Bachelorarbeit
im Studiengang Bachelor Informatik

Kommunikationsfehler, Verklemmung und Divergenz bei Interface Automaten

Universität Augsburg
Fakultät für angewandte Informatik

Ayleen Schinko

1. Dezember 2015

Inhaltsverzeichnis

1 Einleitung

Wie schon aus dem Titel hervor geht, sollen in dieser Arbeit Interface Automaten und Fehler, die durch die Kommunikation mehrere solcher enstehen betrachtet werden. Eine entsprechende Definition dieser Interface Automaten findet sich in [AH04]. Es handelt sich also um Systeme, die via Inputs und Outputs mit anderen Systemen kommunizieren können. Jedoch wir in ausgeschlossen, dass es Fehler in einem Interface Automaten gibt. Es müsste also jedes mal, wenn ein Fehler durch eine Kommunikation, die nicht so möglich ist wie gewollt, eine Veränderung an der Komposition von zwei Systemen vorgenommen werden. Es müssten also alle Wege, die nicht beeinflusst werden können und die zu einem solchen Fehler-Zustand führen aus dem Automaten entfernt werden. Dies ist ein sehr umständliches Vorgehen, wenn man ausschließlich solche Systeme betrachten möchte. Deshalb wurden Transitionssysteme eingeführt, die Fehler-Zustände zulassen. Die sich jedoch trotzdem noch entsprechend den Interface Automaten verhalten. Es kann dann nach der Betrachtung trotzdem die Fehlerfreiheit eines Transitionssystems mit Fehler hergestellt werden, in dem die entsprechenden Wege entfernt werden.
Die Betrachtung der Transitionssysteme mit Fehler-Zuständen hat auch den Vorteil, dass die Inputs nicht als deterministisch vorausgesetzt werden müssen um sicher zu stellen, dass nach dem entfernen eines Weges zu einem Fehler, der gleiche Input nicht noch zu einem anderen Zustand führt.
Um die Begrifflichkeiten hier eindeutiger zu machen, wird im weiteren Verlauf das Wort Error für Kommunikationsfehler verwendet und für Verklemmung das Wort Ruhe. Als Fehler werden im weiteren Kommunikationsfehler, Verklemmung und Divergenz bezeichnet.
Der Anfang dieser Arbeit orientiert sich sehr stark an [BV14]. Jedoch wird hier darauf verzichtet die Input-Mengen der Error-IO-Transitionssysteme (EIOs) als disjunkt anzunehmen und alle Definitionen und Sätze werden erst einmal ohne das Verbergen der synchronisierten Aktionen betrachtet.
Dadurch dass die synchronisierten Aktionen nicht verborgen werden, wird hier ein Modell betrachtet, mit dem nicht nur zwei Systeme miteinander kommunizieren können, sondern beliebig viele. Ein Output eines Systems ist somit eine Art Multicast. Jedes System, das diesen Output als Input verarbeiten kann, empfängt ihn somit auch, da bei jeder Komposition der Output weitergeleitet wird an andere Systeme. Kann jedoch ein System den Output nicht als Input aufnehmen, wird dieses System von der Nachricht nicht beeinträchtigt.
Anschießend werden die Auswirkung von Hiding auf diese Struktur untersucht und somit das Verbergen in der Parallelkomposition nachgebildet. Durch das Hiding können Outputs durch interne Aktionen ersetzt werden.

Diese Art der Betrachtung der EIOs wurde auch bereits in [Sch12] gewählt, jedoch wurde diese Arbeit nicht als direkte Quelle genutzt, bis auf den Abschnitt des Hidings. Die Feststellungen im Definitionskapitel und dem Kapitel über Errors stimmen mit dieser Quelle überein, jedoch wurden alle Beweise davon unabhängig neu geführt.
In dieser Arbeit wird ein optimistischer Ansatz für die Erreichbarkeit der jeweils betrachteten Zuständen verwendet. Ein Zustand gilt nach der Definition in dieser Arbeit als erreichbar, wenn er lokal erreicht werden kann, d.h. durch lokale Aktionen. Die Menge bestehend aus der internen Aktion τ und den Output-Aktionen wird hier als Menge der lokale Aktionen bezeichnet. Alle Elemente aus dieser Menge können ohne weiteres Zutun von außen ausgeführt werden. Somit kann nicht beeinflusst werden, ob diese Transitionen genutzt werden oder nicht. Es besteht also die Möglichkeit, dass das EIO in einen der betrachteten Zustände übergeht, sobald dieser lokal erreichbar ist. Diese Art der Erreichbarkeit von Zuständen wird auch in Kapitel 3 von [BV14] für Error-Zustände behandelt.
Neben dem hier betrachteten optimistischen Ansatz gibt es noch zwei weitere Ansätze in [BV14] für die Erreichbarkeit von Error-Zuständen: einen hyper-optimistischen Ansatz, bei dem ein Error 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 der Error-Zustand vom Startzustand aus erreicht werden kann.
Es wird versucht bei allen betrachteten Zustandsmengen die gröbste Präkongruenz zu finden, die in der jeweiligen Basisrelation enthalten ist und die eine Präkongruenz bezüglich der Parallelkomposition ist.
Es werden im Verlauf dieser Arbeit Ruhe-Zustände betrachtet, die keine Outputs und keine τs zulassen. Somit befindet sich das betrachtete Transitionssystem in einer Art Verklemmung, wenn es in einem Ruhe-Zustand ist. Das System ist dann auf einen Input von Außen angewiesen um sich wieder aus diesem Zustand befreien zu können. Es kann ohne diesen Input keinen Fortschritt mehr geben, in Form von Outputs. Da aber auch die τ-Transitionen verboten sind, kann das System auch keine interne Aktion zu einem anderen Zustand ausführen.
Eine andere Betrachtungsweise zeigt dann die Hinzunahme von unendlichen Traces, die mit der Eigenschaft der Divergenz genauer betrachtet werden sollen. Hierbei kann ein System unendliche viele τ-Transitionen ausführen.

TODO: erweitern/umformulieren (bis jetzt nur Teile aus anderen Kapitel in Einleitung verschoben)

2 Definitionen und Notationen

Die Definitionen dieses Kapitels sind größtenteils aus [BV14] übernommen, mit den in der Einleitung erwähnten Abänderungen. In diesen Definitionen werden die Grundlagen der Transitionssysteme, mit denen hier gearbeitet werden soll behandelt.

2.1 Error-IO-Transitionssystem

Die hier betrachteten EIOs sind Systeme, deren Transitionen mit Inputs und Outputs beschriftet sind. Jede Transition ist dabei mit einem Input oder einem Output versehen. Ebenfalls zulässig ist eine Transitionsbeschriftung mit τ, einer *internen*, unbeobachtbaren *Aktion*. Diese interne Aktion lässt also keine Interaktion mit der Umwelt, d.h. mit anderen Systemen, zu. In [BV14] entsteht das τ in vielen Fällen durch das Verbergen der Inputs und Outputs, die in einer Komposition synchronisiert werden. Hier werden diese Aktionen hingegen nicht verborgen. Jedoch wird im weiteren Verlauf noch das Hiding betrachten, in dem Outputs durch interne Aktionen ersetzt werden.

Definition 2.1 (*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 Output-Aktionen,*
- $\delta \subseteq Q \times (I \cup O \cup \{\tau\}) \times Q$ *– die Transitionsrelation,*
- $q_0 \in Q$ *– 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 Transitionssystem zuzuordnen, werden für die Komponenten die gleichen Indizes wie für ihr zugehöriges System verwendet, z.B. wird I_1 für die Inputmenge des Transitionssystems S_1 geschrieben. Diese Notation wird später

analog für die Sprachen, Traces und Zustandsmengen eines Systems verwendet.
Die Elemente der Transitionsrelation δ werden wie folgt notieren:

- $p \xrightarrow{\alpha} q$ für $(p, \alpha, q) \in \delta$,
- $p \xrightarrow{\alpha}$ für $\exists q : (p, \alpha, q) \in \delta$,
- $p \xrightarrow{w} q$ für $p \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} p_2 \ldots \xrightarrow{\alpha_n} q$ mit $w \in (\Sigma \cup \{\tau\})^*, w = \alpha_1\alpha_2 \ldots \alpha_n$,
- $p \xrightarrow{w}$ für $p \xrightarrow{\alpha_1}\xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n}$ mit $w \in (\Sigma \cup \{\tau\})^*, w = \alpha_1\alpha_2 \ldots \alpha_n$,
- $w|_B$ 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,
- $p \overset{w}{\Rightarrow} q$ für $w \in \Sigma^*$ mit $\exists w' \in (\Sigma \cup \{\tau\})^* : w'|_\Sigma = w \wedge p \xrightarrow{w'} q$,
- $p \overset{w}{\Rightarrow}$ für $\exists q : p \overset{w}{\Rightarrow} q$.

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

2.2 Parallelkomposition

Zwei EIOs sind komponierbar, wenn ihre Output-Mengen disjunkt sind. Die Error-Zustände der Parallelkomposition setzen sich aus den Errors der beiden zusammengesetzten Komponenten (geerbte Errors) und den Zuständen, die Outputs aus der Menge der synchronisierten Aktionen besitzen, für die im zu komponierenden System jedoch kein passender Input vorhanden ist, (neue Errors) zusammen.
In der folgenden Definition muss eine Veränderung gegenüber [BV14] an der Menge der synchronisierten Aktionen vorgenommen werden. Da nicht mehr $I_1 \cap I_2 = \emptyset$ gelten muss, werden die gemeinsamen Inputs synchronisiert. Somit handelt es sich in der Parallelkomposition bei synchronisierten Aktionen nicht mehr nur um Outputs, wie in [BV14], sondern im Fall von $I_1 \cap I_2$ auch um Inputs. Falls es bei Inputs aus $I_1 \cap I_2$ zu einem fehlenden Input für die Synchronisation kommt, ist die Transition für die Parallelkomposition nicht ausführbar, jedoch handelt es sich auch nicht um einen neuen Error, da es zwischen den beiden Systemen dadurch nicht zu einem Kommunikations-Fehler kommt. Die beiden Transitionssysteme können über die beiden Inputs nicht miteinander kommunizieren, sondern nur mit anderen Systemen.

Definition 2.2 (*Parallelkomposition*). *Zwei EIOs S_1, S_2 sind* komponierbar*, falls $O_1 \cap O_2 = \emptyset$ gilt. Die* Parallelkomposition *ist $S_{12} := S_1 \| S_2 = (Q, I, O, \delta, q_0, E)$ mit den Komponenten:*

- $Q = Q_1 \times Q_2$,
- $I = (I_1 \backslash O_2) \cup (I_2 \backslash O_1)$,
- $O = O_1 \cup O_2$,

- $q_0 = (q_{01}, q_{02})$,
- $\delta = \{((q_1, q_2), \alpha, (p_1, q_2)) \mid (q_1, \alpha, p_1) \in \delta_1, \alpha \in (\Sigma_1 \cup \{\tau\}) \backslash Synch(S_1, S_2)\}$
  $\cup \{((q_1, q_2), \alpha, (q_1, p_2)) \mid (q_2, \alpha, p_2) \in \delta_2, \alpha \in (\Sigma_2 \cup \{\tau\}) \backslash Synch(S_1, S_2)\}$
  $\cup \{((q_1, q_2), \alpha, (p_1, p_2)) \mid (q_1, \alpha, p_1) \in \delta_1, (q_2, \alpha, p_2) \in \delta_2, \alpha \in Synch(S_1, S_2)\}$,
- $E = (Q_1 \times E_2) \cup (E_1 \times Q_2)$ geerbte Errors
  $\cup \{(q_1, q_2) \mid \exists a \in O_1 \cap I_2 : q_1 \xrightarrow{a} \wedge q_2 \not\xrightarrow{a}\}$
  $\cup \{(q_1, q_2) \mid \exists a \in I_1 \cap O_2 : q_1 \not\xrightarrow{a} \wedge q_2 \xrightarrow{a}\}$
  } neue Errors.

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 bzw. im Fall von $I_1 \cap I_2$ als Inputs der Komposition beibehalten.*
S_1 *wird ein* Partner *von S_2 genannt, wenn ihre Parallelkomposition geschlossen ist, d.h. wenn sie duale Signaturen $Sig(S_1) = (I, O)$ und $Sig(S_2) = (O, I)$ haben.*
Die oben definierte Notation $S_{12} = S_1 \| S_2$ wird auf für andere Indizierungen der Systeme analog angewendet, so gilt also allgemein $S_{ij} := S_i \| S_j$ für $i, j \in \mathbb{N}$.

Die Parallelkomposition kann nicht nur für Transitionssysteme betrachtet werden, wie bisher in dieser Arbeit, sondern auch über Aktionsfolgen. *Traces* sind die möglichen Wege des Systems, mit ihrer Transitionsbeschriftung. Diese Transitionsbeschriftung besteht aus Inputs und Outputs, mit denen die Folge ab dem Startzustand q_0 beschriftet ist. Somit kann ein Trace auch als das Wort aufgefasst werden, dass verarbeitet wird während des Ablaufs des Systems.

Definition 2.3 (*Parallelkomposition auf Traces*). *Gegeben zwei EIOs S_1 und S_2, sowie $w_1 \in \Sigma_1, w_2 \in \Sigma_2, W_1 \subseteq \Sigma_1^*, W_2 \subseteq \Sigma_2^*$:*

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

Die Semantik der späteren Kapitel basiert darauf die jeweiligen Zustände, die zu Problemen führen, mit den Traces zu betrachten, mit denen man diese Zustände erreicht. Um dies besser umsetzen zu können, wird eine *prune*-Funktion definiert, die alle Outputs am Ende eines Traces entfernt. Zusätzlich werden Funktionen definiert, die die Traces beliebig fortsetzen. Da nicht nur beliebige Fortsetzungen von gekürzten Traces betrachtet werden sollen, wir ebenfalls noch eine andere Funktion zum kürzen und verlängern eingeführt, die nur Trace-Verlängerungen zulässt, die bereits im Eingrabe-Trace möglich waren.

Definition 2.4 (*Pruning- und Fortsetzungs-Funktion*). *Für ein EIO S wird definiert:*

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

- $prune' : \mathfrak{P}(\Sigma^*) \to \mathfrak{P}(\Sigma^*), L \mapsto \bigcup\{prune'(w) \mid w \in L\}$,
- $cont : \Sigma^* \to \mathfrak{P}(\Sigma^*), w \mapsto \{wu \mid u \in \Sigma^*\}$,
- $cont : \mathfrak{P}(\Sigma^*) \to \mathfrak{P}(\Sigma^*), L \mapsto \bigcup\{cont(w) \mid w \in L\}$.

Für zwei komponierbare EIOs S_1 und S_2 ist ein Ablauf ihrer Parallelkomposition S_{12} 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 Projektionen 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 der Form $p_i \stackrel{w_i}{\Rightarrow} q_i$ mit $w|_{\Sigma_i} = w_i$ für $i = 1, 2$, Projektionen von einem Ablauf in S_{12} der Form $(p_1, p_2) \stackrel{w}{\Rightarrow} (q_1.q_2)$. Es ist dafür nötig, dass die Abläufe der beiden Systeme und die Systeme selbst komponierbar sind. Das w wurde so gewählt, dass die Projektion auf die einzelnen Alphabete die jeweiligen Wörter ergibt. Falls keine interne Aktionen zugelassen wären, würde sogar nur genau ein Ablauf möglich sein in S_{12}. Da jedoch auch interne Aktionen zulässig sind, sind mehrere Abläufe möglich, da nicht klar ist, wann ein τ in einem Trace ausgeführt wird. Daraus ergibt sich das folgende Lemma.

Lemma 2.5 (*Sprache der Parallelkomposition*). *Für zwei komponierbare EIOs S_1 und S_2 gilt:*

$$L_{12} := L(S_{12}) = L_1 \| L_2.$$

2.3 Hiding

Hiding wurde in dem hier verwendeten Kontext bereits in [CJK13] auf Traces betrachtet. Da hier die Betrachtungsweise von Transitionssystemen aus startet, wird auch Hiding aus der Sicht dieser Systeme definieren, wie in [Sch12]. Eine ähnliche Betrachtung für Hiding bei LTS mit Inputs und Outputs wurde auch bereits in [Lyn96] umgesetzt. Dort werden nur Output-Aktionen internalisiert, jedoch gibt es eine Menge an internen Aktionen und nicht nur eine. Das Hiding wird durch einen Internalisierungsoperator umgesetzt. Es sollen dadurch Aktionen versteckt werden können, d.h. durch τs ersetzt werden. In [CJK13] ist es in der Definition des Hidings möglich Outputs und Inputs zu verstecken. Durch das Verstecken von Outputs sind diese nach außen nicht mehr sichtbar. Werden jedoch Inputs versteckt sind alle Traces, die diesen Input benötigen, nicht mehr ausführbar. Sie sind dann ab dem versteckten Input nicht mehr weiterführbar. Es handelt sich also um echte Einschränkungen des Systems. Die Transitionen werden durch das Hiding von Inputs ähnlich wie bei der Anwendung von Restriktionen in CSS, siehe dazu [Mil89] ,verboten. Diese Art der Einschränkung der Transitionssysteme sollen hier jedoch nicht behandelt werden. Somit wird in der folgenden Definition nur die Internalisierung von Outputs erlaubt, entsprechend Quelle [Sch12].

Definition 2.6 (*Internalisierungsoperator*). *Für ein EIO $S = (Q, I, O, \delta, q_0.E)$ ist S/X, mit dem* Internalisierungsoperator $\cdot/\cdot$, *definiert als $S' = (Q, I, O', \delta', q_0, E)$ mit:*

- $\tau \notin X$,

- $X \subseteq O$,
- $O' = O \backslash X$,
- $\delta' = (\delta \cup \{(q, \tau, q') \mid (q, x, q') \in \delta, x \in X\}) \backslash \{(q, x, q') \mid x \in X\}$.

Die Menge hinter dem Internalisierungsoperator ist in dieser Definition auf Outputs beschränkt. Diese Einschränkung wurde vorgenommen um die weitere Betrachtung zu erleichtern. Jedoch kann es sinnvoll sein die Möglichkeit zu haben dort weitere Aktionen aufnehmen zu können. Dies wird jedoch nicht mehr Teil dieser Arbeit sein.
In [BV14] wird die Parallelkomposition nur mit Verbergen der synchronisierten Aktionen betrachtet, die durch die Synchronisation von einem Input mit einem Output entstehen. Diese Parallelkomposition kann nun mit dem Internalisierungsoperator durch Hiding der synchronisierten Aktionen, die in der Parallelkomposition zu Outputs werden, nachbildet werden. Da in dieser Arbeit die Inputmengen der Systeme, die komponiert werden, nicht disjunkt sein müssen, ergeben sich auch Inputs aus der Synchronisation von Aktionen. Diese können jedoch mit der hier verwendeten Definition des Internalisierungsoperators nicht verborgen werden. Dies wäre auch nicht sinnvoll, da diese Synchronisation von Inputs keine Kommunikation zwischen den Systemen ist, sondern nur eine Zusammenfügung, damit die Parallelkomposition über diesen Input mit weiteren Systemen kommunizieren kann. Somit ergibt sich die folgende Definition, mit der die Parallelkomposition aus [BV14] nachgebildet werden kann.

Definition 2.7 (*Parallelkomposition mit Internalisierung*). *Seinen S_1 und S_2 komponierbare EIOs, dann ist $S_1|S_2 = S_{12}/(Synch(S_1, S_2) \cap O_{12})$.*

3 Verfeinerung für Error-Freiheit

3.1 Präkongruenz für Error

Da es in dieser Arbeit vor allem um die Erreichbarkeit und die Kommunikation zwischen EIOs geht, wurden die nächsten beiden Definitionen explizit getrennt und erweitert im Vergleich zu denen in [BV14]. Ebenfalls wurde die Parallelkomposition geändert, wie in [Sch12].

Definition 3.1 (*error-freie Kommunikation*). *Ein Error ist* lokal erreichbar *in einem EIO S, wenn $\exists w \in O^* : q_0 \overset{w}{\Rightarrow} q \in E$.*
Zwei EIOs S_1 und S_2 kommunizieren error-frei*, wenn in ihrer Parallelkomposition S_{12} keine Errors lokal erreicht werden können.*

Mittels der lokalen Erreichbarkeit von Errors kann ein Verfeinerungsrelation definiert werden. Zusätzlich wird bereits die gröbste Präkongruenz definiert, die charakterisiert werden soll.

Definition 3.2 (*Error-Verfeinerungs-Basisrelation*). *Für EIOs S_1 und S_2 mit der gleichen Signatur wird $S_1 \sqsubseteq_E^B S_2$ geschrieben, wenn ein Error in S_1 nur dann lokal erreichbar ist, wenn er auch in S_2 lokal erreichbar ist. Es handelt sich dabei um eine* Basisrelation *für die* Verfeinerung *im Bezug auf* Errors*.*
$\sqsubseteq_E^C$ bezeichnet die vollständig abstrakte Präkongruenz *von $\sqsubseteq_E^B$ bezüglich $\cdot \| \cdot$, d.h. die gröbste Präkongruenz bezüglich $\cdot \| \cdot$, die in $\sqsubseteq_E^B$ enthalten ist.*

Um sich näher mit den Präkongruenzen auseinandersetzen zu können, müssen bestimmte Traces aus der Struktur hervor gehoben werden. Die strikten Errortraces entsprechen Wegen, die direkt vom Startzustand zu einem Zustand in der Menge E führen. Da Outputs Aktionen sind, die von außen nicht verhindert werden können, wird auch noch die Menge der Traces benötigt, die zu einem Zustand führen können, 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 möglicherweise nicht fortgesetzt werden können. Diese führen zwar nicht direkt zu einem Error, jedoch in Komposition mit einem anderen Transitionssystem sind dies gefährdete Stellen. Sie führen zu einem neuen Error, sobald dieser Input für die Synchronisation fehlt.

Definition 3.3 (*Errortraces*). *Für ein EIO S wird definiert:*

- strikte Errortraces*: $StET(S) = \{w \in \Sigma^* \mid q_0 \overset{w}{\Rightarrow} q \in E\}$,*

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

In der folgenden Definition wird festgehalten, was als Errortrace auffasst wird. Diese Menge ist dadurch, dass sie die fortgesetzten Traces aus $PrET$ enthält, deutlich allgemeiner als die Menge $StET$. Zusätzlich wird auch noch die geflutete Sprache definiert, in der die Informationen aus der Sprache und den Errortraces vereint werden und somit bei der Inklusion nicht mehr explizit unterscheiden werden.

Definition 3.4 (*Error-Semantik*). *Sei S ein EIO.*

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

Für zwei EIOs S_1, S_2 mit der gleichen Signatur wird $S_1 \sqsubseteq_E S_2$ geschrieben, wenn $ET_1 \subseteq ET_2$ und $EL_1 \subseteq EL_2$ 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, wird hier eher auf die Erwähnung der Unterschiede zu [BV14] Wert gelegt.

Satz 3.5 (*Error-Semanik für Parallelkompositionen*). *Für zwei komponierbare EIOs S_1, S_2 und ihre Komposition S_{12}, 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 $I_{12} \stackrel{2.2}{=} (I_1 \backslash O_2) \cup (I_2 \backslash O_1) = (I_1 \cup I_2) \backslash (O_1 \cup O_2)$ ist, folgt $a \in (I_1 \cup I_2)$ und $a \notin (O_1 \cup O_2)$. Es wird 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).
  - Fall 1a) ($a \in (I_1 \cap I_2)$): Nun kann den Ablauf der Komposition auf die Transitionssysteme projiziert werden und man erhält 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 kann $x_1 a \in cont(MIT_1) \subseteq ET_1 \subseteq EL_1$ und $x_2 a \in EL_2$ ($x_2 a \in MIT_2$ oder $x_2 a \in L_2$)

gefolgert werden. 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 erhält man: $q_{01} \overset{x_1}{\Rightarrow} q_1 \overset{a}{\not\rightarrow}$ und $q_{02} \overset{x_2}{\Rightarrow} q_2$ 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 Definitionen von $PrET$ und *prune* weiß man, dass es ein $v \in O_{12}^*$ gibt, so dass $(q_{01}, q_{02}) \overset{w}{\Rightarrow} (q_1, q_2) \overset{v}{\Rightarrow} (q_1', q_2')$ gilt mit $(q_1', q_2') \in E_{12}$ und $w = prune(wv)$. Durch Projektion erhält man $q_{01} \overset{w_1}{\Rightarrow} q_1 \overset{v_1}{\Rightarrow} q_1'$ und $q_{02} \overset{w_2}{\Rightarrow} q_2 \overset{v_2}{\Rightarrow} q_2'$ 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 q_1 bzw. q_2 ein Error-Zustand gewesen. Der neue Error hingegen wäre durch die fehlende Möglichkeit entstanden, eine synchronisierte Aktion auszuführen.

  - Fall 2a) (geerbter Error): OBdA gilt $q_1' \in E_1$. Daraus folgt $w_1 v_1 \in StET_1 \subseteq cont(PrET_1) \subseteq ET_1$. Da gilt $q_{02} \overset{w_2 v_2}{\Rightarrow}$, erhält man $w_2 v_2 \in L_2 \subseteq EL_2$. Dadurch ergibt sich $wv \in ET_1 \| EL_2$ mit $w = prune(wv)$ und somit ist w in der rechten Seite der Gleichung enthalten.

  - Fall 2b) (neuer Error): OBdA gilt $a \in I_1 \cap O_2$ mit $q_1' \overset{a}{\not\rightarrow} \wedge\ q_2' \overset{a}{\rightarrow}$. 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* wird auch in diesem Fall nur ein präfix-minimales Element $x \in prune((ET_1 \| EL_2) \cup (EL_1 \| ET_2))$ betrachtet. 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 wird davon ausgegangen, 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 das präfix-minimale Element noch mit Outputs fortsetzt werden kann, unterscheidet sich dieser Beweis von dem in [Sch12]. In dieser Quelle wird nicht weiter darauf eingegangen, dass die *prune*-Funktion an dieser Stelle noch zur Anwendung kommt. Da jedoch später nur Präfixe von x betrachtet werden, ist dieser Unterschied irrelevant.
Im Folgenden wird für alle Fälle von xy gezeigt, dass es ein $v \in PrET_{12} \cup MIT_{12}$ gibt, das ein Präfix von xy ist und v entweder auf einen 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 v mindestens einen Buchstaben enthält, muss das Ende von v vor dem Anfang von $y \in O_{12}^*$ liegen. Dadurch hat x ein Präfix in $PrET_{12} \cup MIT_{12}$, damit 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. Es wird nun davon ausgegangen, dass entweder $v_2 = w_2 \in L_2$ gilt oder die letzte Aktion von v_1 vor oder gleichzeitig mit der letzten Aktion von v_2 statt findet. 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 ($v_1 = \varepsilon$): Da $\varepsilon \in PrET_1 \cup MIT_1$, 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. Mit der Wahl $v_2' = v' = \varepsilon$ 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 $v_2 = w_2 \in L_2$, dann ist v_2' ein Präfix von v_2. 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 $v_2 \in MIT_2$ kann durch die gleiche Argumentation ebenfalls geschlossen, dass v_2' ein Präfix von v_2 ist. Zusätzlich weiß man, dass v_2 auf $b \in I_2$ endet, jedoch muss nicht mehr wie in [BV14] $b \neq a$ gelten. Es kann also keine Aussage mehr darüber getroffen, ob es sich um ein echtes Präfix handelt.

In allen Fällen erhält man: $v_2' = v'|_{\Sigma_2}$ 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, kann nicht mehr für alle Fälle $q_{02} \overset{v_2'}{\Rightarrow}$ gefolgert werden, wie das in [BV14] möglich war, sondern nur wenn $a \notin I_2$ gilt.

- 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 bezüglich [BV14] zwei weitere Fälle (Ib) und Ic)) einfügt werden, da es zulässig ist, dass 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 $v'' \in v_1' \| v_2'$. Dadurch erhält man $(q_{01}, q_{02}) \overset{v''}{\Rightarrow} (q_1, q_2) \overset{a}{\not\rightarrow}$ mit $a \in I_{12}$. Somit wird $v := v''a = v' \in MIT_{12}$ gewählt.
  - 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. Daraus folgt, dass $(q_1, q_2) \overset{a}{\not\rightarrow}$ gilt. Es wird $v := v''a = v' \in MIT_{12}$ gewählt.
  - 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 die Menge der synchronisierten Aktionen bezüglich [BV14] erweitert wurde liegt a in $Synch(S_1, S_2)$, also folgt $(q_1, q_2) \overset{a}{\not\rightarrow}$ schon aus $q_1 \overset{a}{\not\rightarrow}$. Somit kann hier $v := v''a = v' \in MIT_{12}$ gewählt werden.
  - Fall Id) ($a \in O_2$): Es gilt $v_2' = v_2''a$ und $q_{02} \overset{v_2'}{\Rightarrow}$. Man erhält 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 $(q_1, q_2) \in E_{12}$. Es wird $v := prune(v'') \in PrET_{12}$ gewählt.

- 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 es hier keine disjunkten Inputmengen wie in [BV14] gibt kann das a, auf das v_1 im Fall $v_1 \neq \varepsilon$ 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. 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 $u_1'c|_{I_2} = u_2c$ weiß man, 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. Es wird $v := prune(v'u_1') \in PrET_{12}$ gewählt, 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. Es wird nun $v := prune(v'u_1) \in PrET_{12}$ gewählt, das wiederum ein Präfix von v' ist.

2.:
Der Beweis für diesen Punkt konnte bezüglich [BV14] fast unverändert übernommen werden, bis auf die Ersetzung der Zeichen der Parallelkomposition.
Es ist durch die Definition klar, dass $L_i \subseteq EL_i$ und $ET_i \subseteq EL_i$ gilt. Die Argumentation wird von der rechten Seite der Gleichung aus begonnen:

$$
\begin{aligned}
&(EL_1\|EL_2) \cup ET_{12} \\
&\overset{3.4}{=} (L_1 \cup ET_1)\|(L_2 \cup ET_2) \cup ET_{12} \\
&= \underbrace{(L_1\|ET_2)}_{\substack{\subseteq(EL_1\|ET_2) \\ \overset{1.}{\subseteq} ET_{12}}} \cup \underbrace{(ET_1\|L_2)}_{\substack{\subseteq(ET_1\|EL_2) \\ \overset{1.}{\subseteq} ET_{12}}} \cup (L_1\|L_2) \cup \underbrace{(ET_1\|ET_2)}_{\substack{\subseteq(EL_1\|ET_2) \\ \overset{1.}{\subseteq} ET_{12}}} \cup ET_{12} \\
&= (L_1\|L_2) \cup ET_{12} \\
&\overset{2.5}{=} L_{12} \cup ET_{12} \\
&\overset{3.4}{=} EL_{12}.
\end{aligned}
$$

□

Die folgende Proposition wurde hier noch explizit mit Beweis eingefügt im Gegensatz zu den Ausführungen in [BV14], in denen diese Präkongruenz nur als Folgerung aus dem letzten Satz erwähnt wird. Die Feststellung, dass es sich um eine Präkongruenz handelt, ist wichtig, da dann die erste Eigenschaft erfüllt ist, um eine operationale Beschreibung der vollständig abstrakten Präkongruenz $\sqsubseteq_E^C$ zu erhalten.

Proposition 3.6 (*Error-Präkongruenz*). *$\sqsubseteq_E$ ist eine Präkongruenz bezüglich $\cdot\|\cdot$.*

Beweis. Es muss gezeigt werden: Wenn $S_1 \sqsubseteq_E S_2$ gilt, dann für jedes komponierbare S_3 auch $S_{31} \sqsubseteq_E S_{32}$. D.h. es ist zu zeigen, dass aus $ET_1 \subseteq ET_2$ und $EL_1 \subseteq EL_2$, $ET_{31} \subseteq ET_{32}$ und $EL_{31} \subseteq EL_{32}$ folgt. Dies ergibt sich aus der Monotonie von *cont*, *prune* und $\cdot\|\cdot$ auf Sprachen wie folgt:

- $ET_{31} \overset{3.5\ 1.}{=} cont(prune((ET_3\|EL_1) \cup (EL_3\|ET_1)))$
  $\overset{\substack{ET_1\subseteq ET_2 \\ \text{und} \\ EL_1\subseteq EL_2}}{\subseteq} cont(prune((ET_3\|EL_2) \cup (EL_3\|ET_2)))$
  $\overset{3.5\ 1.}{=} ET_{32},$
- $EL_{31} \overset{3.5\ 2.}{=} (EL_3\|EL_1) \cup E_{31}$
  $\overset{\substack{EL_1\subseteq EL_2 \\ \text{und} \\ ET_{31}\subseteq ET_{32}}}{\subseteq} (EL_3\|EL_2) \cup ET_{32}$
  $\overset{3.5\ 2.}{=} EL_{32}.$

□

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 sollen die synchronisierten Aktionen nicht verborgen werden. 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 3.8 kombiniert wurde. Jedoch ist die Aussage dieses Lemmas trotzdem Teil dessen, was gezeigt wird und somit finden sich die Teile dieses Beweises auch dort wieder.
Die Verfeinerungsrelation, die in dem nächsten Lemma betrachtet werden soll, verfeinert über guter Kommunikation im Sinne der error-freien Kommunikation.

Lemma 3.7 (*Verfeinerung mit Errors*). *Gegeben sind zwei EIOs S_1 und S_2 mit der gleichen Signatur. Wenn $U\|S_1 \sqsubseteq_E^B U\|S_2$ für alle Partner U gilt, dann folgt daraus die Gültigkeit von $S_1 \sqsubseteq_E S_2$.*

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

- $ET_1 \subseteq ET_2$,
- $EL_1 \subseteq EL_2$.

Für ein gewähltes präfix-minimales Element $w \in ET_1$ wird gezeigt, dass dieses w oder eines seiner Präfixe in ET_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. Für U wird ein Transitionssystem verwendet, das 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 die Definition von U kann dieser Error nur von S_2 geerbt sein. Es muss also in S_2 ein Error-Zustand durch interne Aktionen und Outputs erreichbar sein, d.h. es gilt $\varepsilon \in PrET_2$.
- Fall 2 ($w = x_1 \dots x_n x_{n+1} \in \Sigma^+$ mit $n \geq 0$ und $x_{n+1} \in I$): Es wird der folgenden Partner U bedachtet (siehe auch Abbildung 3.1):
  - $Q_U = \{q_0, q_1, \dots, q_{n+1}\}$,
  - $q_{0U} = q_0$,
  - $E_U = \emptyset$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i \leq n\}$
    $\cup \{(q_i, x, q_{n+1}) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i \leq n\}$
    $\cup \{(q_{n+1}, x, q_{n+1}) \mid x \in I_U\}$.

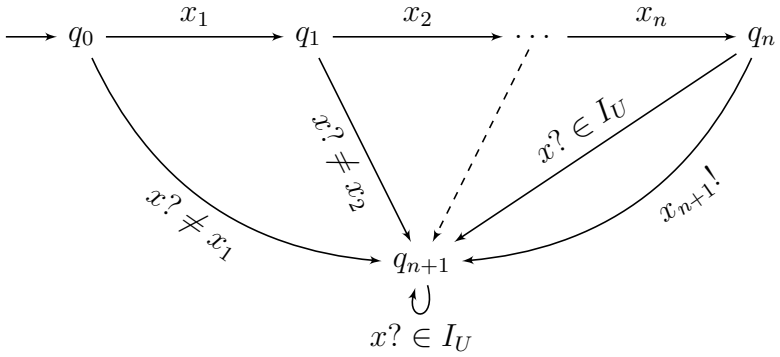

Abbildung 3.1: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$

Für w können zwei Fälle unterscheiden werden. 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) ($w \in MIT_1$): In $U \| S_1$ erhält man $(q_0, q_{01}) \overset{x_1 \dots x_n}{\Rightarrow} (q_n, q')$ mit $q' \overset{x_{n+1}}{\not\rightarrow}$ und $q_n \overset{x_{n+1}}{\rightarrow}$. 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) ($w \in PrET_1$): In $U \| S_1$ erhält man $(q_0, q_{01}) \overset{w}{\Rightarrow} (q_{n+1}, q'') \overset{u}{\Rightarrow} (q_{n+1}, q')$ 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 $x_1, \ldots, x_n, x_{n+1} \in O_{U\|S_1}$ und, da $u \in O^*$, folgt $u \in O^*_{U\|S_1}$. Somit ergibt sich $\varepsilon \in PrET(U\|S_1)$.

Da $\varepsilon \in PrET(U\|S_1)$ gilt, kann durch $U\|S_1 \sqsubseteq^B_E U\|S_2$ geschlossen werden, 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 der Form sein, dass 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 $i \leq n, q' \overset{x_{i+1}}{\not\to}$ und $x_{i+1} \in O_U = I$. Durch Projektion erhält man dann $q_{02} \overset{x_1 \ldots x_i}{\Rightarrow} q' \overset{x_{i+1}}{\not\to}$ und damit gilt $x_1 \ldots x_{i+1} \in MIT_2 \subseteq ET_2$. Somit ist ein Präfix von w in ET_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 Errors geerbt werden können. Es gilt dann $prune(x_1 \ldots x_i u) = prune(x_1 \ldots x_i) \in PrET_2 \subseteq ET_2$. Da $x_1 \ldots x_i$ ein Präfix von w ist, führt auch in diesem Fall ein Präfix von w zu einem Error.

Um die zweiten Inklusion zu beweisen, reicht es aufgrund der ersten Inklusion und der Definition von EL aus zu zeigen, dass $L_1 \backslash ET_1 \subseteq EL_2$ gilt.
Es wird dafür ein beliebiges $w \in L_1 \backslash ET_1$ gewählt und gezeigt, dass es in EL_2 enthalten ist.

- Fall 1 ($w = \varepsilon$): Da ε immer in EL_2 enthalten ist, ist hier nichts zu zeigen.
- Fall 2 ($w = x_1 \ldots x_n$ mit $n \geq 1$): Es wird einen Partner U wie folgt konstruiert (siehe dazu auch Abbildung 3.2):
  - $Q_U = \{q, q_0, q_1, \ldots, q_n\}$,
  - $q_{0U} = q_0$,
  - $E_U = \{q_n\}$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
    $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i < n\}$
    $\cup \{(q, x, q) \mid x \in I_U\}$.

  Da $q_{01} \overset{w}{\Rightarrow} q'$ gilt, kann man schließen, dass $U\|S_1$ einen lokal erreichbaren geerbten 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 $q_{02} \overset{x_1 \ldots x_{i-1}}{\Rightarrow} q'' \overset{x_i}{\not\to}$): Es gilt $x_1 \ldots x_i \in MIT_2$ und somit $w \in EL_2$. Anzumerken ist, dass nur auf diesem Weg Outputs von U möglich sind, deshalb gibt es keine anderen Outputs von U, die zu einem neuen Error führen können.

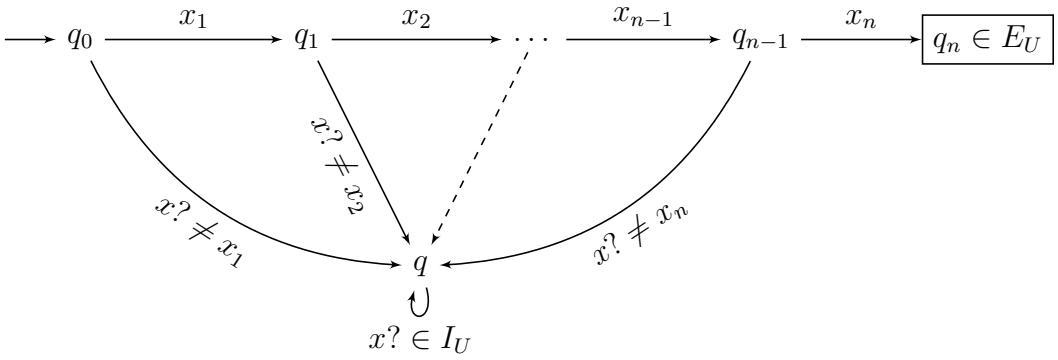

Abbildung 3.2: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$, q_n ist der einzige Error-Zustand

- 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 das komponierte EIO einen geerbten Error und es gilt $w \in L_2 \subseteq EL_2$, wegen Fall 2c).
- 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 $q_{02} \overset{x_1 \ldots x_n}{\Rightarrow}$. In diesem Fall gilt, $w \in L_2 \subseteq EL_2$.
- Fall 2d) (geerbter Error von S_2): Es gilt dann $q_{02} \overset{x_1 \ldots x_i u}{\Rightarrow} q' \in E_2$ für $i \geq 0$ und $u \in O^*$. Somit ist $x_1 \ldots x_i u \in StET_2$ und damit $prune(x_1 \ldots x_i u) = prune(x_1 \ldots x_i) \in PrET_2 \subseteq EL_2$. Somit gilt $w \in EL_2$.

□

Der folgende Satz sagt aus, dass $\sqsubseteq_E$ die gröbste Präkongruenz ist, die charakterisiert werden soll, also gleich der vollständig abstrakten Präkongruenz $\sqsubseteq_E^C$.

Satz 3.8 (*Full Abstractness für 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. „⇐“: Nach Definition gilt, genau dann wenn $\varepsilon \in ET(S)$, ist ein Error lokal erreichbar in S. $S_1 \sqsubseteq_E S_2$ impliziert, dass $\varepsilon \in ET_2$ gilt, wenn $\varepsilon \in ET_1$. Somit ist ein Error in S_1 nur dann lokal erreichbar, wenn dieser auch in S_2 lokal erreichbar ist. Dadurch folgt, dass $S_1 \sqsubseteq_E^B S_2$ gilt, da $\sqsubseteq_E^B$ in Definition 3.2 über die lokale Erreichbarkeit der Error-Zustände definiert wurde. Somit ist $\sqsubseteq_E$ in $\sqsubseteq_E^B$ enthalten. Wie in Proposition 3.6 gezeigt, ist $\sqsubseteq_E$ eine Präkongruenz. Da $\sqsubseteq_E^C$ die gröbste Präkongruenz bezüglich $\cdot \| \cdot$ ist, die in $\sqsubseteq_E^B$ enthalten ist, muss $\sqsubseteq_E$ in $\sqsubseteq_E^C$ enthalten sein. Es folgt also aus $S_1 \sqsubseteq_E S_2$, dass auch $S_1 \sqsubseteq_E^C S_2$ gilt.

„⇒“: Durch die Definition von $\sqsubseteq_E^C$ als Präkongruenz in 3.2 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. Da $\sqsubseteq_E^C$ nach Definition

auch in $\sqsubseteq_E^B$ enthalten sein soll folgt aus $U\|S_1 \sqsubseteq_E^C U\|S_2$ auch die Gültigkeit von $U\|S_1 \sqsubseteq_E^B U\|S_2$ für alle diese EIOs U. Mit Lemma 3.7 folgt dann $S_1 \sqsubseteq_E S_2$. □

Es wurde somit jetzt eine Kette an Folgerungen gezeigt, die sich zu einem Ring schließt. Dies ist in Abbildung 3.3 dargestellt.

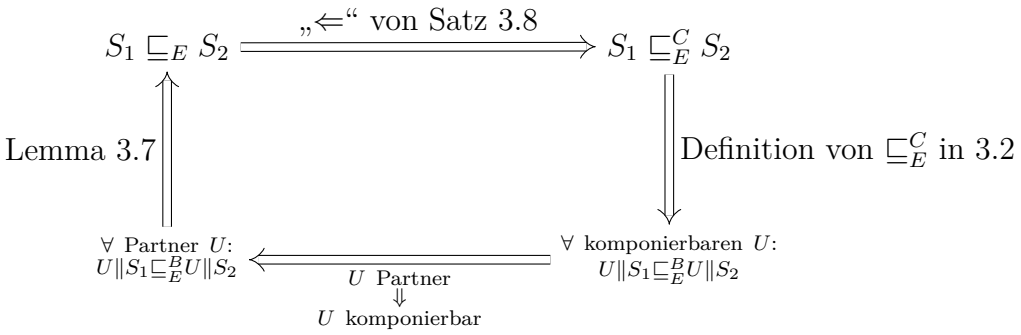

Abbildung 3.3: Folgerungskette

Aus Satz 3.8 und Lemma 3.7 ergibt sich das folgende Korollar. Angenommen man definiert, dass S_1 S_2 verfeinern soll, genau dann wenn für alle Partner EIOs U, für die S_2 error-frei mit U kommuniziert, folgt, dass S_1 ebenfalls error-frei mit U kommuniziert. Dann wird auch diese Verfeinerung durch $\sqsubseteq_E$ charakterisiert.

Korollar 3.9. *Es gilt:* $S_1 \sqsubseteq_E S_2 \Leftrightarrow U\|S_1 \sqsubseteq_E^B U\|S_2$ *für alle Partner* U.

3.2 Hiding und Error-Freiheit

Es soll nun untersucht werden, was für Auswirkungen Hiding auf die Verfeinerungsrelationen hat. Es werden also Outputs der Systeme internalisiert.

Proposition 3.10 (*Error-Basisrelation bzgl. Internalisierung*). *Wenn* $S_1 \sqsubseteq_E^B S_2$ *gilt, dann folgt daraus, dass auch* $S_1/X \sqsubseteq_E^B S_2/X$ *gilt.*

Beweis. Da die Definition der lokalen Erreichbarkeit auf lokalen Aktionen beruht, die aus den Outputs und der internen Aktion besteht, ändert sich durch das Verbergen von Outputs nichts an der Error-Erreichbarkeit. Somit ist jeder Error, der in S_i lokal erreichbar ist über ein Trace, das Outputs aus X enthält, auch in S_i/X erreichbar, jedoch enthält das Trace nicht mehr diesen Output. Alle Traces, die keine Outputs aus der Menge hinter dem Internalisierungsoperator enthalten, bleiben unverändert erhalten. Es ist auch nicht möglich, dass durch das Verbergen von Outputs neue Errors entstehen. Auch in die umgekehrte Richtung kann durch das Ersetzten von τs durch Outputs nicht an der Erreichbarkeit oder der Menge der Errors geändert werden. Es ist also jeder Error, der in S_i/X lokal erreichbar ist auch in S_i erreichbar. Somit folgt die Behauptung. □

Satz 3.11 (*Error-Präkongurenz bzgl. Interalisierung*). *Seinen S_1 und S_2 zwei EIOs für die $S_1 \sqsubseteq_E S_2$ gilt, somit gilt auch $S_1/X \sqsubseteq_E S_2/X$. Daraus folgt insbesondere, dass $\sqsubseteq_E$ eine Präkongruenz bezüglich $\cdot/\cdot$ ist. Es gilt für die Sprachen und Traces:*

(i) $L(S/X) = \{w \in (\Sigma\backslash X)^* \mid \exists w' \in L(S) : w'|_{\Sigma\backslash X} = w\}$,
(ii) $ET(S/X) = \{w \in (\Sigma\backslash X)^* \mid \exists w' \in ET(S) : w'|_{\Sigma\backslash X} = w\}$,
(iii) $EL(S/X) = \{w \in (\Sigma\backslash X)^* \mid \exists w' \in EL(S) : w'|_{\Sigma\backslash X} = w\}$.

Beweis. Zuerst wird hier die Richtigkeit der Aussagen (i) bis (iii) gezeigt. Daraus kann dann der Rest des Satzes gefolgert werden.

(i) Für ein Wort aus der Sprache L eines Transitionssystems S gilt nach Definition $q_0 \stackrel{w'}{\Rightarrow} q$ mit $q \in Q$. Es gibt also zu jedem $w' = a_1 a_2 \dots a_n \in L(S)$ ein Ablauf $q_0 \stackrel{a_1}{\Rightarrow} q_1 \stackrel{a_2}{\Rightarrow} \dots \stackrel{a_n}{\Rightarrow} q_n$ mit $q_n = q$. Hier ist wichtig zu beachten, dass die jeweiligen Zustände nicht exakt über eine Transition erreicht werden müssen. Es kann sich hier um eine Transitionsfolge aus beliebig vielen τs und dem jeweiligen $a_i \in \Sigma$ handeln. Dabei ist egal, an welcher Stelle das a_i auftaucht. Dies ist notwendig, da auf Trace-Ebene nicht mehr festgehalten wird, wann τ-Transitionen auszuführen sind um mit einer bestimmten Transition den Weg fortsetzen zu können. Dies ändert jedoch nichts daran, dass alle a_i atomare Aktionen darstellen.

- Fall 1 ($n = 0$): Es gilt $w' = \varepsilon$. Somit enthält w' keine Aktionen aus X. Es werden also durch die Anwendung des Internalisierungsoperators in diesem w' keine Aktionen verborgen. Es gilt also $w' = w \in L(S/X)$. Somit ist für diesen Fall die Aussage über L korrekt.
- Fall 2 ($n \geq 1$): Nach der Internalisierung bleiben von dem Ablauf nur noch die Aktionen übrig, die nicht Elemente aus X sind. Der Ablauf reduziert sich also auf $q_0 \stackrel{\tau \text{ falls } a_1 \in X \text{ sonst } a_1}{\Longrightarrow} q_1 \stackrel{\tau \text{ falls } a_2 \in X \text{ sonst } a_2}{\Longrightarrow} \dots \stackrel{\tau \text{ falls } a_n \in X \text{ sonst } a_n}{\Longrightarrow} q_n$. Dabei bleibt durch das Hiding von Aktionen aus X in w' nur noch $w := w'|_{\Sigma\backslash X}$ erhalten. Diese Projektion des Wortes w' auf die eingeschränkte Aktionenmenge ist dann in $L(S/X)$ enthalten, da immer noch der selbe Zustand durch das Worte erreicht wird. Es gilt also auch für diesen Fall die Aussage über die Sprache L.

Für ein Wort w aus der Sprache L des Transitionssystems S/X existiert wie oben auch ein Ablauf. Hier ist es jedoch wichtig, dass auch τ-Transitionen gemacht werden um dieses Wort auszuführen. In dem ein Teil dieser τ-Transitionen durch Transitionen mit Elementen aus X ersetzt werden erhält man ein Trace w' aus der Sprache $L(S)$.

(ii) Es wird ein Trace $w' = a_1 a_2 \dots a_n \in ET(S)$ gewählt. Dieses Trace muss nicht wie bei Punkt (i) einem Ablauf in S entsprechen. Jedoch kann ein Präfix von w' gefunden werden, dass besondere Eigenschaften erfüllen soll und für dass es einen Ablauf gibt. Hierfür muss jedoch unterschieden werden, aus welchem Grund das w' in $ET(S)$ enthalten ist.

- Fall 1 ($w' \in cont(PrET(S))$): In diesem Fall gibt es einen Ablauf für ein Präfix dieses w's, der zu einem Ablauf zu einem Zustand aus E ergänzt werden kann. Der Ablauf hat also die Form $q_0 \overset{a_1}{\Rightarrow} q_1 \overset{a_2}{\Rightarrow} \ldots \overset{a_m}{\Rightarrow} q_m \overset{b_1}{\Rightarrow} q_{m+1} \overset{b_2}{\Rightarrow} \ldots \overset{b_l}{\Rightarrow} q_{m+l}$ mit $m \leq n$, $l \geq 0$ und $\forall i \in \{1, \ldots, l\} : b_i \in O(S)$. Es gilt dann $a_1 a_2 \ldots a_m b_1 b_2 \ldots b_l \in StET(S)$. Analog wie im Beweisteil zu (i) wird dieser Ablauf durch die Internalisierung reduziert. Somit ist die Projektion von $a_1 a_2 \ldots a_m b_1 b_2 \ldots b_l$ auf die eingeschränkte Aktionenmenge auf jeden Fall in $StET(S/X) \subseteq ET(S/X)$ enthalten. Da $b_1 b_2 \ldots b_l \in O(S)^*$, gilt nach der Projektion auch $(b_1 b_2 \ldots b_l)|_{\Sigma \setminus X} \in O(S/X)^*$ und somit $prune((a_1 a_2 \ldots a_m)|_{\Sigma \setminus X}) = prune((a_1 a_2 \ldots a_m b_1 b_2 \ldots b_l)|_{\Sigma \setminus X})$. Da ET eine Menge ist, die nach Definition unter *cont* abgeschlossen ist, sind alle Verlängerung von $(a_1 a_2 \ldots a_m)|_{\Sigma \setminus X}$ ebenfalls in $ET(S/X)$ enthalten. Es gilt also speziell auch $w := w'|_{\Sigma \setminus X} \in ET(S/X)$. Da alle Elemente aus $ET(S/X)$ nur Aktionen aus $\Sigma \setminus X$ enthalten, ist ausgeschlossen, dass eine Fortsetzung mit Aktionen außerhalb dieser Menge möglich ist.
- Fall 2 ($w' \in cont(MIT(S))$): In diesem Fall ist bereits für ein Präfix von w' ein Ablauf zu einem Zustand möglich, der nicht für alle Inputs eine Transitionsmöglichkeit bietet. Der Ablauf hat also die Form $q_0 \overset{a_1}{\Rightarrow} q_1 \overset{a_2}{\Rightarrow} \ldots \overset{a_{m-1}}{\Rightarrow} q_{m-1} \overset{a_m}{\not\Rightarrow}$ mit $m \leq n$ und $a_1 a_2 \ldots a_m \in MIT(S)$. Ab hier Verläuft die Argumentation für diesen Fall analog zu der vom letzten Fall, nur das die *prune*-Funktion nicht verwendet werden muss, da es hier keine Verlängerungen um Outputs gab.

Ein Trace $w \in ET(S/X)$ muss wie oben nicht im Transitionssystem enthalten sein. Es kann jedoch ein Präfix von w gefunden werden wie oben, mit dem ein Error-Zustand, über eine Verlängerung, oder ein Zustand, für den nicht alle Input-Transitionen möglich sind, erreicht werden. Für dieses Präfix von w bzw. für das Präfix von w mit der entsprechenden Verlängerung existiert ein Ablauf im Transitionssystem S/X, der wie in (i) beschrieben, durch das ersetzen von τ-Transitionen, auf einen Ablauf in S erweitert werden kann. Dieser Ablauf kann dann analog zu den beiden Fällen oben verkürzt und Aufgrund des Abschlusses gegenüber *cont* verlängert werden, so dass ein $w' \in ET(S)$ entsteht, dass sich von w nur durch hinzugefügte Aktionen aus X unterscheidet.

(iii) Für ein Trace $w' = a_1 a_2 \ldots a_n \in EL(S)$ gilt $w' \in L(S)$ oder $w' \in ET(S)$. Für beide Fälle wurde oben bereits gezeigt, dass dann $w := w'|_{\Sigma \setminus X}$ in der entsprechenden Menge des Transitionssystems S/X enthalten ist. Da EL als Vereinigung aus den Mengen L und ET definiert ist, ist dadurch dann auch gezeigt, dass $w \in EL(S/X)$ gilt.
Ebenfalls analog zu den beiden vorangegangen Punkten kann auch argumentiert werden, dass zu jedem $w \in EL(S/X)$ durch hinzufügen von Aktionen aus X ein $w' \in EL(S)$ gefunden werden kann.

Da $S_1 \sqsubseteq_E S_2$ gilt, weiß man, dass $ET_1 \subseteq ET_2$ und $EL_1 \subseteq EL_2$ gilt. Durch die Aussagen (i) bis (iii) kann draus direkt gefolgert werden, dass auch $ET(S_1/X) \subseteq ET(S_2/X)$ und $EL(S_1/X) \subseteq EL(S_2/X)$ gilt, da zu jedem Trace aus $ET(S)$ bzw. $EL(S)$ ein entsprechendes Trace aus $ET(S/X)$ bzw. $EL(S/X)$ gefunden werden kann und umgekehrt.

Es folgt also insgesamt, dass die Relation $\sqsubseteq_E$ trotz Hiding erhalten bleibt und somit diese Relation bezüglich des Internalisierungsoperator eine Präkongruenz darstellt. □

Aus 3.6 ist bekannt, dass $\sqsubseteq_E$ eine Präkongruenz bezüglich $\cdot\|\cdot$ ist, und aus 3.11, dass $\sqsubseteq_E$ auch eine Präkongruenz bezüglich $\cdot/\cdot$ ist. Da sich nach Definition 2.7 die Parallelkomposition mit Internalisierung nur aus diesen Operatoren zusammensetzt, erhalten man das folgende Korollar.

Korollar 3.12 (*Error-Präkongurenz mit Internalisierung*). *$\sqsubseteq_E$ ist eine Präkongruenz bezüglich $\cdot|\cdot$.*

4 Verfeinerung für Error- und Ruhe-Freiheit

4.1 Präkongruenz für Ruhe

In diesem Kapitel wird es nicht mehr nur um die Erreichbarkeit von Error-Zuständen gehen, sondern auch um die Erreichbarkeit von Ruhe-Zuständen. Es wird dabei ein analoge Herangehensweise zur der im letzten Kapitel angewendet, jedoch wird [CJK13] als Quelle verwendet. Darin werden ähnliche Konzepte beschrieben, jedoch aus Sicht der Traces. Es werden dort zudem gleichzeitig auch noch Traces mit Divergenz betrachtet. Diese Eigenschaft wird hier zunächst nicht betrachtet.
Die Zustände, die keine Outputs und keine Transitionsmöglichkeit für eine interne Aktion haben, werden als eine Art Verklemmung angesehen, da sie ohne einen Input von einem Kommunikationspartner den Zustand nicht mehr verlassen können.

Definition 4.1 (*Ruhe*). *Ein* Ruhe-Zustand *ist ein Zustand in einem EIO, der keine Outputs und kein τ zulässt.*
Somit ist die Menge der Ruhe-Zustände in einem EIO wie folgt formal definiert: $Qui := \left\{ q \in Q \mid \forall \alpha \in (O \cup \{\tau\}) : q \overset{\alpha}{\not\to} \right\}$.

Für die Erreichbarkeit wird wie im letzten Kapitel wieder der optimistischen Ansatz der lokalen Erreichbarkeit für die Error-Zustände verwendet. Ruhe ist kein unabwendbarer Fehler, sondern kann durch einen Input repariert werden. Daraus ergibt sich, dass ein Ruhe-Zustand als nicht so „schlimmer Fehler“ anzusehen wie ein Error. Somit ist ein Ruhe-Zustand ebenso wie ein Error-Zustand erreichbar, sobald er durch Outputs und τs erreicht werden kann, jedoch ist nicht jede beliebige Fortsetzung eines Traces, das durch lokale Aktionen zu einem Ruhe-Zustand führt, ein Ruhetrace.

Definition 4.2 (*error- und ruhe-freie Kommunikation*). *Zwei EIOs S_1 und S_2* kommunizieren error- und ruhe-frei*, wenn in ihrer Parallelkomposition S_{12} keine Errors und keine Ruhe-Zustände lokal erreichbar sind.*

Definition 4.3 (*Ruhe-Verfeinerungs-Basisrelation*). *Für EIOs S_1 und S_2 mit der gleichen Signatur wird $S_1 \sqsubseteq^B_{Qui} S_2$ geschrieben, wenn ein Error oder Ruhe-Zustand in S_1 nur dann lokal erreichbar ist, wenn ein solcher auch in S_2 lokal erreichbar ist. Diese* Basisrelation *stellt eine* Verfeinerung *bezüglich* Error *und* Ruhe *dar.*
$\sqsubseteq^C_{Qui}$ bezeichnet die vollständig abstrakte Präkongruenz *von $\sqsubseteq^B_{Qui}$ bezüglich $\cdot \| \cdot$.*

Um eine genauere Auseinandersetzung mit den Präkongruenzen zu ermöglichen, benötigt man wie im letzten Kapitel die Definition von Traces auf der Struktur. Dadurch erhält man die Möglichkeit, die gröbste Präkongruenz charakterisieren zu können.
Wie bereits oben erwähnt, sind Ruhe-Zustände reparierbare Fehler im Gegensatz zu Errors. Somit werden keine gekürzten Ruhetraces benötigt, bei denen die *prune*-Funktion zur Anwendung käme und auch keine beliebigen Verlängerungen.

Definition 4.4 (*Ruhetraces*). *Sei S ein EIO und definiere:*

- strikte Ruhetraces*:* $StQT(S) := \{w \in \Sigma^* \mid q_0 \stackrel{w}{\Rightarrow} q \in Qui\}$.

Es wird nur eine Semantik für die Ruhe definiert, die Error-Semantik wird aus dem letzten Kapitel übernommen. Somit gelten für ET und EL die Definitionen aus dem letzten Kapitel.

Definition 4.5 (*Ruhe-Semantik*). *Sei S ein EIO.*

- *Die Menge der* error-gefluteten Ruhetraces *von S ist* $QT(S) := StQT(S) \cup ET(S)$.

Für zwei EIOs S_1, S_2 mit der gleichen Signatur wird $S_1 \sqsubseteq_{Qui} S_2$ geschrieben, wenn $S_1 \sqsubseteq_E S_2$ und $QT_1 \subseteq QT_2$ gilt.

Für die Menge der error-gefluteten Ruhetraces QT wurde eine Informationsvermischung mit den Errortraces vorgenommen wie beim Fluten der Sprache EL. Da jedoch durch die Ruhetraces keine neuen Traces entstehen, die nicht bereits in der gefluteten Sprache EL enthalten wären, würde eine neue Flutung von EL nichts ändern. Es wird also durch die Relation $\sqsubseteq_{Qui}$ nur die bereits existierende Präkongruenz $\sqsubseteq_E$ eingeschränkt.
Das folgende Lemma soll explizit festhalten, wie Ruhezustände sich unter der Parallelkomposition verhalten. Dies ist vor allem für den danach folgenden Satz relevant.

Lemma 4.6 (*Ruhe-Zustände unter Parallelkomposition*).

1. *Ein Zustand (q_1, q_2) aus der Parallelkomposition S_{12} ist ruhig, wenn es auch die Zustände q_1 und q_2 in S_1 bzw. S_2 sind.*
2. *Wenn der Zustand (q_1, q_2) ruhig ist und nicht in E_{12} enthalten ist, dann sind auch die auf die Teilsysteme projizierten Zustände q_1 und q_2 ruhig.*

Beweis. 1.:
Da $q_1 \in Qui_1$ und $q_2 \in Qui_2$ gilt, haben diese beiden Zustände jeweils höchstens die Möglichkeiten Transitionen auszuführen, die mit Inputs beschriftet sind, jedoch keine Möglichkeiten, für Outputs oder τs.
Angenommen der Zustand, der durch Parallelkomposition aus den Zuständen q_1 und q_2 entsteht, ist nicht ruhig, d.h. er hat die Möglichkeit für eine Transition mit einem Output oder einem τ.

- Fall 1 ($(q_1, q_2) \stackrel{\tau}{\rightarrow}$): Ein τ ist eine interne Aktion und kann in dieser Arbeit nicht durch das verbergen von Aktionen bei der Synchronisation entstehen. Ein τ in der Parallelkomposition ist also nur möglich, wenn dies bereits für einen der beiden

Zustände ausführbar war, aus denen der Zustand zusammen gesetzt ist. Jedoch ist eine τ-Transition für q_1 und q_2 ausgeschlossen, deshalb kann auch (q_1, q_2) keine solche Transition ausführen.

- Fall 2 ($(q_1, q_2) \xrightarrow{a}$ mit $a \in O \backslash Synch(S_1, S_2)$): Da es sich bei a um einen Output handelt, der nicht in $Synch(S_1, S_2)$ enthalten ist, kann dieser nicht aus der Synchronisation von zwei Aktionen entstanden sein, sondern muss bereits für S_1 oder S_2 ausführbar gewesen sein. Es gilt also oBdA $q_1 \xrightarrow{a}$ mit $a \in O_1$. Dies ist jedoch Aufgrund der Voraussetzungen nicht möglich. Somit kann die Parallelkomposition diese Transition für (q_1, q_2) ebenfalls nicht ausführt werden.
- Fall 3 ($(q_1, q_2) \xrightarrow{a}$ mit $a \in O \cap Synch(S_1, S_2)$): Der Output a ist in diesem Fall durch Synchronisation von einem Output mit einem Input entstanden. OBdA gilt $a \in O_1 \cap I_2$. Für die einzelnen Systeme muss also gelten, dass $q_1 \xrightarrow{a}$ und $q_2 \xrightarrow{a}$ gilt. Die Transition für das System S_1 ist jedoch in der Voraussetzung ausgeschlossen worden. Somit ist es nicht möglich, dass S_{12} diese Transition für den Zustand (q_1, q_2) ausführen kann.

Da alle diese Fälle zu einem Widerspruch mit der Voraussetzung führen, folgt, dass bereits die Annahme, dass der Zustand (q_1, q_2) nicht ruhig ist, falsch war. Es folgt also, dass $(q_1, q_2) \in Qui_{12}$ gilt.

2.:
Es gilt $(q_1, q_2) \in Qui_{12} \backslash E_{12}$, somit hat dieser Zustand allen falls die Möglichkeit, Transitionen für Inputs auszuführen.
Angenommen $q_1 \notin Qui_1$, dann ist für q_1 entweder eine τ-Transition oder eine Output-Transition möglich.

- Fall 1 ($q_1 \xrightarrow{\tau}$): Da diese Transition für S_1 möglich ist, kann auch S_{12} im Zustand (q_1, q_2) diese Transition ausführen. Dies ist jedoch durch die Voraussetzung verboten. Somit gilt auch $q_1 \overset{\tau}{\not\rightarrow}$.
- Fall 2 ($q_1 \xrightarrow{a}$ mit $a \in O_1 \backslash Synch(S_1, S_2)$): Da es sich bei a um einen Output handelt, der nicht zu synchronisieren ist, wird dieser einfach in die Parallelkomposition übernommen. Es müsste also $(q_1, q_2) \xrightarrow{a}$ gelten, was jedoch verboten ist. Somit kann die Transition für S_1 in diesem Fall auch nicht möglich sein.
- Fall 3 ($q_1 \xrightarrow{a}$ mit $a \in O_1 \cap Synch(S_1, S_2)$ und $q_2 \xrightarrow{a}$): In diesem Fall ist die Synchronisation des Outputs a von S_1 mit dem Input a von S_2 möglich, so dass in der Parallelkomposition der Output a als Transition für (q_1, q_2) entsteht. Diese Transitionsmöglichkeit ist jedoch für S_{12} nach Voraussetzung nicht erlaubt. Es folgt also auch, dass dieser Fall nicht eintreten kann.
- Fall 4 ($q_1 \xrightarrow{a}$ mit $a \in O_1 \cap Synch(S_1, S_2)$ und $q_2 \overset{a}{\not\rightarrow}$): Da S_2 diese Transition nicht ausführen kann, handelt es sich hier um einen neuen Error, da die Synchronisation an dieser Stelle nicht möglich ist. Es würde also $(q_1, q_2) \in E_{12}$ gelten, dies wurde

jedoch in der Voraussetzung bereits ausgeschlossen. Somit ist dieser Fall nicht möglich.

Alle aufgeführten Fälle führen zu einem Widerspruch mit der Voraussetzung, somit folgt, dass die Annahme bereits falsch war und $q_1 \in Qui_1$ gelten muss. Analog kann für q_2 argumentiert werden, so dass dann auch $q_2 \in Qui_2$ folgt. □

In dem folgenden Satz sind Punkt 1. und 3. nur zur Vollständigkeit aufgeführt. Sie entsprechen Punkt 1. und 2. aus Satz 3.5.

Satz 4.7 (*Error- und Ruhe-Semantik für Parallelkompositonen*). *Für zwei komponierbare EIOs S_1, S_2 und ihre Komposition S_{12} gilt:*

1. $ET_{12} = cont(prune((ET_1 \| EL_2) \cup (EL_1 \| ET_2)))$,
2. $QT_{12} = (QT_1 \| QT_2) \cup ET_{12}$,
3. $EL_{12} = (EL_1 \| EL_2) \cup ET_{12}$.

Beweis. Es wird nur der 2. Punkt bewiesen.

„$\subseteq$“:
Hier muss unterschieden werden, ob ein $w \in StQT_{12} \backslash ET_{12}$ oder ein $w \in ET_{12}$ betrachtet wird. Im zweiten Fall ist das w in der rechten Seite enthalten. Somit wird ab jetzt ein $w \in StQT_{12} \backslash ET_{12}$ betrachtet und dessen Zugehörigkeit zur rechten Menge versucht zu zeigen. Aufgrund von Definition 4.4 weiß man, dass $(q_{01}, q_{02}) \overset{w}{\Rightarrow} (q_1, q_2)$ gilt mit $(q_1, q_2) \in Qui_{12} \backslash E_{12}$. Durch Projektion erhält man $q_{01} \overset{w_1}{\Rightarrow} q_1$ und $q_{02} \overset{w_2}{\Rightarrow} q_2$ mit $w \in w_1 \| w_2$. Aus $(q_1, q_2) \in Qui_{12} \backslash E_{12}$ kann mit dem zweiten Punkt von Lemma 4.6 gefolgert werden, 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.

„$\supseteq$“:
Es muss wieder danach unterschieden werden, aus welcher Menge das betrachtete Element stammt. Falls $w \in ET_{12}$ gilt, so kann die Zugehörigkeit zur linken Seite direkt gefolgert werden. Somit wird für den weiteren Beweis dieser Inklusionsrichtung ein Element $w \in QT_1 \| QT_2$ betrachtet und gezeigt, dass es in der linken Menge enthalten ist. Da $QT_i = StQT_i \cup ET_i$ gilt, existieren für w_1 und w_2 mit $w \in w_1 \| w_2$ unterschiedliche Möglichkeiten:

- Fall 1 ($w_1 \in ET_1 \vee w_2 \in ET_2$): OBdA gilt $w_1 \in ET_1$. Nun kann $w_2 \in StQT_2 \subseteq L_2$ gelten oder $w_2 \in ET_2$ und somit gilt auf jeden Fall $w_2 \in EL_2$. Daraus kann dann mit dem ersten Punkt von Satz 3.5 gefolgert werden, dass $w \in ET_{12}$ gilt und somit w in der linken Seite der Gleichung enthalten ist.

- Fall 2 ($w_1 \in StQT_1 \backslash ET_1 \wedge w_2 \in StQT_2 \backslash ET_2$): Es gilt in diesem Fall ist $q_{01} \stackrel{w_1}{\Rightarrow} q_1 \in Qui_1$ und $q_{02} \stackrel{w_2}{\Rightarrow} q_2 \in Qui_2$. Da q_1 und q_2 in der jeweiligen Ruhe-Menge enthalten sind, ist auch der Zustand, der aus ihnen zusammengesetzt ist, in der Parallelkomposition ruhig, wie bereits in ersten Punkt von Lemma 4.6 gezeigt. Es gilt also für die Komposition $(q_{01}, q_{02}) \stackrel{w}{\Rightarrow} (q_1, q_2) \in Qui_{12}$ und dadurch ist w in der linken Seite der Gleichung enthalten, da $w \in StQT_{12} \subseteq QT_{12}$ gilt.

□

Die folgende Proposition ist eine direkte Folgerung aus dem letzten Satz. Jedoch ist es eine wichtige Feststellung um zu zeigen, dass es sich bei der Relation $\sqsubseteq_{Qui}$ um die gröbste Präkongruenz handeln könnte, die in diesem Kapitel charakterisiert werden soll.

Proposition 4.8 (*Ruhe-Präkongruenz*). *$\sqsubseteq_{Qui}$ ist eine Präkongruenz bezüglich $\cdot\|\cdot$.*

Beweis. Es muss gezeigt werden: Wenn $S_1 \sqsubseteq_{Qui} S_2$ gilt, so auch $S_{31} \sqsubseteq_{Qui} S_{32}$ für jedes komponierbare System S_3. D.h. es ist zu zeigen, dass aus $S_1 \sqsubseteq_E S_2$ und $QT_1 \subseteq QT_2$ sowohl $S_{31} \sqsubseteq_E S_{32}$ als auch $QT_{31} \subseteq QT_{32}$ folgt. Dies ergibt sich wie im Beweis zu Proposition 3.6 aus der Monotonie von $\cdot\|\cdot$ auf Sprachen wie folgt:

- $S_{31} \overset{\substack{\text{Proposition 3.6}\\ \text{und}\\ S_1 \sqsubseteq_E S_2}}{\sqsubseteq_E} S_{32}$,
- $QT_{31} \stackrel{4.7\ 2.}{=} (QT_3 \| QT_1) \cup ET_{31}$
  $\overset{\substack{ET_{31} \subseteq ET_{32}\\ \text{und}\\ QT_1 \subseteq QT_2}}{\subseteq} (QT_3 \| QT_2) \cup ET_{32}$
  $\stackrel{4.7\ 2.}{=} QT_{32}$.

□

Im nächsten Lemma soll eine Verfeinerungsrelation bezüglich guter Kommunikation mit Partnern im Sinne von error- und ruhe-freier Kommunikation betrachtet werden.

Lemma 4.9 (*Verfeinerung mit Ruhe-Zuständen*). *Gegeben sind zwei EIOs S_1 und S_2 mit der gleichen Signatur. Wenn $U\|S_1 \sqsubseteq^B_{Qui} U\|S_2$ für alle Partner U gilt, dann folgt daraus $S_1 \sqsubseteq_{Qui} S_2$.*

Beweis. Da davon ausgegangen wird, dass S_1 und S_2 die gleiche Signatur haben, definiert man $I := I_1 = I_2$ 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 die folgenden Punkte nachgewiesen werden:

- $S_1 \sqsubseteq_E S_2$,

- $QT_1 \subseteq QT_2$.

In Lemma 3.7 wurde bereits etwas Ähnliches gezeigt, jedoch wurde dort als Voraussetzung $U\|S_1 \sqsubseteq_E^B U\|S_2$ für alle Partner U verwendet und hier die selbe Aussage mit der Basisrelation der Ruhe. Dadurch, dass die hier verwendete Basisrelation nichts über die Art der erreichbaren Fehlers in den Komponenten aussagt, kann der Beweis aus 3.7 nicht verwendet werden. Es kann also aus der lokalen Erreichbarkeit eines Errors in S_1' und dem relationalen Zusammenhang von $S_1' \sqsubseteq_{Qui}^B S_2'$ nur geschlossen werden, dass in S_2' auch ein Fehler lokal erreichbar ist, jedoch kann dieser Fehler ein Error oder ein Ruhe-Zustand sein. Analog verhält es sich, wenn in S_1' ein Ruhe-Zustand lokal erreichbar ist. Es muss also für den ersten Punkt folgendes nachgewiesen werden:

- $ET_1 \subseteq ET_2$,
- $EL_1 \subseteq EL_2$.

Es wird nun damit begonnen, den ersten Unterpunkt des ersten Beweispunktes zu zeigen, d.h. es wird unter der Voraussetzung, dass $U\|S_1 \sqsubseteq_{Qui}^B U\|S_2$ gilt, gezeigt, dass $ET_1 \subseteq ET_2$ gilt. Da beide ET-Mengen unter *cont* abgeschlossen sind, reicht es ein präfix-minimales Element $w \in ET_1$ zu betrachten und zu zeigen, dass dieses w oder eines seiner Präfixe in ET_2 enthalten ist.

- Fall 1 ($w = \varepsilon$): Es handelt sich um einen lokal erreichbaren Error in S_1. Für U wird ein Transitionssystem verwendet, das nur aus dem Startzustand, einer Schleife für alle Inputs $x \in I_U$ und einer Schleife für τ besteht. Somit kann S_1 die im Prinzip gleichen Error-Zustände lokal erreichen wie $U\|S_1$. Es folgt also, dass in $U\|S_2$ ein Fehler lokal erreichbar ist. Es kann sich bei dem Fehler nur um einen Error handeln, da es in der Komposition mit U keine Ruhe-Zustände geben kann. Da U keinen Error-Zustand und auch keine fehlenden Input-Möglichkeiten enthält, kann der Error nur von S_2 geerbt sein. Somit muss in S_2 ein Error-Zustand lokal erreichbar sein, d.h. es gilt $\varepsilon \in PrET_2 \subseteq ET_2$.
- Fall 2 ($w = x_1 \dots x_n x_{n+1} \in \Sigma^+$ mit $n \geq 0$ und $x_{n+1} \in I$): Es wird der folgende Partner U betrachtet (siehe auch Abbildung 4.1):
  - $Q_U = \{q_0, q_1, \dots, q_{n+1}\}$,
  - $q_{0U} = q_0$,
  - $E_U = \emptyset$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i \leq n\}$
    $\cup \{(q_i, x, q_{n+1}) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i \leq n\}$
    $\cup \{(q_{n+1}, x, q_{n+1}) \mid x \in I_U\}$
    $\cup \{(q_i, \tau, q_i) \mid 0 \leq i \leq n+1\}$.

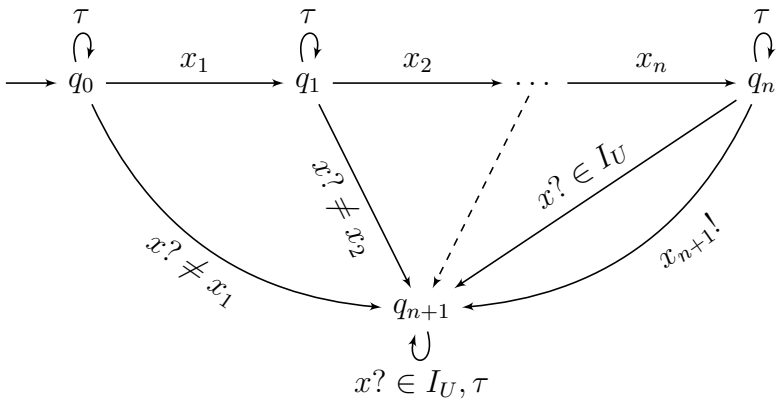

Abbildung 4.1: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$

Die Menge der Ruhe-Zustände des hier betrachteten Us ist leer. Da im Vergleiche zum Transitionssystem in Abbildung 3.1 nur die τ-Schlingen ergänzt wurden, ändert sich nichts an den Fällen 2a) und 2b). Die Begründungen, wieso in den beiden Fällen $\varepsilon \in PrET(U \| S_1)$ gilt bleibt also analog zum Beweis von Lemma 3.7. Durch die τ-Schlingen wurde, genau wie im letzten Fall, nur erreicht, dass in einer Parallelkomposition mit U keine Ruhe-Zustände möglich sind. Es kann also auch hier aus der lokalen Erreichbarkeit eines Error-Zustandes in $U \| S_1$ auf die lokale Erreichbarkeit eines Errors in $U \| S_2$ geschlossen werden. Die weitere Argumentation verläuft dann analog zu Fall 2 der selben Inklusion im Beweis zu Lemma 3.7. Da τs nur interne Aktionen eines einzelnen Systems sind, verändert sich auch nichts an den Traces, über die argumentiert wird. Es können zwar möglicherweise τ-Transitionen ausgeführt werden, diese können jedoch weder zu einem Fehler führen noch beeinflussen, dass ein anderes Trace nicht ausgeführt werden kann.

Nun wird mit dem zweiten Unterpunkt des ersten Beweispunktes begonnen. Genau wie im Beweis zu 3.7 ist hier jedoch auf Grund des bereits geführten Beweisteils nur noch $L_1 \backslash ET_1 \subseteq EL_2$ zu zeigen. Es wird also für ein beliebig gewähltes $w \in L_1 \backslash ET_1$ gezeigt, dass es auch in EL_2 enthalten ist.

- Fall 1 ($w = \varepsilon$): Ebenso wie in 3.7 gilt auch hier, dass ε immer in EL_2 enthalten ist.
- Fall 2 ($w = x_1 \dots x_n$ mit $n \geq 1$): Die Konstruktion des Partners U weicht wie im letzten Beweisteil nur durch die τ-Schleifen an den Zuständen des Transitionssystems vom Beweis zu Lemma 3.7 ab. Somit ist der Partner U dann wie folgt definiert (siehe dazu auch Abbildung 4.2):
  - $Q_U = \{q, q_0, q_1, \dots, q_n\}$,
  - $q_{0U} = q_0$,
  - $E_U = \{q_n\}$,

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

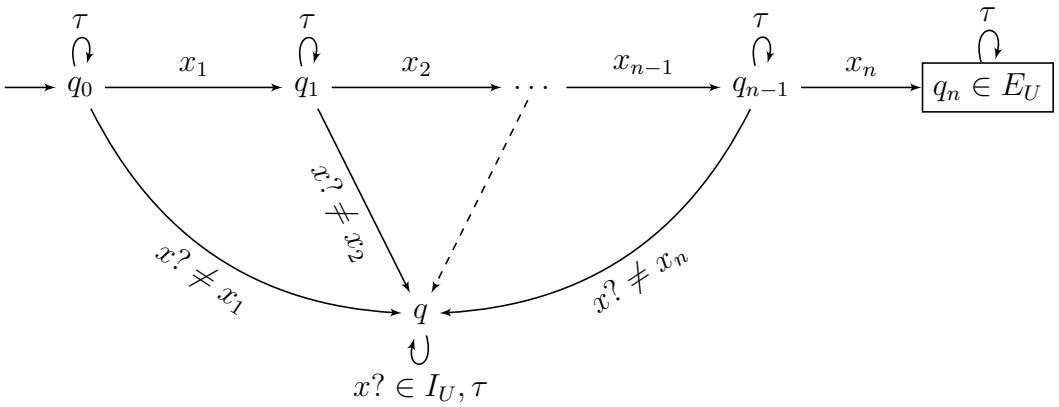

Abbildung 4.2: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$, q_n ist der einzige Error-Zustand

Da durch die τ-Schlingen an den Zuständen wie oben vermieden wird, dass es in einer Komposition mit U und auch in U selbst Ruhe-Zustände gibt, verläuft der Rest des Beweises dieses Punktes analog zum Beweis von Lemma 3.7. Und somit gilt für alle Fälle (2a) bis 2d)), dass w in EL_2 enthalten ist.

So bleibt nun nur noch der letzte Beweispunkt zu zeigen, d.h. die Inklusion $QT_1 \subseteq QT_2$. Diese Inklusion kann jedoch noch, anlog zum Beweis der Inklusion der error-gefluteten Sprachen, weiter eingeschränkt werden. Da bereits bekannt ist, dass $ET_1 \subseteq ET_2$ gilt, muss nur noch $StQT_1 \backslash ET_1 \subseteq QT_2$ gezeigt werden.
Es wird ein $w \in StQT_1 \backslash ET_1$ gewählt und gezeigt, dass es auch in QT_2 enthalten ist.
Durch die Wahl des ws wird vom Startzustand von S_1 durch das Wort w ein ruhiger Zustand erreichbar. Dies hat nur Auswirkungen auf die Parallelkomposition $U \| S_1$, wenn in U ebenfalls ein Ruhe-Zustand durch w erreichbar ist.

- Fall 1 ($w = \varepsilon$): Es ist ein Ruhe-Zustand intern erreichbar in S_1, da jedoch $\varepsilon \notin ET_1$ gilt, ist kein Error lokal erreichbar. Für U wird ein Transitionssystem verwendet, das nur aus dem Startzustand ohne Transitionsmöglichkeiten besteht. Somit ist auch in U ein Ruhe-Zustand lokal erreichbar. Es folgt also mit Lemma 4.6, dass auch in $U \| S_1$ ein Ruhe-Zustand lokal erreichbar ist. Es muss also auch in $U \| S_2$ ein Fehler lokal erreichbar sein. Bei diesem Fehler kann es sich nun um einen Error oder um einen Ruhe-Zustand handeln, dies ist jedoch Aufgrund der Definition der Menge QT nicht relevant. Da für U keine Transitionen möglich sind, kann nur ε ein striktes Ruhetrace sein, falls es sich bei dem Fehler in $U \| S_2$ um einen lokal erreichbaren Ruhe-Zustand handelt. Somit folgt in beiden Fällen, dass w in QT_2 enthalten ist.

- Fall 2 ($w = x_1 \dots x_n \in \Sigma^+$ mit $n \geq 1$): Es wird der folgende Partner U betrachtet (siehe auch Abbildung 4.3):
  - $Q_U = \{q_0, q_1, \dots, q_n, q\}$,
  - $q_{0U} = q_0$,
  - $E_U = \emptyset$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
    $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i < n\}$
    $\cup \{(q_i, \tau, q_i) \mid 0 \leq i < n\}$
    $\cup \{(q_n, x, q) \mid x \in I_U\}$
    $\cup \{(q, \alpha, q) \mid \alpha \in I_U \cup \{\tau\}\}$.

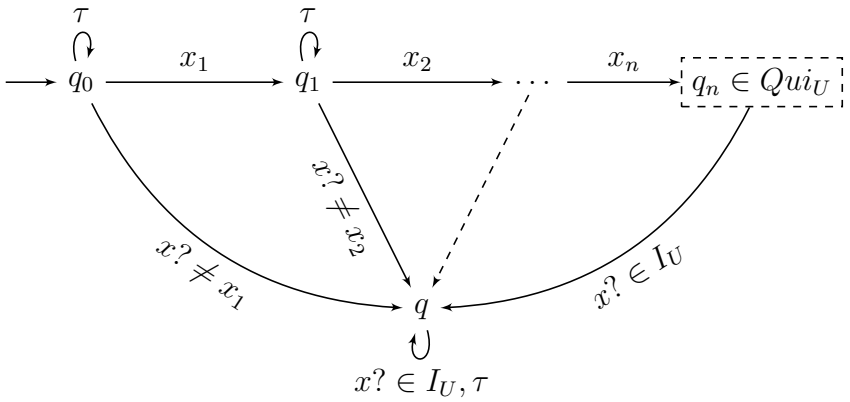

Abbildung 4.3: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$

Der Zustand q_n aus U ist der einzige ruhige Zustand in U. Es gilt wegen Lemma 4.6, dass auch in der Parallelkomposition $U \| S_1$ ein Ruhe-Zustand mit w erreicht wird. Da es sich bei allen in w befindlichen Aktionen um synchronisierte Aktionen handelt und $I_U \cap I = \emptyset$, folgt $w \in O^*_{U \| S_1}$ und $w \in StQT(U \| S_1)$. Es kann also in der Parallelkomposition durch w ein Ruhe-Zustand lokal erreicht werden. Da $w \notin ET_1$ gilt, kann auf dem Weg, der mit w im Transitionssystem S_1 zurück gelegt wird kein Error lokal erreicht werden. Es kann also weder von S_1 noch von U ein Error auf diesem Weg geerbt werden und durch den Aufbau von U kann auch kein neuer Error entstehen. Da ein Ruhe-Zustand in $U \| S_1$ lokal erreichbar ist, muss auch ein Fehler in $U \| S_2$ lokal erreichbar sein. Hier kann jedoch zunächst keine Aussage darüber getroffen werden, ob das w ausführbar ist und ob es sich bei dem Fehler um Ruhe oder Error handelt.

  - Fall 2a) ($\varepsilon \in ET(U \| S_2)$): Es handelt sich bei dem lokal erreichbaren Fehler um einen Error. Es ist somit ist nicht relevant, ob das w ausführbar ist. Der Error kann sowohl von S_2 geerbt sein, wie auch durch fehlende synchronizations Möglichkeiten als neuer Error in der Parallelkomposition entstanden sein.

Es gilt also, dass bereits in S_2 ein Präfix von w in ET_2 enthalten ist, wegen des Beweises des ersten Punktes aus Lemma 3.7 und da U nur Synchronisations-Fehler auf dem Trace w zulässt. Da die Menge ET unter *cont* abgeschlossen ist, gilt also auch $w \in ET_2 \subseteq QT_2$.

- Fall 2b) (Ruhe-Zustand lokal erreichbar in $U \| S_2$ und $\varepsilon \notin ET(U \| S_2)$): Da in U nur durch w ein ruhiger Zustand erreicht werden kann, muss es sich bei dem lokal erreichbaren Ruhe-Zustand in $U \| S_2$ um einen handeln, der mit w erreicht werden kann. Mit Lemma 4.6 kann somit gefolgert werden, dass auch in S_2 ein Ruhe-Zustand mit w erreichbar sein muss. Es gilt also $w \in StQT_2 \subseteq QT_2$.

□

Mit dem folgenden Satz wird festgehalten, dass mit $\sqsubseteq_{Qui}$ die gröbste Präkongruenz charakterisiert wurde bezüglich $\cdot\|\cdot$, die in $\sqsubseteq^B_{Qui}$ enthalten ist.

Satz 4.10 (*Full Abstractness für Ruhe-Semantik*). *Seien S_1 und S_2 zwei EIOs mit derselben Signatur. Dann gilt $S_1 \sqsubseteq^C_{Qui} S_2 \Leftrightarrow S_1 \sqsubseteq_{Qui} S_2$, insbesondere ist $\sqsubseteq_{Qui}$ eine Präkongruenz.*

Beweis. „⇐“: Nach Definition gilt $w \in QT(S)$ mit $w \in O^*$, genau dann wenn in S ein Ruhe-Zustand oder ein Error-Zustand lokal erreichbar ist. $S_1 \sqsubseteq_{Qui} S_2$ impliziert, dass $w \in QT_2$ gilt, wenn $w \in QT_1$ gilt. Somit ist ein Ruhe-Zustand oder ein Error-Zustand nur dann in S_1 lokal erreichbar, wenn auch ein solcher in S_2 lokal erreichbar ist. Daraus folgt, dass $S_1 \sqsubseteq^B_{Qui} S_2$ gilt. Somit ist $\sqsubseteq_{Qui}$ in $\sqsubseteq^B_{Qui}$ enthalten. In Proposition 4.8 wurde festgestellt, dass $\sqsubseteq_{Qui}$ eine Präkongruenz ist. Da jedoch $\sqsubseteq^C_{Qui}$ nach Definition die gröbste Präkongruenz bezüglich $\cdot\|\cdot$ ist, die in $\sqsubseteq^B_{Qui}$ enthalten ist, muss $\sqsubseteq_{Qui}$ in $\sqsubseteq^C_{Qui}$ enthalten sein. Es folgt also aus $S_1 \sqsubseteq_{Qui} S_2$, dass auch der relationale Zusammenhang $S_1 \sqsubseteq^C_{Qui} S_2$ gilt.

„⇒“: Durch die Definition von $\sqsubseteq^C_{Qui}$ als Präkongruenz in 4.3 folgt aus $S_1 \sqsubseteq^C_{Qui} S_2$, dass $U\|S_1 \sqsubseteq^C_{Qui} U\|S_2$ für alle EIOs U gilt, die mit S_1 komponierbar sind. Somit folgt auch die Gültigkeit von $U\|S_1 \sqsubseteq^B_{Qui} U\|S_2$ für alle diese EIOs U. Mit Lemma 4.9 folgt dann $S_1 \sqsubseteq_{Qui} S_2$. □

Es wurde somit, wie im letzten Kapitel, eine Kette an Folgerungen gezeigt, die sich zu einem Ring schließen. Dies ist in Abbildung 4.4 dargestellt.

Aus Satz 4.10 und Lemma 4.9 erhält man das folgende Korollar. Angenommen man definiert, dass S_1 S_2 verfeinern soll, genau dann wenn für alle Partner EIOs U, für die S_2 error- und ruhe-frei mit U kommuniziert, folgt, dass S_1 ebenfalls error- und ruhe-frei mit U kommuniziert. Dann wird auch diese Verfeinerung durch $\sqsubseteq_{Qui}$ charakterisiert.

Korollar 4.11. *Es gilt: $S_1 \sqsubseteq_{Qui} S_2 \Leftrightarrow U\|S_1 \sqsubseteq^B_{Qui} U\|S_2$ für alle Partner U.*

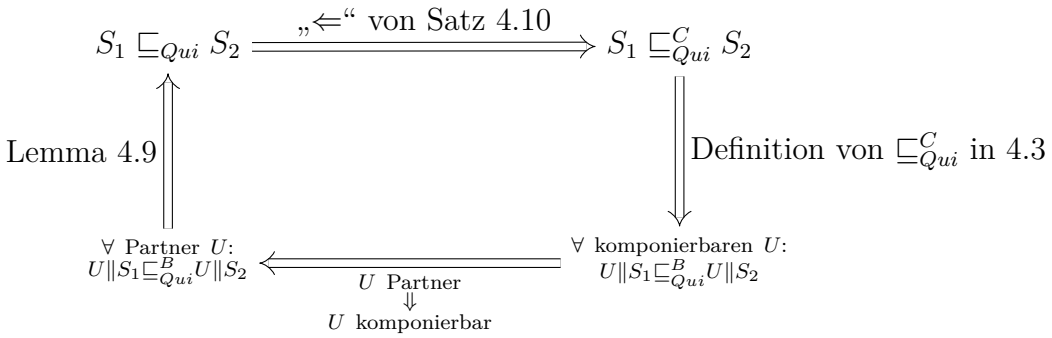

Abbildung 4.4: Folgerungskette

4.2 Hiding und Ruhe-Freiheit

Es soll nun auch hier die Auswirkungen der Internalisierung von Aktionen auf die Verfeinerungsrelationen untersucht werden. Es werden Outputs in interne Aktionen umgewandelt. Da jedoch bei den Ruhe-Zuständen auch τ-Transitionen verboten wurden, verändert sich nichts an der Menge der ruhigen Zustände. Da die Erreichbarkeit von Ruhe-Zuständen mittels lokaler Aktionen betrachtet wurde, kann sich auch nichts an der Erreichbarkeit der Ruhe-Zustände ändern. Somit kann eine analoge Proposition zu 3.10 formuliert werden.

Proposition 4.12 (*Ruhe-Basisrelation bzgl. Internalisierung*). *Wenn $S_1 \sqsubseteq^B_{Qui} S_2$ gilt, dann folgt daraus, dass auch $S_1/X \sqsubseteq^B_{Qui} S_2/X$ gilt.*

Beweis. Dass die Error-Erreichbarkeit unverändert bleibt unter Hiding wurde bereits im Beweis zu Proposition 3.10 gezeigt. Mit der analogen Argumentation folgt auch, dass sich nichts an der Erreichbarkeit der Ruhe-Zustände ändert. Es können durch Hiding nämlich nur Outputs verborgen werden, die bereits in der Menge der lokalen Aktionen enthalten sind. Die Menge der Ruhe-Zustände kann sich durch das Internalisieren auch nicht vergrößern oder verkleinern, wie oben bereits festgestellt. Also gilt: Wenn S_i einen Error oder Ruhe-Zustand lokal erreichen kann, dann kann das auch S_i/X und umgekehrt, da dabei nur τs durch Outputs ersetzt werden. Es folgt also aus der Relation $S_1 \sqsubseteq^B_{Qui} S_2$ auch der Zusammenhang $S_1/X \sqsubseteq^B_{Qui} S_2/X$. □

Satz 4.13 (*Ruhe-Präkongurenz bzgl. Internalisierung*). *Seien S_1 und S_2 zwei EIOs für die $S_1 \sqsubseteq_{Qui} S_2$ gilt, somit gilt auch $S_1/X \sqsubseteq_{Qui} S_2/X$. Es ist also $\sqsubseteq_{Qui}$ eine Präkongruenz bezüglich $\cdot/\cdot$. Es gilt für die Sprachen und Traces:*

(i) $L(S/X) = \{w \in (\Sigma \backslash X)^* \mid \exists w' \in L(S) : w'|_{\Sigma \backslash X} = w\}$,
(ii) $ET(S/X) = \{w \in (\Sigma \backslash X)^* \mid \exists w' \in ET(S) : w'|_{\Sigma \backslash X} = w\}$,
(iii) $EL(S/X) = \{w \in (\Sigma \backslash X)^* \mid \exists w' \in EL(S) : w'|_{\Sigma \backslash X} = w\}$,
(iv) $QT(S/X) = \{w \in (\Sigma \backslash X)^* \mid \exists w' \in QT(S) : w'|_{\Sigma \backslash X} = w\}$.

Beweis. Als erstes sollte man sich von der Richtigkeit der Aussagen (i) bis (iv) überzeugen. Da jedoch im Beweis zu Satz 3.11 bereits die ersten drei Punkte gezeigt wurden, muss hier nur noch (iv) betrachtet werden.

(iv) Die Korrektheit dieser Aussage kann analog zum Beweis der Punkte (i) bis (iii) im Satz 3.11 gezeigt werden. Für ein $w' \in QT(S)$ gilt $w' \in ET(S)$ oder $w' \in StQT(S) \subseteq L(S)$. Es gilt also mit Punkt (i) und (ii), dass $w := w'|_{\Sigma \setminus X}$ in $ET(S/X)$ oder in $StQT(S/X)$ enthalten ist. Für ein $w' \in StQT(S)$ ist jedoch anzumerken, dass im Beweis zu Punkt (i) durch die Einschränkung des Traces immer noch der gleiche Zustand erreicht wird. Es folgt also insgesamt, dass $w := w'_{\Sigma \setminus X} \in QT(S/X)$ gilt.
Für ein $w \in QT(S/X)$ kann ebenfalls wie im Beweis zu Punkt (i) und (iii) durch ersetzen von τs im jeweiligen Ablauf ein Ablauf gefunden werden, der in S enthalten ist. Es gibt also eine Erweiterung w' von w um Aktionen aus X, sodass diese in $QT(S)$ enthalten ist.

Da $S_1 \sqsubseteq_{Qui} S_2$ gilt, kann geschlossen werden, dass $S_1 \sqsubseteq_E S_2$ und $QT_1 \subseteq QT_2$ gilt. Aufgrund von Satz 3.11 ist bekannt, dass daraus $S_1/X \sqsubseteq_E S_2/X$ folgt. Mit der Argumentation für den Punkt (iv) von oben, kann aus der Voraussetzung $QT_1 \subseteq QT_2$ ebenfalls $QT(S_1/X) \subseteq QT(S_2/X)$ gefolgert werden.
Es folgt also insgesamt, dass die Relation $\sqsubseteq_{Qui}$ trotz Hiding erhalten bleibt und somit bezüglich des Hiding diese Relation eine Präkongruenz ist. □

In Definition 2.7 wurde mit Hilfe des Internalisierungsoperator aus der Parallelkomposition ohne Verbergen die Parallelkomposition mit Verbergen der synchronisierten Aktionen nachgebildet. Es kann deren Eigenschaft als Präkongruenz aus den Präkongruenz-Eigenschaften von $\cdot \| \cdot$ und $\cdot / \cdot$ bezüglich $\sqsubseteq_{Qui}$ aus der Proposition 4.8 und dem Satz 4.13 geschlossen werden.

Korollar 4.14 (*Ruhe-Präkongruenz mit Internalisierung*). *$\sqsubseteq_{Qui}$ ist eine Präkongruenz bezüglich $\cdot|\cdot$.*

5 Verfeinerung für Error-, Ruhe- und Divergenz-Freiheit

5.1 Präkongruenz für Divergenz

In diesem Kapitel soll die Menge der betrachteten Zustände noch einmal erweitert werden. Somit werden dann Errors, Ruhe-Zustände und Divergente-Zustände betrachtet. Es eignet sich also [CJK13] als Quelle, da nun auch noch die Divergenz betrachtet wird. Diese wurde dort gleichzeitig mit der Ruhe eingeführt und betrachtet. Da es sich nur um eine Erweiterung der Präkongruenzen aus den letzten beiden Kapiteln handelt, wird dabei ähnlich vorgegangen wie in den letzten beiden Kapiteln.
Wie bereits oben und im letzten Kapitel erwähnt wurden in [CJK13] auch noch divergente Zustände als Fehler-Zustände betrachtet. Um zu klären, was darunter verstanden wird, wird nun noch eine Definition für Divergenz gegeben.

Definition 5.1 (*Divergenz*). *Ein* Divergenz-Zustand *ist ein Zustand in einem EIO, der eine unendliche Folge an τs ausführen kann.*

Als Erreichbarkeitsbegriff wird wieder die lokale Erreichbarkeit verwendet. Da das Divergieren eines Systems nicht mehr verhindert werden kann, sobald ein divergenter Zustand lokal erreichbar ist, ist Divergenz als ähnlich „schlimm“ zu bewerten wie ein Error.

Definition 5.2 (*error-, ruhe- und divergenz-freie Kommunikation*). *Zwei EIOs S_1 und S_2 kommunizieren* error-, ruhe- und divergenz-frei*, wenn in ihre Parallelkomposition S_{12} keine Errors, Ruhe-Zustände und Divergenz-Zustände lokal erreichbar sind.*

Definition 5.3 (*Divergenz-Verfeinerungs-Basisrelation*). *Für EIOs S_1 und S_2 mit der gleichen Signatur wird $S_1 \sqsubseteq^B_{Div} S_2$ geschrieben, wenn ein Error, Ruhe-Zustand oder Divergenz-Zustand in S_1 nur dann lokal erreichbar ist, wenn er auch in S_2 lokal erreichbar ist. Diese* Basisrelation *stellt eine* Verfeinerung *bezüglich* Error, Ruhe *und* Divergenz *dar.*
$\sqsubseteq^C_{Div}$ bezeichnet die vollständige abstrakte Präkongruenz *von $\sqsubseteq^B_{Div}$ bezüglich $\cdot\|\cdot$.*

Da nun die grundlegenden Definitionen für Divergenz festgehalten sind, kann man sich nun einen Begriff für die Traces von divergenten Zuständen bilden. Da oben bereits festgehalten wurde, dass Divergenz als ähnlich „schlimmer“ Fehler anzusehen ist, wie

Error und das Divergieren eines Systems nicht mehr verminderbar ist, sobald ein divergenter Zustand erreichbar ist, kommt für die Divergenztraces wieder die *prune*-Funktion zu Einsatz. Da jedoch nur die Präkongruenz aus dem Error-Kapitel weiter verfeinert werden soll, sollen keine neuen Traces entstehen, die nicht bereits in EL enthalten sind. Es kann also nicht möglich sein abgeschnittene Traces beliebig Fortzusetzen.

Definition 5.4 (*Divergenztraces*). *Sei S ein EIO und definiere:*

- strikte Divergenztraces*:* $StDT(S) := \{w \in \Sigma^* \mid q_0 \stackrel{w}{\Rightarrow} q \in Div\}$,
- gekürzte Divergenztraces*:* $PrDT(S) := \bigcup\{prune'(w) \mid w \in StDT(S)\}$.

Da in [CJK13] bereits direkt Divergenz mit betrachtet wurde, wird dort die Flutung der Traces so vorgenommen, dass $ET \subseteq DT \subseteq QT$ gilt. Dies soll auch hier in diesem Kapitel erreicht werden. Somit kann zwar die Semantik aus dem Error-Kapitel übernehmen werden, jedoch wird für die Ruhe eine andere Semantik benötigt, die sich von der im letzten Kapitel unterscheidet. Die Inklusionskette der Fehlertraces scheint auch von der Hierarchie her auf die Bewertung zu passen, als wie kritisch die einzelnen Fehler zu bewerten sind.

Definition 5.5 (*Ruhe- und Divergenz-Semantik*). *Sei S ein EIO.*

- *Die Menge der* error-gefluteten Divergenztraces *von S ist* $DT(S) := PrDT(S) \cup ET(S)$.
- *Die Menge der* divergenz-gefluteten Ruhetraces *von S ist* $QT(S) := StQT(S) \cup DT(S)$.

Für zwei EIOs S_1, S_2 mit der gleichen Signatur schreibt man $S_1 \sqsubseteq_{Div} S_2$, wenn $S_1 \sqsubseteq_E S_2$, $DT_1 \subseteq DT_2$ und $QT_1 \subseteq QT_2$ gilt.

In der letzten Definition wurde wieder durch das Fluten eine Informationsvermischung vorgenommen. Im Fall von DT mit den Errortraces und im Fall von QT mit den Divergenztraces. Jedoch entstehen hier wie im letzten Kapitel keine neuen Traces, die nicht bereits in der error-gefluteten Sprache EL aus den Error-Kapitel enthalten wären. Somit kann diese Sprache ohne weitere Flutung verwendet werden. Es folgt, dass die Relation $\sqsubseteq_{Div}$ ebenso wie $\sqsubseteq_{Qui}$ eine Einschränkung der Relation $\sqsubseteq_E$ ist.
Ebenso wie in Satz 4.7 wird im nächsten Satz nur der Vollständigkeit halber der erste und letzte Punkt erwähnt der Beweis dazu findet sich in Satz 3.5.

Satz 5.6 (*Error-, Ruhe- und Divergenz-Semantik für Parallelkompostionen*). *Für zwei komponierbare EIOs S_1, S_2 und ihre Komposition S_{12} gilt:*

1. $ET_{12} = cont(prune((ET_1 \| EL_2) \cup (EL_1 \| ET_2)))$,
2. $DT_{12} = prune'((DT_1 \| EL_2) \cup (EL_1 \| DT_2)) \cup ET_{12}$,
3. $QT_{12} = (QT_1 \| QT_2) \cup DT_{12}$,

4. $EL_{12} = (EL_1 \| EL_2) \cup ET_{12}$.

Beweis. Es wird hier nur der 2. und 3. Punkt bewiesen.

2. „$\subseteq$“:
Es muss hier unterschieden, ob $w \in PrDT_{12} \backslash ET_{12}$ oder $w \in ET_{12}$ betrachtet wird. Im zweiten Fall ist das w in der rechten Seite der Gleichung enthalten. Deshalb wird im weiteren Verlauf dieses Beweises davon ausgegangen, dass $w \in PrDT_{12} \backslash ET_{12}$ gilt, und es wird versucht zu zeigen, dass dieses w ebenfalls in der rechten Seite enthalten ist. Aus der Definition 5.5 weiß man, dass ein $v \in O_{12}^*$ existiert, so dass $(q_{01}, q_{02}) \overset{w}{\Rightarrow} (q_1, q_2) \overset{v}{\Rightarrow} (q_1', q_2')$ mit $(q_1', q_2') \in Div_{12}$ gilt. Durch die Projektion auf die Transitionssysteme S_1 und S_2 erhält man $q_{01} \overset{w_1}{\Rightarrow} q_1 \overset{v_1}{\Rightarrow} q_1'$ und $q_{02} \overset{w_2}{\Rightarrow} q_2 \overset{v_2}{\Rightarrow} q_2'$ mit $w \in w_1 \| w_2$ und $v \in v_1 \| v_2$. Aus $(q_1', q_2') \in Div_{12}$ folgt, dass oBdA $q_1' \in Div_1$ gilt, d.h. $w_1 v_1 \in StDT_1$. Da $q_{02} \overset{w_2 v_2}{\Rightarrow}$ gilt, erhält man $w_2 v_2 \in L_2 \subseteq EL_2$ gelten. Somit gilt insgesamt $wv \in DT_1 \| EL_2$ und w ist in der rechten Seite der Gleichung enthalten, da $v \in O_{12}^*$ gilt und dieser Wortteil somit durch die $prune'$-Funktion entfernt werden kann.

2. „$\supseteq$“:
Falls $w \in ET_{12}$ gilt, ist dieses w auch in der linken Seite der Gleichung enthalten. Somit wird für den Rest des Beweises dieser Inklusion davon ausgegangen, dass $w \in prune'((DT_1 \| EL_2) \cup (EL_1 \| DT_2))$ gilt. Dadurch lässt sich ein v aus O_{12}^* finden, sodass $wv \in (DT_1 \| EL_2) \cup (EL_1 \| DT_2)$ gilt. Es wird nun noch die Einschränkung vorausgesetzt, dass oBdA $wv \in DT_1 \| EL_2$ gilt, d.h. es existieren $w_1 v_1 \in DT_1$ und $w_2 v_2 \in EL_2$ mit $w \in w_1 \| w_2$ und $v \in v_1 \| v_2$.

- Fall 1 ($w_1 v_1 \in ET_1 \vee w_2 v_2 \in ET_2$): OBdA gilt $w_1 v_1 \in ET_1$. Es kann sich nun bei $w_2 v_2$ um ein Errortrace, Divergenztrace, Ruhetrace oder um ein ausführbares Trace aus dem Transitionssystem handeln, es gilt in jedem Fall $w_2 v_2 \in EL_2$. Somit kann der erste Punkt von Satz 3.5 angewendet werden und es folgt damit, dass $w \in ET_{12}$ gilt und w deshalb in der linken Seite der Gleichung enthalten ist.
- Fall 2 ($w_1 v_1 \in PrDT_1 \backslash ET_1 \wedge w_2 \in EL_2 \backslash ET_2$): Es gilt in diesem Fall, dass ein $u \in O_1^* \subseteq O_{12}^*$ existiert mit $q_{01} \overset{w_1}{\Rightarrow} q_1 \overset{v_1}{\Rightarrow} q_1' \overset{u}{\Rightarrow} q_1'' \in Div_1$ und $q_{02} \overset{w_2}{\Rightarrow} q_2 \overset{v_2}{\Rightarrow} q_2'$. An dieser Stelle kann nichts darüber gesagt werden, wie sich S_2 verhält, wenn S_1 das u ausführt.
  - Fall 2a) ($q_2' \overset{u|_{\Sigma_2}}{\not\Rightarrow}$): S_2 hat also nicht die Möglichkeit das u gemeinsam mit S_1 auszuführen, es kommt also zu einem Kommunikationsfehler in der Parallelkomposition. Es existiert ein Präfix t von $u|_{\Sigma_2}$ mit $w_2 v_2 t \in MIT_2$. Somit folgt insgesamt, da v und u in O_{12}^* enthalten sind, dass $w \in ET_{12}$ gilt. Das w ist also in der linken Seite der Gleichung enthalten.
  - Fall 2b) ($q_2' \overset{u|_{\Sigma_2}}{\Rightarrow} q_2''$): Da q_1'' ein unendliche Folge an τs ausführen kann, ist dies auch für den zusammengesetzten Zustand von q_1'' und q_2'' in der Parallelkomposition möglich. Es gilt also für die Komposition $(q_{01}, q_{02}) \overset{w}{\Rightarrow} (q_1, q_2) \overset{v}{\Rightarrow} (q_1', q_2') \overset{o}{\Rightarrow} (q_1'', q_2'') \in Div_{12}$ und somit ist wvu in $StDT_{12}$ enthalten. Durch die

parallel Ausführung von u und $u|_{\Sigma_2}$ entstehen nur Outputs und es gilt auch $u \in u \| u|_{\Sigma_2}$. Es folgt also, dass $w \in PrDT_{12}$ gilt und somit die Zugehörigkeit zur linken Seite erfüllt ist.

3. „$\subseteq$“:
Diese Inklusionsrichtung kann analog zum Beweis der selben Inklusionsrichtung von Satz 4.7 gezeigt werden. Es muss dabei nur in der Argumentation die Menge ET_{12} durch die Menge DT_{12} ersetzt werden. Dadurch kann ebenso gefolgert werden, dass der erreichte Zustand (q_1, q_2) kein Error-Zustand sein kann, da $ET_{12} \subseteq DT_{12}$ gilt.

3. „$\supseteq$“:
Es muss wieder danach unterschieden werden, aus welcher Menge das betrachtete Element stammt. Falls w ein Element von DT_{12} ist, so folgt die Zugehörigkeit zur linken Seite der Gleichung direkt. Somit wird für den weiteren Punkt dieses Beweises davon ausgegangen, dass $w \in QT_1 \| QT_2$ gilt. Für dieses w soll dann gezeigt werden, dass es auch in QT_{12} enthalten ist. Da $QT_i = StQT_i \cup DT_i$ gilt, existieren für w_1 und w_2 mit $w \in w_1 \| w_2$ unterschiedliche Möglichkeiten:

- Fall 1 ($w_1 \in DT_1 \vee w_2 \in DT_2$): OBdA gilt $w_1 \in DT_1$. Es kann nun $w_2 \in StQT_2 \subseteq L_2$ gelten oder $w_2 \in DT_2$ und somit gilt auf jeden Fall $w_2 \in EL_2$. Daraus kann mit dem zweiten Punkt dieses Satzes gefolgert werden, dass $w \in DT_{12}$ gilt und somit w in der linken Seite der Gleichung enthalten ist.
- Fall 2 ($w_1 \in StQT_1 \backslash DT_1 \wedge w_2 \in StQT_2 \backslash DT_2$): Dieser Fall läuft analog zu Fall 2 der selben Inklusionsrichtung des Beweises von Satz 4.7.

□

Analog wie in den beiden vorgegangenen Kapiteln, ergibt sich aus diesem Satz als direkte Folgerung, dass es sich bei der Relation $\sqsubseteq_{Div}$ um eine Präkongruenz handelt.

Proposition 5.7 (*Divergenz-Präkongurenz*). *$\sqsubseteq_{Div}$ ist eine Präkongruenz bezüglich $\cdot\|\cdot$.*

Beweis. Um zu zeigen, dass es sich bei $\sqsubseteq_{Div}$ um eine Präkongruenz handelt, muss nachgewiesen werden, dass $S_{31} \sqsubseteq_{Div} S_{32}$ für jedes komponierbare System S_3 gilt, wenn $S_1 \sqsubseteq_{Div} S_2$ erfüllt ist. D. h. es ist zu zeigen, dass auch $S_1 \sqsubseteq_E S_2$, $DT_1 \subseteq DT_2$ und $QT_1 \subseteq QT_2$ sowohl $S_{31} \sqsubseteq_E S_{32}$, $DT_{31} \subseteq DT_{32}$ als auch $QT_{31} \subseteq QT_{32}$ folgt. Dies ergibt sich wie in den Beweisen zu den Propositionen 3.6 und 4.8 aus der Monotonie von $\cdot\|\cdot$ auf Sprachen wie folgt:

- $S_{31} \overset{\substack{\text{Proposition 3.6}\\ \text{und}\\ S_1 \sqsubseteq_E S_2}}{\sqsubseteq_E} S_{32}$

- $DT_{31} \overset{5.6\ 2.}{=} prune'((DT_3 \| EL_1) \cup (EL_3 \| DT_1)) \cup ET_{31}$
  $\overset{\substack{ET_{31} \subseteq ET_{32}, \\ EL_1 \subseteq EL_2 \\ \text{und} \\ DT_1 \subseteq DT_2}}{\subseteq} prune'((DT_3 \| EL_2) \cup (EL_3 \| DT_2)) \cup ET_{32}$
  $\overset{5.6\ 2.}{=} DT_{32},$
- $QT_{31} \overset{5.6\ 3.}{=} (QT_3 \| QT_1) \cup DT_{31}$
  $\overset{\substack{DT_{31} \subseteq DT_{32}, \\ \text{und} \\ QT_1 \subseteq QT_2}}{\subseteq} (QT_3 \| QT_2) \cup DT_{32}$
  $\overset{5.6\ 3.}{=} QT_{32}.$

□

Als nächstes soll nun eine Verfeinerungsrelation bezüglich guter Kommunikation mit Partnern im Sinne von error-, ruhe- und divergenz-freier Kommunikation betrachtet werden. Es muss in diesem Lemma eine Veränderung zu den analogen Lemmata aus den vorangegangen Kapiteln vorgenommen. Die Einschränkung, dass U ein Partner sein muss, kann nicht mehr beibehalten werden, da die Vermeidung von Ruhe im Beweis aus dem letzten Kapitel hier zu Divergenz führen würde. Somit werden dafür Aktionen außerhalb der Menge *Synch* benötigt.

Lemma 5.8 (*Vereinerung mit Divergenz-Zuständen*). *Gegeben sind zwei EIOs S_1 und S_2 mit der gleichen Signatur. Wenn $U \| S_1 \sqsubseteq^B_{Div} U \| S_2$ für komponierbaren U gilt, dann folgt daraus $S_1 \sqsubseteq_{Div} S_2$.*

Beweis. $ET_1 \subseteq ET_2$ kann nicht garantiert werden, da divergenz in S_2 nicht verhindert werden kann durch ein U. □

Der folgenden Satz hält fest, dass $\sqsubseteq_{Div}$ die gröbste Präkongruenz bezüglich $\cdot\|\cdot$ charakterisiert, die in $\sqsubseteq^B_{Div}$ enthalten ist.

Satz 5.9 (*Full Abstractness für Divergenz-Semantik*). *Seinen S_1 und S_2 zwei EIOs mit derselben Signatur. Dann gilt $S_1 \sqsubseteq^C_{Div} S_2 \Leftrightarrow S_1 \sqsubseteq_{Div} S_2$, insbesondere ist $\sqsubseteq_{Div}$ eine Präkongruenz.*

Beweis. „⇐“: Nach Definition gilt $w \in DT(S)$ mit $w \in O^*$, genau dann wenn in S ein Divergenz-Zustand, ein Ruhe-Zustand oder ein Error-Zustand lokal erreichbar ist. $S_1 \sqsubseteq_{Div} S_2$ impliziert, dass $w \in DT_2$ gilt, wenn $w \in DT_1$ gilt. Somit ist ein Divergenz-Zustand, ein Ruhe-Zustand oder ein Error-Zustand nur dann in S_1 lokal erreichbar, wenn auch ein solcher in S_2 lokal erreichbar ist. Daraus folgt, dass $S_1 \sqsubseteq^B_{Div} S_2$ gilt. Somit ist $\sqsubseteq_{Div}$ in $\sqsubseteq^B_{Div}$ enthalten. In Proposition 5.7 wurde festgestellt, dass $\sqsubseteq_{Div}$ eine Präkongruenz ist. Da jedoch $\sqsubseteq^C_{Div}$ nach Definition die gröbste Präkongruenz bezüglich $\cdot\|\cdot$ ist, die in $\sqsubseteq^B_{Div}$ enthalten ist, muss $\sqsubseteq_{Div}$ in $\sqsubseteq^C_{Div}$ enthalten sein. Es folgt also aus $S_1 \sqsubseteq_{Div} S_2$, dass auch der relationale Zusammenhang $S_1 \sqsubseteq^C_{Div} S_2$ gilt.

„⇒“: Durch die Definition von $\sqsubseteq^C_{Div}$ als Präkongruenz in 5.3 folgt aus $S_1 \sqsubseteq^C_{Div} S_2$, dass $U \| S_1 \sqsubseteq^C_{Div} U \| S_2$ für alle EIOs U gilt, die mit S_1 komponierbar sind. Somit folgt auch die Gültigkeit von $U \| S_1 \sqsubseteq^B_{Div} U \| S_2$ für alle diese EIOs U. Mit Lemma 5.8 folgt dann $S_1 \sqsubseteq_{Qui} S_2$. □

Es wurde somit, wie in den letzten beiden Kapiteln, eine Kette an Folgerungen gezeigt, die sich zu einem Ring schließen. Jedoch wird dafür ein Schritt weniger benötigt, da in Lemma 5.8 bereits direkt U nur als komponierbar und nicht als Partner vorausgesetzt wurde. Diese Folgerungskette ist in Abbildung 5.1 dargestellt.

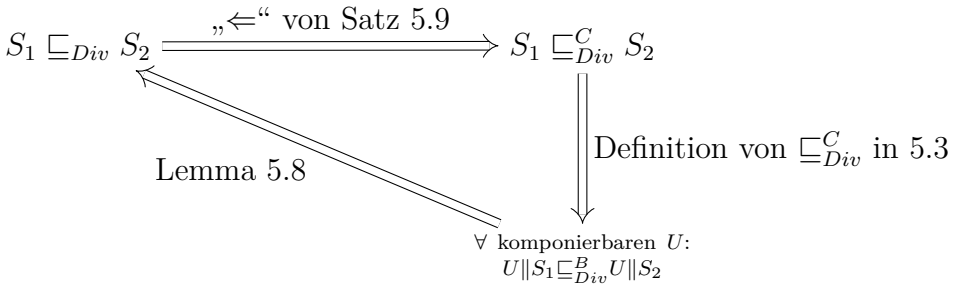

Abbildung 5.1: Folgerungskette

Aus Satz 5.9 und Lemma 5.8 erhält man das folgende Korollar. Angenommen man definiert, dass S_1 S_2 verfeinern soll, genau dann wenn für alle Partner EIOs U, für die S_2 error-, ruhe- und divergenz-frei mit U kommuniziert, folgt, dass S_1 ebenfalls error-, ruhe- und divergenz-frei mit U kommuniziert. Dann wird auch diese Verfeinerung durch $\sqsubseteq_{Div}$ charakterisiert.

Korollar 5.10. *Es gilt:* $S_1 \sqsubseteq_{Div} S_2 \Leftrightarrow U \| S_1 \sqsubseteq^B_{Div} U \| S_2$ *für alle komponierbaren* U.

5.2 Hiding und Divergenz-Freiheit

Da durch den Internalisierungsoperator Outputs in τs umgewandelt werden, hat das Hiding auf die Divergenz-Eigenschaft eine recht große Auswirkung. Die Menge der divergenten Zustände kann sich somit durch das Internalisieren vergrößern. Es kann ein Zustand divergent werden, wenn von diesem aus bereits lokal ein divergenter Zustand erreichbar war oder wenn er eine unendliche Folge von Aktionen aus $X \cup \{\tau\}$ ausführen konnte, jedoch nur endlich viele davon τs waren. Somit kann die Basisrelation für Divergenz auf jeden Fall nicht erhalten bleiben unter Internalisierung.

Literaturverzeichnis

[AH04] Luca De Alfaro und Thomas A. Henzinger, *Interface-based design*, In Engineering Theories of Software Intensive Systems, Kluwer, 2004.

[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.

[Lyn96] Nancy A. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[Mil89] Robin Milner, *Communication and concurrency*, PHI Series in computer science, Prentice Hall, 1989.

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