

# 1 Definitionen

Stand: 28. Juni 2017

Kombination aus [BV15] und [Sch16] mit Einflüssen von [BFLV16]:

**Definition 1.1 (*Modal Error-I/O-Transitionssystem*).** Ein Modal Error-I/O-Transitionssystem (MEIO) ist ein Tupel  $(P, I, O, \longrightarrow, \dashrightarrow, p_0, E)$  mit:

- $P$ : Menge der Zustände,
- $p_0 \in P$ : Startzustand,
- $I, O$ : disjunkte Mengen der (sichtbaren) Input- und Output-Aktionen,
- $\longrightarrow \subseteq P \times \Sigma_\tau \times P$ : must-Transitions-Relation,
- $\dashrightarrow \subseteq P \times \Sigma_\tau \times P$ : may-Transitions-Relation,
- $E \subseteq P$ : Menge der Fehler-Zustände.

Es wird vorausgesetzt, dass  $\longrightarrow \subseteq \dashrightarrow$  (syntaktische Konsistenz) gilt.

Das Alphabet bzw. die Aktionsmenge eines MEIO ist  $\Sigma = I \cup O$ . Die interne Aktion  $\tau$  ist nicht in  $\Sigma$  enthalten. Jedoch gilt  $\Sigma_\tau := \Sigma \cup \{\tau\}$ .

Falls  $\longrightarrow = \dashrightarrow$  gilt, wird  $P$  auch Implementierung genannt.

Must-Transitionen sind Transitionen, die von einer Verfeinerung implementiert werden müssen. Die may-Transitionen sind hingegen die zulässigen Transitionen für eine Verfeinerung.

MEIOs werden in dieser Arbeit durch ihre Zustandsmenge (z.B.  $P$ ) identifiziert und falls notwendig werden damit auch die Komponenten indiziert (z.B.  $I_P$  anstatt  $I$ ). Falls der MEIO selbst bereits einen Index hat (z.B.  $P_1$ ) kann an der Komponente die Zustandsmenge als Index wegfallen und nur noch der Index des gesamten Automaten verwendet werden (z.B.  $I_1$  anstatt  $I_{P_1}$ ). Zusätzlich stehen  $i, o, a, \omega$  und  $\alpha$  für Buchstaben aus den Alphabeten  $I, O, \Sigma, O \cup \{\tau\}$  und  $\Sigma_\tau$ .

Es wird die Notation  $p \xrightarrow{\alpha} p'$  für  $(p, \alpha, p') \in \dashrightarrow$  und  $p \xrightarrow{\alpha}$  für  $\exists p' : (p, \alpha, p') \in \dashrightarrow$  verwendet. Dies kann entsprechend auf Buchstaben-Sequenzen  $w \in \Sigma_\tau^*$  erweitert werden zu  $p \xrightarrow{w} p'$  ( $p \xrightarrow{w}$ ) steht für die Existenz eines Laufes  $p \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} \dots p_{n-1} \xrightarrow{\alpha_n} p'$  ( $p \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} \dots p_{n-1} \xrightarrow{\alpha_n}$ ) mit  $w = \alpha_1 \dots \alpha_n$ .

Desweiteren soll  $w|_B$  die Aktions-Sequenz bezeichnen, die man erhält, wenn man aus  $w$  alle Aktionen löscht, die nicht in  $B \subseteq \Sigma$  enthalten sind.  $\hat{w}$  steht für  $w|_\Sigma$ . Es wird  $p \xRightarrow{w} p'$

## 1 Definitionen

für ein  $w \in \Sigma^*$  geschrieben, falls  $\exists w' \in \Sigma_\tau^* : \widehat{w'} = w \wedge p \xrightarrow{w'} p'$ , und  $p \xRightarrow{w} p'$ , falls  $p \xRightarrow{w} p'$  für ein beliebiges  $p'$  gilt. Falls  $p_0 \xRightarrow{w} p$  gilt, dann wird  $w$  *Trace* genannt und  $p$  ist ein *erreichbarer Zustand*.

Analog zu  $\xrightarrow{\cdot}$  und  $\xRightarrow{\cdot}$  werden  $\longrightarrow$  und  $\Longrightarrow$  für die entsprechenden Relationen der must-Transition verwendet.

Outputs und die interne Aktion werden *lokale Aktionen* genannt, da sie lokal vom ausführenden MEIO kontrolliert sind. Um eine Erleichterung der Notation zu erhalten, soll gelten, dass  $p \xrightarrow{a} p'$  und  $p \xrightarrow{a!} p'$  für  $\nexists p' : p \xrightarrow{a} p'$  und  $\nexists p' : p \xrightarrow{a!} p'$  stehen soll. In Graphiken wird eine Aktion  $a$  als  $a?$  notiert, falls  $a \in I$  und  $a!$ , falls  $a \in O$ . Must-Transitionen (may-Transitionen) werden als durchgezogener Pfeil gezeichnet (gestrichelter Pfeil). Entsprechend der syntaktischen Konsistenz repräsentiert jede gezeichnete must-Transition auch gleichzeitig die zugrundeliegende may-Transitionen.

**Definition 1.2 (*Parallelkomposition*).** Zwei MEIOs  $P_1 = (P_1, I_1, O_1, \longrightarrow_1, \xrightarrow{\cdot}_1, p_{01}, E_1)$  und  $P_2 = (P_2, I_2, O_2, \longrightarrow_2, \xrightarrow{\cdot}_2, p_{02}, E_2)$  sind komponierbar, falls  $O_1 \cap O_2 = \emptyset$ . Für solche MEIOs ist die Parallelkomposition  $P_{12} := P_1 \parallel P_2 = ((P_1 \times P_2), I, O, \longrightarrow_{12}, \xrightarrow{\cdot}_{12}, (p_{01}, p_{02}), E)$  definiert mit: TODO: erzwungenen Zeilenumbrüche kontrollieren

- $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2),$
- $O = (O_1 \cup O_2),$
- $\longrightarrow_{12} = \left\{ ((p_1, p_2), \alpha, (p'_1, p'_2)) \mid p_1 \xrightarrow{\alpha}_1 p'_1, \alpha \in \Sigma_\tau \setminus \text{Synch}(P_1, P_2) \right\} \\ \cup \left\{ ((p_1, p_2), \alpha, (p_1, p'_2)) \mid p_2 \xrightarrow{\alpha}_2 p'_2, \alpha \in \Sigma_\tau \setminus \text{Synch}(P_1, P_2) \right\} \\ \cup \left\{ ((p_1, p_2), \alpha, (p'_1, p'_2)) \mid p_1 \xrightarrow{\alpha}_1 p'_1, p_2 \xrightarrow{\alpha}_2 p'_2, \alpha \in \text{Synch}(P_1, P_2) \right\}$
- $\xrightarrow{\cdot}_{12} = \left\{ ((p_1, p_2), \alpha, (p'_1, p'_2)) \mid p_1 \xrightarrow{\alpha}_1 p'_1, \alpha \in \Sigma_\tau \setminus \text{Synch}(P_1, P_2) \right\} \\ \cup \left\{ ((p_1, p_2), \alpha, (p_1, p'_2)) \mid p_2 \xrightarrow{\alpha}_2 p'_2, \alpha \in \Sigma_\tau \setminus \text{Synch}(P_1, P_2) \right\} \\ \cup \left\{ ((p_1, p_2), \alpha, (p'_1, p'_2)) \mid p_1 \xrightarrow{\alpha}_1 p'_1, p_2 \xrightarrow{\alpha}_2 p'_2, \alpha \in \text{Synch}(P_1, P_2) \right\}$
- $E = (P_1 \times E_2) \cup (E_1 \times P_2) \quad \text{geerbte Fehler} \\ \left. \begin{array}{l} \cup \left\{ (p_1, p_2) \mid \exists a \in O_1 \cap I_2 : p_1 \xrightarrow{a} \wedge p_2 \xrightarrow{a} \right\} \\ \cup \left\{ (p_1, p_2) \mid \exists a \in I_1 \cap O_2 : p_1 \xrightarrow{a} \wedge p_2 \xrightarrow{a} \right\} \end{array} \right\} \quad \text{neue Kommunikationsfehler}$

Dabei bezeichnet  $\text{Synch}(P_1, P_2) = (I_1 \cap O_2) \cup (O_1 \cap I_2) \cup (I_1 \cap I_2)$  die Menge der zu synchronisierenden Aktionen. Die synchronisierten Aktionen werden als Output bzw. Input der Komposition beibehalten.

**Definition 1.3 (*Simulation*).** Eine (starke) alternierende Simulation ist eine Relation  $R \subseteq P \times Q$  auf zwei MEIOs  $P$  und  $Q$ , falls für alle  $(p, q) \in R$  gilt:

1.  $p \xrightarrow{\alpha} p'$  impliziert  $q \xrightarrow{\alpha} q'$  für ein  $q'$  mit  $pRq'$ ,

2. TODO:  $\xrightarrow{\cdot}$

## 1 Definitionen

*TODO: Simulation als Grundlage für die Definition der Implementierung aus [BV15]*

# Literaturverzeichnis

- [BFLV16] Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen, und Walter Vogler, *Non-deterministic Modal Interfaces*, Theor. Comput. Sci. **642** (2016), 24–53.
- [BV15] Ferenc Bujtor und Walter Vogler, *Failure Semantics for Modal Transition Systems*, ACM Trans. Embedded Comput. Syst. **14** (2015), no. 4, 67:1–67:30.
- [Sch16] Ayleen Schinko, *Kommunikationsfehler, Verklemmung und Divergenz bei Interface-Automaten*, Bachelorarbeit, Universität Augsburg, 2016.