1 Definitionen und Notationen

Die Definitionen dieses Kapitels sind größtenteils aus [BV14] übernommen. Hierbei handelt es sich um die Grundlagen der Transitionssysteme mit denen hier gearbeitet werden soll. Jedoch wurde angepasst, dass für die Parallelkomposition die Input-Aktionen der Error-IO-Transitionssysteme (EIOs) nicht disjunkt sein müssen. Dies wäre eine unnötige Einschränkung. Die nicht synchronisierten Inputs der zu komponierenden EIOs werden als Inputs der Parallelkomposition übernommen. Zusätzlich verzichten wir hier auf das verbergen der synchronisierten Aktionen. Die gleiche Betrachtungsweise wurde bereits in [Sch12] gewählt, deshalb stimmen die Definitionen in diesem Kapitel mit denen aus [Sch12] überein, jedoch wurde diese Arbeit nicht als direkte Quelle verwendet.

1.1 Error-IO-Transitionssystem

Die hier betrachteten EIOs sind Systeme, deren Übergänge mit Inputs und Outputs beschriftet sind. Jeder Übergang ist dabei mit einem Input oder einem Output versehen. Ebenfalls zulässig ist eine Kantenbeschriftung mit τ, einer internen, unbeobachtbaren Aktion. Diese interne Aktion lässt also keine Interaktion mit der Umwelt zu. In vielen Fällen entstehen sie durch das Verbergen der Inputs und Outputs dieses Übergangs, da diese in einer Komposition synchronisiert wurden.

Definition 1.1 (*Error-IO-Transitionssystem*). *Ein Error-IO-Transitionssystem (EIO) ist ein Tupel $S = (Q, I, O, \delta, q_0, E)$, mit den Komponenten:*

- Q *– die Menge der Zustände,*
- I, O *– die disjunkten Mengen der (sichtbaren) Input- und Outputaktionen,*
- $\delta \subseteq Q \times (I \cup O \cup \{\tau\}) \times Q$ *– die Übergangsrelation,*
- $q_0 \in Q$ *– der Startzustand,*
- $E \subseteq Q$ *– die Menge der Error-Zustände.*

Die Aktionsmenge eines EIOs S ist $\Sigma = I \cup O$ und die Signatur $Sig(S) = (I, O)$.
Um in graphischen Veranschaulichungen Inputs und Outputs zu unterscheiden wird folgende Notation verwendet: $x?$ für den Input x und $x!$ für den Output x. Falls ein x ohne ? oder ! verwendet wird, steht dies für eine Aktion, bei der nicht festgelegt ist, ob sie ein Input oder ein Output ist.
Um die Komponenten der entsprechenden Struktur zuzuordnen, werden für die Komponenten die gleichen Indizes wie für ihre zugehörigen Struktur verwendet, z.B. für die

Inputmenge des Transitionssystem S_1 schreiben wir I_1. Diese Notation verwenden wir später analog für die Sprachen, die einer Struktur zugeordnet sind.
Die Elemente der Übergangsrelation δ werden wir wie folgt notieren:

- $p \xrightarrow{a} q$ für $(p, a, q) \in \delta$,
- $p \xrightarrow{a}$ für $\exists q : (p, a, q) \in \delta$,
- $p \xrightarrow{w} q$ für $p \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \ldots \xrightarrow{a_n} q$ mit $w \in (\Sigma \cup \{\tau\})^*, w = a_1 a_2 \ldots a_n$,
- $p \xrightarrow{w}$ für $p \xrightarrow{a_1}\xrightarrow{a_2} \ldots \xrightarrow{a_n}$ mit $w \in (\Sigma \cup \{\tau\})^*, w = a_1 a_2 \ldots a_n$,
- $w|_B$ steht für die Zeichenfolge, die aus w entsteht durch löschen aller Zeichen, die nicht in $B \subseteq \Sigma$ enthalten sind, d.h. es bezeichnet die Projektion von w auf die Menge B,
- $p \stackrel{w}{\Rightarrow} q$ für $w \in \Sigma^*$ mit $\exists w' \in (\Sigma \cup \{\tau\})^* : w'|_\Sigma = w \wedge p \xrightarrow{w'} q$,
- $p \stackrel{w}{\Rightarrow}$ für $\exists q : p \stackrel{w}{\Rightarrow} q$.

Die Sprache von S ist $L(S) = \{w \in \Sigma^* \mid q_0 \stackrel{w}{\Rightarrow}\}$.

1.2 Parallelkomposition

Zwei EIOs sind komponierbar, wenn ihre Outputaktionsmengen disjunkt sind. Die Error-Zustände der Parallelkomposition setzten sich aus den Error-Zuständen der beiden zusammengesetzten Komponenten (geerbte Errors) und den Outputs zusammen, die von der anderen Komponente nicht als Inputs angenommen werden können (neue Errors).
Die nächste Definition ist noch analog zu [BV14], nur wird hier darauf verzichtet die Inputmengen als disjunkt anzunehmen und der zweite Definitionsteil über das verbergen der synchronisierten Aktionen wird komplett weg gelassen. Das verzichten auf die analogen Definitionen für das Verbergen wird sich auch bei den folgenden Definitionen zeigen. Zusätzlich nehmen wir in der folgenden Definition eine Änderung an der Menge der synchronisierten Aktionen vor, da nun nicht mehr $I_1 \cap I_2 = \emptyset$ gelten muss, werden wir diese gemeinsamen Inputs synchronisieren.

Definition 1.2 (*Parallelkomposition*). *Zwei EIOs S_1, S_2 sind komponierbar, falls $O_1 \cap O_2 = \emptyset$ gilt. Die Parallelkomposition ist $S_1 \| S_2 = (Q, I, O, \delta, q_0, E)$ mit den Komponenten:*

- $Q = Q_1 \times Q_2$,
- $I = (I_1 \backslash O_2) \cup (I_2 \backslash O_1)$,
- $O = O_1 \cup O_2$,
- $q_0 = (q_{01}, q_{02})$,

- $\delta = \{((q_1,q_2),\alpha,(p_1,q_2)) \mid (q_1,\alpha,p_1) \in \delta_1, \alpha \in (\Sigma_1 \cup \{\tau\}) \backslash Synch(S_1,S_2)\}$
  $\cup \{((q_1,q_2),\alpha,(q_1,p_2)) \mid (q_2,\alpha,p_2) \in \delta_2, \alpha \in (\Sigma_2 \cup \{\tau\}) \backslash Synch(S_1,S_2)\}$
  $\cup \{((q_1,q_2),\alpha,(p_1,p_2)) \mid (q_1,\alpha,p_1) \in \delta_1, (q_2,\alpha,p_2) \in \delta_2, \alpha \in Synch(S_1,S_2)\}$,
- $$\begin{aligned} E = &(Q_1 \times E_2) \cup (E_1 \times Q_2) && \text{geerbte Errors} \\ &\left.\begin{aligned} &\cup \{(q_1,q_2) \mid \exists a \in O_1 \cap I_2 : q_1 \xrightarrow{a} \wedge q_2 \stackrel{a}{\not\rightarrow}\} \\ &\cup \{(q_1,q_2) \mid \exists a \in I_1 \cap O_2 : q_1 \stackrel{a}{\not\rightarrow} \wedge q_2 \xrightarrow{a}\} \end{aligned}\right\} && \text{neue Errors.} \end{aligned}$$

Dabei werden die synchronisierten Aktionen $Synch(S_1,S_2) = (I_1 \cap O_2) \cup (O_1 \cap I_2) \cup (I_1 \cap I_2)$ nicht versteckt, sondern als Outputs der Komposition beibehalten.

Nun werden wir drauf eingehen, dass eine Parallelkomposition nicht nur für Transitionssysteme betrachtet werden kann, sondern auch über Transitionsfolgen. Ein *Trace* ist dann das Wort, das aus den Inputs und Outputs besteht, mit denen die Übergängen beschriftet sind.

Definition 1.3 (*Parallelkomposition auf Traces*). *Gegeben zwei EIOs S_1 und S_2, $w_1 \in \Sigma_1, w_2 \in \Sigma_2, W_1 \subseteq \Sigma_1^*, W_2 \subseteq \Sigma_2^*$:*

- $w_1 \| w_2 := \{w \in (\Sigma_1 \cup \Sigma_2)^* \mid w|_{\Sigma_1} = w_1 \wedge w|_{\Sigma_2} = w_2\}$,
- $W_1 \| W_2 := \bigcup \{w_1 \| w_2 \mid w_1 \in W_1 \wedge w_2 \in W_2\}$.

Die Semantik der späteren Kapitel basiert darauf die jeweiligen Zustände, die zu Problemen führen, mit ihren Traces zu betrachten. Um dies besser umsetzten zu können, definieren wir eine *prune*-Funktion, die alle Outputs am Ende eines Traces entfernt. Zusätzlich werden Funktionen definiert, die die Traces beliebig fortsetzen.

Definition 1.4 (*Pruning und Fortsetzungs Funktionen*). *Für einen EIO S definieren wir:*

- $prune : \Sigma^* \to \Sigma^*, w \mapsto u,$ *mit* $w = uv, u = \varepsilon \vee u \in \Sigma^* \cdot I$ *und* $v \in O^*$,
- $cont : \Sigma^* \to \mathfrak{P}(\Sigma^*), w \mapsto \{wu \mid u \in \Sigma^*\}$,
- $cont : \mathfrak{P}(\Sigma^*) \to \mathfrak{P}(\Sigma^*), L \mapsto \bigcup \{cont(w) \mid w \in L\}$.

Für zwei komponierbare EIOs S_1 und S_2 ist ein Ablauf ihrer Parallelkomposition $S_{12} = S_1 \| S_2$ eine Transitionsfolge der Form $(p_1,p_2) \stackrel{w}{\Rightarrow} (q_1,q_2)$ für ein $w \in \Sigma_{12}^*$. So ein Ablauf kann auf Abläufe von S_1 und S_2 projiziert werden. Diese Projektion erfüllen $p_i \stackrel{w_i}{\Rightarrow} q_i$ mit $w|_{\Sigma_i} = w_i$ für $i = 1,2$. Umgekehrt sind zwei Abläufe von S_1 und S_2, die wie oben aufgebaut sind, Projektionen von Abläufen in S_{12}, die ebenfalls wie oben aufgebaut sind. Wir können nicht garantieren, dass es in S_{12} genau einen Ablauf gibt, da wir die Reihenfolge der unsynchronisierten Aktionen verändern können. Daraus folgt das folgende Lemma.

Lemma 1.5 (*Sprache der Parallelkomposition*). *Für zwei komponierbare EIOs S_1 und S_2 gilt:*

$$L_{12} := L(S_1 \| S_2) = L(S_1) \| L(S_2).$$

Dieses Lemma ist bereits in [BV14] enthalten. Hier wurde jedoch noch die Benennung der Sprache explizit definiert und es wurde keine Sprache für die Parallelkomposition mit Verbergen erwähnt.

2 Verfeinerung über Errortraces

In diesem Kapitel wählen wir einen optimistischen Ansatz für die Fehlererreichbarkeit. Ein Error gilt hier als erreichbar, wenn er lokal erreicht werden kann, d.h. durch lokale Aktionen. Die Menge, bestehend aus der internen Aktion τ und den Outputaktionen, bezeichnen wir hier als lokale Aktionen. Alle Elemente aus dieser Menge können ohne weiteres Zutun von außen aufgeführt werden. Somit kann nicht beeinflusst werden ob diese Übergänge genommen werden oder nicht. Es besteht also die Möglichkeit, dass der EIO in einen Error-Zustand übergeht, sobald dieser lokal erreichbar ist. Diese Art der Erreichbarkeit von Fehler wird auch in Kapitel 3 von [BV14] dargestellt.
Neben dem hier betrachteten optimistischen Ansatz gibt es noch zwei weitere Ansätze in [BV14]. Einen hyper-optimistischen Ansatz, bei dem ein Fehler als erreichbar gilt, wenn er durch interne Aktionen erreicht werden kann, und einen pessimistischen Ansatz, bei dem ein Error als erreichbar gilt, sobald es eine Folge an Inputs und Outputs gibt, mit denen ein Error-Zustand vom Startzustand aus erreicht werden kann.
Die gleiche Betrachtung wie hier wurde bereits in [Sch12] gewählt, somit stimmen die Definitionen und Ergebnisse überein. Jedoch wurden alle Beweise unabhängig von dieser Arbeit neu geführt.
Da es in dieser Arbeit vor allem und die Erreichbarkeit und die Kommunikation zwischen EIOs geht, wurden die nächsten beiden Definitionen explizit getrennt und erweitert zu denen in [BV14]. Ebenfalls wurde die Parallelkomposition geändert, wie in [Sch12].

Definition 2.1 (*lokal errorfreie Kommunikation*). *Ein Error ist lokal erreichbar in einem EIO S, wenn $\exists w \in O^* : q_0 \stackrel{w}{\Rightarrow} q \in E$.*
Zwei EIOs S_1 und S_2 kommunizieren gut, wenn in ihrer Parallelkomposition $S_1 \| S_2$ keine lokalen Errors erreicht werden können.

Über der lokalen Erreichbarkeit von Fehlern können wir eine Verfeinerungsrelation definieren.

Definition 2.2 (*lokale Basisrelation*). *Für EIOs S_1 und S_2 mit der gleichen Signatur schreiben wir $S_1 \sqsubseteq_E^B S_2$, wenn ein Error in S_1 nur dann lokal erreichbar ist, wenn er auch in S_2 lokal erreichbar ist.*
$\sqsubseteq_E^C$ bezeichnet die vollständig abstrakte Präkongruenz von $\sqsubseteq_E^B$ bezüglich $\cdot\|\cdot$.

Um uns nun näher mit den Präkongruenzen auseinandersetzen zu können, müssen wir bestimmte Traces aus unser Struktur hervorheben. Die strikten Errortraces sind Wege, die direkt vom Startzustand zu einem Zustand in der Menge E führen. Da Outputs Aktionen sind, die von außen nicht verhindert werden können, benötigen wir auch noch die Menge der Traces, die zu einem Zustand führen, von dem aus mit lokalen Aktionen ein Error erreicht werden kann. Zusätzlich ist auch noch die Menge der Traces interessant, für die es einen Input $a \in I$ gibt, durch den sie nicht fortgesetzt werden können. Diese führen zwar nicht direkt zu einem Fehler, jedoch in Komposition mit einem anderen Transitionssystem sind dies gefährdete Stellen. Sie führen zu einem neuen Error, sobald dieser Input für die Synchronisation fehlt. Die folgenden beiden Definitionen wurden aus [BV14] übernommen.

Definition 2.3 (*Errortraces*). *Sei S ein EIO und definiere:*

- *strikte Errortraces:*
  $StET(S) = \{w \in \Sigma^* \mid q_0 \overset{w}{\Rightarrow} q \in E\}$,
- *gekürzte Errortraces:*
  $PrET(S) = \{prune(w) \mid w \in StET(S)\}$,
- *fehlende Input-Traces:* $MIT(S) = \{wa \in \Sigma^* \mid q_0 \overset{w}{\Rightarrow} q \wedge a \in I \wedge q \overset{a}{\not\rightarrow}\}$.

In der folgenden Definition halten wir fest, was wir als Errortraces auffassen. Diese Menge ist dadurch, dass sie die fortgesetzten Traces aus $PrET$ enthält deutlich allgemeiner wie die Menge $StET$. Zusätzlich definieren wir auch noch die geflutete Sprache, in der wir die Informationen aus der Sprache und den Errortraces vereinen und somit bei der Inklusion dann nicht mehr explizit unterscheiden.

Definition 2.4 (*Lokale Error Semantik*). *Sei S ein EIO.*

- *Die Menge der Errortraces von S ist $ET(S) := cont(PrET(S)) \cup cont(MIT(S))$.*
- *Die geflutete Sprache von S ist $EL(S) := L(S) \cup ET(S)$.*

Für zwei EIOs S_1, S_2 mit der gleichen Signatur schreiben wir $S_1 \sqsubseteq_E S_2$, wenn $ET(S_1) \subseteq ET(S_2)$ und $EL(S_1) \subseteq EL(S_2)$ gilt.

Der folgende Satz wurde in [BV14] nur für die Parallelkomposition mit verborgenen synchronisierten Aktionen formuliert, jedoch entspricht er dem analogen Satz aus [Sch12]. Da der Beweis jedoch ohne Beachtung von [Sch12] neu geführt wurde, werden hier eher auf die Erwähnung der Unterschiede zu [BV14] wert gelegt.

Satz 2.5 (*Lokale Error Semanik für Parallelkompositionen*). *Für zwei komponierbare EIOs S_1, S_2 und $S_{12} = S_1 \| S_2$, gilt:*

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

Beweis.

1. „$\subseteq$“:

Da beide Seiten der Gleichung unter der Fortsetzung *cont* abgeschlossen sind, genügt es ein präfix-minimales Element w von ET_{12} zu betrachten. Dieses Element ist aufgrund der Definition der Menge der Errortraces entweder in MIT_{12} oder in $PrET_{12}$ enthalten.

- Fall 1 ($w \in MIT_{12}$): Aus der Definition von MIT folgt, dass es eine Aufteilung $w = xa$ gibt mit $(q_{01}, q_{02}) \stackrel{x}{\Rightarrow} (q_1, q_2) \wedge a \in I_{12} \wedge (q_1, q_2) \stackrel{a}{\not\rightarrow}$. Da $I_{12} \stackrel{\text{Def}}{=} (I_1 \backslash O_2) \cup (I_2 \backslash O_1) = (I_1 \cup I_2) \backslash (O_1 \cup O_2)$ ist $a \in (I_1 \cup I_2)$ und $a \notin (O_1 \cup O_2)$. Somit müssen wir unterscheiden, ob $a \in (I_1 \cap I_2)$ oder $a \in (I_1 \cup I_2) \backslash (I_1 \cap I_2)$ ist. Diese Unterscheidung ist in [BV14] nicht nötig, da dort $I_1 \cap I_2 = \emptyset$ gilt, somit gibt es dort nur den Fall 1b). Der Fall 1a) wir auch in [Sch12] behandelt.
  - Fall 1a) ($a \in (I_1 \cap I_2)$): Nun können wir den Ablauf der Komposition auf die Transitionssysteme projizieren und erhalten dann oBdA $q_{01} \stackrel{x_1}{\Rightarrow} q_1 \stackrel{a}{\not\rightarrow}$ und $q_{02} \stackrel{x_2}{\Rightarrow} q_2 \stackrel{a}{\not\rightarrow}$ oder $q_{02} \stackrel{x_2}{\Rightarrow} q_2 \stackrel{a}{\rightarrow}$ mit $x \in x_1 \| x_2$. Daraus können wir $x_1 a \in cont(MIT_1) \subseteq ET_1 \subseteq EL_1$ und $x_2 a \in EL_2$ ($x_2 a \in MIT_2$ oder $x_2 a \in L_2$) folgern. Damit folgt $w \in (x_1 \| x_2) \cdot \{a\} \subseteq (x_1 a) \| (x_2 a) \subseteq ET_1 \| EL_2$, und somit ist w in der rechten Seite der Gleichung enthalten.
  - Fall 1b) ($a \in (I_1 \cup I_2) \backslash (I_1 \cap I_2)$): OBdA gilt $a \in I_1$. Durch Projektion erhalten wir: $q_{01} \stackrel{x_1}{\Rightarrow} q_1 \stackrel{a}{\not\rightarrow}$ und $q_{02} \stackrel{x_2}{\Rightarrow} q_2$ mit $x \in x_1 \| x_2$. Daraus folgt $x_1 a \in cont(MIT_1) \subseteq ET_1$ und $x_2 \in L_2 \subseteq EL_2$. Somit gilt $w \in (x_1 \| x_2) \cdot \{a\} \subseteq (x_1 a) \| x_2 \subseteq ET_1 \| EL_2$. Dies ist eine Teilmenge der rechten Seite der Gleichung.
- Fall 2 ($w \in PrET_{12}$): Durch die Definition von $PrET$ und *prune* wissen wir, dass es ein $v \in O_{12}^*$ gibt, so dass $(q_{01}, q_{02}) \stackrel{w}{\Rightarrow} (q_1, q_2) \stackrel{v}{\Rightarrow} (q_1', q_2')$ gilt mit $(q_1', q_2') \in E_{12}$ und $w = prune(wv)$. Durch Projektion erhalten wir $q_{01} \stackrel{w_1}{\Rightarrow} q_1 \stackrel{v_1}{\Rightarrow} q_1'$ und $q_{02} \stackrel{w_2}{\Rightarrow} q_2 \stackrel{v_2}{\Rightarrow} q_2'$ mit $w \in w_1 \| w_2$ und $v \in v_1 \| v_2$. Aus $(q_1', q_2') \in E_{12}$ folgt, dass es sich entweder um einen geerbten oder einen neuen Error handelt. Bei einem geerbten wäre bereits einer der beiden Zustände ein Error-Zustand gewesen. Der neue Error hingegen wäre durch die fehlende Möglichkeit entstanden eine synchronisierte Aktion auszuführen.
  - Fall 2a) (geerbter Error): OBdA $q_1' \in E_1$. Daraus folgt $w_1 v_1 \in StET_1 \subseteq cont(PrET_1) \subseteq ET_1$. Da gilt $q_{02} \stackrel{w_2 v_2}{\Rightarrow}$, erhalten wir $w_2 v_2 \in L_2 \subseteq EL_2$. Dadurch ergibt sich $wv \in ET_1 \| EL_2$ mit $w = prune(wu)$ und somit ist w in der rechten Seite der Gleichung enthalten.
  - Fall 2b) (neuer Error): OBdA $a \in I_1 \cap O_2$ mit $q_1' \stackrel{a}{\not\rightarrow} \wedge q_2' \stackrel{a}{\rightarrow}$. Daraus folgt $w_1 v_1 a \in MIT_1 \subseteq ET_1$ und $w_2 v_2 a \in L_2 \subseteq EL_2$. Damit ergibt sich $wva \in ET_1 \| EL_2$, da $a \in O_2 \subseteq O_{12}$ gilt $w = prune(wva)$ und somit ist w in der rechten Seite der Gleichung enthalten.

1. „$\supseteq$“:
Wegen der Abgeschlossenheit beider Seiten der Gleichung gegenüber *cont* betrachten wir auch in diesem Fall nur ein präfix-minimales Element $x \in prune((ET_1 \| EL_2) \cup (EL_1 \| ET_2))$. Da x durch die Anwendung der *prune*-Funktion entstanden ist, existiert ein $y \in O_{12}^*$ mit $xy \in (ET_1 \| EL_2) \cup (EL_1 \| ET_2)$. OBdA gehen wir davon aus, dass $xy \in ET_1 \| EL_2$ gilt, d.h. es gibt $w_1 \in ET_1$ und $w_2 \in EL_2$ mit $xy \in w_1 \| w_2$. In dem Punkt, dass wir das präfix-minimale Element noch mit Outputs fortsetzten können, unterscheidet sich dieser Beweis von dem in [Sch12]. Dort wird nicht weiter darauf eingegangen, dass die *prune*-Funktion hier noch zur Anwendung kommt, da wir jedoch später nur Präfixe von x betrachten werden, ist dieser Unterschied irrelevant.
Im Folgenden werden wir für alle Fälle von xy zeigen, dass es ein $v \in PrET(S_1 \| S_2) \cup MIT(S_1 \| S_2)$ gibt, das ein Präfix von xy ist und v entweder auf ein Input aus I_{12} endet oder $v = \varepsilon$. Da v entweder leer ist oder auf einen Input endet, muss v ein Präfix von x sein. ε ist Präfix von jedem Wort und sobald x mindestens einen Buchstaben enthält, kann y durch die Definition von *prune* nur aus Outputs besteht und somit muss das Ende von v vor dem Anfang von y liegen. x hat dadurch ein Präfix in $PrET(S_1 \| S_2) \cup MIT(S_1 \| S_2)$, dann ist x in der Fortsetzung dieser Menge enthalten und somit gilt $x \in ET_{12}$.
Sei v_1 das kürzeste Präfix von w_1 in $PrET_1 \cup MIT_1$. Falls $w_2 \in L_2$, so sei $v_2 = w_2$, sonst soll v_2 das kürzeste Präfix von w_2 in $PrET_2 \cup MIT_2$ sein. Jede Aktion in v_1 und v_2 hängt mit einer aus xy zusammen. Wir gehen nun davon aus, dass entweder $v_2 = w_2 \in L_2$ gilt oder die letzte Aktion von v_1 findet vor oder gleichzeitig mit der letzten Aktion von v_2 statt. Ansonsten endet $v_2 \in PrET_2 \cup MIT_2$ vor v_1 und somit ist dieser Fall analog zu v_1 endet vor v_2.

- Fall 1 ($v_1 = \varepsilon$): Dadurch dass $\varepsilon \in PrET_1 \cup MIT_1$, ist bereits in S_1 ein Error lokal erreichbar. $\varepsilon \in MIT_1$ ist nicht möglich, da jedes Element aus MIT nach Definition mindestens die Länge 1 haben muss. Wir wähle $v'_2 = v' = \varepsilon$, somit ist v'_2 ein Präfix von v_2.

- Fall 2 ($v_1 \neq \varepsilon$): Aufgrund der Definitionen von $PrET$ und MIT endet v_1 auf ein $a \in I_1$, d.h. $v_1 = v'_1 a$. v' sei das Präfix von xy, das mit der letzten Aktion von v_1 endet, d.h. mit a, und $v'_2 = v'|_{\Sigma_2}$. Falls $v_2 \in L_2$, dann ist v'_2 ein Präfix von v_2, da in diesem Fall in der Parallelkomposition kein Fehler möglich ist und somit maximal das gesamte Wort v_2 bereits in v' enthalten sein kann. Falls $v_2 \in PrET_2 \cup MIT_2$ gilt, dann ist durch die Annahme, dass v_2 nicht vor v_1 endet, v'_2 ein Präfix von v_2. Im Fall $v_2 \in MIT_2$ können wir durch die gleiche Argumentation ebenfalls schließen, dass v'_2 ein Präfix von v_2 ist. Wir wissen zusätzlich, dass v_2 auf $b \in I_2$ endet, jedoch muss nicht mehr wie in [BV14] $b \neq a$ gelten. Wir können also keine Aussage mehr darüber treffen, ob es sich um ein echtes Präfix handelt.

In allen Fällen erhalten wir $v_2' = v'|_{\Sigma_2}$ ist ein Präfix von v_2 und $v' \in v_1 \| v_2'$ ist ein Präfix von xy. Da nicht mehr $b \neq a$ gelten muss, können wir nicht mehr für alle Fälle $q_{02} \overset{v_2'}{\Rightarrow}$ folgern, wie das in [BV14] möglich war. Dies wurde auch bereits in [Sch12] festgestellt.

- Fall I ($v_1 \in MIT_1$ und $v_1 \neq \varepsilon$): Es gibt $q_{01} \overset{v_1'}{\Rightarrow} q_1 \overset{a}{\not\rightarrow}$ und sei $v' = v''a$. Bei der folgenden Fallunterscheidung müssen wir bezüglich [BV14] einen weiteren Fall (Ib)) einfügen, da es zulässig ist, das a sowohl in I_1, wie auch in I_2 enthalten ist.
  - Fall Ia) ($a \notin \Sigma_2$): Es gilt $q_{02} \overset{v_2'}{\Rightarrow} q_2$ mit $v'' \in v_1' \| v_2'$, da v_2' und v_2 nicht auf a enden können. Dadurch erhalten wir $(q_{01}, q_{02}) \overset{v''}{\Rightarrow} (q_1, q_2) \overset{a}{\not\rightarrow}$ mit $a \in I_{12}$. Somit können wir wählen $v := v''a = v' \in MIT_{12}$.
  - Fall Ib) ($a \in I_2$ und $v_2' \in MIT_2$): Es gilt $v_2' = v_2''a$ mit $q_{02} \overset{v_2''}{\Rightarrow} q_2 \overset{a}{\not\rightarrow}$ und $v'' \in v_1' \| v_2''$. a ist für S_2, ebenso wie für S_1, ein fehlender Input. Wir können somit schließen, dass $(q_1, q_2) \overset{a}{\not\rightarrow}$ gilt. Wir wählen $v := v''a = v' \in MIT_{12}$.
  - Fall Ic) ($a \in I_2$ und $v_2' \in L_2$): Es gilt $q_{02} \overset{v_2''}{\Rightarrow} q_2 \overset{a}{\rightarrow}$ mit $v_2' = v_2''a$. Da jedoch $a \in Synch(S_1, S_2)$ liegt, da die Menge der synchronisierten Aktionen bezüglich [BV14] erweitert wurde, somit reicht schon, dass $q_1 \overset{a}{\not\rightarrow}$ gilt, um folgendes schließen zu können $(q_1, q_2) \overset{a}{\not\rightarrow}$. Somit können wir hier $v := v''a = v' \in MIT_{12}$ wählen.
  - Fall Id) ($a \in O_2$): Es gilt $v_2' = v_2''a$ und $q_{02} \overset{v_2'}{\Rightarrow}$, da der Input b von v_2 hier nicht dem a entsprechen kann. Wir erhalten also $q_{02} \overset{v_2''}{\Rightarrow} q_2 \overset{a}{\rightarrow}$ mit $v'' \in v_1' \| v_2''$. Daraus ergibt sich $(q_{01}, q_{02}) \overset{v''}{\Rightarrow} (q_1, q_2)$ mit $q_1 \overset{a}{\not\rightarrow}, a \in I_1, q_2 \overset{a}{\rightarrow}, a \in O_2$, somit gilt $(q_1, q_2) \in E_{12}$. Wir wählen $v := prune(v'') \in PrET_{12}$.
- Fall II ($v_1 \in PrET_1$): $\exists u_1 \in O_1^* : q_{01} \overset{v_1}{\Rightarrow} q_1 \overset{u_1}{\Rightarrow} q_1'$ mit $q_1' \in E_1$. Da wir hier keine disjunkten Inputmengen mehr haben, kann das a, auf das v_1 endet ebenfalls der letzte Buchstabe von v_2 sein. Im Fall von $v_2 \in MIT_2$ kann somit $a = b$ gelten und somit wäre $v_2 = v_2'$. Dieser Fall verläuft jedoch analog zu Fall Ic) und wird somit hier nicht weiter betrachtet. Dieser Unterschied zu [BV14] wurde auch bereits in [Sch12] angemerkt. Es gilt somit für alle anderen Fälle hier $q_{02} \overset{v_2'}{\Rightarrow} q_2$ mit $(q_{01}, q_{02}) \overset{v'}{\Rightarrow} (q_1, q_2)$.
  - Fall IIa) ($u_2 \in (O_1 \cap I_2)^*, c \in (O_1 \cap I_2)$, sodass u_2c Präfix von $u_1|_{I_2}$ mit $q_2 \overset{u_2}{\Rightarrow} q_2' \overset{c}{\not\rightarrow}$): Für das Präfix $u_1'c$ von u_1 mit $u_1'c|_{I_2} = u_2c$ wissen wir, dass $q_1 \overset{u_1'}{\Rightarrow} q_1'' \overset{c}{\rightarrow}$. Somit gilt $u_1' \in u_1' \| u_2$ und $(q_1, q_2) \overset{u_1'}{\Rightarrow} (q_1'', q_2') \in E_{12}$, da für S_2 der entsprechende Input fehlt, der mit dem c Output von S_1 zu koppeln wäre. Es handelt sich also um einen neuen Error. Wir wählen $v := prune(v'u_1') \in PrET(S_1 \| S_2)$, dies ist ein Präfix von v', da $u_1 \in O_1^*$.

- Fall IIb) ($q_2 \overset{u_2}{\Rightarrow} q_2'$ mit $u_2 = u_1|_{I_2}$): Somit ist $u_1 \in u_1 \| u_2$ und $(q_1, q_2) \overset{u_1}{\Rightarrow} (q_1', q_2') \in E_{12}$, da $q_1' \in E_1$ und somit handelt es sich um einen geerbten Error. Wir wählen nun $v := prune(v'u_1) \in$
$PrET(S_1 \| S_2)$, das wiederum ein Präfix von v' ist.

2.:
Der Beweis für diesen Punkt musste bezüglich [BV14] nicht verändert werden, bis auf die Ersetzung der Zeichen der Parallelkomposition. Somit entspricht dieser Beweis dem aus [BV14] und dem aus [Sch12].
Es ist durch die Definition klar, dass gilt $L_i \subseteq EL_i$ und $ET_i \subseteq EL_i$. Wir beginnen mit der Argumentation von der rechten Seite der Gleichung aus:

$$\begin{aligned}
&(EL_1 \| EL_2) \cup ET_{12} \\
&\overset{2.4}{=} (L_1 \cup ET_1) \| (L_2 \cup ET_2) \cup ET_{12} \\
&= \underbrace{(L_1 \| ET_2)}_{\substack{\subseteq (EL_1 \| ET_2) \\ \overset{1.}{\subseteq} ET_{12}}} \cup \underbrace{(ET_1 \| L_2)}_{\substack{\subseteq (ET_1 \| EL_2) \\ \overset{1.}{\subseteq} ET_{12}}} \cup (L_1 \| L_2) \cup \underbrace{(ET_1 \| ET_2)}_{\substack{\subseteq (EL_1 \| ET_2) \\ \overset{1.}{\subseteq} ET_{12}}} \cup ET_{12} \\
&= (L_1 \| L_2) \cup ET_{12} \\
&\overset{1.5}{=} L_{12} \cup ET_{12} \\
&\overset{2.4}{=} EL_{12}
\end{aligned}$$

□

Die folgende Proposition wurde hier noch explizit mit Beweis eingefügt im Gegensatz zu den Ausführungen in [BV14], in denen diese Präkongruenz nur als Resultat aus dem letzten Satz erwähnt wird.

Proposition 2.6 (*Präkongruenz*). *$\sqsubseteq_E$ ist eine Präkongruenz.*

Beweis. Es muss gezeigt werden, wenn $S_1 \sqsubseteq_E S_2$ gilt, für jedes S_3 auch $S_3 \| S_1 \sqsubseteq_E S_3 \| S_1$ gilt. D.h. es ist zu zeigen, dass aus $ET_1 \subseteq ET_2$ und $EL_1 \subseteq EL_2$ folgt, $ET(S_3 \| S_1) \subseteq ET(S_3 \| S_2)$ und $EL(S_3 \| S_1) \subseteq EL(S_3 \| S_2)$.

- $$\begin{aligned} ET(S_3 \| S_1) &\overset{2.5\ 1.}{=} cont(prune((ET_3 \| EL_1) \cup (EL_3 \| ET_1))) \\ &\overset{\substack{ET_1 \subseteq ET_2 \\ \text{und} \\ EL_1 \subseteq EL_2}}{\subseteq} cont(prune((ET_3 \| EL_2) \cup (EL_3 \| ET_2))) \\ &\overset{2.5\ 1.}{=} ET(S_3 \| S_2) \end{aligned}$$

- $$\begin{aligned} EL(S_3 \| S_1) &\overset{2.5\ 2.}{=} (EL_3 \| EL_1) \cup E_{31} \\ &\overset{\substack{EL_1 \subseteq EL_2 \\ \text{und} \\ ET_{31} \subseteq ET_{32}}}{\subseteq} (EL_3 \| EL_2) \cup ET_{32} \\ &\overset{2.5\ 2.}{=} EL(S_3 \| S_2) \end{aligned}$$

□

In [BV14] wurde auch die Verfeinerung von EIOs als Relation betrachtet mit Spezifikation und Implementierung. Hier soll ebenfalls eine Verfeinerungsrelation über EIOs betrachtet werden, jedoch in leicht abgewandelter Form, da die synchronisierten Aktionen nicht verborgen werden sollen. Dadurch ändern sich natürlich auch Teile des Beweises, vor allem muss statt mit
$StET$ mit der Menge
$PrET$ argumentiert werden. Dieses Lemma existiert in dieser Form nicht in [Sch12], da es dort mit der Aussage von Satz 2.8 kombiniert wurde. Jedoch ist die Aussage diese Lemmas trotzdem Teil dessen, was gezeigt wird und somit finden sich die Teile dieses Beweises auch dort wieder.

Lemma 2.7 (*Verfeinerung mit Errors*). *Gegeben sind zwei EIOs S_1 und S_2 mit der gleichen Signatur. Wenn alle EIOs U, für die S_2 und U gut kommunizieren auch S_1 und U gut kommunizieren, dann verfeinert S_1 den EIO S_2. Diese Verfeinerung entspricht der Relation $\sqsubseteq_E$ von oben: Wenn $U \| S_1 \sqsubseteq_E^B U \| S_2$ für alle U, dann gilt $S_1 \sqsubseteq_E S_2$.*

Beweis. Da S_1 und S_2 die gleichen Signaturen haben, definieren wir: $I := I_1 = I_2$ und $O := O_1 = O_2$. Für jeden der Partner U gilt $I_U = O$ und $O_U = I$.
Um zu zeigen, dass $S_1 \sqsubseteq_E S_2$ wird nachgeprüft, dass gilt:

- $ET(S_1) \subseteq ET(S_2)$,
- $EL(S_1) \subseteq EL(S_2)$.

Wir wählen ein präfix-minimales Element $w \in ET(S_1)$ und zeigen, dass dieses w oder eines seiner Präfixe in $ET(S_2)$ enthalten ist. Dies ist möglich, da beide Mengen durch *cont* abgeschlossen sind.

- Fall 1 ($w = \varepsilon$): Es handelt sich um einen lokal erreichbaren Error in S_1. Wir nehmen für U ein Transitionssystem, der nur aus dem Startzustand und einer Schleife für alle Inputs $x \in I_U$ besteht. Somit kann S_1 die gleichen Error-Zustände lokal erreichen wie $U \| S_1$. Daraus folgt, dass auch $U \| S_2$ einen lokal erreichbaren Error-Zustand haben muss. Durch unsere Definition von U kann dieser Fehler nur von S_2 geerbt werden. Es muss also in S_2 ein Error-Zustand durch interne Aktionen und Outputs erreichbar sein, d.h. es gilt $\varepsilon \in$
$PrET(S_2)$.
- Fall 2 ($w = x_1 \ldots x_n x_{n+1} \in \Sigma^+$ mit $n \geq 0$ und $x_{n+1} \in I$): Wir betrachten die folgenden Partner U (siehe auch Abbildung 2.1):
  - $Q_U = \{q_0, q_1, \ldots, q_{n+1}\}$
  - $q_{0U} = q_0$
  - $E_U = \emptyset$

– $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i \leq n\}$
$\cup \{(q_i, x, q_{n+1}) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i \leq n\}$
$\cup \{(q_{n+1}, x, q_{n+1}) \mid x \in I_U\}$

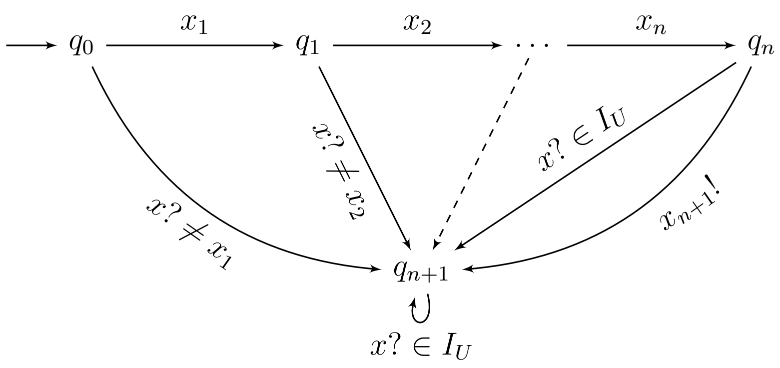

Abbildung 2.1: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$

Wir können für w zwei Fälle unterscheiden. Beide führen zu $\varepsilon \in PrET(U \| S_1)$. Dieses Resultat unterscheidet sich von dem in [BV14], da hier die synchronisierten Aktionen als Outputs vorhanden bleiben und somit kann nicht $\varepsilon \in StET(U \| S_1)$ gelten.

- Fall 2a) ($w \in MIT(S_1)$): In $U \| S_1$ erhalten wir $(q_0, q_{01}) \overset{x_1 \dots x_n}{\Rightarrow} (q_n, q')$ mit $q' \overset{x_{n+1}}{\not\to}$ und $q_n \overset{x_{n+1}}{\to}$. Deshalb gilt $(q_n, q') \in E_{U \| S_1}$ und $x_1 \dots x_n \in StET(U \| S_1)$. Da alle Aktionen aus w bis auf x_{n+1} synchronisiert werden gilt $x_1, \dots, x_n \in O_{U \| S_1}$. Daraus ergibt sich dann $\varepsilon \in PrET(U \| S_1)$.
- Fall 2b) ($w \in PrET(S_1)$): In $U \| S_1$ erhalten wir $(q_0, q_{01}) \overset{w}{\Rightarrow} (q_{n+1}, q'') \overset{u}{\Rightarrow} (q_{n+1}, q')$ für $u \in O^*$ und $q' \in E_1$. Daraus folgt $(q_{n+1}, q') \in E_{U \| S_1}$ und somit $wu \in StET(U \| S_1)$. Da alle Aktionen aus w synchronisiert werden gilt $x_1, \dots, x_n, x_{n+1} \in O_{U \| S_1}$ und da $u \in O^*$ folgt $u \in O^*_{U \| S_1}$. Somit ergibt sich $\varepsilon \in PrET(U \| S_1)$.

Da wir wissen, dass $\varepsilon \in PrET(U \| S_1)$ gilt, können wir durch $U \| S_1 \sqsubseteq_E^B U \| S_2$ schließen, dass auch in $U \| S_2$ ein Error lokal erreichbar sein muss.
Dieser Error kann geerbt oder neu sein.

- Fall 2i) (neuer Error): Da jeder Zustand von U alle Inputs $x \in O = I_U$ zulässt, muss ein lokal erreichbarer Error einer sein, bei dem ein Output $a \in O_U$ von U möglich ist, der nicht mit einem passenden Input aus S_2 synchronisiert werden kann. Durch die Konstruktion von U sind in q_{n+1} keine Outputs möglich.

Ein neuer Error muss also die Form (q_i, q') haben mit $i \leq n, q' \overset{x_{i+1}}{\not\rightarrow}$ und $x_{i+1} \in O_U = I$. Durch Projektion erhalten wir dann $q_{02} \overset{x_1 \ldots x_i}{\Rightarrow} q' \overset{x_{i+1}}{\not\rightarrow}$ und damit gilt $x_1 \ldots x_{i+1} \in MIT(S_2) \subseteq ET(S_2)$. Somit ist ein Präfix von w in $ET(S_2)$ enthalten.

- Fall 2ii) (geerbter Error): U hat $x_1 \ldots x_i u$ ausgeführt mit $u \in I_U^* = O^*$ und ebenso hat S_2 diesen Weg ausgeführt. Durch dies hat S_2 einen Zustand in E_2 erreicht, da von U keine Fehler geerbt werden können. Es gilt dann $prune(x_1 \ldots x_i u) = prune(x_1 \ldots x_i) \in$
  $PrET(S_2) \subseteq ET(S_2)$. Da $x_1 \ldots x_i$ ein Präfix von w ist, führt auch in diesem Fall ein Präfix von w zu einem Error.

Um uns von der zweiten Inklusion zu überzeugen reicht es aufgrund der ersten Inklusion und der Definition von EL, zu zeigen, dass $L(S_1) \backslash ET(S_1) \subseteq EL(S_2)$ gilt.
Wir nehmen uns dafür ein beliebiges $w \in L(S_1) \backslash ET(S_1)$ und zeigen, dass es in $EL(S_2)$ enthalten ist.

- Fall 1 ($w = \varepsilon$): Da ε immer in $EL(S_2)$ enthalten ist, haben wir hier nichts zu zeigen.

- Fall 2 ($w = x_1 \ldots x_n$ mit $n \geq 1$): Wir konstruieren einen Partner U wie folgt (siehe dazu auch Abbildung 2.2):
  - $Q_U = \{q, q_0, q_1, \ldots, q_n\}$
  - $q_{0U} = q_0$
  - $E_U = q_n$
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
    $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i \leq n\}$
    $\cup \{(q, x, q) \mid x \in I_U\}$

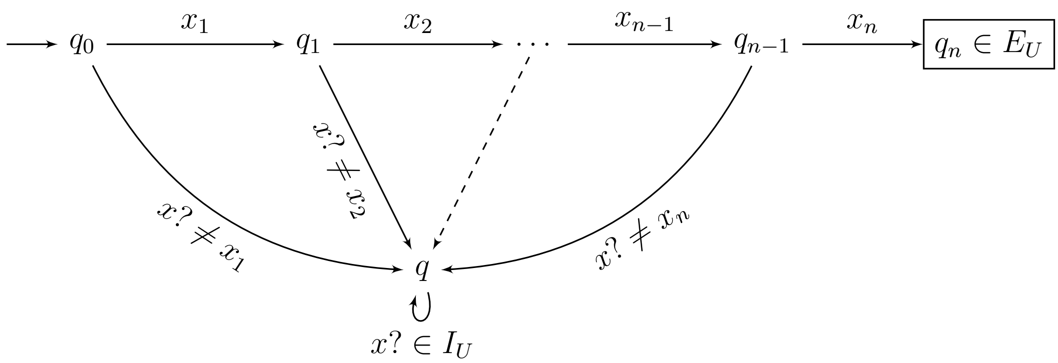

Abbildung 2.2: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$, q_n ist der einzige Error-Zustand

Da $q_{01} \overset{w}{\Rightarrow} q'$ gilt, wissen wir, dass $U \| S_1$ einen lokal erreichbaren Error hat. Somit muss $U \| S_2$ ebenfalls einen lokal erreichbaren Error haben.

- Fall 2a) (neuer Error aufgrund von $x_i \in O_U$ und $q_{02} \stackrel{x_1 \dots x_{i-1}}{\Rightarrow} q'' \stackrel{x_i}{\not\rightarrow}$): Es gilt $x_1 \dots x_i \in MIT(S_2)$ und somit $w \in EL(S_2)$. Anzumerken ist, dass nur auf diesem Weg Outputs von U möglich sind, deshalb gibt es keine anderen Outputs von U, die zu einem neuen Fehler führen können.
- Fall 2b) (neuer Error aufgrund von $a \in O_2$): Der einzige Zustand, in dem U nicht alle Inputs erlaubt sind, ist q_n, der bereits ein Error-Zustand ist. Falls dieser Zustand erreichbar ist in $U \| S_2$, dann besitzt der komponierte EIO einen geerbten Error und es gilt $w \in L(S_2) \subseteq EL(S_2)$.
- Fall 2c) (geerbter Error von U): Da der einzige Zustand aus E_U q_n ist und alle Aktionen synchronisiert sind, ist dies nur möglich, wenn gilt $q_{02} \stackrel{x_1 \dots x_n}{\Rightarrow}$. In diesem Fall gilt, wie im letzten, $w \in L(S_2) \subseteq EL(S_2)$.
- Fall 2d) (geerbter Error von S_2): Es gilt dann $q_{02} \stackrel{x_1 \dots x_i u}{\Rightarrow} q' \in E_2$ für $i \geq 0$ und $u \in O^*$. Somit ist $x_1 \dots x_i u \in StET(S_2)$ und damit $prune(x_1 \dots x_i u) = prune(x_1 \dots x_i) \in PrET(S_2) \subseteq EL(S_2)$. Somit gilt $w \in EL(S_2)$.

□

Satz 2.8 (*Full Abstractness für lokale Error Semanik*). *Seien S_1 und S_2 zwei EIOs mit derselben Signatur. Dann gilt $S_1 \sqsubseteq_E^C S_2 \Leftrightarrow S_1 \sqsubseteq_E S_2$, insbesondere ist $\sqsubseteq_E$ eine Präkongruenz.*

Beweis. Wie bereits in Korollar 2.6 festgehalten, ist $\sqsubseteq_E$ eine Präkongruenz.

„$\Leftarrow$“: Im Beweis von Lemma 2.7 wurde bereits festgestellt, wenn $\varepsilon \in ET(S)$ gilt, ist ein Error lokal erreichbar in S. Somit impliziert $S_1 \sqsubseteq_E S_2$, dass $\varepsilon \in ET(S_2)$ gilt, wenn $\varepsilon \in ET(S_1)$. Dadurch folgt ebenfalls, dass $S_1 \sqsubseteq_E^B S_2$ gilt. Somit folgt aus $S_1 \sqsubseteq_E S_2$ der relationale Zusammenhang $S_1 \sqsubseteq_E^C S_2$.

„$\Rightarrow$“: Durch die Definition von $\sqsubseteq_E^C$ folgt aus $S_1 \sqsubseteq_E^C S_2$, dass $U \| S_1 \sqsubseteq_E^C U \| S_2$ für alle EIOs U, die mit S_1 komponierbar sind. Somit folgt auch die Gültigkeit von $U \| S_1 \sqsubseteq_E^B U \| S_2$ für alle diese EIOs U. Mit Lemma 2.7 folgt dann $S_1 \sqsubseteq_E S_2$. □

Aus Satz 2.8 und Lemma 2.7 erhalten wir das folgende Korollar.

Korollar 2.9. *Ein EIO S_1 verfeinert einen EIO S_2 genau dann, wenn für alle EIOs U für die S_2 gut mit U kommuniziert folgt S_1 kommuniziert ebenfalls gut mit U.*
Dies lässt sich formal wie folgt ausdrücken: $S_1 \sqsubseteq_E S_2 \Leftrightarrow U \| S_1 \sqsubseteq_E^B U \| S_2$ für alle Partner U.

3 Verfeinerung über Error- und Quiescenttraces

In diesem Kapitel werden wir und nicht nur um die Erreichbarkeit von Error-Zuständen kümmern, sondern auch um die Erreichbarkeit von Quiescent-Zuständen. Wir werden dabei ähnlich vorgehen wie im letzten Kapitel, jedoch halten wir uns als Quelle an [CJK13]. Darin werden ähnliche Konzepte beschrieben, jedoch aus Sicht der Traces.

Definition 3.1 (*Quiescent*). *Ein Quiescent-Zustand ist ein Zustand in einem EIO der keine Outputs besitzt oder ein Zustand, von dem aus über eine interne Handlung* τ *ein Zustand erreicht werden kann, der keine Outputs zulässt.*
Somit ist die Menge der Quiescent-Zustände in einem EIO wie folgt formal definiert: $Qui := \{q \in Q \mid q \in K \vee \exists p \in Q : q \overset{\tau}{\Rightarrow} p \in K\}$ *mit* $K := \{q \in Q \mid \forall a \in O : q \overset{a}{\not\rightarrow}\}$.

Für die Erreichbarkeit verwenden wir wie im letzten Kapitel wieder den optimistischen Ansatz der lokalen Erreichbarkeit.

Definition 3.2 (*lokal error- und quiescentfreie Kommunikation*). *Zwei EIOs* S_1 *und* S_2 *kommunizieren gut, wenn keine Errors und Quiescents lokal erreichbar sind in ihrer Parallelkomposition* $S_1 \| S_2$

Definition 3.3 (*lokale Basisrelation*). *Für EIOs* S_1 *und* S_2 *mit der gleichen Signatur schreiben wir* $S_1 \sqsubseteq^B_{Qui} S_2$*, wenn ein Error oder Quiescent in* S_1 *nur dann lokal erreichbar ist, wenn er auch in* S_2 *lokal erreichbar ist.*
$\sqsubseteq^C_{Qui}$ *bezeichnet die vollständig abstrakte Präkongruenz von* $\sqsubseteq^B_{Qui}$ *bezüglich* $\cdot\|\cdot$.

Definition 3.4 (*Error und Quiescenttraces*). *Sei S ein EIO und definiere:*

- *die Traces bezüglich Errors entsprechen denen aus 2.3,*
- *strickte Quiescenttraces:* $StQT(S) := \{w \in \Sigma^* \mid q_0 \overset{w}{\Rightarrow} q \in Qui\}$.

Definition 3.5 (*Lokale Error und Quiescent Semantik*). *Sei S ein EIO.*

- *Die Menge der Errortraces ist wie in 2.4 definiert.*
- *Die Menge der Quiescenttraces von S ist* $QT(S) := StQT(S)$.
- *Die geflutete Sprache von S ist* $QL(S) := L(S) \cup ET(S)$ *und entspricht somit der gefluteten Sprache* $EL(S)$ *in 2.4.*

Für zwei EIOs S_1, S_2 mit der gleichen Signatur schreiben wir $S_1 \sqsubseteq_{Qui} S_2$, wenn $ET(S_1) \subseteq ET(S_2)$, $QT(S_1) \subseteq QT(S_2)$ und $QL(S_1) \subseteq QL(S_2)$ gilt.

Da QT nur die strickten Quiescenttraces enthält, also die Traces, die im Automaten vorhanden sind und zu einem Quiescent-Zustand führen, gilt $QT \subseteq L \subseteq QL$. Somit musste QT nicht mehr explizit in die Definition von QL aufgenommen werden und trotzdem ist jeder Traces, der in QT möglich ist auch in der gefluteten Sprache enthalten.

Satz 3.6 (*Lokale Error und Quiescent Semantik für Parallelkompositonen*). *Für zwei komponierbare EIOs S_1, S_2 und $S_{12} = S_1 \| S_2$ gilt:*

1. $ET_{12} = cont(prune((ET_1 \| QL_2) \cup (QL_1 \| ET_2)))$,
2. $QT_{12} = (QT_1 \| QT_2)$,
3. $QL_{12} = (QL_1 \| QL_2) \cup ET_{12}$.

Beweis.

1.:
Der Beweis diese Punktes entspricht dem Beweis von Punkt 1. im Beweis von Satz 2.5.
2. "$\subseteq$":
Wir betrachten $w \in StQT_{12}$ und versuchen dessen Zugehörigkeit zur rechten Menge zu zeigen. Aufgrund von Definition 3.4 wissen wir es gilt $(q_{01}, q_{02}) \overset{w}{\Rightarrow} (q_1, q_2)$ mit $(q_1, q_2) \in Qui_{12}$. Durch Projektion erhalten wir $q_{01} \overset{w_1}{\Rightarrow} q_1$ und $q_{02} \overset{w_2}{\Rightarrow} q_2$ mit $w \in w_1 \| w_2$. Aus $(q_1, q_2) \in Qui_{12}$ können wir folgern, dass bereits $q_1 \in Qui_1$ und $q_2 \in Qui_2$ gilt. Somit gilt $w_1 \in StQT_1 \subseteq QT_1$ und $w_2 \in StQT_2 \subseteq QT_2$. Daraus folgt dann $w \in QT_1 \| QT_2$ und somit ist w in der rechten Seiten der Gleichung enthalten.
2. "$\supseteq$":
Für diese Inklusionsrichtung betrachten wir ein Element $w \in QT_1 \| QT_2$ und zeigen, dass es in der linken Menge enthalten ist. Da $QT_i = StQT_i$ gilt, existieren w_1 und w_2 für die gilt $q_{01} \overset{w_1}{\Rightarrow} q_1 \in Qui_1$ und $q_{02} \overset{w_2}{\Rightarrow} q_2 \in Qui_2$ mit $w \in w_1 \| w_2$. Da für die Zustände q_1 und q_2 die Zugehörigkeit zur Quiescent Menge gilt, können wir folgern, dass der aus ihnen zusammengesetzte Zustand in der Parallelkomposition ebenfalls keine Outputs zulässt. Somit gilt also für die Komposition $(q_{01}, q_{02}) \overset{w}{\Rightarrow} (q_1, q_2) \in Qui_{12}$ und dadurch ist w in der linken Seite der Gleichung enthalten.
3.:
Der Beweis diese Punktes entspricht dem Beweis von Punkt 2. im Beweis von Satz 2.5.
$\square$

QT_{12} ist hier in QL_{12} enthalten, da wie wir bereits oben festgestellt haben $QT_i \subseteq QL_i$ für $i = 1, 2$ gilt. Somit gilt auch $(QT_1 \| QT_2) \subseteq (QL_1 \| QL_2)$.

Proposition 3.7 (*Präkongruenz*). *$\sqsubseteq_{Qui}$ ist eine Präkongruenz.*

Beweis. Es muss gezeigt werden, wenn $S_1 \sqsubseteq_{Qui} S_2$ gilt, für jedes S_3 auch $S_3 \| S_1 \sqsubseteq_{Qui} S_3 \| S_2$ gilt. D.h. es ist zu zeigen, dass aus $ET_1 \subseteq ET_2$, $QT_1 \subseteq QT_2$ und $QL_1 \subseteq QL_2$ folgt, $ET_{31} \subseteq ET_{32}$, $QT_{31} \subseteq QT_{32}$ und $QL_{31} \subseteq QL_{32}$.

- $ET_{31} \overset{\text{Beweis 2.6 Punkt 1}}{\subseteq} ET_{32}$
- $QT_{31} \overset{3.6\ 2.}{=} (QT_3 \| QT_1)$
  $\overset{QT_1 \subseteq QT_2}{\subseteq} (QT_3 \| QT_2)$
  $\overset{3.6\ 2.}{=} QT_{32}$
- $QL_{31} \overset{\text{Beweis 2.6 Punkt 2}}{\subseteq} QL_{32}$

□

Lemma 3.8 (*Verfeinerung mit Quiescents*). *Gegeben sind zwei EIOs S_1 und S_2 mit der gleichen Signatur. Wenn alle EIOs U für die S_2 und U gut kommunizieren auch S_1 und U gut kommunizieren, dann verfeinert S_1 den EIO S_2. Diese Verfeinerung entspricht der Relation $\sqsubseteq_{Qui}$ von oben: Wenn $U \| S_1 \sqsubseteq_{Qui}^{B} U \| S_2$ für alle U, dann gilt $S_1 \sqsubseteq_{Qui} S_2$.*

Beweis. Da wir davon ausgehen, dass S_1 und S_2 die gleiche Signatur haben, definieren wir $I := I_1 = I_2$ und $O := O_1 = O_2$. Für jeden Partner U gilt $I_U = O$ und $O_U = I$.
Um zu zeigen, dass die Relation $S_1 \sqsubseteq_{Qui} S_2$ gilt, müssen wir die folgenden Punkte nachweisen:

- $ET(S_1) \subseteq ET(S_2)$,
- $QT(S_1) \subseteq QT(S_2)$,
- $QL(S_1) \subseteq QL(S_2)$.

Der erste und der letzte Punkt wurden bereits in Lemma 2.7 gezeigt. So bleibt uns nur noch die Inklusion $QT_1 \subseteq QT_2$ zu zeigen. □

Literaturverzeichnis

[BV14] Ferenc Bujtor und Walter Vogler, *Error-Pruning in Interface Automata*, preprint, Universität Augsburg, 2014.

[CJK13] Chris Chilton, Bengt Jonsson, und Marta Z. Kwiatkowska, *An Algebraic Theory of Interface Automata*, preprint, University of Oxford, 2013.

[Sch12] Christoph Franz Schlosser, *EIO-Automaten mit Parallelkomposition ohne Internalisierung*, Bachelorarbeit, Universität Augsburg, 2012.