) &\triangleq \mathcal{O}_F(S) \cap \mathcal{O}_G(S) \\
\mathcal{O}_{F \vee G}(S) &\triangleq \mathcal{O}_F(S) \cup \mathcal{O}_G(S) \\
\mathcal{O}_{\langle\alpha\rangle F}(S) &\triangleq \langle\cdot\alpha\cdot\rangle \mathcal{O}_F(S) \\
\mathcal{O}_{[\alpha] F}(S) &\triangleq [\cdot\alpha\cdot] \mathcal{O}_F(S)
\end{aligned}$$

6.6 Lemma
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}_{\{X\}}$ definiert über Act.
Dann gilt für alle $F \in \mathcal{M}_{\{X\}}$, dass $\mathcal{O}_F$ monoton bzgl. $\left(2^{\mathsf{Proc}}, \subseteq\right)$ ist.

6.7 Theorem (Theorem 6.1 in [AILS07])
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}_{\{X\}}$ definiert über Act.
Sei Proc endlich, d.h., $\#(\mathsf{Proc}) \in \mathbb{N}$.
Sei $F_X \in \mathcal{M}_{\{X\}}$.
Dann gilt (mit Theorem 4.18) für eine Gleichung $X = F_X$, dass

$$\begin{aligned}
\mathrm{FIX}\,\mathcal{O}_{F_X} &= (\mathcal{O}_{F_X})^{M}(\mathsf{Proc}) \\
\mathrm{fix}\,\mathcal{O}_{F_X} &= (\mathcal{O}_{F_X})^{m}(\emptyset)
\end{aligned}$$

für $m, M \in \mathbb{N}$.

6.8 Definition (Denotationelle Semantik von 1HML)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}_{\{X\}}$ definiert über Act.
Die Funktion $[\![\cdot]\!] : \mathcal{M}_{\{X\}} \to 2^{\mathsf{Proc}}$ ist induktiv definiert durch:

$$
\begin{aligned}
[\![tt]\!] &\triangleq \mathsf{Proc} \\
[\![ff]\!] &\triangleq \emptyset \\
[\![F \wedge G]\!] &\triangleq [\![F]\!] \cap [\![G]\!] \\
[\![F \vee G]\!] &\triangleq [\![F]\!] \cup [\![G]\!] \\
[\![\langle \alpha \rangle F]\!] &\triangleq \langle \cdot \alpha \cdot \rangle [\![F]\!] \\
[\![[\alpha] F]\!] &\triangleq [\cdot \alpha \cdot] [\![F]\!] \\
[\![X]\!] &\triangleq \begin{cases} \mathrm{FIX}\, \mathcal{O}_{F_X} & \text{falls } X \overset{\max}{=} F_X \\ \mathrm{fix}\, \mathcal{O}_{F_X} & \text{falls } X \overset{\min}{=} F_X \end{cases}
\end{aligned}
$$

6.9 Definition
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Es gibt zwei Arten *vollständiger Transitionssequenzen*:

- endliche Sequenzen
  $p_0 \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} p_n$ mit $p_n \not\to$, und
- unendliche Sequenzen
  $p_0 \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} p_n \xrightarrow{\alpha_{n+1}} \ldots$

6.10 Definition
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$, mit $\mathcal{M}_{\{X\}}$ definiert über Act.
Häufig benutzte Formelschemata sind gegeben wie folgt:

$$
\begin{aligned}
\mathit{Inv}(F) &: X \overset{\max}{=} F \wedge [\mathsf{Act}]X \\
\mathit{Pos}(F) &: X \overset{\min}{=} F \vee \langle \mathsf{Act} \rangle X \\
\mathit{Safe}(F) &: X \overset{\max}{=} F \wedge ([\mathsf{Act}]ff \vee \langle \mathsf{Act} \rangle X) \\
\mathit{Even}(F) &: X \overset{\min}{=} F \vee (\langle \mathsf{Act} \rangle tt \wedge [\mathsf{Act}]X) \\
G\, \mathcal{U}^w\, F &: X \overset{\max}{=} F \vee (G \wedge [\mathsf{Act}]X) \\
G\, \mathcal{U}^s\, F &: X \overset{\min}{=} F \vee (G \wedge \langle \mathsf{Act} \rangle tt \wedge [\mathsf{Act}]X)
\end{aligned}
$$

6.11 Definition (HML Syntax, mit mehreren (Rekursions-)Variablen)
Sei $\mathcal{X} = \{ X_1, \ldots, X_n \}$ eine nicht-leere Menge von Variablen.
Die Menge $\mathcal{M}_{\mathcal{X}}$ der HML-Formeln über Act mit Variablen in $\mathcal{X}$,
ist gegegeben durch:

$$F \ ::= \ X \mid tt \mid ff \mid F \wedge F \mid F \vee F \mid \langle \alpha \rangle F \mid [\alpha] F$$

wobei $\alpha \in \mathsf{Act}$ und $X \in \mathcal{X}$.

6.12 Bemerkung (Gleichungssysteme: Syntax)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{X} = \{ X_1, \ldots, X_n \}$ eine nicht-leere Menge von Variablen.
Ein *gegenseitig rekursives Gleichungssystem* hat die Form

$$\begin{aligned} X_1 &= F_{X_1} \\ &\vdots \\ X_n &= F_{X_n} \end{aligned}$$

wobei $F_{X_i} \in \mathcal{M}_{\mathcal{X}}$ für $i \in [1, n]$.
Es wird abkürzend beschrieben durch eine *Deklaration*

$$\begin{aligned} \boldsymbol{F} : \mathcal{X} &\to \mathcal{M}_{\mathcal{X}} \\ X &\mapsto F_X \end{aligned}$$

und wir gehen zunächst davon aus, dass wir
bei allen Einzelgleichungen den gleichen Fixpunkt intendieren.

6.13 Definition (HML Semantik, mit mehreren (Rekursions-)Variablen)
Sei $T = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{X} = \{ X_1, \ldots, X_n \}$ eine nicht-leere Menge von Variablen.
Die Funktion $\mathcal{O}_F : (2^{\mathsf{Proc}})^n \to 2^{\mathsf{Proc}}$ ist induktiv definiert durch:

$$\begin{aligned} \mathcal{O}_{X_i}(S_1, \ldots, S_n) &\triangleq S_i \\ \mathcal{O}_{tt}(S_1, \ldots, S_n) &\triangleq \mathsf{Proc} \\ \mathcal{O}_{ff}(S_1, \ldots, S_n) &\triangleq \emptyset \\ \mathcal{O}_{F \wedge G}(S_1, \ldots, S_n) &\triangleq \mathcal{O}_F(S_1, \ldots, S_n) \cap \mathcal{O}_G(S_1, \ldots, S_n) \\ \mathcal{O}_{F \vee G}(S_1, \ldots, S_n) &\triangleq \mathcal{O}_F(S_1, \ldots, S_n) \cup \mathcal{O}_G(S_1, \ldots, S_n) \\ \mathcal{O}_{\langle \alpha \rangle F}(S_1, \ldots, S_n) &\triangleq \langle \cdot \alpha \cdot \rangle \mathcal{O}_F(S_1, \ldots, S_n) \\ \mathcal{O}_{[\alpha] F}(S_1, \ldots, S_n) &\triangleq [\cdot \alpha \cdot] \mathcal{O}_F(S_1, \ldots, S_n) \end{aligned}$$

6.14 Bemerkung (Gleichungssysteme: Semantik)
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{X} = \{X_1, \ldots, X_n\}$ eine nicht-leere Menge von Variablen.
Das syntaktische Gleichungssystem aus 6.12
entspricht dem semantischen Gleichungssystem

$$\begin{aligned} S_1 &= \mathcal{O}_{F_{X_1}}(S_1, \ldots, S_n) \\ &\vdots \\ S_n &= \mathcal{O}_{F_{X_n}}(S_1, \ldots, S_n) \end{aligned}$$

mit $S_i \in 2^{\mathsf{Proc}}$.

6.15 Lemma
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{D} \triangleq (2^{\mathsf{Proc}})^n$ für $n \in \mathbb{N}$.
Sei $\sqsubseteq : (\mathcal{D}, \mathcal{D})$ definiert durch

$$\begin{pmatrix} S_1 \\ \vdots \\ S_n \end{pmatrix} \sqsubseteq \begin{pmatrix} S_1' \\ \vdots \\ S_n' \end{pmatrix} \quad \text{falls } S_i \subseteq S_i' \text{ für alle } i \in [1, n].$$

Dann ist $(\mathcal{D}, \sqsubseteq)$ ist ein vollständiger Verband, mit

$$\bigsqcup \left\{ \begin{pmatrix} S_1^i \\ \vdots \\ S_n^i \end{pmatrix} \mid i \in I \right\} \triangleq \begin{pmatrix} \bigcup\limits_{i \in I} S_1^i \\ \vdots \\ \bigcup\limits_{i \in I} S_n^i \end{pmatrix}$$

$$\bigsqcap \left\{ \begin{pmatrix} S_1^i \\ \vdots \\ S_n^i \end{pmatrix} \mid i \in I \right\} \triangleq \begin{pmatrix} \bigcap\limits_{i \in I} S_1^i \\ \vdots \\ \bigcap\limits_{i \in I} S_n^i \end{pmatrix}$$

wobei I eine beliebige Indexmenge ist.

6.16 Lemma
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
Sei $\mathcal{X} = \{X_1, \ldots, X_n\}$ eine nicht-leere Menge von Variablen.
Sei $\mathcal{D} = (2^{\mathsf{Proc}})^n$.
Sei $\boldsymbol{F} : \mathcal{X} \to \mathcal{M}_{\mathcal{X}}$ eine Deklaration.
Dann ist $\mathcal{O}_{\boldsymbol{F}} : \mathcal{D} \to \mathcal{D}$ gegeben durch

$$\begin{pmatrix} S_1 \\ \vdots \\ S_n \end{pmatrix} \mapsto \begin{pmatrix} \mathcal{O}_{\boldsymbol{F}(X_1)}(S_1, \ldots, S_n) \\ \vdots \\ \mathcal{O}_{\boldsymbol{F}(X_n)}(S_1, \ldots, S_n) \end{pmatrix}$$

eine monotone Funktion auf $(\mathcal{D}, \sqsubseteq)$.

6.17 Definition (Denotationelle Semantik von HML)
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$ endliches LTS.
Sei $\mathcal{X} = \{ X_1, \ldots, X_n \}$ eine nicht-leere Menge von Variablen.
Sei $\boldsymbol{F} : \mathcal{X} \to \mathcal{M}_{\mathcal{X}}$ eine Deklaration
mit $X_i \overset{m}{=} \boldsymbol{F}(X_i)$ für alle $1 \leqslant i \leqslant n$ und $m \in \{\max, \min\}$.
Die Funktion $[\![\cdot]\!] : \mathcal{M}_{\mathcal{X}} \to 2^{\mathsf{Proc}}$ ist induktiv definiert durch:

$$
\begin{aligned}
[\![tt]\!] &\triangleq \mathsf{Proc} \\
[\![ff]\!] &\triangleq \emptyset \\
[\![F \wedge G]\!] &\triangleq [\![F]\!] \cap [\![G]\!] \\
[\![F \vee G]\!] &\triangleq [\![F]\!] \cup [\![G]\!] \\
[\![\langle \alpha \rangle F]\!] &\triangleq \langle \cdot \alpha \cdot \rangle [\![F]\!] \\
[\![[\alpha] F]\!] &\triangleq [\cdot \alpha \cdot] [\![F]\!] \\
[\![X_i]\!] &\triangleq \begin{cases} (\mathrm{FIX}\, \mathcal{O}_{\boldsymbol{F}})_i & \text{falls } m = \max \\ (\mathrm{fix}\, \mathcal{O}_{\boldsymbol{F}})_i & \text{falls } m = \min \end{cases}
\end{aligned}
$$

wobei FIX und fix (modulo Typ) analog zu Theorem 6.7 gegeben sind.

6.18 Definition (Charakteristische Deklaration; Teil von Theorem 6.4 in [AILS07])
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$ ein endliches LTS.
Sei $\mathcal{X}_{\mathsf{T}} \triangleq \{ X_p \mid p \in \mathsf{Proc} \}$ eine Menge von Variablen,
deren Bedeutung wir durch Rekursionsgleichungen bestimmen.
Die *charakteristische Deklaration* ist hierzu gegeben durch:

$$
\begin{aligned}
\boldsymbol{F}_{\mathsf{T}} : \mathcal{X}_{\mathsf{T}} &\to \mathcal{M}_{\mathcal{X}_{\mathsf{T}}} \\
X_p &\mapsto \bigwedge_{\alpha \in \mathsf{Act}} \Big(\bigwedge_{p' \in \mathrm{Der}(p,\alpha)} \langle \alpha \rangle X_{p'} \Big) \wedge \bigwedge_{\alpha \in \mathsf{Act}} [\alpha] \Big(\bigvee_{p' \in \mathrm{Der}(p,\alpha)} X_{p'} \Big)
\end{aligned}
$$

6.19 Theorem (Charakteristische Formeln; Teil von Theorem 6.4 in [AILS07])
Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$ ein endliches LTS.
Sei $\boldsymbol{F}_{\mathsf{T}}$ die charakteristische Deklaration, wie in 6.18 definiert.
Sei

$$X_p \overset{\max}{=} \boldsymbol{F}_{\mathsf{T}}(X_p)$$

für alle $p \in \mathsf{Proc}$.
Dann gilt[9]

$$[\![X_p]\!] = [p]_{\sim}$$

weshalb X_p jeweils die *charakteristische Formel für* p genannt wird.

[9] Im Vergleich zu [AILS07] wählen wir hier die Variante aus Lemma 6.3, da sie (ohne Verweis auf andere Prozesse) die charakterische Eigenschaft expliziter widerspiegelt.

6.20 Definition (Min-Max-Mix)
Sei $\mathcal{X} \triangleq (\mathcal{X}_k)_{k\in[1,n]}$ eine Folge
von Vektoren $\mathcal{X}_k = (X_{k,1}, \ldots, X_{k,z_k})$ unterschiedlicher Variablen.
Sei $\mathcal{X}^{\leqslant i} \triangleq (\mathcal{X}_k)_{k\in[1,i]}$ ein Präfix von $\mathcal{X}$. Sei $\mathcal{X}^{<i} \triangleq \mathcal{X}^{\leqslant i-1}$.
Sei $\underline{\mathcal{X}^{\leqslant i}} \triangleq \{\, X_{k,l} \mid k\in[1,i] \text{ und } l\in[1,z_k] \,\}$ die *Menge* der Variablen in $\mathcal{X}^{\leqslant i}$.

Sei $(\boldsymbol{F}_i)_{i\in[1,n]}$ eine Folge von Deklarationen mit $\boldsymbol{F}_i : \mathcal{X}_i \to \mathcal{M}_{\underline{\mathcal{X}^{\leqslant i}}}$.

Sei $(m_i)_{i\in[1,n]}$ eine Folge von Fixpunkt-Indikatoren
mit $m_i \in \{\max, \min\}$ und $m_i \neq m_{i+1}$ für alle $i \in [1, n-1]$.

Dann heißt $\langle(\boldsymbol{F}_1, \mathcal{X}_1, m_1), \cdots, (\boldsymbol{F}_n, \mathcal{X}_n, m_n)\rangle$
n-*verschachteltes gegenseitig rekursives Gleichungssystem*,
wobei $(\boldsymbol{F}_i, \mathcal{X}_i, m_i)$ i-ter Block genannt wird.

Sei $\mathsf{T} = (\mathsf{Proc}, \mathsf{Act}, \mathsf{Tran})$.
In Bezug auf die Abbildung $[\![\cdot]\!] : \mathcal{M}_{\underline{\mathcal{X}^{\leqslant n}}} \to 2^{\mathsf{Proc}}$ (siehe unten) sei
$[\![\mathcal{X}^{\leqslant i}]\!] \triangleq ([\![\mathcal{X}_1]\!], \ldots, [\![\mathcal{X}_i]\!])$ mit $[\![\mathcal{X}_k]\!] \triangleq ([\![X_{k,1}]\!], \ldots, [\![X_{k,z_k}]\!])$.

Die Abbildung $\mathcal{O}_{\boldsymbol{F}_i(X_{i,j});[\![\mathcal{X}^{<i}]\!]} : (2^{\mathsf{Proc}})^{z_i} \to 2^{\mathsf{Proc}}$ für $j\in[1,z_i]$
ist induktiv definiert durch:

$$
\begin{aligned}
\mathcal{O}_{X_{i,j};[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) &\triangleq S_j \\
\mathcal{O}_{X_{k,l};[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) &\triangleq [\![X_{k,l}]\!] \quad \text{falls } k\in[1,i-1] \text{ und } l\in[1,z_k]. \\
\mathcal{O}_{tt;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) &\triangleq \mathsf{Proc} \\
\mathcal{O}_{ff;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) &\triangleq \emptyset \\
\mathcal{O}_{F\wedge G;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) &\triangleq \mathcal{O}_{F;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) \cap \mathcal{O}_{G;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) \\
\mathcal{O}_{F\vee G;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) &\triangleq \mathcal{O}_{F;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) \cup \mathcal{O}_{G;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) \\
\mathcal{O}_{\langle\alpha\rangle F;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) &\triangleq \langle\cdot\alpha\cdot\rangle \mathcal{O}_{F;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) \\
\mathcal{O}_{[\alpha]F;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i}) &\triangleq [\cdot\alpha\cdot] \mathcal{O}_{F;[\![\mathcal{X}^{<i}]\!]}(S_1, \ldots, S_{z_i})
\end{aligned}
$$

und wie in Lemma 6.16 geliftet auf $\mathcal{O}_{\boldsymbol{F}_i;[\![\mathcal{X}^{<i}]\!]} : (2^{\mathsf{Proc}})^{z_i} \to (2^{\mathsf{Proc}})^{z_i}$.

Die Funktion $[\![\cdot]\!] : \mathcal{M}_{\underline{\mathcal{X}^{\leqslant n}}} \to 2^{\mathsf{Proc}}$ ist schließlich induktiv definiert durch:

$$
\begin{aligned}
[\![tt]\!] &\triangleq \mathsf{Proc} \\
[\![ff]\!] &\triangleq \emptyset \\
[\![F \wedge G]\!] &\triangleq [\![F]\!] \cap [\![G]\!] \\
[\![F \vee G]\!] &\triangleq [\![F]\!] \cup [\![G]\!] \\
[\![\langle\alpha\rangle F]\!] &\triangleq \langle\cdot\alpha\cdot\rangle [\![F]\!] \\
[\![[\alpha] F]\!] &\triangleq [\cdot\alpha\cdot] [\![F]\!] \\
[\![X_{i,j}]\!] &\triangleq \begin{cases} \left(\mathrm{FIX}\, \mathcal{O}_{\boldsymbol{F}_i;[\![\mathcal{X}^{<i}]\!]}\right)_j & \text{falls } m = \max \\ \left(\mathrm{fix}\, \mathcal{O}_{\boldsymbol{F}_i;[\![\mathcal{X}^{<i}]\!]}\right)_j & \text{falls } m = \min \end{cases}
\end{aligned}
$$

wobei FIX und fix wie in Theorem 6.17 gegeben sind.