1 Definitionen

Stand: 2. Juli 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 τ *ist nicht in* Σ *enthalten. Jedoch gilt* $\Sigma_\tau := \Sigma \cup \{\tau\}$*. Die* Signatur *eines MEIOs entspricht* $\mathrm{Sig}(S) = (I, O)$*.*
Falls $\longrightarrow = \dashrightarrow$ *gilt, wird* P *auch* Implementierung *genannt.*

Implementierungen entsprechen den in [Sch16] behandelten EIOs.
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, ω und α für Buchstaben aus den Alphabeten $I, O, \Sigma, O \cup \{\tau\}$ und Σ_τ.
Es wir die Notation $p \overset{\alpha}{\dashrightarrow} p'$ für $(p, \alpha, p') \in \dashrightarrow$ und $p \overset{\alpha}{\dashrightarrow}$ für $\exists p' : (p, \alpha, p') \in \dashrightarrow$ verwendet. Dies kann entsprechend auf Buchstaben-Sequenzen $w \in \Sigma_\tau^*$ erweitert werden zu $p \overset{w}{\dashrightarrow} p'$ ($p \overset{w}{\dashrightarrow}$) steht für die Existenz eines Laufes $p \overset{\alpha_1}{\dashrightarrow} p_1 \overset{\alpha_2}{\dashrightarrow} \ldots p_{n-1} \overset{\alpha_n}{\dashrightarrow} p'$ ($p \overset{\alpha_1}{\dashrightarrow} p_1 \overset{\alpha_2}{\dashrightarrow} \ldots p_{n-1} \overset{\alpha_n}{\dashrightarrow}$) mit $w = \alpha_1 \ldots \alpha_n$.

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

Analog zu $\dashrightarrow$ und $\Longrightarrow$ 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 $p \stackrel{a}{\not\longrightarrow}$ und $p \stackrel{a}{\not\dashrightarrow}$ für $\nexists p' : p \stackrel{a}{\longrightarrow} p'$ und $\nexists p' : p \stackrel{a}{\dashrightarrow} p'$ 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* $P_1 = (P_1, I_1, O_1, \longrightarrow_1, \dashrightarrow_1, p_{01}, E_1)$ *und* $P_2 = (P_2, I_2, O_2, \longrightarrow_2, \dashrightarrow_2, p_{02}, E_2)$ *sind* komponierbar*, falls* $O_1 \cap O_2 = \emptyset$. *Für solche MEIOs ist die* Parallelkomposition $P_{12} := P_1 \| P_2 = ((P_1 \times P_2), I, O, \longrightarrow_{12}, \dashrightarrow_{12}, (p_{01}, p_{02}), E)$ *definiert mit:* TODO: erzwungenen Zeilenumbrüche kontrollieren

- $I = (I_1 \cup I_2) \backslash (O_1 \cup O_2)$,
- $O = (O_1 \cup O_2)$,
- $$\begin{aligned} \longrightarrow_{12} = & \Big\{((p_1,p_2),\alpha,(p_1',p_2)) \mid p_1 \stackrel{\alpha}{\longrightarrow}_1 p_1', \alpha \in \Sigma_\tau \backslash \mathrm{Synch}(P_1,P_2)\Big\} \\ & \cup \Big\{((p_1,p_2),\alpha,(p_1,p_2')) \mid p_2 \stackrel{\alpha}{\longrightarrow}_2 p_2', \alpha \in \Sigma_\tau \backslash \mathrm{Synch}(P_1,P_2)\Big\} \\ & \cup \Big\{((p_1,p_2),\alpha,(p_1',p_2')) \mid p_1 \stackrel{\alpha}{\longrightarrow}_1 p_1', p_2 \stackrel{\alpha}{\longrightarrow}_2 p_2', \alpha \in \mathrm{Synch}(P_1,P_2)\Big\} \end{aligned}$$
- $$\begin{aligned} \dashrightarrow_{12} = & \Big\{((p_1,p_2),\alpha,(p_1',p_2)) \mid p_1 \stackrel{\alpha}{\dashrightarrow}_1 p_1', \alpha \in \Sigma_\tau \backslash \mathrm{Synch}(P_1,P_2)\Big\} \\ & \cup \Big\{((p_1,p_2),\alpha,(p_1,p_2')) \mid p_2 \stackrel{\alpha}{\dashrightarrow}_2 p_2', \alpha \in \Sigma_\tau \backslash \mathrm{Synch}(P_1,P_2)\Big\} \\ & \cup \Big\{((p_1,p_2),\alpha,(p_1',p_2')) \mid p_1 \stackrel{\alpha}{\dashrightarrow}_1 p_1', p_2 \stackrel{\alpha}{\dashrightarrow}_2 p_2', \alpha \in \mathrm{Synch}(P_1,P_2)\Big\} \end{aligned}$$
- $$\begin{aligned} E = & (P_1 \times E_2) \cup (E_1 \times P_2) && \textit{geerbte Kommunikationsfehler} \\ & \left.\begin{aligned} & \cup \Big\{(p_1,p_2) \mid \exists a \in O_1 \cap I_2 : p_1 \stackrel{a}{\dashrightarrow} \wedge p_2 \stackrel{a}{\not\longrightarrow}\Big\} \\ & \cup \Big\{(p_1,p_2) \mid \exists a \in I_1 \cap O_2 : p_1 \stackrel{a}{\not\longrightarrow} \wedge p_2 \stackrel{a}{\dashrightarrow}\Big\} \end{aligned}\right\} && \textit{neue Kommunikationsfehler} \end{aligned}$$

Dabei bezeichnet $\mathrm{Synch}(P_1, P_2) = (I_1 \cap O_2) \cup (O_1 \cap I_2) \cup (I_1 \cap I_2)$ *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. $p \stackrel{\alpha}{\longrightarrow} p'$ *impliziert* $q \stackrel{\alpha}{\longrightarrow} q'$ *für ein* q' *mit* pRq',

2. $q \overset{\alpha}{\dashrightarrow} q'$ *impliziert* $p \overset{\alpha}{\dashrightarrow} p'$ *für ein* p' *mit* pRq',

Die Vereinigung $\sqsubseteq_{as}$ *aller dieser Relationen wird als* (starke) as-Verfeinerung(-s Relation) *(auch modal Verfeinerung) bezeichnet. Es wird* $P \sqsubseteq_{as} Q$ *geschrieben, falls* $p_0 \sqsubseteq_{as} q_0$ *gilt, und* P as-verfeinert Q (stark) *oder* P *ist eine* (starke) as-Verfeinerung *von* Q.
Für einen MEIO Q *und eine Implementierung mit* P *mit* $P \sqsubseteq_{as} Q$, *ist* P *eine* as-Implementierung *von* Q *wund es wird* as-impl(Q) *für die Menge aller as-Implementierungen von* Q *verwendet.*

Die Funktionen prune und cont zum Abschneiden und Verlängern von Traces werden hier genauso verwendet, wie sie in [Sch16] definiert wurden. Man muss dazu nur EIO in der Defintion durch MEIO ersetzten. Ebenso wird die Parallelkomposition von Wörtern und Mengen von Wörtern bzw. Sprachen analog übernommen.

Definition 1.4 (*Sprache*). *Die* (minimale) Sprache *eines MEIOs* P *ist* $L(P) := \left\{ w \in \Sigma^* \mid \text{für alle Implementierungen von } P : p_0 \overset{w}{\Longrightarrow} \right\}$.
Für zwei komboinerbare MEIOs P_1 *und* P_2 *gilt:* $L_{12} := L(P_{12}) = L_1 \| L_2$.

2 Verfeinerungen für Kommunikationsfehler-Freiheit

Dieses Kapitel versucht die Präkongruenz für Error bei EIOs aus [Sch16] auf die hier betrachten MEIOs zu erweitern.

Definition 2.1 (*Fehler-freie Kommunikation*). *Ein Kommunikationsfehler-Zustand ist* lokal erreichbar *in einer Implementierung P' eines MEIO P, wenn ein $w \in O^*$ existiert mit $p'_0 \stackrel{w}{\Longrightarrow} p' \in E'$.*
Zwei MEIOs P_1 und P_2 kommunizieren Fehler-frei*, wenn alle Implementierungen ihrer Parallelkomposition P_{12} keine Kommunikationsfehler-Zustände lokal erreichen können.*

Definition 2.2 (*Kommunikationsfehler-Verfeinerungs-Basirelation*). *Für zwei MEIOs P_1 und P_2 mit der gleichen Signatur wird $P_1 \overline{\sqsubset}_E^{\mathrm{B}} P_2$ geschrieben, wenn ein Kommunikationsfehler-Zustand in einer Implementierung von P_1 nur dann lokal erreichbar ist, wenn es auch eine Implementierung von P_2 gibt, in der dieser Kommunikationsfehler-Zustand auch lokal erreichbar ist. Die Basisrelation stellt eine* Verfeinerung *bezüglich* Kommunikationsfehlern *dar.*
$\overline{\sqsubset}_E^{\mathrm{C}}$ bezeichnet die vollständig abstrakte Präkongruenz *von $\overline{\sqsubset}_E^{\mathrm{B}}$ bezüglich $\cdot\|\cdot$, d.h. die gröbste Präkongruenz bezüglich $\cdot\|\cdot$, die in $\overline{\sqsubset}_E^{\mathrm{B}}$ enthalten ist.*

Für P_1 und P_2 Implementierungen entspricht $\overline{\sqsubset}_E^{\mathrm{B}}$ der Relation $\sqsubseteq_E^{\mathrm{B}}$ aus [Sch16].

Wie in [Sch16] werden die Fehler hier Trace basiert betrachtet. Da jedoch die Basisrelation über Implementierungen spricht, sind die Trace Mengen auch nicht für die MEIOs mit may-Transitionen definiert sondern nur für die Menge der möglichen Implementierungen eines solchen MEIOs.

Definition 2.3 (*Kommunikationsfehler-Traces*). *Für einen MEIO P wird definiert:*

- strikte Kommunikationsfehler-Traces*: $StET(P) := \left\{ w \in \Sigma^* \mid \exists \textit{Implementierung von } P : p_0 \stackrel{w}{\Longrightarrow} p \in E \right\}$*
- gekürzte Kommunikationsfehler-Traces*: $PrET(P) := \{\mathrm{prune}(w) \mid w \in StET(P)\}$*
- Input-kritische-Traces*: $MIT(P) := \left\{ wa \in \Sigma^* \mid \exists \textit{Implementierung von } P : p_0 \stackrel{w}{\Longrightarrow} p \wedge a \in I \wedge p \stackrel{a}{\not\longrightarrow} \right\}$*

Definition 2.4 (*Kommunikationsfehler-Semantik*). *Sie P ein MEIO.*

- *Die Menge der* Kommunikationsfehler-Traces *von* P *ist* $ET(P) := \mathrm{cont}(PrET(P)) \cup \mathrm{cont}(MIT(P))$.
- *Die* Kommunikationsfehler-geflutete Sprache *von* P *ist* $EL(P) := L(P) \cup ET(P)$.

Für zwei MEIOs P_1, P_2 *mit der gleichen Signatur wird* $P_1 \sqsubseteq_E P_2$ *geschrieben, wenn* $ET_1 \subseteq ET_2$ *und* $EL_1 \subseteq EL_2$ *gilt.*

Hierbei ist zu beachten, dass die Mengen $StET$, $PrET$, MIT, ET und EL nur denen aus [Sch16] entsprechen, wenn P bereits eine Implementierung ist.

Satz 2.5 (*Kommunikationsfehler-Semantik für Parallelkompositionen*). *Für zwei komponierbare MEIOs* P_1, P_2 *und ihre Komposition* P_{12} *gilt:*

1. $ET_{12} = \mathrm{cont}(\mathrm{prune}((ET_1 \| EL_2) \cup (EL_1 \| ET_2)))$,
2. $EL_{12} = (EL_1 \| EL_2) \cup ET_{12}$.

Literaturverzeichnis

[BFLV16] Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen, und Walter Vogler, *Nondeterministic 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.