

# 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 Inputaktionen 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 Handlungen.

## 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 beschriftet. Ebenfalls zulässig ist eine Kantenbeschriftung mit einem  $\tau$ , das dann eine interne, unbeobachtbare Aktion darstellt. Diese interne Aktion lässt also keine Interaktion mit der Umwelt zu. In vielen Fällen entstehen sie dadurch, dass die Inputs und Outputs dieses Übergangs verborgen wurden, da diese in einer Komposition synchronisiert wurden.

**Definition 1.1 (*Error-IO-Transitionssystem*).** *Ein Error-IO-Transitionssystem (EIO) ist als Tupel  $S = (Q, I, O, \delta, q_0, E)$  definiert, 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 Handlungsmenge 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 Handlung, bei der nicht festgelegt ist, ob sie ein Input oder ein Output ist.

Um die Komponenten den entsprechenden Automaten zuzuordnen, werden für die Komponenten die gleichen Indizes wie für ihre zugehörigen Automaten verwendet, z.B. für die Inputmenge des Automaten  $S_1$  schreiben wir  $I_1$ . Diese Notation verwenden wir später auch analog für die Sprachen, die einem Automaten zugeordnet sind.

Die Elemente der Übergangsrelation  $\delta$  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  $p \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \dots \xrightarrow{a_n} q$  mit  $w \in (\Sigma \cup \{\tau\})^*$ ,  $w = a_1 a_2 \dots a_n$ ,
- $p \xrightarrow{w}$  für  $p \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \dots \xrightarrow{a_n} p_n$  mit  $w \in (\Sigma \cup \{\tau\})^*$ ,  $w = a_1 a_2 \dots a_n$ ,
- $w|_B$  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  $B$ ,
- $p \xRightarrow{w} q$  für  $w \in \Sigma^*$  mit  $\exists w' \in (\Sigma \cup \{\tau\})^* : w'|_\Sigma = w \wedge p \xrightarrow{w'} q$ ,
- $p \xRightarrow{w}$  für  $\exists q : p \xRightarrow{w} q$ .

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

## 1.2 Parallelkomposition

Zwei EIOs sind komponierbar, wenn ihre Outputaktionsmengen disjunkt sind. Die Error-Zustände der Parallelkomposition setzen 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).

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

- $Q = Q_1 \times Q_2$
- $I = (I_1 \setminus O_2) \cup (I_2 \setminus O_1)$
- $O = O_1 \cup O_2$
- $q_0 = (q_{01}, q_{02})$
- $\delta = \{((q_1, q_2), \alpha, (p_1, p_2)) \mid (q_1, \alpha, p_1) \in \delta_1, \alpha \in (\Sigma_1 \cup \{\tau\}) \setminus \text{Synch}(S_1, S_2)\} \cup$   
 $\{((q_1, q_2), \alpha, (q_1, p_2)) \mid (q_2, \alpha, p_2) \in \delta_2, \alpha \in (\Sigma_2 \cup \{\tau\}) \setminus \text{Synch}(S_1, S_2)\} \cup$   
 $\{((q_1, q_2), \alpha, (p_1, p_2)) \mid (q_1, \alpha, p_1) \in \delta_1, (q_2, \alpha, p_2) \in \delta_2, \alpha \in \text{Synch}(S_1, S_2)\}$
- $E = (Q_1 \times E_2) \cup (E_1 \times Q_2) \cup$   
 $\left. \begin{array}{l} \{(q_1, q_2) \mid \exists a \in O_1 \cap I_2 : q_1 \xrightarrow{a} \wedge q_2 \not\xrightarrow{a}\} \cup \\ \{(q_1, q_2) \mid \exists a \in I_1 \cap O_2 : q_1 \not\xrightarrow{a} \wedge q_2 \xrightarrow{a}\} \end{array} \right\}$ 

geerbte Errors

neue Errors

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

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

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

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

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

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

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

Für zwei komponierbare EIOs  $S_1, S_2$  ist ein Ablauf ihrer Parallelkomposition  $S_1 \parallel S_2$  eine Transitionsfolge der Form  $(p_1, p_2) \xrightarrow{w} (q_1, q_2)$  für ein  $w \in \Sigma_{12}^*$ . So ein Ablauf kann auf Abläufe von  $S_1$  und  $S_2$  projiziert werden. Diese Projektion erfüllen  $p_i \xrightarrow{w_i} q_i$  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 genau einem Ablauf  $S_1 \parallel S_2$ , der ebenfalls wie oben aufgebaut ist. Formuliert ist dies in folgendem Lemma.

**Lemma 1.5 (*Sprache der Parallelkomposition*).** Für zwei komponierbare EIOs  $S_1$  und  $S_2$  gilt:  $L_{12} := L(S_1 \parallel S_2) = L(S_1) \parallel L(S_2).$

## 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  $\tau$  und den Outputaktionen bezeichnen wir hier als lokale Aktionen. Alle Elemente aus dieser Menge können ausgeführt werden ohne weiteres Zutun von außen, somit kann auch nicht beeinflusst werden ob diese Übergänge genommen werden oder nicht. Es besteht also die Möglichkeit, dass der EIO einen Error-Zustand übergeht, sobald dieser lokal erreichbar ist. Diese Art der Erreichbarkeit von Fehler wird auch in Kapitel 3 von [BV14] dargestellt.

**Definition 2.1 (lokal errorfreie Kommunikation).** Ein Error ist lokal erreichbar in einem EIO  $S$ , wenn  $\exists w \in O^* : q_0 \xrightarrow{w} q \in E$ .

Zwei EIOs  $S_1, S_2$  kommunizieren gut, wenn keine lokalen Errors erreicht werden können in ihrer Parallelkomposition  $S_1 \parallel S_2$ .

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  $\parallel$ .

Um uns nun näher mit den Präkongruenzen auseinandersetzen zu können, müssen wir bestimmte Traces aus unseren Automaten hervorheben. Die strikten Errortraces sind Wege, die direkt vom Startzustand zu einem Zustand in  $E$  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 vorgesetzt werden können. Diese führen zwar nicht direkt zu einem Fehler, jedoch in Komposition mit einem anderen Automaten sind dies gefährdete Stellen für neue Errors, da für die Synchronisation dieser Input fehlt.

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

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

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

- Die Menge der Errortraces von  $S$  ist  $ET(S) = cont(PrT(S)) \cup cont(MIT(S))$ .
- 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.

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

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

*Beweis.*

1. " $\subseteq$ ":

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

- Fall 1 ( $w \in MIT_{12}$ ): Aus der Definition von  $MIT$  folgt, dass es eine Aufteilung  $w = xa$  gibt mit  $(q_{01}, q_{02}) \xrightarrow{x} (q_1, q_2) \wedge a \in I_{12} \wedge (q_1, q_2) \not\xrightarrow{a}$ . Da  $I_{12} \stackrel{Def}{=} (I_1 \setminus O_2) \cup (I_2 \setminus O_1) = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$  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  $a \in (I_1 \cup I_2) \setminus (I_1 \cap I_2)$  ist.
  - Fall 1a) ( $a \in (I_1 \cap I_2)$ ): Nun können wir den Ablauf der Komposition auf die Automaten projizieren und erhalten dann:  $q_{01} \xrightarrow{x_1} q_1 \not\xrightarrow{a}$  und  $q_{02} \xrightarrow{x_2} q_2 \not\xrightarrow{a}$  mit  $x \in x_1 \parallel x_2$ , sobald einer der Automaten einen Übergang für  $a$  machen könnte, wäre dies auch für die Komposition der beiden möglich. Daraus können wir folgern  $x_1 a \in cont(MIT_1) \subseteq ET_1 \subseteq EL_1$  und  $x_2 \in cont(MIT_2) \subseteq ET_2 \subseteq EL_2$ . Damit folgt  $w \in (x_1 \parallel x_2) \cdot \{a\} \subseteq x_1 a \parallel x_2 a \subseteq ET_1 \parallel ET_2 \subseteq (ET_1 \parallel EL_2) \cup (EL_1 \parallel ET_2)$ , und somit ist  $w$  in der rechten Seite der Gleichung enthalten.
  - Fall 1b) ( $a \in (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ ): OBdA gilt  $a \in I_1$ . Durch Projektion erhalten wir:  $q_{01} \xrightarrow{x_1} q_1 \not\xrightarrow{a}$  und  $q_{02} \xrightarrow{x_2} q_2$  mit  $x \in x_1 \parallel x_2$ . Daraus folgt  $x_1 a \in cont(MIT_1) \subseteq ET_1$  und  $x_2 \in L_2 \subseteq EL_2$ . Somit gilt  $w \in (x_1 \parallel x_2) \cdot \{a\} \subseteq x_1 a \parallel x_2 \subseteq ET_1 \parallel EL_2$ . Dies ist eine Teilmenge der rechten Seite der Gleichung.
- Fall 2 ( $w \in PrT_{12}$ ): Durch die Definition von  $PrT$  und  $prune$  wissen wir, dass es ein  $v \in O_{12}^*$  gibt, so dass  $(q_{01}, q_{02}) \xrightarrow{w} (q_1, q_2) \xrightarrow{v} (q'_1, q'_2)$  gilt mit  $(q'_1, q'_2) \in E_{12}$  und  $w = prune(wv)$ . Durch Projektion erhalten wir  $q_{01} \xrightarrow{w_1} q_1 \xrightarrow{v_1} q'_1$  und  $q_{02} \xrightarrow{w_2} q_2 \xrightarrow{v_2} q'_2$  mit  $w \in w_1 \parallel w_2$  und  $v \in v_1 \parallel v_2$ . Aus  $(q'_1, q'_2) \in E_{12}$  folgt, dass entweder einer der beiden Zustände bereits ein Error-Zustand gewesen sein muss und der Fehler somit geerbt ist oder dass der Error durch die fehlende Möglichkeit entstanden ist eine synchronisierte Handlung auszuführen, da der Input nicht möglich war und es sich somit um einen neuen Fehler handelt.

- Fall 2a) (geerbter Error): OBdA  $q'_1 \in E_1$ . Daraus folgt  $w_1v_1 \in StT_1 \subseteq cont(PrT_1) \subseteq ET_1$ . Da gilt  $q_{02} \xrightarrow{w_2v_2}$ , erhalten wir  $w_2v_2 \in L_2 \subseteq EL_2$ . Dadurch ergibt sich  $wv \in ET_1 \parallel EL_2$  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  $q'_1 \xrightarrow{a} \wedge q'_2 \xrightarrow{a}$ . Daraus folgt  $w_1v_1a \in MIT_1 \subseteq ET_1$  und  $w_2v_2a \in L_2 \subseteq EL_2$ . Damit ergibt sich  $wva \in ET_1 \parallel EL_2$ , 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  $x \in prune((ET_1 \parallel EL_2) \cup (EL_1 \parallel ET_2))$ . Da  $x$  durch die Anwendung der  $prune$ -Funktion entstanden ist, existiert ein  $y \in O_{12}$  mit  $xy \in (ET_1 \parallel EL_2) \cup (EL_1 \parallel ET_2)$ . OBdA gehen wir davon aus, dass  $xy \in ET_1 \parallel EL_2$  gilt, d.h. es gibt  $w_1 \in ET_1$  und  $w_2 \in EL_2$  mit  $xy \in w_1 \parallel w_2$ .

Weiterführend werden wir für alle Fälle von  $xy$  zeigen, dass es ein  $v \in PrT(S_1 \parallel S_2) \cup MIT(S_1 \parallel S_2)$  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.  $\varepsilon$  ist Präfix von jedem Wort und sobald  $x$  mindestens einen Buchstaben enthält, kann  $y$  durch die Definition von  $prune$  nur aus Outputs bestehen und somit muss das Ende von  $v$  vor dem Anfang von  $y$  liegen.  $x$  hat dadurch ein Präfix in  $PrT(S_1 \parallel S_2) \cup MIT(S_1 \parallel S_2)$ , 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  $PrT_1 \cup MIT_1$ . Falls  $w_2 \in L_2$ , so sei  $v_2 = w_2$ , sonst soll  $v_2$  das kürzeste Präfix von  $w_2$  in  $PrT_2 \cup MIT_2$  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  $v_2 \in PrT_2 \cup MIT_2$  vor  $v_1$  und somit ist dieser Fall analog zu  $v_1$  endet vor  $v_2$ .

- Fall 1 ( $v_1 = \varepsilon$ ): Dadurch dass  $\varepsilon \in PrT_1 \cup MIT_1$ , ist bereits in  $S_1$  ein Error lokal erreichbar. 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  $PrT$  und  $MIT$  endet  $v_1$  auf ein  $a \in I_1$ , d.h.  $v_1 = v'_1a$ .  $v'$  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 kein Fehler möglich ist in der Parallelkomposition und somit maximal das gesamte Wort  $v_2$  bereits in  $v'$  enthalten sein kann durch Projektion auf  $\Sigma_2$ . Falls  $v_2 \in PrT_2 \cup MIT_2$  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 sogar schließen, dass  $v'_2$  ein echtes Präfix von  $v_2$  ist, da  $v_2$  auf  $b \in I_2$  endet und somit der letzte Input  $b$  noch nicht der fehlende sein kann, da auch  $v_1$  auf einen Input endet.

In allen Fällen erhalten wir  $q_{02} \xrightarrow{v'_2} (*)$  und desweiteren ist  $v'_2 = v'|_{\Sigma_2}$  ein Präfix von  $v_2$  und  $v' \in v_1 \parallel v'_2$  ist ein Präfix von  $xy$ .

- Fall 1 ( $v_1 \in MIT_1$  und  $v_1 \neq \varepsilon$ ): Es gibt  $q_{01} \xrightarrow{v'_1} q_1 \xrightarrow{a}$  und sei  $v' = v''a$ .
  - Fall 1a) ( $a \notin Synch(S_1, S_2)$ ): Es folgt  $a \notin \Sigma_2$  und durch (\*) folgt  $q_{02} \xrightarrow{v'_2} q_2$  mit  $v'' \in v'_1 \| v'_2$ . Dadurch erhalten wir  $(q_{01}, q_{02}) \xrightarrow{v''} (q_1, q_2) \xrightarrow{a}$  mit  $a \in I_{12}$ . Somit können wir wählen  $v := v''a = v' \in MIT(S_1 \| S_2)$ .
  - Fall 1b) ( $a \in \Sigma_2$ ): Es folgt  $a \in O_2$  und  $v'_2 = v''_2 a$ . Durch (\*) erhalten wir  $q_{02} \xrightarrow{v''_2} q_2 \xrightarrow{a}$  mit  $v'' \in v'_1 \| v'_2$ . Daraus ergibt sich  $(q_{01}, q_{02}) \xrightarrow{v''} (q_1, q_2)$  mit  $q_1 \xrightarrow{a}$ ,  $a \in I_1$ ,  $q_2 \xrightarrow{a}$ ,  $a \in O_2$ , somit gilt  $(q_1, q_2) \in E_{12}$ . Wir wählen  $v := prune(v'') \in PrT(S_1 \| S_2)$ .
- Fall 2 ( $v_1 \in PrT_1$ ):  $\exists u_1 \in O_1^* : q_{01} \xrightarrow{u_1} q_1 \xrightarrow{u'_1} q'_1$  mit  $q'_1 \in E_1$ . Es gilt  $q_{02} \xrightarrow{v'_2} q_2$  mit  $(q_{01}, q_{02}) \xrightarrow{v'} (q_1, q_2)$ .
  - Fall 2a) ( $u_2 \in (O_1 \cap I_2)^*$ ,  $c \in (O_1 \cap I_2)$ , sodass  $u_2 c$  Präfix von  $u_1|_{I_2}$  mit  $q_2 \xrightarrow{u_2} q'_2 \xrightarrow{c}$ ): Für das Präfix  $u'_1 c$  von  $u_1$  mit  $u'_1 c|_{I_2} = u_2 c$  wissen wir, dass  $q_1 \xrightarrow{u'_1} q'_1 \xrightarrow{c}$ . Somit gilt  $u'_1 \in u'_1 \| u_2$  und  $(q_1, q_2) \xrightarrow{u'_1} (q'_1, q'_2) \in E_{12}$ , 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  $v := prune(v' u'_1) \in PrT(S_1 \| S_2)$ , dies ist ein Präfix von  $v'$ , da  $u_1 \in O_1^*$ .
  - Fall 2b) ( $q_{02} \xrightarrow{u_2} q'_2$  mit  $u_2 = u_1|_{I_2}$ ): Somit ist  $u_1 \in u_1 \| u_2$  und  $(q_1, q_2) \xrightarrow{u_1} (q'_1, q'_2) \in E_{12}$ , da  $q'_1 \in E_1$  und somit handelt es sich um einen geerbten Error. Wir wählen nun  $v := prune(v' u_1) \in PrT(S_1 \| S_2)$ , das wiederum ein Präfix von  $v'$  ist.

2.:

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:

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

□

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

*Beweis.* Es muss gezeigt werden, dass wenn  $S_1 \sqsubseteq_E S_2$  gilt, dass auch  $S_3 \| S_1 \sqsubseteq S_3 \| S_2$  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  $EL(S_3 \| S_1) \subseteq EL(S_3 \| S_2)$ .

$$\begin{aligned}
 & \bullet \quad ET(S_3 \| S_1) \stackrel{2.5}{=}^{1.} cont(prune((ET_3 \| EL_1) \cup (EL_3 \| ET_1))) \\
 & \quad \quad \quad \begin{array}{c} ET_1 \subseteq ET_2 \\ \text{und} \\ EL_1 \subseteq EL_2 \\ \subseteq \end{array} cont(prune((ET_3 \| EL_2) \cup (EL_3 \| ET_2))) \\
 & \quad \quad \quad \stackrel{2.5}{=}^{1.} ET(S_3 \| S_2) \\
 & \bullet \quad EL(S_3 \| S_1) \stackrel{2.5}{=}^{2.} (EL_3 \| EL_1) \cup E_{31} \\
 & \quad \quad \quad \begin{array}{c} EL_1 \subseteq EL_2 \\ \text{und} \\ ET_{31} \subseteq ET_{32} \\ \subseteq \end{array} (EL_3 \| EL_2) \cup ET_{32} \\
 & \quad \quad \quad \stackrel{2.5}{=}^{2.} EL(S_3 \| S_2)
 \end{aligned}$$

□

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.

**Lemma 2.7 (Verfeinerung mit Errors).** *Gegeben sind zwei EIOs  $S_1$  und  $S_2$  mit der gleichen Signatur. Wenn gilt, für alle EIOs  $U$  für die  $S_2$  und  $U$  kommunizieren gut, folgt  $S_1$  und  $U$  kommunizieren gut, dann verfeinert  $S_1$   $S_2$ . Diese Verfeinerung entspricht der Relation  $\sqsubseteq_E$  von oben, die hier dann wie folgt definiert ist: 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  $S_1 \sqsubseteq_E S_2$  müssen wir zeigen, dass gilt:

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

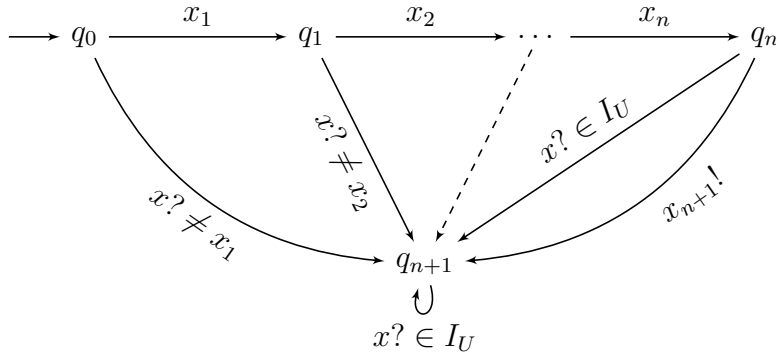
Wir wählen ein präfix-minimales Element  $w \in ET(S_1)$  und müssen dann zeigen, dass dieses  $w$  oder eines seiner Präfixe in  $ET(S_2)$  enthalten ist, um die erste Inklusion zu zeigen.

- Fall 1 ( $w = \varepsilon$ ): Es handelt sich um einen lokal erreichbaren Error in  $S_1$ . Wir nehmen für  $U$  einen Automaten, der nur aus dem Startzustand und einer Schleife für alle  $x \in I_U$  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 geerbt werden von  $S_2$ . Es muss also in  $S_2$  ein Error-Zustand durch interne Handlungen und Outputs erreichbar sein, d.h. es gilt  $\varepsilon \in PrT(S_2)$ .



- Fall 2 ( $w = x_1 \dots x_n x_{n+1} \in \Sigma^+$  mit  $n \geq 0$  und  $x_{n+1} \in I$ ): Wir betrachten die folgenden Partner  $U$ , siehe auch Abbildung 2.1:

- $Q_U = \{q_0, q_1, \dots, q_{n+1}\}$
- $q_{0U} = q_0$
- $E_U = \emptyset$
- $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i \leq n\} \cup$   
 $\{(q_i, x, q_{n+1}) \mid x \in I_U \setminus \{x_{i+1}\}, 0 \leq i \leq n\} \cup$   
 $\{(q_{n+1}, x, q_{n+1}) \mid x \in I_U\}$


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

Wir können für  $w$  zwei Fälle unterscheiden. Beide führen zu  $\varepsilon \in PrT(U \parallel S_1)$ .

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

Da wir wissen, dass  $\varepsilon \in StT(U \parallel S_1)$  gilt, können wir durch  $U \parallel S_1 \sqsubseteq_E^B U \parallel S_2$  schließen, dass auch in  $U \parallel S_2$  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  $i \leq n, q' \not\stackrel{x_{i+1}}{\rightarrow}$  und  $x_{i+1} \in O_U = I$ . Durch Projektion erhalten wir dann  $q_{02} \not\stackrel{x_1 \dots x_i}{\Rightarrow} q' \not\stackrel{x_{i+1}}{\rightarrow}$  und damit gilt  $x_1 \dots x_{i+1} \in MIT(S_2) \subseteq ET(S_2)$ . Somit ist ein Präfix von  $w$  in  $ET(S_2)$  enthalten.

- Fall 2ii) (geerbter Error):  $U$  hat  $x_1 \dots x_i u$  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  $prune(x_1 \dots x_i u) = prune(x_1 \dots x_i) \in PrT(S_2) \subseteq ET(S_2)$ . Da  $x_1 \dots x_i$  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 zu zeigen, dass  $L(S_1) \setminus ET(S_1) \subseteq EL(S_2)$  gilt, da wir die erste Inklusion bereits bewiesen haben und die Definition von  $EL$  gilt.

Wir nehmen uns dafür ein beliebiges  $w \in L(S_1) \setminus ET(S_1)$  und zeigen, dass es in  $EL(S_2)$  enthalten ist.

- Fall 1 ( $w = \varepsilon$ ): Da  $\varepsilon$  immer in  $EL(S_2)$  enthalten ist, haben wir hier nichts zu zeigen.
- Fall 2 ( $w = x_1 \dots x_n$  mit  $n \geq 1$ ): Wir konstruieren einen Partner  $U$  wie folgt, siehe dazu auch Abbildung 2.2:
  - $Q_U = \{q, q_0, q_1, \dots, q_n\}$
  - $q_{0U} = q_0$
  - $E_U = q_n$
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\} \cup \{(q_i, x, q) \mid x \in I_U \setminus \{x_{i+1}\}, 0 \leq i \leq n\} \cup \{(q, x, q) \mid x \in I_U\}$

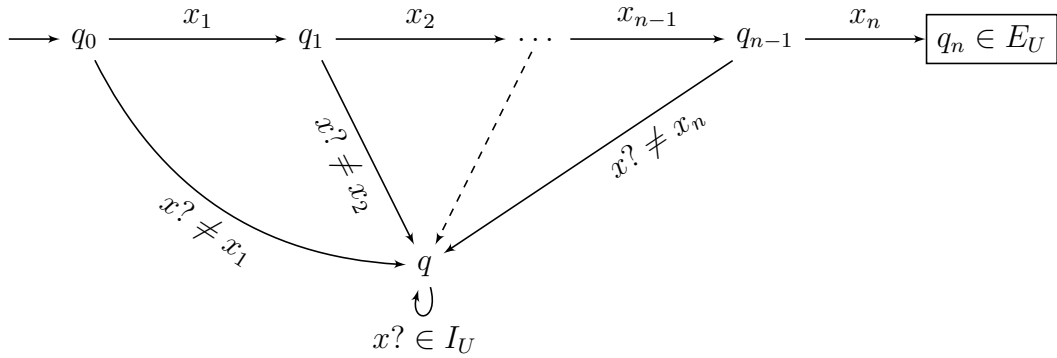


Abbildung 2.2:  $x? \neq x_i$  steht für alle  $x \in I_U \setminus \{x_i\}$ ,  $q_n$  ist der einzige Error-Zustand

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

- Fall 2a) (neuer Error aufgrund von  $x_i \in O_U$  und  $q_{02} \xrightarrow{x_1 \dots x_{i-1}} q' \not\xrightarrow{x_i}$ ): 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  $U \parallel S_2$ , 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  $q_{02} \xrightarrow{x_1 \dots x_n}$ . 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  $q_{02} \xrightarrow{x_1 \dots x_i u} q' \in E_2$  für  $i \geq 0$  und  $u \in O^*$ . Somit ist  $x_1 \dots x_i u \in StT(S_2)$  und damit  $prune(x_1 \dots x_i u) = prune(x_1 \dots x_i) \in PrT(S_2) \subseteq EL(S_2)$ . Damit gilt  $w \in EL(S_2)$ .

□

**Satz 2.8 (Full Abstractness für lokale Error Semantik).** 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 2.6 festgehalten, ist  $\sqsubseteq_E$  eine Präkongruenz.

“ $\Leftarrow$ ”: Im Beweis von 2.7 wurde bereits festgestellt wenn  $\varepsilon \in ET(S)$  gilt, dass ein Error lokal erreichbar ist 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 auch, dass ebenfalls  $S_1 \sqsubseteq_E^B S_2$  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  $U \parallel S_1 \subseteq U \parallel S_2$  gilt für alle EIOs  $U$ , die mit  $S_1$  komponierbar sind. Somit folgt auch die Gültigkeit von  $U \parallel S_1 \sqsubseteq_E^B U \parallel S_2$  für alle diese EIOs  $U$ . Mit Lemma 2.7 ergibt sich dann  $S_1 \sqsubseteq_E S_2$ . □

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

*Beweis.* “ $\Rightarrow$ ”: Diese Richtung folgt aus dem Beweis von Satz 2.8.

“ $\Leftarrow$ ”: Diese Richtung wurde in Lemma 2.7 bewiesen. □

# Literaturverzeichnis

- [BV14] Ferenc Bujtor und Walter Vogler, *Error-pruning in interface automata*, Tech. report, Universität Augsburg, 2014.