Universität Augsburg
Fakultät für Angewandte
Informatik

Bachelorarbeit
im Studiengang Bachelor Informatik

Kommunikationsfehler, Verklemmung und Divergenz bei Interface-Automaten

Universität Augsburg
Fakultät für Angewandte Informatik
Theorie verteilter Systeme

Aufgabensteller: Prof. Dr. Walter Vogler

eingereicht von: Ayleen Schinko
eingereicht am: 24. Februar 2016

Inhaltsverzeichnis

1 Einleitung

Interface-Automaten werden im Allgemeinen dafür eingesetzt, um Verhalten vom System oder Teilen davon zu modellieren und zu untersuchen. Sie beschreiben abstrakt des Kommunikationsverhaltens eines Systems oder einer Komponente durch Inputs und Outputs. Es können durch Interface-Automaten parallel arbeitende Softwarekomponenten simuliert werden. Die Kommunikation zwischen Systemen wird durch eine Parallelkomposition der beteiligten Komponenten modelliert. Grundsätzlich wird hier ein Output bzw. eine interne Aktion als vom jeweiligen System kontrolliert angesehen und ein Input als von der Umwelt kontrolliert. Falls in einer Parallelkomposition ein Output von einem System ausgeführt wird, muss dieser von allen anderen beteiligten System, die diese Aktion in ihrer Inputmenge haben, durch einen Input synchronisiert werden. Falls dies nicht möglich ist, tritt ein Kommunikationsfehler ein, der verhindert werden sollte, da das Verhalten des Systems dann unvorhersehbar wird. In [Lyn96] wird die Abwesenheit dieser Kommunikationsfehler für die darin betrachteten I/O-Automaten garantiert, in dem für alle Zustände alle Inputs möglich sind. In Interface-Automaten ist ein fehlender Input jedoch durch die Voraussetzung zu begründen, dass die Umwelt diesen Input in diesem Zustand nicht senden darf. Es ist also nicht sinnvoll, für die Vermeidung von Kommunikationsfehler in Interface-Automaten, alle Inputs für alle Zustände zuzulassen, da dann nicht mehr alle zulässigen Voraussetzungen erfüllt werden könnten.
Die Definition der Interface-Automaten in [DAH05] lässt in der Parallelkomposition von zwei dieser Automaten Kommunikationsfehlern zu, in dem eine Komponente einen Output macht, den die andere nicht als Input aufnehmen kann. Die daraus entstehen Fehler-Zustände, sind jedoch nicht zulässig und müssen inklusive aller Zustände, durch die man mit lokalen Aktionen einen dieser Fehler-Zustände erreichen kann, mit einer speziell definieren Operation gestutzt werden.
Um dem Problem des aufwendigen Stutzens entgegenzuwirken, soll hier eine andere Modellierung betrachtet werden. Die Komponenten sollen als beschriftetes Transitionssystem (LTS) mit disjunkten Input- und Output-Aktionen und einer internen Aktion τ modelliert werden. Diese Modelle werden dann Error-IO-Transitionssysteme (EIOs) genannt. Für die Systeme sollen explizite Error-Zustände möglich sein. Durch die Parallelkomposition können dann wie oben beschrieben neue Error-Zustände entstehen und es werden zusätzlich auch noch die Error-Zustände der einzelnen Komponenten geerbt. Darüber sind dann Verfeinerungsrelationen möglich, diese sollen die Voraussetzung erfüllen, dass eine fehlerfrei Spezifikation nur durch ein fehlerfreies System verfeinert werden kann. Dies wird hier als Basisrelation verstanden, die durch die explizite Definition der jeweiligen Fehler-Art parametrisiert ist. Die Verfeinerungsrelation kann auch als Implementierung aufgefasst werden.

Da Modularität bzw. Austauschbarkeit der Komponenten gewünscht ist, sollte die Verfeinerungsrelationen eine Präkongruenz sein. Falls eine Komponente einer Parallelkomposition durch eine Verfeinerung ersetzt wird, soll die Komposition selbst auch verfeinert werden. Da die Basisrelation keine Präkongruenz ist, wird eine größte Präkongruenz bezüglich der Parallelkomposition charakterisiert, die in der Basisrelation enthalten ist.
In [BV15] wurde für Kommunikationsfehler-Freiheit nachgewiesen, dass jedes EIO äquivalent ist zu einem ohne Error-Zustände unter Berücksichtigung der Präkongruenz. Dieses erhält man durch Stutzen wie in [DAH05]. Damit ist es also möglich die EIOs nur für die Untersuchung zu verwenden und die Fehler-Zustände am Ende zu entfernen, um wieder Interface-Automaten zu erhalten.
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 zulässigen Zustand führt. Jedoch muss dann vor der Umwandlung in einen Interface-Automaten durch das Entfernen der Fehler-Zustände auf diese Inputs geachtet werden.
Um die Begrifflichkeiten hier eindeutiger und intuitiver 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 [BV15]. Jedoch wird hier darauf verzichtet die Input-Mengen der EIOs als disjunkt anzunehmen und alle Definitionen und Sätze werden zunächst ohne das Verbergen der synchronisierten Aktionen betrachtet. Da die Input-Mengen im Gegensatz zu [BV15] einen nicht leeren Schnitt haben können, ergibt sich in natürlicherweise die Voraussetzung, dass die Komposition aus zwei Systemen einen Input aus diesem Schnitt nur ausführen kann, wenn beide Systeme die Möglichkeit für diesen Input haben.
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 auch, da bei jeder Komposition der Output an andere Systeme weitergeleitet wird. 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. Es wird also die Kommunikation durch diese Outputs mit anderen Systemen verboten, bzw. die Kommunikationskanäle werden geschlossen, da die Aktionen internalisiert werden.
Diese Art der Betrachtung der EIOs wurde auch bereits in [Sch12] gewählt, jedoch dient diese Arbeit nur im Abschnitt des Hidings in Kapitel 3 als direkte Quelle. 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 Zustände 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 lokalen 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 [BV15] für Error-Zustände behandelt.
Allgemein werden hier nur optimistische Ansätze betrachtet, d.h. es wird davon ausgegangen, dass ein Fehler kein Problem ist, solang er in einer hilfreichen Umgebung nicht erreicht wird. Jedoch ist auch die pessimistische Ansicht weit verbreitet, dass ein System mit einem Fehler niemals zulässig sein kann. Da man nicht immer von einer fehlerfreien Spezifikation ausgehen kann, scheint der optimistische Ansatz näher an der Realität zu liegen.
Neben dem hier betrachteten optimistischen Ansatz gibt es noch zwei weitere Ansätze in [BV15] 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 den bereits oben erwähnten pessimistischen Ansatz.
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 und 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. Somit wurden die Ruhe-Zustände als Art Deadlocks umgesetzt, jedoch falls man die τs nicht komplett verbieten würde, könnten auch Livelocks damit dargestellt werden.
Die Hinzunahme unendlicher Traces führt zu einer anderen Betrachtungsweise, mit der die Eigenschaft der Divergenz genauer betrachtet werden sollen. Hierbei kann ein System unendlich viele τ-Transitionen ausführen. Es kann dann auch relevant sein, dass die hier betrachteten Transitionssysteme nicht endlich sein müssen.

Kapitel 2 erläutert die grundlegenden Definitionen. In Kapitel 3 bis 5 werden nacheinander die Fehler-Mengen, die betrachtet werden, immer weiter vergrößert. Zuerst werden in Kapitel 3 nur Errors betrachtet, in Kapitel 4 auch noch zusätzlich Ruhe. Im letzten Kapitel werden dann Error, Ruhe und Divergenz betrachtet.

2 Definitionen und Notationen

Die Definitionen dieses Kapitels basieren größtenteils auf dem Paper von Bujtor und Vogler [BV15], mit den in der Einleitung bereits erwähnten Modifikationen. In diesen Definitionen werden die Grundlagen der Transitionssysteme, mit denen im folgenden gearbeitet werden soll, beschrieben.

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 [BV15] entsteht das τ meist 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 $\mathrm{Sig}(S) = (I, O)$.

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 $\exists q \in Q : (p, \alpha, q) \in \delta$,
- $p \xrightarrow{w} q$ 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* eines EIOs S ist $L(S) = \left\{w \in \Sigma^* \mid q_0 \overset{w}{\Rightarrow}\right\}$.

2.2 Parallelkomposition

Zwei EIOs sind komponierbar, wenn ihre Output-Mengen disjunkt sind. Die Error-Zustände der Parallelkomposition setzen sich aus den Error-Zuständen 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 [BV15] an der Menge der synchronisierten Aktionen vorgenommen werden. Da nicht mehr $I_1 \cap I_2 = \emptyset$ gelten muss, werden die gemeinsamen Inputs der Systeme synchronisiert. Somit handelt es sich in der Parallelkomposition bei synchronisierten Aktionen nicht mehr nur um Outputs, wie in [BV15], 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 nicht um einen neuen Error, da es zwischen den beiden Systemen dadurch nicht zu einem Kommunikationsfehler 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 *der EIOs S_1 und S_2 ist $S_{12} := S_1 \| S_2 = (Q, I, O, \delta, q_0, E)$ mit den Komponenten:*

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

- $q_0 = (q_{01}, q_{02})$,

- $\delta = \{((q_1, q_2), \alpha, (p_1, q_2)) \mid (q_1, \alpha, p_1) \in \delta_1, \alpha \in (\Sigma_1 \cup \{\tau\}) \backslash \text{Synch}(S_1, S_2)\}$
  $\cup \{((q_1, q_2), \alpha, (q_1, p_2)) \mid (q_2, \alpha, p_2) \in \delta_2, \alpha \in (\Sigma_2 \cup \{\tau\}) \backslash \text{Synch}(S_1, S_2)\}$
  $\cup \{((q_1, q_2), \alpha, (p_1, p_2)) \mid (q_1, \alpha, p_1) \in \delta_1, (q_2, \alpha, p_2) \in \delta_2, \alpha \in \text{Synch}(S_1, S_2)\}$,

- $E = (Q_1 \times E_2) \cup (E_1 \times Q_2)$ geerbte Errors
  $\cup \left\{(q_1, q_2) \mid \exists a \in O_1 \cap I_2 : q_1 \xrightarrow{a} \wedge q_2 \overset{a}{\not\to}\right\}$
  $\cup \left\{(q_1, q_2) \mid \exists a \in I_1 \cap O_2 : q_1 \overset{a}{\not\to} \wedge q_2 \xrightarrow{a}\right\}$ neue Errors.

Dabei werden die synchronisierten Aktionen $\text{Synch}(S_1, S_2) = (I_1 \cap O_2) \cup (O_1 \cap I_2) \cup (I_1 \cap I_2)$ *nicht versteckt, sondern als Outputs bzw. im Fall von* $I_1 \cap I_2$ *als Inputs der Komposition beibehalten.*
Die oben definierte Notation $S_{12} = S_1 \| S_2$ *wird auch für andere Indizierungen der Systeme analog angewendet, so gilt also allgemein* $S_{ij} := S_i \| S_j$ *für* $i, j \in \mathbb{N}$.
S_1 *wird* Partner *von* S_2 *genannt, wenn die Parallelkomposition von* S_1 *und* S_2 *geschlossen ist, d.h. wenn sie duale Signaturen* $\text{Sig}(S_1) = (I, O)$ *und* $\text{Sig}(S_2) = (O, I)$ *haben.*

Die Parallelkomposition kann nicht nur für Transitionssysteme betrachtet werden, wie bisher in dieser Arbeit, sondern auch über Aktionsfolgen. Diese Aktionsfolgen werden *Traces* genannt. 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 *gilt:*

- *Für zwei Wörter* $w_1 \in \Sigma_1$ *und* $w_2 \in \Sigma_2$ *ist deren Parallelkomposition definiert als:* $w_1 \| w_2 := \{w \in (\Sigma_1 \cup \Sigma_2)^* \mid w|_{\Sigma_1} = w_1 \wedge w|_{\Sigma_2} = w_2\}$.

- *Für zwei Mengen von Wörtern bzw. Sprachen* $W_1 \subseteq \Sigma_1^*$ *und* $W_2 \subseteq \Sigma_2^*$ *ist deren Parallelkomposition definiert als:* $W_1 \| W_2 := \bigcup \{w_1 \| w_2 \mid w_1 \in W_1 \wedge w_2 \in W_2\}$.

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.

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

- prune : $\Sigma^* \to \Sigma^*, w \mapsto u$, *mit* $w = uv, u = \varepsilon$[1] $\vee\, u \in \Sigma^* \cdot I$ *und* $v \in O^*$,
- cont : $\Sigma^* \to \mathfrak{P}(\Sigma^*)$[2], $w \mapsto \{wu \mid u \in \Sigma^*\}$,
- cont : $\mathfrak{P}(\Sigma^*) \to \mathfrak{P}(\Sigma^*), L \mapsto \bigcup \{\text{cont}(w) \mid w \in L\}$.

Für zwei komponierbare EIOs S_1 und S_2 ist ein Ablauf ihrer Parallelkomposition S_{12} eine Transitionsfolge der Form $(p_1, p_2) \overset{w}{\Rightarrow} (q_1, q_2)$ für ein $w \in \Sigma_{12}^*$. So ein Ablauf kann auf Abläufe von S_1 und S_2 projiziert werden. Diese Projektionen erfüllen $p_i \overset{w_i}{\Rightarrow} q_i$ mit $w|_{\Sigma_i} = w_i$ für $i = 1, 2$. Umgekehrt sind zwei Abläufe von S_1 und S_2 der Form $p_i \overset{w_i}{\Rightarrow} q_i$ mit $w|_{\Sigma_i} = w_i$ für $i = 1, 2$, Projektionen von einem Ablauf in S_{12} der Form $(p_1, p_2) \overset{w}{\Rightarrow} (q_1.q_2)$. Es ist dafür nötig, dass die Abläufe der beiden Systeme S_1, S_2 und die Systeme selbst komponierbar sind. Das w wurde so gewählt, dass die Projektion auf die einzelnen Alphabete Σ_1 und Σ_2 die jeweiligen Wörter w_1 und w_2 ergibt. Falls keine internen 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. Aus diesen Feststellungen 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 [CJK14] 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, wie hier. Das Hiding wird durch einen Internalisierungsoperator umgesetzt. Es sollen dadurch Aktionen versteckt werden können, d.h. durch τs ersetzt werden. In [CJK14] 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 verboten, ähnlich wie bei der Anwendung von Restriktionen in CSS, siehe dazu [Mil89]. 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].

[1] ε bezeichnet das leere Wort

[2] $\mathfrak{P}(M)$ bezeichnet die Potenzmenge der Menge M

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* $(Q, I, O', \delta', q_0, E)$ *mit:*

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

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 [BV15] werden bei der Parallelkomposition die synchronisierten Aktionen, welche durch die Synchronisation von einem Input mit einem Output entstehen, verborgen. Diese Parallelkomposition wird nun mit dem Internalisierungsoperator durch Hiding der synchronisierten Aktionen, die in der Parallelkomposition zu Outputs werden, nachbildet. 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 [BV15] nachgebildet werden kann.

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

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 [BV15]. Ebenfalls wurde die Parallelkomposition geändert, wie in [Sch12].

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

Mittels der lokalen Erreichbarkeit von Error-Zuständen kann eine 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 $S_1 \sqsubseteq_E^{\mathrm{B}} S_2$ geschrieben, wenn ein Error-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 *da.*
$\sqsubseteq_E^{\mathrm{C}}$ bezeichnet die vollständig abstrakte Präkongruenz *von $\sqsubseteq_E^{\mathrm{B}}$ bezüglich $\cdot\|\cdot$, d.h. die gröbste Präkongruenz bezüglich $\cdot\|\cdot$, die in $\sqsubseteq_E^{\mathrm{B}}$ enthalten ist.*

Um sich näher mit den Präkongruenzen auseinandersetzen zu können, müssen bestimmte Traces aus der Struktur hervorgehoben werden. Die strikten Errortraces entsprechen Wegen, die direkt vom Startzustand zu einem Zustand in der Menge E führen. Da Outputs und τs 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-Zustand 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-Zustand, 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*:* $StET(S) = \left\{w \in \Sigma^* \mid q_0 \overset{w}{\Rightarrow} q \in E\right\}$,
- gekürzte Errortraces*:* $PrET(S) = \{\text{prune}(w) \mid w \in StET(S)\}$,
- Input-kritische Traces*:* $MIT(S) = \left\{wa \in \Sigma^* \mid q_0 \overset{w}{\Rightarrow} q \wedge a \in I \wedge q \overset{a}{\not\rightarrow}\right\}$.

In der folgenden Definition wird festgehalten, was als Errortrace aufgefasst wird. Diese Menge ist dadurch, dass sie die fortgesetzten Traces aus $PrET$ enthält, deutlich allgemeiner als die Menge $StET$. Die fortgesetzten Traces aus MIT sind aufgrund der Möglichkeit für Kommunikationsfehler ebenfalls in den Errortraces enthalten. 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 unterschieden werden.

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

- *Die Menge der* Errortraces *von S ist* $ET(S) := \text{cont}(PrET(S)) \cup \text{cont}(MIT(S))$.
- *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 [BV15] für die Parallelkomposition mit verborgenen synchronisierten Aktionen formuliert, jedoch entspricht, der hier aufgeführte Satz für die Parallelkomposition ohne verbergen der synchronisierten Aktionen, dem analogen Satz aus [Sch12]. Da der Beweis ohne Beachtung von [Sch12] neu geführt wurde, wird hier eher auf die Erwähnung der Unterschiede zu [BV15] Wert gelegt.

Satz 3.5 (*Error-Semanik für Parallelkompositionen*). *Für zwei komponierbare EIOs S_1, S_2 und ihre Komposition S_{12}, gilt:*

1. $ET_{12} = \text{cont}\left(\text{prune}\left((ET_1 \| EL_2) \cup (EL_1 \| ET_2)\right)\right)$,
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 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 $(q_{01}, q_{02}) \overset{x}{\Rightarrow} (q_1, q_2) \wedge a \in I_{12} \wedge (q_1, q_2) \overset{a}{\not\rightarrow}$. Da $I_{12} \overset{2.2}{=} (I_1 \backslash O_2) \cup (I_2 \backslash O_1) = (I_1 \cup I_2) \backslash (O_1 \cup O_2)$ ist, folgt $a \in (I_1 \cup I_2)$ und $a \notin (O_1 \cup O_2)$. Es wird unterschieden, 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 [BV15] nicht nötig, da dort $I_1 \cap I_2 = \emptyset$ gilt, somit gibt es dort nur den Fall 1b).

  - Fall 1a) ($a \in (I_1 \cap I_2)$): Der Ablauf der Komposition kann nun 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 $x_1 a \in \text{cont}(MIT_1) \subseteq ET_1$ 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: $q_{01} \stackrel{x_1}{\Rightarrow} q_1 \stackrel{a}{\not\rightarrow}$ und $q_{02} \stackrel{x_2}{\Rightarrow} q_2$ mit $x \in x_1 \| x_2$. Daraus folgt $x_1 a \in \text{cont}(MIT_1) \subseteq ET_1$ 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 ($w \in PrET_{12}$): Aus den Definitionen von $PrET$ und prune folgt, dass es ein $v \in O_{12}^*$ gibt, so dass $(q_{01}, q_{02}) \stackrel{w}{\Rightarrow} (q_1, q_2) \stackrel{v}{\Rightarrow} (q_1', q_2')$ gilt mit $(q_1', q_2') \in E_{12}$ und $w = \text{prune}(wv)$. Durch Projektion erhält man $q_{01} \stackrel{w_1}{\Rightarrow} q_1 \stackrel{v_1}{\Rightarrow} q_1'$ und $q_{02} \stackrel{w_2}{\Rightarrow} q_2 \stackrel{v_2}{\Rightarrow} q_2'$ mit $w \in w_1 \| w_2$ und $v \in v_1 \| v_2$. Aus $(q_1', q_2') \in E_{12}$ folgt, dass es sich entweder um einen geerbten oder einen neuen Error handelt. Bei einem geerbten wäre bereits einer der beiden Zustände q_1 bzw. q_2 ein Error-Zustand gewesen. Ein 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 $w_1 v_1 \in StET_1 \subseteq \text{cont}(PrET_1) \subseteq ET_1$. Da $q_{02} \stackrel{w_2 v_2}{\Longrightarrow}$ gilt, erhält man $w_2 v_2 \in L_2 \subseteq EL_2$. Dadurch ergibt sich $wv \in ET_1 \| EL_2$ mit $w = \text{prune}(wv)$ 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 $q_1' \stackrel{a}{\not\rightarrow} \wedge\ q_2' \stackrel{a}{\rightarrow}$. Daraus folgt $w_1 v_1 a \in MIT_1 \subseteq ET_1$ und $w_2 v_2 a \in L_2 \subseteq EL_2$. Damit ergibt sich $wva \in ET_1 \| EL_2$, da $a \in O_2 \subseteq O_{12}$ gilt $w = \text{prune}(wva)$ 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 $x \in \text{prune}\left((ET_1 \| EL_2) \cup (EL_1 \| ET_2)\right)$ 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 $v = \varepsilon$. Damit 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 ist ein Präfix von x in $PrET_{12} \cup MIT_{12}$ enthalten und somit gilt $x \in ET_{12}$, da ET die Fortsetzung der Mengenvereinigung aus $PrET$ und MIT ist.
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 kann nun davon ausgegangen werden, 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-Zustand 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 [BV15] $b \neq a$ 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 [BV15] möglich war, sondern nur wenn $a \notin I_2$ gilt.

- Fall I ($v_1 \in MIT_1$ und $v_1 \neq \varepsilon$): Es gibt einen Ablauf der Form $q_{01} \overset{v_1'}{\Rightarrow} q_1 \overset{a}{\not\rightarrow}$ und es gilt $v' = v''a$. Bei der folgenden Fallunterscheidung müssen im Gegensatz zu [BV15] 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 ebenfalls $v := v''a = v' \in MIT_{12}$ 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 [BV15] erweitert wurde

liegt a in Synch(S_1, S_2), also folgt $(q_1, q_2) \overset{a}{\not\to}$ schon aus $q_1 \overset{a}{\not\to}$. Somit kann hier nochmals $v := v''a = v' \in MIT_{12}$ 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 $q_{02} \overset{v_2''}{\Rightarrow} q_2 \overset{a}{\to}$ mit $v'' \in v_1' \| v_2''$. Daraus ergibt sich $(q_{01}, q_{02}) \overset{v''}{\Rightarrow} (q_1, q_2)$ mit $q_1 \overset{a}{\not\to}, a \in I_1, q_2 \overset{a}{\to}$ und $a \in O_2$, somit gilt $(q_1, q_2) \in E_{12}$. Es wird $v := \text{prune}(v'') \in PrET_{12}$ 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 [BV15] 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, wodurch $v_2 = v_2'$ gilt. Dieser Fall verläuft jedoch analog zu Fall Ic) und wird hier nicht weiter betrachtet. Es gilt für alle anderen Fälle $q_{02} \overset{v_2'}{\Rightarrow} q_2$ mit $(q_{01}, q_{02}) \overset{v'}{\Rightarrow} (q_1, q_2)$.

  - Fall IIa) $\Big(u_2 \in (O_1 \cap I_2)^*, c \in (O_1 \cap I_2)$, sodass u_2c Präfix von $u_1|_{I_2}$ mit $q_2 \overset{u_2}{\Rightarrow} q_2' \overset{c}{\not\to}\Big)$: Für das Präfix $u_1'c$ von u_1 mit $u_1'c|_{I_2} = u_2c$ weiß man, dass $q_1 \overset{u_1'}{\Rightarrow} q_1'' \overset{c}{\to}$. 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 $v := \text{prune}(v'u_1') \in PrET_{12}$ gewählt, dies ist ein Präfix von v', da $u_1 \in O_1^*$.

  - Fall IIb) $\Big(q_2 \overset{u_2}{\Rightarrow} q_2'$ mit $u_2 = u_1|_{I_2}\Big)$: Es 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 $q_1' \in E_1$ und somit handelt es sich in S_{12} um einen geerbten Error. Nun wird $v := \text{prune}(v'u_1) \in PrET_{12}$ gewählt, das wiederum ein Präfix von v' ist.

2.:
Der Beweis für diesen Punkt konnte bezüglich [BV15] fast unverändert übernommen werden.
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:

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

□

Das folgende Korollar wurde hier noch explizit mit Beweis eingefügt im Gegensatz zu den Ausführungen in [BV15], 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 $\sqsubseteq_E^{\mathrm{C}}$ zu erhalten.

Korollar 3.6 (*Error-Präkongruenz*). *Die Relation $\sqsubseteq_E$ 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:

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

□

In [BV15] wurde auch die Verfeinerung von EIOs als Relation mit Spezifikation und Implementierung betrachtet. Hier soll ebenfalls eine Verfeinerungsrelation über EIOs betrachtet werden, jedoch sollen die synchronisierten Aktionen nicht verborgen werden. Dadurch ändern sich natürlich auch Teile des folgenden 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 des folgenden 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 $U\|S_1 \sqsubseteq_E^{\mathrm{B}} U\|S_2$ 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 die beide Mengen ET_1 und ET_2 durch cont abgeschlossen sind.

- Fall 1 ($w = \varepsilon$): Es handelt sich um einen lokal erreichbaren Error-Zustand 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 im Prinzip 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 $x_{n+1} \in I = O_U$): Es wird der folgende Partner U bedachtet (siehe auch Abbildung 3.1):
  - $Q_U = \{q_0, q_1, \dots, q_{n+1}\}$,
  - $q_{0U} = q_0$,
  - $E_U = \emptyset$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i \leq n\}$
    $\cup \{(q_i, x, q_{n+1}) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i \leq n\}$
    $\cup \{(q_{n+1}, x, q_{n+1}) \mid x \in I_U\}$.

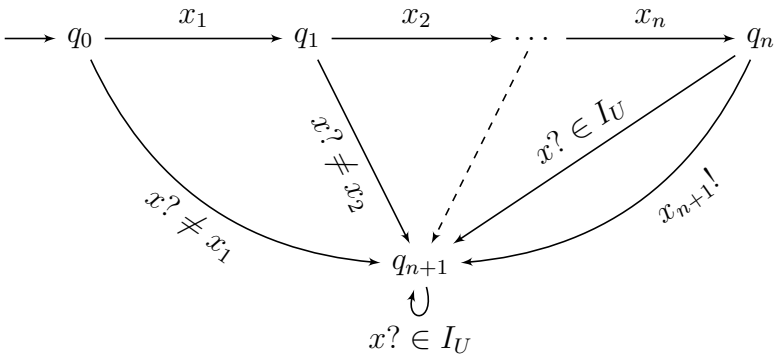

Abbildung 3.1: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$

Für w können nun zwei Fälle unterscheiden werden. Aus beiden wird folgen, dass $\varepsilon \in PrET(U \| S_1)$. Dieses Vorgehen unterscheidet sich von dem in [BV15], da

hier die synchronisierten Aktionen als Outputs vorhanden bleiben und somit nicht $\varepsilon \in StET(U\|S_1)$ gelten kann.

- Fall 2a) ($w \in MIT_1$): In $U\|S_1$ erhält man $(q_0, q_{01}) \overset{x_1 \dots x_n}{\Longrightarrow} (q_n, q')$ mit $q' \overset{x_{n+1}}{\not\longrightarrow}$ und $q_n \overset{x_{n+1}}{\longrightarrow}$. 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 und $I \cap I_U = \emptyset$, 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 und $I \cap I_U = \emptyset$, gilt $x_1, \dots, x_n, x_{n+1} \in O_{U\|S_1}$ und, da $u \in O^*$, folgt $u \in O^*_{U\|S_1}$. Somit ergibt sich $\varepsilon \in PrET(U\|S_1)$.

Da $\varepsilon \in PrET(U\|S_1)$ gilt, kann durch $U\|S_1 \sqsubseteq^{\mathrm{B}}_E U\|S_2$ geschlossen werden, dass auch in $U\|S_2$ ein Error-Zustand 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-Zustand 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 $i \leq n, q' \overset{x_{i+1}}{\not\longrightarrow}$ und $x_{i+1} \in O_U = I$. Durch Projektion erhält man dann $q_{02} \overset{x_1 \dots x_i}{\Longrightarrow} q' \overset{x_{i+1}}{\not\longrightarrow}$ und damit gilt $x_1 \dots x_{i+1} \in MIT_2 \subseteq ET_2$. Somit ist ein Präfix von w in ET_2 enthalten.
- Fall 2ii) (geerbter Error): U hat $x_1 \dots x_i u$ mit $u \in I^*_U = O^*$ ausgeführt und ebenso hat S_2 diesen Wort abgearbeitet. Durch dies hat S_2 einen Zustand in E_2 erreicht, da von U keine Errors geerbt werden können. Es gilt dann $\mathrm{prune}(x_1 \dots\ x_i u) = \mathrm{prune}(x_1 \dots x_i) \in PrET_2 \subseteq ET_2$. Da $x_1 \dots x_i$ ein Präfix von w ist, führt in diesem Fall eine Verlängerung um lokale Aktionen von einem Präfix von w zu einem Error-Zustand. Da ET der Menge aller Verlängerungen von gekürzten Errortraces entspricht, ist $x_1 \dots x_i$ in ET_2 enthalten und somit ist ein Präfix von w in ET_2 enthalten.

Um die andere 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, muss hier nichts gezeigt werden.
- Fall 2 ($w = x_1 \dots x_n$ mit $n \geq 1$): Es wird ein Partner U wie folgt konstruiert (siehe dazu auch Abbildung 3.2):
  - $Q_U = \{q_0, q_1, \dots, q_n, q\}$,

- $q_{0U} = q_0$,
- $E_U = \{q_n\}$,
- $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
  $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i < n\}$
  $\cup \{(q, x, q) \mid x \in I_U\}$.

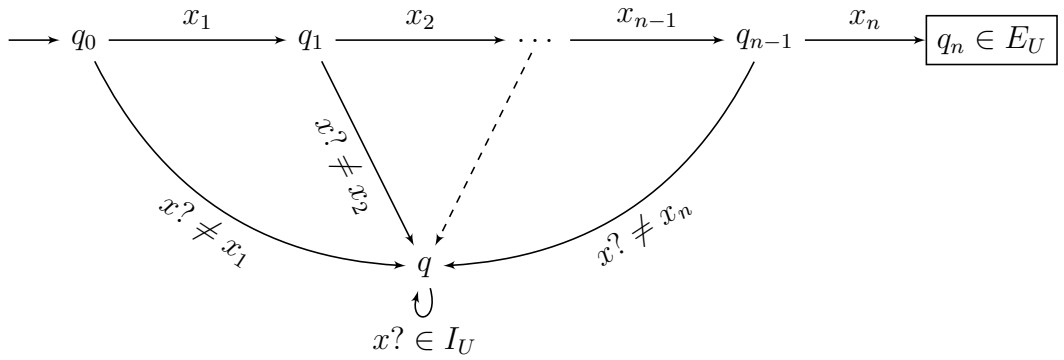

Abbildung 3.2: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$, q_n ist der einzige Error-Zustand

Da $q_{01} \stackrel{w}{\Rightarrow} q'$ gilt, kann man schließen, dass $U \| S_1$ einen lokal erreichbaren geerbten Error hat. Somit muss $U \| S_2$ ebenfalls einen lokal erreichbaren Error-Zustand haben.

- Fall 2a) (neuer Error aufgrund von $x_i \in O_U$ und $q_{02} \stackrel{x_1 \ldots x_{i-1}}{\Longrightarrow} q'' \stackrel{x_i}{\not\rightarrow}$): 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.
- Fall 2b) (neuer Error aufgrund von $a \in O = I_U$): Der einzige Zustand, in dem U nicht alle Inputs erlaubt sind, ist q_n, der bereits ein Error-Zustand ist. Da hier dieser Zustand in $U \| S_2$ erreichbar ist, besitzt das komponierte EIO einen geerbten Error und es gilt $w \in L_2 \subseteq EL_2$, wegen dem folgenden Fall 2c).
- Fall 2c) (geerbter Error von U): Da q_n der einzige Error-Zustand in U ist und alle Aktionen synchronisiert sind, ist dies nur möglich, wenn $q_{02} \stackrel{x_1 \ldots x_n}{\Longrightarrow}$ gilt. In diesem Fall gilt $w \in L_2 \subseteq EL_2$.
- Fall 2d) (geerbter Error von S_2): Es gilt dann $q_{02} \stackrel{x_1 \ldots x_i u}{\Longrightarrow} q' \in E_2$ für $i \geq 0$ und $u \in O^*$. Somit ist $x_1 \ldots x_i u \in StET_2$ und damit $\text{prune}(x_1 \ldots x_i u) = \text{prune}(x_1 \ldots x_i) \in PrET_2 \subseteq EL_2$. 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 $\sqsubseteq_E^{\mathrm{C}}$.

Satz 3.8 (*Vollständige Abstraktheit für Error-Semanik*). *Für zwei EIOs S_1 und S_2 mit derselben Signatur gilt $S_1 \sqsubseteq_E^{\mathrm{C}} S_2 \Leftrightarrow S_1 \sqsubseteq_E S_2$.*

Beweis. „$\Leftarrow$“: Nach Definition gilt, genau dann wenn $\varepsilon \in ET(S)$, ist ein Error-Zustand 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-Zustand in S_1 nur dann lokal erreichbar, wenn dieser auch in S_2 lokal erreichbar ist. Dadurch folgt, dass $S_1 \sqsubseteq_E^{\mathrm{B}} S_2$ gilt, da $\sqsubseteq_E^{\mathrm{B}}$ in Definition 3.2 über die lokale Erreichbarkeit der Error-Zustände definiert wurde. Es ist also $\sqsubseteq_E$ in $\sqsubseteq_E^{\mathrm{B}}$ enthalten. Wie in Korollar 3.6 gezeigt, ist $\sqsubseteq_E$ eine Präkongruenz bezüglich $\cdot\|\cdot$. Da $\sqsubseteq_E^{\mathrm{C}}$ die gröbste Präkongruenz bezüglich $\cdot\|\cdot$ ist, die in $\sqsubseteq_E^{\mathrm{B}}$ enthalten ist, muss $\sqsubseteq_E$ in $\sqsubseteq_E^{\mathrm{C}}$ enthalten sein. Es folgt also aus $S_1 \sqsubseteq_E S_2$, dass auch $S_1 \sqsubseteq_E^{\mathrm{C}} S_2$ gilt.

„$\Rightarrow$“: Durch die Definition von $\sqsubseteq_E^{\mathrm{C}}$ als Präkongruenz in 3.2 folgt aus $S_1 \sqsubseteq_E^{\mathrm{C}} S_2$, dass $U\|S_1 \sqsubseteq_E^{\mathrm{C}} U\|S_2$ für alle EIOs U gilt, die mit S_1 komponierbar sind. Da $\sqsubseteq_E^{\mathrm{C}}$ nach Definition auch in $\sqsubseteq_E^{\mathrm{B}}$ enthalten sein soll, folgt aus $U\|S_1 \sqsubseteq_E^{\mathrm{C}} U\|S_2$ auch die Gültigkeit von $U\|S_1 \sqsubseteq_E^{\mathrm{B}} U\|S_2$ 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.

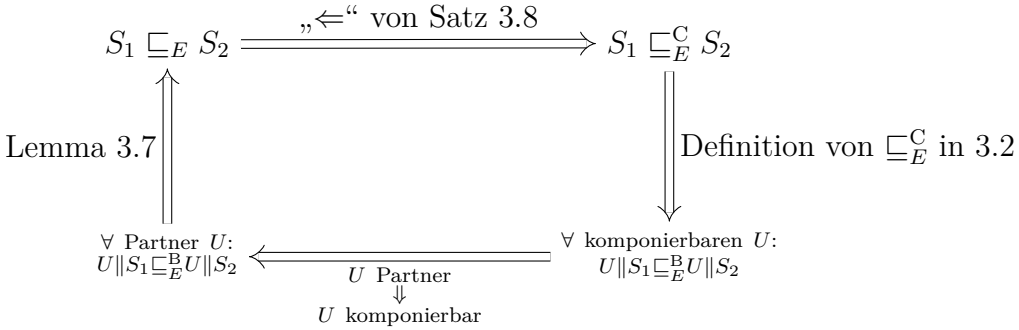

Abbildung 3.3: Folgerungskette

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.
Aus Satz 3.8 und Lemma 3.7 ergibt sich das folgende Korollar.

Korollar 3.9. *Es gilt: $S_1 \sqsubseteq_E S_2 \Leftrightarrow U\|S_1 \sqsubseteq_E^{\mathrm{B}} U\|S_2$ 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 betrachteten Systeme internalisiert.

Proposition 3.10 (*Error-Basisrelation bzgl. Internalisierung*). *Wenn $S_1 \sqsubseteq_E^{\mathrm{B}} S_2$ gilt, dann folgt daraus, dass auch $S_1/X \sqsubseteq_E^{\mathrm{B}} S_2/X$ 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-Zustand, 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 diese Outputs. 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 nichts an der Erreichbarkeit oder der Menge der Error-Zustände geändert werden. Es ist also jeder Error-Zustand, 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) $L(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in L(S) : w'|_{\Sigma\backslash X} = w\Big\}$,
(ii) $ET(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in ET(S) : w'|_{\Sigma\backslash X} = w\Big\}$,
(iii) $EL(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in EL(S) : w'|_{\Sigma\backslash X} = w\Big\}$.

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 $q_0 \overset{w'}{\Rightarrow} q$ mit $q \in Q$. Es gibt also zu jedem $w' = a_1 a_2 \dots a_n \in L(S)$ ein Ablauf $q_0 \overset{a_1}{\Rightarrow} q_1 \overset{a_2}{\Rightarrow} \dots \overset{a_n}{\Rightarrow} q_n$ 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 an der Ausführungsreihenfolge der a_is und auch 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 $q_0 \overset{\substack{\tau \text{ falls } a_1 \in X \\ \text{sonst } a_1}}{\Longrightarrow} q_1 \overset{\substack{\tau \text{ falls } a_2 \in X \\ \text{sonst } a_2}}{\Longrightarrow} \dots \overset{\substack{\tau \text{ falls } a_n \in X \\ \text{sonst } a_n}}{\Longrightarrow} q_n$. Dabei bleibt durch das Hiding von

Aktionen aus X in w' nur noch $w := w'|_{\Sigma\setminus X}$ erhalten. Diese Projektion des Wortes w' auf die eingeschränkte Aktionenmenge ist dann in $L(S/X)$ enthalten, da immer noch derselbe 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, das besondere Eigenschaften erfüllen soll und für das es einen Ablauf gibt. Hierfür muss jedoch unterschieden werden aus welchem Grund das w' in $ET(S)$ enthalten ist.

- Fall 1 ($w' \in \text{cont}(PrET(S))$): In diesem Fall gibt es einen Ablauf für ein Präfix dieses w's, der zu einem Ablauf ergänzt werden kann, der zu einem Zustand aus E führt. Der Ablauf hat also die Form $q_0 \stackrel{a_1}{\Rightarrow} q_1 \stackrel{a_2}{\Rightarrow} \dots \stackrel{a_m}{\Rightarrow} q_m \stackrel{b_1}{\Rightarrow} q_{m+1} \stackrel{b_2}{\Rightarrow} \dots \stackrel{b_l}{\Rightarrow} q_{m+l}$ mit $m \leq n$, $l \geq 0$ und $\forall i \in \{1, \dots, l\} : b_i \in O(S)$. Es gilt dann $a_1 a_2 \dots a_m b_1 b_2 \dots b_l \in StET(S)$. Analog wie im Beweisteil zu (i) wird dieser Ablauf durch die Internalisierung reduziert. Somit ist die Projektion von $a_1 a_2 \dots a_m b_1 b_2 \dots b_l$ auf die eingeschränkte Aktionenmenge auf jeden Fall in $StET(S/X) \subseteq ET(S/X)$ enthalten. Da $b_1 b_2 \dots b_l \in O(S)^*$, gilt nach der Projektion auch $(b_1 b_2 \dots b_l)|_{\Sigma\setminus X} \in O(S/X)^*$ und somit $\text{prune}((a_1 a_2 \dots a_m)|_{\Sigma\setminus X}) = \text{prune}((a_1 a_2 \dots a_m b_1 b_2 \dots b_l)|_{\Sigma\setminus X})$. Da ET eine Menge ist, die nach Definition unter cont abgeschlossen ist, sind alle Verlängerungen von $(a_1 a_2 \dots a_m)|_{\Sigma\setminus X}$ 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 ($w' \in \text{cont}(MIT(S))$): 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 $q_0 \stackrel{a_1}{\Rightarrow} q_1 \stackrel{a_2}{\Rightarrow} \dots \stackrel{a_{m-1}}{\Rightarrow} q_{m-1} \stackrel{a_m}{\not\Rightarrow}$ mit $m \leq n$ und $a_1 a_2 \dots a_m \in MIT(S)$. Analog zum letzten Fall und zum Teil (i) wird dieser Ablauf durch die Internalisierung reduziert. Somit ist die Projektion von $a_1 a_2 \dots a_m$ auf die eingeschränkte Aktionenmenge auf jeden Fall in $MIT(S/X) \subseteq ET(S/X)$ enthalten. Da ET unter cont abgeschlossen ist, sind alle Verlängerungen von $(a_1 a_2 \dots a_m)|_{\Sigma\setminus X}$ ebenfalls in $ET(S/X)$ enthalten. Speziell gilt also auch $w := w'|_{\Sigma\setminus X} \in ET(S/X)$.

Ein Trace $w \in ET(S/X)$ muss wie oben nicht im Transitionssystem enthalten sein. Es kann jedoch ein Präfix von w wie oben gefunden werden, 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 durch Outputs aus der Menge X 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, das sich von w nur durch hinzugefügte Aktionen aus X unterscheidet.

(iii) Für ein Trace $w' = a_1 a_2 \dots a_n \in EL(S)$ 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 wurde, ist dadurch auch gezeigt, dass $w \in EL(S/X)$ gilt.
Ebenfalls analog zu den beiden vorangegangenen 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 Korollar 3.6 ist bekannt, dass $\sqsubseteq_E$ eine Präkongruenz bezüglich $\cdot \| \cdot$ ist, und aus Satz 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, erhält man das folgende Korollar.

Korollar 3.12 (*Error-Präkongurenz mit Internalisierung*). *Die Relation $\sqsubseteq_E$ 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, wie im letzten Kapitel, sondern auch um die Erreichbarkeit von Ruhe-Zuständen. Es wird dabei eine analoge Herangehensweise zur der im letzten Kapitel angewendet, wobei nun [CJK14] als Quelle verwendet wird. 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.
Zustände, die keine Outputs ohne einen Input ausführen können, werden als in einer Art Verklemmung angesehen, da sie ohne Zutun von Außen den Zustand nicht mehr verlassen können. So ein Zustand hat also keine Transitions-Möglichkeiten für einen Output. Falls dieser Zustand die Möglichkeit für eine interne Aktion hat, darf durch diese τs niemals ein Zustand erreicht werden, von dem aus ein Output möglich ist. Ein Zustand, der keine Outputs und τs ausführen kann, ist also ein Deadlock-Zustand, in denen das System nichts mehr tun kann ohne einen Input. Wenn man eine Erweiterung um τs zu Zuständen ohne Outputs zulässt, hat man zusätzliche noch Verklemmungen der Art Livelock, da diese Zustände möglicherweise beliebig viele interne Aktionen ausführen können, jedoch nie aus eigener Kraft einen wirklichen Fortschritt in Form eines Outputs bewirken können. Die Menge der Zustände, die sich in einer Verklemmung befinden, würde also durch $\left\{ q \in Q \mid \forall a \in O : q \overset{a}{\not\Rightarrow} \right\}$ beschrieben werden. Somit wären dies alle Zustände, die keine Möglichkeit haben ohne einen Input von Außen je wieder einen Output machen zu können. Falls man diese Definition verwenden würde, müsste man immer alle Zustände betrachten, die durch τs erreichbar sind. Dies würde einige Betrachtungen deutlich aufwendiger machen und soll deshalb hier nicht behandelt werden. Die Definition für die betrachteten Verklemmungen, hier Ruhe genannt, beschränkt sich auf Zustände, die keine Outputs und τs ausführen 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 $:= \left\{ q \in Q \mid \forall \alpha \in (O \cup \{\tau\}) : q \overset{\alpha}{\not\rightarrow} \right\}$.

Für die Erreichbarkeit wird wie im letzten Kapitel wieder der optimistische 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 Ruhe im Vergleich zu Error als weniger „schlimmer Fehler“ anzusehen ist. 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 Error- 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 $S_1 \sqsubseteq^{\mathrm{B}}_{Qui} S_2$ 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.*
$\sqsubseteq^{\mathrm{C}}_{Qui}$ bezeichnet die vollständig abstrakte Präkongruenz *von $\sqsubseteq^{\mathrm{B}}_{Qui}$ 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 Error-Zuständen. Es genügt deshalb für Ruhe die strikten Traces ohne Kürzung zu betrachten.

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

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

Es wird nur eine Ruhe-Semantik 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 $QET(S) := StQT(S) \cup ET(S)$.*

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 $QET_1 \subseteq QET_2$ gilt.

Für die Menge der error-gefluteten Ruhetraces QET 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 $\left((q_1, q_2) \xrightarrow{\tau}\right)$: 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 $\left((q_1, q_2) \xrightarrow{a} \text{ mit } a \in O \backslash \text{Synch}(S_1, S_2)\right)$: Da es sich bei a um einen Output handelt, der nicht in $\text{Synch}(S_1, S_2)$ 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 ausführen.

- Fall 3 $\left((q_1, q_2) \xrightarrow{a} \text{ mit } a \in O \cap \text{Synch}(S_1, S_2)\right)$: 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 $q_2 \xrightarrow{a}$. Die Transition für das System S_1 ist jedoch in der Voraussetzung ausgeschlossen worden. Somit ist es nicht möglich, dass S_{12} diese in diesem Fall angenommene 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 gilt also, dass aus $q_i \in Qui_i$ für $i \in \{1, 2\}$ $(q_1, q_2) \in Qui_{12}$ folgt.

2.:
Es gilt $(q_1, q_2) \in Qui_{12} \backslash E_{12}$, somit hat dieser Zustand allenfalls 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 $\left(q_1 \xrightarrow{\tau}\right)$: 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 und somit kann dieser Fall nicht eintreten.
- Fall 2 $\left(q_1 \xrightarrow{a} \text{ mit } a \in O_1 \backslash \text{Synch}(S_1, S_2)\right)$: Da es sich bei a um einen Output handelt, der nicht zu synchronisieren ist, wird dieser einfach in die Parallelkomposition übernommen. Es müsste also $(q_1, q_2) \xrightarrow{a}$ mit $a \in O_{12}$ gelten, was jedoch verboten ist. Somit kann die Transition für S_1 in diesem Fall nicht möglich sein.
- Fall 3 $\left(q_1 \xrightarrow{a} \text{ mit } a \in O_1 \cap \text{Synch}(S_1, S_2) \text{ und } q_2 \xrightarrow{a}\right)$: 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 Transitions-Möglichkeit ist jedoch für S_{12} nach Voraussetzung nicht erlaubt. Es folgt also auch, dass dieser Fall nicht eintreten kann.
- Fall 4 $\left(q_1 \xrightarrow{a} \text{ mit } a \in O_1 \cap \text{Synch}(S_1, S_2) \text{ und } q_2 \overset{a}{\not\rightarrow}\right)$: Da S_2 diese Transition nicht ausführen kann, handelt es sich hier um einen neuen Error, da die Synchronisation des Outputs a von S_1 mit einem Input a von S_2 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. $ET_{12} = \text{cont}\,(\text{prune}\,((ET_1 \| EL_2) \cup (EL_1 \| ET_2)))$,
2. $QET_{12} = (QET_1 \| QET_2) \cup ET_{12}$,
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 offensichtlich in der rechten Seite enthalten. Somit wird ab jetzt ein $w \in StQT_{12} \backslash ET_{12}$ betrachtet und es wird versucht dessen Zugehörigkeit zur rechten Menge 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 $w_1 \in StQT_1 \subseteq$

QET_1 und $w_2 \in StQT_2 \subseteq QET_2$. Daraus folgt dann $w \in QET_1 \| QET_2$ 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 $w \in QET_1 \| QET_2$ betrachtet und gezeigt, dass es in der linken Menge enthalten ist. Da $QET_i = StQT_i \cup ET_i$ 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 $w_2 \in StQT_2 \subseteq L_2$ oder $w_2 \in ET_2$ gelten und somit ist auf jeden Fall w_2 in EL_2 enthalten. Daraus kann dann mit dem ersten Punkt von Satz 3.5 gefolgert werden, dass $w \in ET_{12}$ gilt und damit ist w in der linken Seite der Gleichung enthalten.
- Fall 2 ($w_1 \in StQT_1 \backslash ET_1 \wedge w_2 \in StQT_2 \backslash ET_2$): Es gilt in diesem Fall $q_{01} \overset{w_1}{\Rightarrow} q_1 \in Qui_1$ und $q_{02} \overset{w_2}{\Rightarrow} q_2 \in Qui_2$. 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 $(q_{01}, q_{02}) \overset{w}{\Rightarrow} (q_1, q_2) \in Qui_{12}$ und dadurch ist w in der linken Seite der Gleichung enthalten, da $w \in StQT_{12} \subseteq QET_{12}$ gilt.

□

Das folgende Korollar 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.

Korollar 4.8 (*Ruhe-Präkongruenz*). *Die Relation $\sqsubseteq_{Qui}$ 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 $QET_1 \subseteq QET_2$ sowohl $S_{31} \sqsubseteq_E S_{32}$ als auch $QET_{31} \subseteq QET_{32}$ folgt. Dies ergibt sich, wie im Beweis zu Korollar 3.6, aus der Monotonie von $\cdot\|\cdot$ auf Sprachen wie folgt:

- $S_{31} \overset{\text{Korollar 3.6 und } S_1 \sqsubseteq_E S_2}{\sqsubseteq_E} S_{32}$,
- $QET_{31} \overset{4.7\ 2.}{=} (QET_3 \| QET_1) \cup ET_{31}$
  $\overset{ET_{31} \subseteq ET_{32} \text{ und } QET_1 \subseteq QET_2}{\subseteq} (QET_3 \| QET_2) \cup ET_{32}$
  $\overset{4.7\ 2.}{=} QET_{32}$.

□

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 $U\|S_1 \sqsubseteq^{\mathrm{B}}_{Qui} U\|S_2$ für alle Partner U gilt, dann folgt daraus $S_1 \sqsubseteq_{Qui} S_2$.*

Beweis. Da S_1 und S_2 die gleiche Signatur haben, wird $I := I_1 = I_2$ und $O := O_1 = O_2$ definiert. 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:

- $S_1 \sqsubseteq_E S_2$,
- $QET_1 \subseteq QET_2$.

In Lemma 3.7 wurde bereits etwas Ähnliches gezeigt, jedoch wurde dort als Voraussetzung $U\|S_1 \sqsubseteq^{\mathrm{B}}_{E} U\|S_2$ für alle Partner U verwendet und hier dieselbe Aussage mit der Basisrelation der Ruhe. Dadurch, dass die hier verwendete Basisrelation nichts über die Art des erreichbaren Fehlers in den Komponenten aussagt, kann der Beweis aus Lemma 3.7 nicht verwendet werden. Es kann also aus der lokalen Erreichbarkeit eines Error-Zustandes in S_1' und dem relationalen Zusammenhang von $S_1' \sqsubseteq^{\mathrm{B}}_{Qui} S_2'$ nur geschlossen werden, dass in S_2' auch ein Fehler lokal erreichbar ist, jedoch kann dieser Fehler Error oder Ruhe sein. Analog verhält es sich, wenn in S_1' ein Ruhe-Zustand lokal erreichbar ist.
Es muss also für den ersten Punkt noch 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 $U\|S_1 \sqsubseteq^{\mathrm{B}}_{Qui} U\|S_2$ 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-Zustand 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 Schlinge 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-Zustand 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):
  - $Q_U = \{q_0, q_1, \dots, q_{n+1}\}$,
  - $q_{0U} = q_0$,
  - $E_U = \emptyset$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i \leq n\}$
    $\cup \{(q_i, x, q_{n+1}) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i \leq n\}$
    $\cup \{(q_{n+1}, x, q_{n+1}) \mid x \in I_U\}$
    $\cup \{(q_i, \tau, q_i) \mid 0 \leq i \leq n+1\}$.

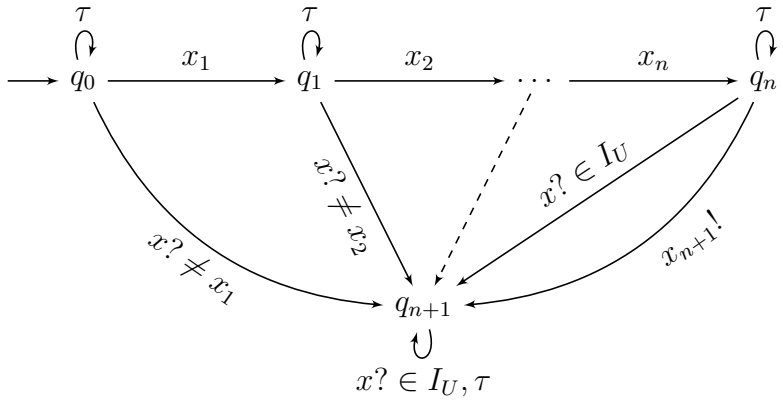

Abbildung 4.1: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$

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 des ersten Punktes 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 Error-Zustandes in $U \| S_2$ geschlossen werden. Die weitere Argumentation verläuft dann analog zu Fall 2, derselben 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 aufgrund 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 dieses 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 des zweiten Punktes aus Lemma 3.7 ab. Somit ist der Partner U dann wie folgt definiert (siehe dazu auch Abbildung 4.2):
  - $Q_U = \{q_0, q_1, \dots, q_n, q\}$,
  - $q_{0U} = q_0$,
  - $E_U = \{q_n\}$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
    $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i < n\}$
    $\cup \{(q_i, \tau, q_i) \mid 0 \leq i \leq n\}$
    $\cup \{(q, \alpha, q) \mid \alpha \in I_U \cup \{\tau\}\}$.

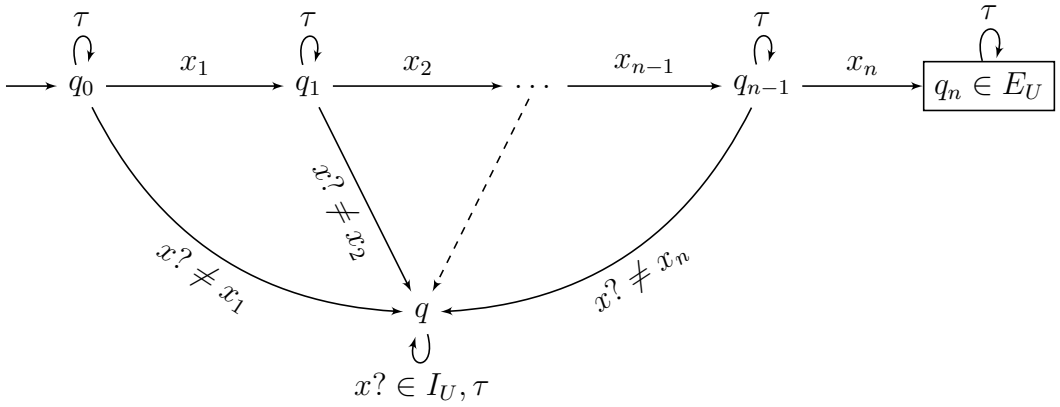

Abbildung 4.2: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$, q_n ist der einzige Error-Zustand

  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 der selben Inklusionsrichtung von Lemma 3.7. Und somit gilt für alle Fälle (2a) bis 2d)), dass w in EL_2 enthalten ist.

So bleibt nur noch der letzte Beweispunkt zu zeigen, d.h. die Inklusion $QET_1 \subseteq QET_2$. Diese Inklusion kann jedoch, anlog zum Beweis der Inklusion der error-gefluteten Sprachen, noch weiter eingeschränkt werden. Da bereits bekannt ist, dass $ET_1 \subseteq ET_2$ gilt, muss nur noch $StQT_1 \backslash ET_1 \subseteq QET_2$ gezeigt werden.
Es wird ein $w \in StQT_1 \backslash ET_1$ gewählt und gezeigt, dass dieses auch in QET_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.
Das betrachtete w hat also die Form $w = x_1 \dots x_n \in \Sigma^*$ mit $n \geq 0$. Es wird der folgende Partner U betrachtet (siehe auch Abbildung 4.3):

- $Q_U = \{q_0, q_1, \dots, q_n, q\}$,
- $q_{0U} = q_0$,
- $E_U = \emptyset$,
- $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
  $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i < n\}$
  $\cup \{(q_i, \tau, q_i) \mid 0 \leq i < n\}$
  $\cup \{(q_n, x, q) \mid x \in I_U\}$
  $\cup \{(q, \alpha, q) \mid \alpha \in I_U \cup \{\tau\}\}$.

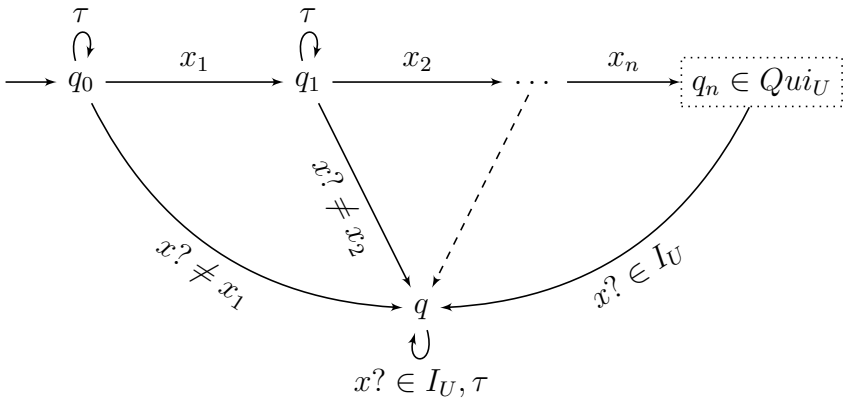

Abbildung 4.3: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$, q_n ist der einzige Ruhe-Zustand

Falls für das betrachtete $w = \varepsilon$ gilt, reduziert sich der Partner U auf den Zustand $q_n = q_0$ und den Zustand q. Es ist also in diesem Fall der Startzustand gleich dem ruhigen Zustand.
Allgemein ist der Zustand q_n aus U der einzige ruhige Zustand in U. Es gilt wegen des ersten Punktes von 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 zurückgelegt wird, kein Error-Zustand 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 in der Parallelkomposition beider Systeme 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 a) ($\varepsilon \in ET(U\|S_2)$): Es handelt sich bei dem lokal erreichbaren Fehler um einen Error-Zustand. Es ist somit nicht relevant, ob das w ausführbar ist. Der Error kann sowohl von S_2 geerbt sein, wie auch durch fehlende Synchronisations-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 $w \in ET_2 \subseteq QET_2$.
- Fall b) (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 $w \in StQT_2 \subseteq QET_2$.

□

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

Satz 4.10 (*Vollständige Abstraktheit für Ruhe-Semantik*). *Seien S_1 und S_2 zwei EIOs mit derselben Signatur. Dann gilt $S_1 \sqsubseteq^{\mathrm{C}}_{Qui} S_2 \Leftrightarrow S_1 \sqsubseteq_{Qui} S_2$.*

Beweis. „⇐“: Nach Definition gilt $w \in QET(S)$ mit $w \in O(S)^*$ genau dann, wenn in S ein Ruhe-Zustand oder ein Error-Zustand lokal erreichbar ist. $S_1 \sqsubseteq_{Qui} S_2$ impliziert, dass $w \in QET_2$ gilt, wenn $w \in QET_1$ 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 $S_1 \sqsubseteq^{\mathrm{B}}_{Qui} S_2$ gilt. Es ist also $\sqsubseteq_{Qui}$ in $\sqsubseteq^{\mathrm{B}}_{Qui}$ enthalten. Im Korollar 4.8 wurde festgestellt, dass $\sqsubseteq_{Qui}$ eine Präkongruenz ist. Da jedoch $\sqsubseteq^{\mathrm{C}}_{Qui}$ nach Definition 4.3 die gröbste Präkongruenz bezüglich $\cdot\|\cdot$ ist, die in $\sqsubseteq^{\mathrm{B}}_{Qui}$ enthalten ist, muss $\sqsubseteq_{Qui}$ in $\sqsubseteq^{\mathrm{C}}_{Qui}$ enthalten sein. Es folgt also aus $S_1 \sqsubseteq_{Qui} S_2$, dass auch $S_1 \sqsubseteq^{\mathrm{C}}_{Qui} S_2$ gilt.

„⇒“: Durch die Definition von $\sqsubseteq^{\mathrm{C}}_{Qui}$ als Präkongruenz in 4.3 folgt aus $S_1 \sqsubseteq^{\mathrm{C}}_{Qui} S_2$, dass $U\|S_1 \sqsubseteq^{\mathrm{C}}_{Qui} U\|S_2$ für alle EIOs U gilt, die mit S_1 komponierbar sind. Da $\sqsubseteq^{\mathrm{C}}_{Qui}$ nach Definition in $\sqsubseteq^{\mathrm{B}}_{Qui}$ enthalte ist, folgt auch die Gültigkeit von $U\|S_1 \sqsubseteq^{\mathrm{B}}_{Qui} U\|S_2$ 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.

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

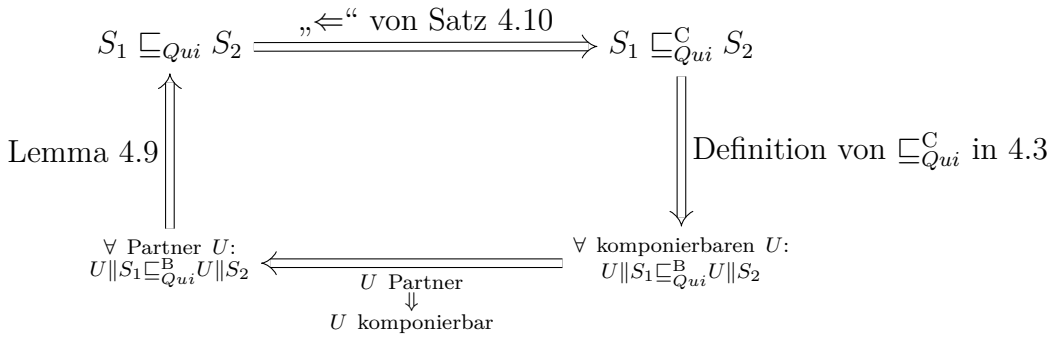

Abbildung 4.4: Folgerungskette

error- und ruhe-frei mit U kommuniziert. Dann wird auch diese Verfeinerung durch $\sqsubseteq_{Qui}$ charakterisiert.
Aus Satz 4.10 und Lemma 4.9 erhält man das folgende Korollar.

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

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, in der hier verwendeten Definition, 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* $S_1 \sqsubseteq^{\mathrm{B}}_{Qui} S_2$ *gilt, dann folgt daraus, dass auch* $S_1/X \sqsubseteq^{\mathrm{B}}_{Qui} S_2/X$ *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 nur Outputs verborgen werden, die bereits in der Menge der lokalen Aktionen enthalten sind. Die Menge der Ruhe-Zustände kann sich durch das Internalisieren 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 $S_1 \sqsubseteq^{\mathrm{B}}_{Qui} S_2$ auch der Zusammenhang $S_1/X \sqsubseteq^{\mathrm{B}}_{Qui} S_2/X$. □

Satz 4.13 (*Ruhe-Präkongruenz 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) $L(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in L(S) : w'|_{\Sigma\backslash X} = w\Big\}$,
(ii) $ET(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in ET(S) : w'|_{\Sigma\backslash X} = w\Big\}$,
(iii) $EL(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in EL(S) : w'|_{\Sigma\backslash X} = w\Big\}$,
(iv) $QET(S/X) = \Big\{w \in (\Sigma\backslash X)^* \mid \exists w' \in QET(S) : w'|_{\Sigma\backslash X} = w\Big\}$.

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) aus Satz 3.11 gezeigt werden. Für ein $w' \in QET(S)$ gilt $w' \in ET(S)$ oder $w' \in StQT(S) \subseteq L(S)$. Es gilt also mit Punkt (i) und (ii), dass $w := w'|_{\Sigma\backslash X}$ 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 $w \in QET(S/X)$ gilt.
Für ein $w \in QET(S/X)$ kann, ebenfalls wie im Beweis zu Punkt (i) und (ii), mittels Ersetzen von τs durch Outputs aus X 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 Erweiterung in $QET(S)$ enthalten ist.

Da $S_1 \sqsubseteq_{Qui} S_2$ gilt, kann geschlossen werden, dass $S_1 \sqsubseteq_E S_2$ und $QET_1 \subseteq QET_2$ 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 $QET_1 \subseteq QET_2$ ebenfalls $QET(S_1/X) \subseteq QET(S_2/X)$ 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 Präkongruenzeigenschaften von $\cdot\|\cdot$ und $\cdot/\cdot$ bezüglich $\sqsubseteq_{Qui}$ aus dem Korollar 4.8 und dem Satz 4.13 geschlossen werden.

Korollar 4.14 (*Ruhe-Präkongruenz mit Internalisierung*). *Die Relation $\sqsubseteq_{Qui}$ 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 Error-, Ruhe- und Divergente-Zustände betrachtet. Es eignet sich also [CJK14] 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 Abwandlung der Präkongruenzen aus den letzten beiden Kapiteln handeln soll, wird dabei ähnlich vorgegangen wie in den letzten beiden Kapiteln.
Wie bereits oben und im letzten Kapitel erwähnt wurden in [CJK14] auch noch divergente Zustände als Fehler-Zustände betrachtet. Um zu klären, was darunter zu verstehen ist, 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.*
Die Menge $Div(S)$ besteht aus all diesen divergenten Zuständen des EIOs S.

Die unendliche Folge an τs kann durch eine Schleife an einem durch τs erreichbaren Zustand ausführbar sein oder durch einen Weg, der mit τs ausführbar ist, mit dem unendliche viele Zustände durchlaufen werden. Es ist jedoch zu beachten, dass ein Zustand, von dem aus unendlich viele Zustände durch τs erreichbar sind, nicht divergent sein muss. Es ist auch möglich, dass dieser Zustand eine unendliche Verzeigung hat und somit keine 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 erreicht werden kann, ist Divergenz als ähnlich „schlimm“ zu bewerten wie ein Error-Zustand.

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 Error-, Ruhe- und Divergenz-Zustände lokal erreichbar sind.*

Definition 5.3 (*Divergenz-Verfeinerungs-Basisrelation*). *Für EIOs S_1 und S_2 mit der gleichen Signatur wird $S_1 \sqsubseteq^{\mathrm{B}}_{Div} S_2$ geschrieben, wenn ein Error-, Ruhe- 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.
$\sqsubseteq^{\mathrm{C}}_{Div}$ *bezeichnet die* vollständige abstrakte Präkongruenz *von* $\sqsubseteq^{\mathrm{B}}_{Div}$ *bezüglich* $\cdot\|\cdot$.

Da nun die grundlegenden Definitionen für Divergenz festgehalten sind, kann man sich einen Begriff für die Traces zu divergenten Zuständen bilden. Da oben bereits festgestellt wurde, dass Divergenz als ähnlich „schlimmer Fehler“ anzusehen ist wie Error und dass das Divergieren eines Systems nicht mehr verhinderbar ist, sobald ein divergenter Zustand lokal erreichbar ist, kommt für die Divergenztraces wieder die prune-Funktion zu Einsatz. Ein System, das unendliche viele τs ausführen kann, ist von außen nicht von so einem System zu unterscheiden, das einen Error-Zustand erreicht. Somit wird in den Trace-Mengen auch nicht zwischen Errortraces und Divergenztraces explizit unterschieden. Dadurch kann man auch nicht mehr nur mit den Errortraces die Sprache fluten, sondern muss sowohl mit den Errortraces wie auch den Divergenztraces fluten. Ebenso werden die strikten Ruhetraces mit diesen beiden Trace-Mengen geflutet.

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

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

Da in [CJK14] bereits direkt Divergenz mit betrachtet wurde, wird dort die Flutung der Traces so vorgenommen, dass die Errortraces in den Divergenztraces enthalten sind und die Divergenztraces in den Ruhetraces. Es wird dort die Error-Präkongruenz erhalten und nur durch die Inklusionen der Divergenztraces und Ruhetraces erweitert. Dies ist aufgrund der Basisrelation hier nicht möglich. Da hier zwischen den Errortraces und den Divergenztraces nicht unterschieden werden soll, kann diese Inklusionskette nicht so umgesetzt werden wie in [CJK14], jedoch entsteht durch das Fluten immer noch derselbe Effekt für die Ruhetraces. Somit kann nur die Semantik der Menge ET aus den früheren Kapiteln übernommen werden, die jedoch für den weiteren Verlauf nur innerhalb der Trace-Menge EDT relevant sein wird.

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

- *Die Menge der* Divergenztraces *von S ist* $DT(S) := \mathrm{cont}(PrDT(S))$.
- *Die Menge der* Error-Divergenztraces *von S ist* $EDT(S) := ET(S) \cup DT(S)$.
- *Die Menge der* error-divergenz-gefluteten Ruhetraces *von S ist* $QDT(S) := StQT(S) \cup EDT(S)$.
- *Die Menge der* error-divergenz-gefluteten Sprache *von S ist* $EDL(S) := L(S) \cup EDT(S)$.

Für zwei EIOs S_1, S_2 mit der gleichen Signatur schreibt man $S_1 \sqsubseteq_{Div} S_2$, wenn $EDT_1 \subseteq EDT_2$, $QDT_1 \subseteq QDT_2$ und $EDL_1 \subseteq EDL_2$ gilt.

In der letzten Definition wurde wieder durch das Fluten eine Informationsvermischung vorgenommen, sowohl im Fall QDT wie auch im Fall EDL mit EDT, den Error-Divergenztraces. Da hier die Errortraces um die Divergenztraces erweitert wurden und die error-geflutete Sprache zur error-divergenz-gefluteten Sprache erweitert wurde, folgt dass die Relation $\sqsubseteq_{Div}$ im Gegensatz zu $\sqsubseteq_{Qui}$ keine direkte Einschränkung der Relation $\sqsubseteq_E$ ist. Durch die Definition der Basisrelation ist auch keine andere Lösung möglich. Dies wird an folgendem Beispiel illustriert. Hierfür werden die nachfolgenden Transitionssysteme benötigt:

- $S_1 = (\{q_{01}\}, \emptyset, \emptyset, \emptyset, q_{01}, \{q_{01}\})$ (siehe auch Abbildung 5.1),
- $S_2 = (\{q_{02}\}, \emptyset, \emptyset, \{(q_{02}, \tau, q_{02})\}, q_{02}, \emptyset)$ (siehe auch Abbildung 5.2).

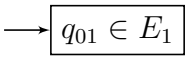

Abbildung 5.1: S_1

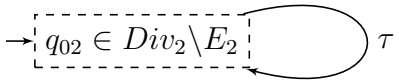

Abbildung 5.2: S_2

Der Startzustand von S_1 ist in der Menge E_1 enthalten und somit ein Error-Zustand. Da keine unendliche τ-Transitionen in S_1 möglich sind, ist q_{01} kein Divergenz-Zustand. Bei S_2 ist der Startzustand durch die τ-Schleife divergent und nicht in der Menge $E_2 = \emptyset$ enthalten.
Somit stehen die beiden Transitionssysteme S_1 und S_2 in der Basisrelation $\sqsubseteq^{\mathrm{B}}_{Div}$, obwohl in S_1 nur ein Error-Zustand und in S_2 nur ein Divergenz-Zustand jedoch kein Error-Zustand lokal erreichbar ist. Um ein analoges Lemma zu den Lemmata 3.7 und 4.9 formulieren zu können sollten somit auch S_1 und S_2 in der Relation $\sqsubseteq_{Div}$ stehen, da ein entsprechendes U, die Identität (Transitionssystem aus einem Startzustand und einer Schleife für ω), gewählt werden kann. Dies wurde in der letzten Definition bereits berücksichtigt. Es kann also unter der Verfeinerungsrelation $\sqsubseteq_{Div}$ nicht unterschieden werden, ob das System Error oder Divergenz als Fehler hat. Somit folgt also aus $S_1 \sqsubseteq_{Div} S_2$ nicht $S_1 \sqsubseteq_E S_2$, da in S_2 kein Error-Zustand lokal erreichbar ist.

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. $EDT_{12} = \mathrm{cont}\,(\mathrm{prune}\,((EDT_1 \| EDL_2) \cup (EDL_1 \| EDT_2)))$,
2. $QDT_{12} = (QDT_1 \| QDT_2) \cup EDT_{12}$,
3. $EDL_{12} = (EDL_1 \| EDL_2) \cup EDT_{12}$.

Beweis. 1. „$\subseteq$“:
Da beide Seiten der Gleichung unter cont abgeschlossen sind, genügt es ein präfix-minimales Element w zu betrachten. Es muss hier unterschieden werden, ob $w \in ET_{12}$ oder $w \in DT_{12} \backslash ET_{12}$ betrachtet wird. Im ersten Fall ist das w in der rechten Seite der

Gleichung enthalten wegen des Beweis des ersten Punktes von Satz 3.5 und da $ET(S) \subseteq EDT(S)$ und $EL(S) \subseteq EDL(S)$ gilt. Deshalb wird im weiteren Verlauf dieses Beweises davon ausgegangen, dass $w \in DT_{12} \backslash ET_{12}$ gilt und es wird versucht zu zeigen, dass dieses w ebenfalls in der rechten Seite enthalten ist. Da das betrachtete w präfix-minimal ist, gilt $w \in PrDT_{12} \backslash ET_{12}$. Aus der Definition 5.5 weiß man, dass ein $v \in O_{12}^*$ existiert, sodass $(q_{01}, q_{02}) \overset{w}{\Rightarrow} (q_1, q_2) \overset{v}{\Rightarrow} (q_1', q_2')$ gilt mit $(q_1', q_2') \in Div_{12}$. 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. $w_1 v_1 \in StDT_1 \subseteq EDT_1$. Da $q_{02} \overset{w_2 v_2}{\Longrightarrow}$ gilt, erhält man $w_2 v_2 \in EDL_2$. Somit gilt insgesamt $wv \in EDT_1 \| EDL_2$ und w ist in der rechten Seite der Gleichung enthalten, da $v \in O_{12}^*$ gilt und somit $\text{prune}(wv) = \text{prune}(w)$.

1. „$\supseteq$“:
Es wird ebenso wie oben nur ein präfix-minimales x betrachtet wegen des Abschlusses beider Seiten der Gleichung unter cont. Es wird also für ein beliebiges $x \in \text{prune}\,(\,(EDT_1 \| EDL_2) \cup (EDL_1 \| EDT_2))$ gezeigt, dass dieses oder einer seiner Präfixe auch in EDT_{12} enthalten ist. Da das x aus der prune-Funktion entstanden ist, lässt sich ein y aus O_{12}^* finden, sodass $xy \in (EDT_1 \| EDL_2) \cup (EDL_1 \| EDT_2)$. Es wird nun noch die Einschränkung vorausgesetzt, dass oBdA $xy \in EDT_1 \| EDL_2$, d.h. es existieren $w_1 \in EDT_1$ und $w_2 \in EDL_2$ mit $xy \in w_1 \| w_2$.
Die folgende Argumentation läuft analog zu dem Beweis der Inklusion $ET_{12} \supseteq \text{cont}\,(\,\text{prune}\,((ET_1 \| EL_2) \cup (EL_1 \| ET_2)))$ aus Satz 3.5. Es muss dazu nur jeweils an den Stellen, an denen $PrET(S) \cup MIT(S)$ steht auch noch eine Vereinigung mit $PrDT(S)$ vorgenommen werden. Für Fall I und II ist jeweils kein weiterer Unterfall für v_2' notwendig da, wenn v_2' nicht ausführbar ist, automatisch ein Error-Zustand in der Parallelkomposition entsteht. Somit ist egal, ob auch noch Divergenz vorlag. Falls v_2' ausführbar, ist nicht relevant, ob eine Divergenz-Möglichkeit bestanden hat, da diese nichts an der Ausführbarkeit ändert. Am Ende ist dann auch noch ein Fall für $v_1 \in PrDT_1$ zu ergänzen:

- Fall III ($v_1 \in PrDT_1$): Es existiert ein u_1 aus O_1^*, sodass $q_{01} \overset{v_1}{\Rightarrow} q_1 \overset{u_1}{\Rightarrow} q_1'$ mit $q_1' \in Div_1$ gilt. Da es hier keine disjunkten Inputmengen 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 damit wäre $v_2 = v_2'$. Dieser Fall verläuft jedoch analog zu Fall Ic) und wird somit hier nicht weiter betrachtet. Deshalb gilt für alle im folgenden betrachteten Fälle $q_{02} \overset{v_2'}{\Rightarrow} q_2$ mit $(q_{01}, q_{02}) \overset{v'}{\Rightarrow}$.

  - Fall IIIa) $\left(u_2 \in (O_1 \cap I_2)^*, c \in (O_1 \cap I_2)\text{, sodass } u_2 c \text{ ein Präfix von } u_1|_{I_2} \text{ mit } q_2 \overset{u_2}{\Rightarrow} q_2' \overset{c}{\not\rightarrow}\right)$: Für ein Präfix $u_1' c$ von u_1 mit $u_1' c|_{I_2} = u_2 c$ 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 $v := \text{prune}(v' u_1') \in PrET_{12}$ gewählt, dies ist ein Präfix von v', da $u_1 \in O_1^*$.

– Fall IIIb) $\left(q_2 \overset{u_2}{\Rightarrow} q_2' \text{ mit } u_2 = u_1|_{I_2}\right)$: Somit ist $u_1 \in u_1 \| u_2$ und $(q_1, q_2) \overset{u_1}{\Rightarrow} (q_1', q_2') \in Div_{12}$, da $q_1 \in Div_1$. S_{12} hat also die Divergenz von S_1 geerbt. Es wird nun $v := \text{prune}(v'u_1) \in PrDT_{12}$ gewählt, das wiederum ein Präfix von v' ist.

2. „$\subseteq$“:
Diese Inklusionsrichtung kann analog zum Beweis derselben Inklusionsrichtung des zweiten Punktes von Satz 4.7 gezeigt werden. Es muss dabei nur in der Argumentation die Menge ET_{12} durch die Menge EDT_{12} und die Mengen $QET(S)$ durch die Mengen $QDT(S)$ für die entsprechenden Transitionssysteme S ersetzt werden. Dadurch kann ebenso gefolgert werden, dass im Fall $w \in StQT_{12} \backslash EDT_{12}$ der erreichte Zustand (q_1, q_2) kein Error-Zustand sein kann, da $ET_{12} \subseteq EDT_{12}$ gilt und somit lässt sich auch hier der zweite Punkt von Lemma 4.6 anwenden.

2. „$\supseteq$“:
Es muss wieder danach unterschieden werden, aus welcher Menge das betrachtete Element stammt. Falls w ein Element von EDT_{12} ist, so folgt die Zugehörigkeit zur linken Seite der Gleichung direkt. Somit wird für den weiteren Verlauf dieses Beweises davon ausgegangen, dass $w \in QDT_1 \| QDT_2$ gilt. Für dieses w soll dann gezeigt werden, dass es auch in QDT_{12} enthalten ist. Da $QDT_i = StQT_i \cup EDT_i$ gilt, existieren für w_1 und w_2 mit $w \in w_1 \| w_2$ unterschiedliche Möglichkeiten:

- Fall 1 ($w_1 \in EDT_1 \vee w_2 \in EDT_2$): OBdA gilt $w_1 \in EDT_1$. Es kann nun $w_2 \in StQT_2 \subseteq L_2$ gelten oder $w_2 \in EDT_2 \subseteq EDL_2$ und somit gilt auf jeden Fall $w_2 \in EDL_2$. Daraus kann mit dem ersten Punkt dieses Satzes gefolgert werden, dass $w \in EDT_{12}$ gilt und somit w in der linken Seite der Gleichung enthalten ist.
- Fall 2 ($w_1 \in StQT_1 \backslash EDT_1 \wedge w_2 \in StQT_2 \backslash EDT_2$): Dieser Fall läuft analog zu Fall 2 derselben Inklusionsrichtung des Beweises von Satz 4.7. Hierfür muss die Menge QET_{12} durch QDT_{12} ersetzt werden.

3.:
Durch die Definition ist klar, dass $L_i \subseteq EDT_i$ und $EDT_i \subseteq EDL_i$ gilt. Die Argumentation wird von der rechten Seite der Gleichung aus begonnen:

$$
\begin{aligned}
&(EDL_1 \| EDL_2) \cup EDT_{12} \\
&\quad \overset{5.5}{=} ((L_1 \cup EDT_1) \,\|\, (L_2 \cup EDT_2)) \cup EDT_{12} \\
&\quad = (L_1 \| L_2) \cup \underbrace{(L_1 \| EDT_2)}_{\substack{\subseteq (EDL_1 \| EDT_2) \\ \overset{1.}{\subseteq} EDT_{12}}} \cup \underbrace{(EDT_1 \| L_2)}_{\substack{\subseteq (EDT_1 \| EDL_2) \\ \overset{1.}{\subseteq} EDT_{12}}} \cup \underbrace{(EDT_1 \| EDT_2)}_{\substack{\subseteq (EDL_1 \| EDT_2) \\ \overset{1.}{\subseteq} EDT_{12}}} \cup EDT_{12} \\
&\quad = (L_1 \| L_2) \cup EDT_{12} \\
&\quad \overset{2.5}{=} L_{12} \cup EDT_{12} \\
&\quad \overset{5.5}{=} EDL_{12}.
\end{aligned}
$$

□

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.

Korollar 5.7 (*Divergenz-Präkongurenz*). *Die Relation $\sqsubseteq_{Div}$ 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 aus $EDT_1 \subseteq EDT_2$, $QDT_1 \subseteq QDT_2$ und $EDL_1 \subseteq EDL_2$, sowohl $EDT_{31} \subseteq EDT_{32}$, $QDT_{31} \subseteq QDT_{32}$ als auch $EDL_{31} \subseteq EDL_{32}$ folgt. Dies ergibt sich, wie in den Beweisen zu den Korollaren 3.6 und 4.8, aus der Monotonie von cont, prune und $\cdot\|\cdot$ auf Sprachen wie folgt:

- $$EDT_{31} \overset{5.6\ 1.}{=} \operatorname{cont}(\operatorname{prune}((EDT_3\|EDL_1) \cup (EDL_3\|EDT_1))) \overset{\substack{EDT_1\subseteq EDT_2 \\ \text{und} \\ EDL_1\subseteq EDL_2}}{\subseteq} \operatorname{cont}(\operatorname{prune}((EDT_3\|EDL_2) \cup (EDL_3\|EDT_2))) \overset{5.6\ 2.}{=} EDT_{32},$$
- $$QDT_{31} \overset{5.6\ 2.}{=} (QDT_3\|QDT_1) \cup EDT_{31} \overset{\substack{EDT_{31}\subseteq EDT_{32}, \\ \text{und} \\ QDT_1\subseteq QDT_2}}{\subseteq} (QDT_3\|QDT_2) \cup EDT_{32} \overset{5.6\ 3.}{=} QDT_{32}.$$

□

Als nächstes soll nun eine Verfeinerungsrelation bezüglich guter Kommunikation von Transitionssystemen 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 werden. Die Einschränkung, dass U ein Partner sein muss, kann nicht mehr beibehalten werden, da die Strategie zur Vermeidung von Ruhe im Beweis aus dem letzten Kapitel hier zu Divergenz führen würde. Somit werden für die Ruhe-Vermeidung in diesem Kapitel Aktionen außerhalb der Menge Synch benötigt. Jedoch müssen trotzdem nicht alle komponierbaren EIOs U betrachtet werden. Es kann eine Einschränkung gemacht werden, sodass U fast ein Partner ist. Zur Vereinfachung von umständlichen Formulierungen im Folgenden wird hierfür nun ein neuer Begriff definiert.

Definition 5.8 (*ω-Partner*). *Ein EIO S_1 ist ein* ω-Partner *von einem EIO S_2, wenn $I_1 = O_2$ und $O_1 = I_2 \cup \{\omega\}$ mit $\omega \notin I_2 \cup O_2$ gilt.*

Ein ω-Partner S_1 von S_2 unterscheidet sich von einem Partner von S_2 nur um den Output ω, der nicht in der Menge $\text{Synch}(S_1, S_2)$ enthalten ist.

Lemma 5.9 (*Verfeinerung mit Divergenz-Zuständen*). *Gegeben sind zwei EIOs S_1 und S_2 mit der gleichen Signatur. Wenn $U\|S_1 \sqsubseteq^{\mathrm{B}}_{Div} U\|S_2$ für alle ω-Partner U gilt, dann folgt daraus $S_1 \sqsubseteq_{Div} S_2$.*

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 ω-Partner U gilt $I_U = O$ und $O_U = I \cup \{\omega\}$ mit $\omega \notin I \cup O$.
Um zu zeigen, dass die Relation $S_1 \sqsubseteq_{Div} S_2$ gilt, müssen die folgenden Punkte nachgewiesen werden:

- $EDT_1 \subseteq EDT_2$,
- $QDT_1 \subseteq QDT_2$,
- $EDL_1 \subseteq EDL_2$.

In den Lemmata 3.7 und 4.9 wurde bereits etwas Ähnliches gezeigt. Jedoch kann aufgrund der unterschiedlichen Basisrelation, die zur Anwendung kommen, nichts über dieses Lemma und dessen Gültigkeit ausgesagt werden. Es kann also in diesem Lemma, ebenso wie in Lemma 4.9, aus der lokalen Erreichbarkeit eines Error-Zustandes in S_1' und dem Zusammenhang von $S_1' \sqsubseteq^{\mathrm{B}}_{Div} S_2'$ nur geschlossen werden, dass in S_2' auch ein Fehler lokal erreichbar ist, jedoch kann dieser Fehler hier ein Error-Zustand, ein Ruhe-Zustand oder ein divergenter Zustand sein. Analog verhält es sich, wenn in S_1' ein Divergenz-Zustand oder ein Ruhe-Zustand lokal erreichbar ist.

Als Erstes wird der erste Beweispunkt gezeigt, also die Inklusion $EDT_1 \subseteq EDT_2$.
Es wird für ein präfix-minimales w aus EDT_1 gezeigt, dass dieses w oder eines seiner Präfixe in EDT_2 enthalten ist. Diese Möglichkeit bietet sich, da beide Mengen unter cont abgeschlossen sind.

- Fall 1 ($w = \varepsilon$): Es handelt sich um einen lokal erreichbaren Error- oder Divergenz-Zustand in S_1. Für U wird ein Transitionssystem verwendet, das nur aus dem Startzustand und einer Schleife für alle Inputs $x \in I_U$ und einer Schlinge für ω besteht. Somit kann S_1 im Prinzip die gleichen Error-Zustände bzw. Divergenz-Zustände lokal erreichen wie $U\|S_1$. Draus folgt, dass auch $U\|S_2$ einen lokal erreichbaren Fehler haben muss, da $U\|S_1 \sqsubseteq^{\mathrm{B}}_{Div} U\|S_2$ gilt. Durch den Aufbau von U ist in einer Parallelkomposition mit U kein Ruhe-Zustand möglich. Der Fehler, der in $U\|S_2$ lokal erreichbar ist, muss also ein Error- oder Divergenz-Zustand sein. Da von U kein Error und keine Divergenz geerbt werden kann und durch die Inputschleife auch kein neuer Error enstehen kann, muss der Fehler von U_2 geerbt sein. Somit gilt also, dass in S_2 ein Error- oder Divergenz-Zustand lokal erreichbar ist. Da $EDT(S) = ET(S) \cup DT(S)$ gilt, folgt $w \in EDT_2$.
- 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 ω-Partner U betrachtet (siehe auch Abbildung 5.3):
  - $Q_U = \{q_0, q_1, \dots, q_{n+1}\}$,

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

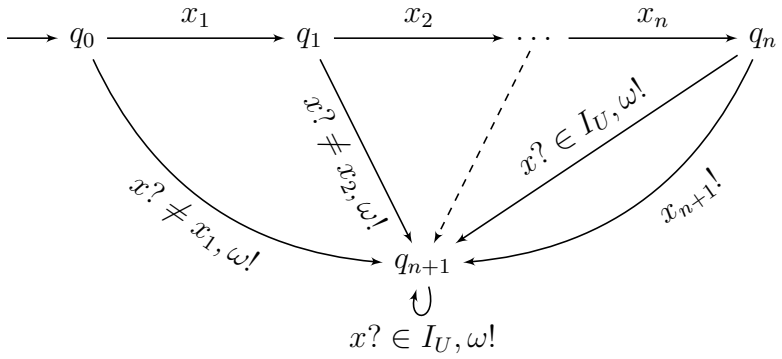

Abbildung 5.3: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$

Die Mengen der Divergenz- und Ruhe-Zustände des hier betrachteten Us sind leer. Da im Vergleiche zum Transitionssystem in Abbildung 3.1 nur die ω-Transitionen zu q_{n+1} ergänzt wurden, ändert sich nichts an dem Fall 2a) im ersten Punkt des Beweises von Lemma 3.7. In Fall 2b) muss die Menge O^* durch $(O \cup \{\omega\})^*$ ersetzt werden. Die Begründungen, wieso in den beiden Fällen $\varepsilon \in PrET(U \| S_1)$ gilt, bleibt also analog zum Beweis von Lemma 3.7. Da nun aber auch Divergenz betrachtet wird, muss ein weiterer Fall ergänzt werden:

- Fall 2c) ($w \in PrDT_1$): In $U \| S_1$ erhält man $(q_0, q_{01}) \stackrel{w}{\Rightarrow} (q_{n+1}, q'') \stackrel{u}{\Rightarrow} (q_{n+1}, q')$ für $u \in (O \cup \{\omega\})^*$ und $q' \in Div_1$. Daraus folgt $(q_{n+1}, q') \in Div_{U \| S_1}$ und somit $wu \in StDT(U \| S_1)$. Da alle Aktionen aus w synchronisiert werden und $I_U \cap I_1 = \emptyset$ gilt $x_1, \ldots, x_n, x_{n+1} \in O_{U \| S_1}$ und, da $u \in (O \cup \{\omega\})^*$, folgt $u \in O^*_{U \| S_1}$. Somit ergibt sich $\varepsilon \in PrDT(U \| S_1)$.

Da ε in $PrET(U \| S_1) \cup PrDT(U \| S_1)$ enthalten ist, kann mit der Relation $U \| S_1 \sqsubseteq^{\mathrm{B}}_{Div} U \| S_2$ geschlossen werden, dass in $U \| S_2$ ein Fehler lokal erreichbar sein muss. Durch die ω-Transitionen an den Zuständen von U kann es in Komposition mit U keine Ruhe-Zustände geben. Der Fehler muss also Error oder Divergenz sein.

- Fall 2i) ($\varepsilon \in ET(U \| S_2)$ wegen neuem Error): Da jeder Zustand von U alle Inputs $x \in I_U = O$ zulässt, muss ein lokal erreichbarer Error-Zustand der Form sein, dass ein Output $a \in O_U \backslash \{\omega\}$ von U möglich ist, der nicht mit einem passenden Input aus S_2 synchronisiert werden kann. Durch die Konstruktion

von U ist in q_{n+1} kein Output außer ω möglich. Ein neuer Error muss also die Form (q_i, q') haben mit $i \leq n, q' \overset{x_{i+1}}{\not\longrightarrow}$ und $x_{i+1} \in O_U \backslash \{\omega\}$. Durch Projektion erhält man dann $q_{02} \overset{x_1 \ldots x_i}{\Longrightarrow} q' \overset{x_{i+1}}{\not\longrightarrow}$ und damit gilt $x_1 \ldots x_{i+1} \in MIT_2 \subseteq ET_2$. Somit ist ein Präfix von w in EDT_2 enthalten.

- Fall 2ii) ($\varepsilon \in ET(U \| S_2)$ wegen geerbtem Error): U hat $x_1 \ldots x_i u$ ausgeführt mit $u \in (O \cup \{\omega\})^*$ und ebenso hat S_2 den Weg $x_1 \ldots x_i u|_{\Sigma_2}$ ausgeführt. Durch dies hat S_2 einen Zustand aus E_2 erreicht, da von U kein Error geerbt werden kann. Es gilt dann $\text{prune}(x_1 \ldots x_i u|_{\Sigma_2}) = \text{prune}(x_1 \ldots x_i) \in PrET_2 \subseteq ET_2$. Da $x_1 \ldots x_i$ ein Präfix von w ist, führt in diesem Fall eine Verlängerung um lokale Aktionen von einem Präfix von w zu einem Error-Zustand. Da ET der Menge aller Verlängerungen von gekürzten Errortraces entspricht, ist $x_1 \ldots x_i$ in EDT_2 enthalten und somit ist ein Präfix von w in EDT_2 enthalten.

- Fall 2iii) ($\varepsilon \in DT(U \| S_2) \backslash ET(U \| S_2)$): Da U nicht unendliche viele Zustände hat und auch keine τ-Schleifen besitzt, kann das Divergenzverhalten nur von S_2 geerbt sein. U hat $x_1 \ldots x_i u$ ausgeführt mit $u \in (O \cup \{\omega\})^*$ und ebenso hat S_2 den Weg $x_1 \ldots x_i u|_{\Sigma_2}$ ausgeführt. Durch dies hat S_2 einen Zustand aus Div_2 erreicht. Es gilt dann $\text{prune}(x_1 \ldots x_i u|_{\Sigma_2}) = \text{prune}(x_1 \ldots x_i) \in PrDT_2 \subseteq DT_2$, da $u|_{\Sigma_2}$ in O^* enthalten ist. Da $x_1 \ldots x_i$ ein Präfix von w ist, führt in diesem Fall eine Verlängerung um lokale Aktionen von einem Präfix von w zu einem divergenten Zustand. Da DT die Menge aller Verlängerungen von gekürzten Divergenztraces ist und $DT_2 \subseteq EDT_2$ gilt, ist in diesem Fall das Präfix $x_1 \ldots x_i$ von w in EDT_2 enthalten.

Als nächstes wird nun der zweite Beweispunkt gezeigt, d.h. die Inklusion $QDT_1 \subseteq QDT_2$. Diese Inklusion kann jedoch noch, anlog zum Beweis der Inklusion der error-gefluteten Sprachen aus dem Error-Kapitel, weiter eingeschränkt werden. Da bereits bekannt ist, dass $EDT_1 \subseteq EDT_2$ gilt, muss nur noch $StQT_1 \backslash EDT_1 \subseteq QDT_2$ gezeigt werden.
Es wird ein $w \in StQT_1 \backslash EDT_1$ gewählt und gezeigt, dass dieses auch in QDT_2 enthalten ist.
Durch die Wahl des ws wird vom Startzustand von S_1 durch das Wort w ein ruhiger Zustand erreicht. Dies hat nur Auswirkungen auf die Parallelkomposition $U \| S_1$, wenn in U ebenfalls ein Ruhe-Zustand durch w erreichbar ist.
Das betrachtete w hat also die Form $w = x_1 \ldots x_n \in \Sigma^*$ mit $n \geq 0$. Es wird der folgende ω-Partner U betrachtet (siehe auch Abbildung 5.4):

- $Q_U = \{q_0, q_1, \ldots, q_n, q\}$,
- $q_{0U} = q_0$,
- $E_U = \emptyset$,

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

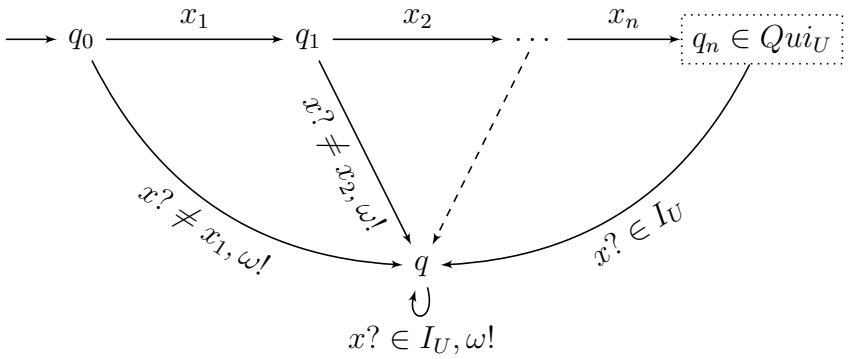

Abbildung 5.4: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$, q_n ist der einzige Ruhe-Zustand

Falls für das betrachtete $w = \varepsilon$ gilt, reduziert sich der ω-Partner U auf den Zustand $q_n = q_0$ und den Zustand q. Es ist also in diesem Fall der Startzustand gleich dem ruhigen Zustand.
Allgemein ist der Zustand q_n aus U ist der einzige ruhige Zustand in U. Es gilt wegen des erstem Punktes von 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 $w \notin EDT_1$ gilt, kann auf dem Weg, der mit w im Transitionssystem S_1 zurück gelegt wird, kein Error- oder Divergenz-Zustand lokal erreicht werden. Es kann also weder von S_1 noch von U Error oder Divergenz auf diesem Weg geerbt werden oder neu 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 Error, Ruhe oder Divergenz handelt.

- Fall a) ($\varepsilon \in ET(U\|S_2)$): Der lokal erreichbare Fehler ist ein Error-Zustand. Das w muss somit nicht ausführbar sein. Der Error kann sowohl von S_2 geerbt sein, wie durch fehlende Synchronisations-Möglichkeiten als neuer Error in der Parallelkomposition entstanden sein. Da nur auf dem Trace w in U Synchronisations-Fehler möglich sind und wegen den Fällen 2i) und 2ii) des ersten Punktes von diesem Beweis ist ein Präfix von w in EDT_2 enthalten. Da die Menge EDT unter cont abgeschlossen ist, gilt auch $w \in EDT_2 \subseteq QDT_2$.

- Fall b) ($\varepsilon \in DT(U\|S_2) \backslash ET(U\|S_2)$): Es handelt sich bei dem lokal erreichbaren Fehler um Divergenz. Der Fehler muss von S_2 geerbt sein, da U keine Divergenz-

Möglichkeiten hat. Es gilt also, dass bereits in S_2 ein Präfix von w in EDT_2 enthalten ist, wegen Fall 2iii) des Beweises des ersten Punktes aus diesem Lemma. Mit dem Abschluss unter cont folgt, dass auch $w \in EDT_2 \subseteq QDT_2$ gilt.

- Fall c) (Ruhe-Zustand lokal erreichbar in $U \| S_2$ und $\varepsilon \notin EDT(U \| S_2)$): 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 dem zweiten Punkt von Lemma 4.6 kann somit gefolgert werden, dass auch in S_2 ein Ruhe-Zustand mit w erreichbar sein muss, da $ET(U \| S_2) \subseteq EDT(U \| S_2)$ ist. Es gilt also $w \in StQT_2 \subseteq QDT_2$.

Nun wird mit dem letzten Punkt des Beweises begonnen. Analog wie in den Beweisen zu den Lemmata 3.7 und 4.9 ist hier jedoch aufgrund der bereits geführten Beweisteile nur noch $L_1 \backslash EDT_1 \subseteq EDL_2$ zu zeigen. Es wird also für ein beliebig gewähltes $w \in L_1 \backslash EDT_1$ gezeigt, dass es auch in EDL_2 enthalten ist.

- Fall 1 ($w = \varepsilon$): Analog zu den Lemmata 3.7 und 4.9 gilt auch hier, dass ε immer in EDL_2 enthalten ist.
- Fall 2 ($w = x_1 \dots x_n$ mit $n \geq 1$): Die Konstruktion des ω-Partners U weicht nur durch die ω-Transition vom Transitionssystem aus dem Beweis der Inklusion der error-gefluteten Sprache EL aus Lemma 3.7 ab. Somit ist der ω-Partner U dann wie folgt definiert (siehe dazu auch Abbildung 5.5):
  - $Q_U = \{q_0, q_1, \dots, q_n, q\}$,
  - $q_{0U} = q_0$,
  - $E_U = \{q_n\}$,
  - $\delta_U = \{(q_i, x_{i+1}, q_{i+1}) \mid 0 \leq i < n\}$
    $\cup \{(q_i, x, q) \mid x \in I_U \backslash \{x_{i+1}\}, 0 \leq i < n\}$
    $\cup \{(q_i, \omega, q) \mid 0 \leq i \leq n\}$
    $\cup \{(q, x, q) \mid x \in I_U \cup \{\omega\}\}$.

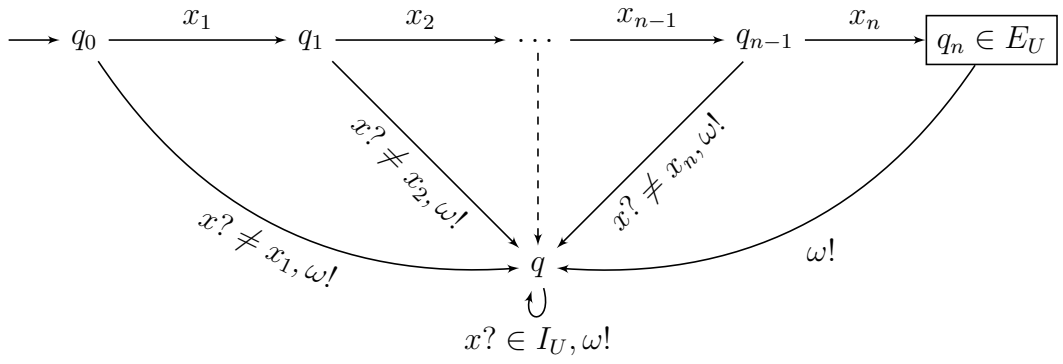

Abbildung 5.5: $x? \neq x_i$ steht für alle $x \in I_U \backslash \{x_i\}$, q_n ist der einzige Error-Zustand

Durch die ω-Transitionen an den Zuständen wird wie oben vermieden, dass es in einer Komposition mit U und auch in U selbst Ruhe-Zustände geben kann. Da $q_{01} \stackrel{w}{\Rightarrow} q_1'$ gilt, kann man schließen, dass $U \| S_1$ einen lokal erreichbaren geerbten Error hat. Somit muss $U \| S_2$ ebenfalls einen lokal erreichbaren Fehler haben. Wie oben bereits erwähnt, kann es sich bei dem Fehler nicht um Ruhe handeln.

- Fall 2a) $\Big($neuer Error aufgrund von $x_i \in O_U \backslash \{\omega\}$ und $q_{02} \stackrel{x_1 \ldots x_{i-1}}{\Longrightarrow} q_2' \stackrel{x_i}{\not\rightarrow}\Big)$: Es gilt $x_1 \ldots x_i \in MIT_2$ und somit $w \in EDL_2$. Anzumerken ist, dass nur auf diesem Weg Outputs von U aus der Menge $\text{Synch}(S_2, U)$ möglich sind, deshalb gibt es keine anderen Outputs von U, die zu einem neuen Error führen könnten.

Die restlichen Fälle sind analog zu Lemma 3.7 möglich. Somit gilt für alle Fälle (2a) bis 2d)), dass w in EDL_2 enthalten ist, da $EL_2 \subseteq EDL_2$ gilt.

- Fall 2e) (Divergenz und kein neuer Error): Da U keine Möglichkeit hat zu divergieren, muss diese Möglichkeit von S_2 geerbt sein. Es gilt dann $q_{02} \stackrel{x_1 \ldots x_i u}{\Longrightarrow} q' \in Div_2$ für $i \geq 0$ und $u \in O^*$. Somit ist $x_1 \ldots x_i u \in StDT_2$ und damit $\text{prune}(x_1 \ldots x_i u) = \text{prune}(x_1 \ldots x_i) \in PrDT_2 \subseteq EDT_2$. Also folgt, dass w in $EDT_2 \subseteq EDL_2$ enthalten ist, da DT unter cont abgeschlossen ist.

□

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

Satz 5.10 (*Vollständige Abstraktheit für Divergenz-Semantik*). *Seinen S_1 und S_2 zwei EIOs mit derselben Signatur. Dann gilt $S_1 \sqsubseteq_{Div}^{\text{C}} S_2 \Leftrightarrow S_1 \sqsubseteq_{Div} S_2$.*

Beweis. „⇐“: Nach Definition gilt $w \in QDT(S)$ mit $w \in O(S)^*$ genau dann, wenn in S ein Divergenz-, Ruhe- oder Error-Zustand lokal erreichbar ist. $S_1 \sqsubseteq_{Div} S_2$ impliziert, dass $w \in QDT_2$ gilt, wenn $w \in QDT_1$ gilt. Somit ist ein Divergenz-, Ruhe- oder Error-Zustand nur dann in S_1 lokal erreichbar, wenn auch ein solcher in S_2 lokal erreichbar ist. Daraus folgt, dass $S_1 \sqsubseteq_{Div}^{\text{B}} S_2$ gilt. Es ist also $\sqsubseteq_{Div}$ in $\sqsubseteq_{Div}^{\text{B}}$ enthalten. In Korollar 5.7 wurde festgestellt, dass $\sqsubseteq_{Div}$ eine Präkongruenz ist. Da jedoch $\sqsubseteq_{Div}^{\text{C}}$ nach Definition 5.3 die gröbste Präkongruenz bezüglich $\cdot \| \cdot$ ist, die in $\sqsubseteq_{Div}^{\text{B}}$ enthalten ist, muss $\sqsubseteq_{Div}$ in $\sqsubseteq_{Div}^{\text{C}}$ enthalten sein. Es folgt also aus $S_1 \sqsubseteq_{Div} S_2$, dass auch der Zusammenhang $S_1 \sqsubseteq_{Div}^{\text{C}} S_2$ gilt.

„⇒“: Durch die Definition von $\sqsubseteq_{Div}^{\text{C}}$ als Präkongruenz in 5.3 folgt aus $S_1 \sqsubseteq_{Div}^{\text{C}} S_2$, dass $U \| S_1 \sqsubseteq_{Div}^{\text{C}} U \| S_2$ für alle EIOs U gilt, die mit S_1 komponierbar sind. Da $\sqsubseteq_{Div}^{\text{C}}$ nach Definition in $\sqsubseteq_{Div}^{\text{B}}$ enthalten ist, folgt auch die Gültigkeit von $U \| S_1 \sqsubseteq_{Div}^{\text{B}} U \| S_2$ für alle diese EIOs U. Mit Lemma 5.9 folgt dann $S_1 \sqsubseteq_{Div} S_2$. □

Es wurde somit, wie in den letzten beiden Kapiteln, eine Kette an Folgerungen gezeigt, die sich zu einem Ring schließen. Jedoch ändert sich an der Begründung für einen der Folgepfeile etwas, da in Lemma 5.9 U kein Partner mehr ist, sondern nur ein ω-Partner. Diese Folgerungskette ist in Abbildung 5.6 dargestellt.

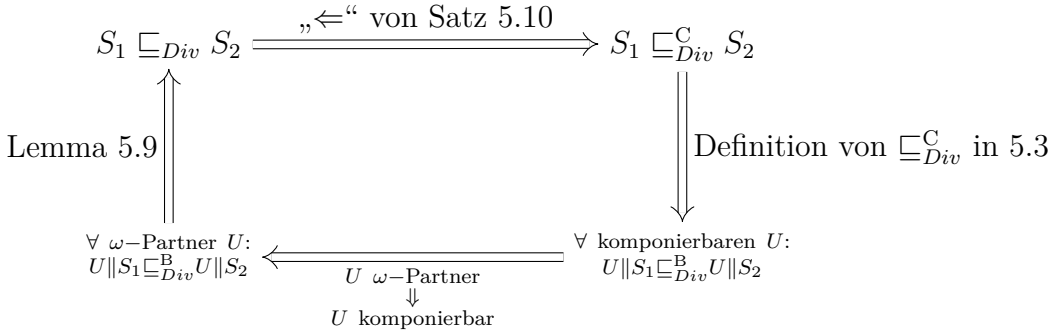

Abbildung 5.6: Folgerungskette

Angenommen man definiert, dass S_1 S_2 verfeinern soll genau dann, wenn für alle ω-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.
Aus Satz 5.10 und Lemma 5.9 erhält man das folgende Korollar.

Korollar 5.11. *Es gilt:* $S_1 \sqsubseteq_{Div} S_2 \Leftrightarrow U\|S_1 \sqsubseteq^{\mathrm{B}}_{Div} U\|S_2$ *für alle komponierbaren* U*.*

Um die gröbste Präkongruenz mit den gewünschten Eigenschaften zu erhalten, wurde hier eine Änderung gegenüber [CJK14] vorgenommen. Es war nötig auf die Verfeinerung der Error-Präkongruenz zu verzichtet und es mussten auf Trace-Ebene Error und Divergenz vermischt werden.
Falls man eine Basisrelation definieren würde, die unterscheiden kann, ob es sich um Error oder Divergenz handelt, könnte man die hier erhaltenen Ergebnisse so verändern, dass sie dem Ansatz entsprechen würden den [CJK14] verwendet hat. Es wäre also möglich, dass man nicht nur die Basisrelation $\sqsubseteq^{\mathrm{B}}_{Div}$ als erfüllt voraussetzt, sondern zusätzlich auch noch die Basisrelation $\sqsubseteq^{\mathrm{B}}_{E}$. Somit wäre garantiert, dass man die beiden Systeme S_1 (siehe dazu Abbildung 5.1) und S_2 (siehe dazu Abbildung 5.2) aus dem Beispiel oben unter der Verfeinerungsrelation unterscheiden könnte. Die Definition für die zu verwendende Basisrelation müsste dann wie folgt lauten.

Definition 5.12 (*alternative Divergenz-Verfeinerungs-Basisrelation*). *Für zwei EIOs* S_1 *und* S_2 *mit der gleichen Signatur wird* $S_1 \sqsubseteq^{\mathrm{B}}_{Div_{\mathrm{alt}}} S_2$ *geschrieben, wenn ein Error-Zustand in* S_1 *nur dann lokal erreichbar ist, wenn er auch in* S_2 *erreichbar ist und wenn ein Divergenz- oder Ruhe-Zustand in* S_1 *nur dann lokal erreichbar ist, wenn auch in* S_2 *ein Error-, Divergenz- oder Ruhe-Zustand lokal erreichbar ist. Die* alternative Basisrelation *stellt ebenfalls eine Verfeinerung bezüglich* Error, Ruhe *und* Divergenz *dar.*

$\sqsubseteq^{\mathrm{C}}_{Div_{\mathrm{alt}}}$ *bezeichnet die* alternative vollständig abstrakte Präkongruenz *von* $\sqsubseteq^{\mathrm{B}}_{Div_{\mathrm{alt}}}$ *bezüglich* $\cdot \| \cdot$.

Die Relation $\sqsubseteq^{\mathrm{B}}_{Div_{\mathrm{alt}}}$ ist somit die Vereinigung der Relationen $\sqsubseteq^{\mathrm{B}}_{E}$ und $\sqsubseteq^{\mathrm{B}}_{Div}$. Die Forderung von $\sqsubseteq^{\mathrm{B}}_{Div}$ für die lokale Erreichbarkeit von Error-Zuständen in S_1 wird bereits durch die stärkere Forderung von $\sqsubseteq^{\mathrm{B}}_{E}$ für die lokale Erreichbarkeit von Error-Zuständen erfüllt und wurde deshalb in der obigen Definition nicht explizit aufgeführt. Die Relation $\sqsubseteq^{\mathrm{C}}_{Div_{\mathrm{alt}}}$ könnte dann durch die im folgenden definierte Relation charakterisiert werden.

Definition 5.13 (*alternative Charakterisierung der volltändig abstrakten Präkonrugenz*). *Für zwei EIOs S_1, S_2 mit der gleichen Signatur schreibt man $S_1 \sqsubseteq_{Div_{\mathrm{alt}}} S_2$, wenn $S_1 \sqsubseteq_E S_2$, $EDT_1 \subseteq EDT_2$ und $QDT_1 \subseteq QDT_2$ gilt.*

Diese Charakterisierung $\sqsubseteq_{Div_{\mathrm{alt}}}$ entspricht dem Ansatz aus [CJK14]. Dadurch wird jedoch Error als „schlimmerer Fehler“ aufgefasst wie Divergenz. Es gibt mit diesen Definitionen also eine echte Hierarchie zwischen den Fehler. Es sieht so aus, dass die Mengeninklusion $EDL_1 \subseteq EDL_2$ fehlen würde, damit $\sqsubseteq_{Div_{\mathrm{alt}}}$ die Vereinigung von $\sqsubseteq_E$ und $\sqsubseteq_{Div}$ wäre. Dies ist jedoch nicht der Fall, da diese Inklusion aus den geforderten Inklusionen bereits folgt. Durch $S_1 \sqsubseteq_E S_2$ weiß man, dass $EL_1 \subseteq EL_2$ gilt. Die Menge EL ist nach Definition die Vereinigung aus der Sprache L und den Errortraces ET. Somit fehlt nur noch die Divergenztraces DT, um daraus die Menge EDL zu erhalten. Jedoch ist durch die Inklusion $EDT_1 \subseteq EDT_2$ bereits klar, dass $ET_1 \cup DT_1 \subseteq ET_2 \cup DT_2$ gilt. Insgesamt folgt dann die Gleichung $EDL = EL \cup EDT$. Falls sich eine der beiden Mengen EL bzw. EDT vergrößert, vergrößert sich auch die Menge EDL echt. Somit ist also $\sqsubseteq_{Div_{\mathrm{alt}}}$ die Vereinigung von $\sqsubseteq_E$ und $\sqsubseteq_{Div}$.

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 bereits lokal ein divergenter Zustand aus 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 unter Internalisierung auf keinen Fall erhalten bleiben.
Außerdem vergrößern sich durch die zusätzlichen Divergenz-Zustände alle Trace-Mengen, die in der Präkongruenz $\sqsubseteq_{Div}$ betrachtet werden. Somit würde die Untersuchung, ob die Relation erhalten bleibt, unter Hiding deutlich aufwendiger werden. Deshalb soll dies nicht mehr in dieser Arbeit betrachtet werden.

Literaturverzeichnis

[BV15] Ferenc Bujtor und Walter Vogler, *Error-Pruning in Interface Automata*, Theoretical Computer Science **597** (2015), 18–39.

[CJK14] Chris Chilton, Bengt Jonsson, und Marta Z. Kwiatkowska, *An Algebraic Theory of Interface Automata*, Theoretical Computer Science **549** (2014), 146–174.

[DAH05] Luca De Alfaro und Thomas A Henzinger, *Interface-based design*, Engineering Theories of Software Intensive Systems, Springer Netherlands, 2005, pp. 83–104.

[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.