

# 1 Definitionen

Stand: 24. Juli 2017

Kombination aus [BV15] und [Sch16] mit Einflüssen von [BFLV16]:

**Definition 1.1 (*Modal Error-I/O-Transitionssystem*).** Ein Modal Error-I/O-Transitionssystem (MEIO) ist ein Tupel  $(P, I, O, \longrightarrow, \dashrightarrow, p_0, E)$  mit:

- $P$ : Menge der Zustände,
- $p_0 \in P$ : Startzustand,
- $I, O$ : disjunkte Mengen der (sichtbaren) Input- und Output-Aktionen,
- $\longrightarrow \subseteq P \times \Sigma_\tau \times P$ : must-Transitions-Relation,
- $\dashrightarrow \subseteq P \times \Sigma_\tau \times P$ : may-Transitions-Relation,
- $E \subseteq P$ : Menge der Fehler-Zustände.

Es wird vorausgesetzt, dass  $\longrightarrow \subseteq \dashrightarrow$  (syntaktische Konsistenz) gilt.

Das Alphabet bzw. die Aktionsmenge eines MEIO ist  $\Sigma = I \cup O$ . Die interne Aktion  $\tau$  ist nicht in  $\Sigma$  enthalten. Jedoch wird  $\Sigma_\tau := \Sigma \cup \{\tau\}$  definiert. Die Signatur eines MEIOs entspricht  $\text{Sig}(P) = (I, O)$ .

Falls  $\longrightarrow = \dashrightarrow$  gilt, wird  $P$  auch Implementierung genannt.

Implementierungen entsprechen den in [Sch16] behandelten EIOs.

Must-Transitions sind Transitions, die von einer Verfeinerung implementiert werden müssen. Die may-Transitions sind hingegen die zulässigen Transitions für eine Verfeinerung.

MEIOs werden in dieser Arbeit durch ihre Zustandsmenge (z.B.  $P$ ) identifiziert und falls notwendig werden damit auch die Komponenten indiziert (z.B.  $I_P$  anstatt  $I$ ). Falls das MEIO selbst bereits einen Index hat (z.B.  $P_1$ ) kann an der Komponente die Zustandsmenge als Index wegfallen und nur noch der Index des gesamten Transitionssystems verwendet werden (z.B.  $I_1$  anstatt  $I_{P_1}$ ). Zusätzlich stehen  $i, o, a, \omega$  und  $\alpha$  für Buchstaben aus den Alphabeten  $I, O, \Sigma, O \cup \{\tau\}$  und  $\Sigma_\tau$ .

Es wird die Notation  $p \xrightarrow{\alpha} p'$  für  $(p, \alpha, p') \in \dashrightarrow$  und  $p \xrightarrow{\alpha}$  für  $\exists p' : (p, \alpha, p') \in \dashrightarrow$  verwendet. Dies kann entsprechend auf Buchstaben-Sequenzen  $w \in \Sigma_\tau^*$  erweitert werden zu  $p \xrightarrow{w} p'$  ( $p \xrightarrow{w}$ ) steht für die Existenz eines Laufes  $p \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} \dots p_{n-1} \xrightarrow{\alpha_n} p'$  ( $p \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} \dots p_{n-1} \xrightarrow{\alpha_n}$ ) mit  $w = \alpha_1 \dots \alpha_n$ .

## 1 Definitionen

Desweiteren soll  $w|_B$  die Aktions-Sequenz bezeichnen, die man erhält, wenn man aus  $w$  alle Aktionen löscht, die nicht in  $B \subseteq \Sigma$  enthalten sind.  $\hat{w}$  steht für  $w|_\Sigma$ . Es wir  $p \stackrel{w}{\Rightarrow} p'$  für ein  $w \in \Sigma^*$  geschrieben, falls  $\exists w' \in \Sigma_\tau^* : \hat{w}' = w \wedge p \xrightarrow{w'} p'$ , und  $p \stackrel{w}{\Rightarrow} p'$  für ein beliebiges  $p'$  gilt. Falls  $p_0 \stackrel{w}{\Rightarrow} p$  gilt, dann wird  $w$  *Trace* genannt und  $p$  ist ein *erreichbarer Zustand*.

Analog zu  $\xrightarrow{\quad}$  und  $\Rightarrow$  werden  $\longrightarrow$  und  $\Longrightarrow$  für die entsprechenden Relationen der must-Transition verwendet.

Outputs und die interne Aktion werden *lokale Aktionen* genannt, da sie lokal vom ausführenden MEIO kontrolliert sind. Um eine Erleichterung der Notation zu erhalten, soll gelten, dass  $p \xrightarrow{a} p'$  und  $p \xrightarrow{a} p'$  für  $\nexists p' : p \xrightarrow{a} p'$  und  $\nexists p' : p \xrightarrow{a} p'$  stehen soll.  $p \xrightarrow{a} p' \stackrel{\varepsilon}{\Rightarrow} p''$  wird geschrieben, wenn sowohl ein  $p'$  wie auch ein  $p''$  existiert, so dass  $p \xrightarrow{a} p' \stackrel{\varepsilon}{\Rightarrow} p''$  gilt. Diese Transition wird auch als *schwache-nachlaufende must-Transition* bezeichnet. Entsprechen steht  $\xrightarrow{a} \stackrel{\varepsilon}{\Rightarrow}$  für die *schwache-nachlaufende may-Transition*.

In Graphiken wird eine Aktion  $a$  als  $a?$  notiert, falls  $a \in I$  und  $a!$ , falls  $a \in O$ . Must-Transitionen (may-Transitionen) werden als durchgezogener Pfeil gezeichnet (gestrichelter Pfeil). Entsprechend der syntaktischen Konsistenz repräsentiert jede gezeichnete must-Transition auch gleichzeitig die zugrundeliegende may-Transitionen.

**Definition 1.2 (Parallelkomposition).** Zwei MEIOs  $P_1 = (P_1, I_1, O_1, \longrightarrow_1, \xrightarrow{\quad}_1, p_{01}, E_1)$  und  $P_2 = (P_2, I_2, O_2, \longrightarrow_2, \xrightarrow{\quad}_2, p_{02}, E_2)$  sind komponierbar, falls  $O_1 \cap O_2 = \emptyset$ . Für solche MEIOs ist die Parallelkomposition  $P_{12} := P_1 \parallel P_2 = ((P_1 \times P_2), I, O, \longrightarrow_{12}, \xrightarrow{\quad}_{12}, (p_{01}, p_{02}), E)$  definiert mit:

- $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2),$
- $O = (O_1 \cup O_2),$
- $\longrightarrow_{12} = \left\{ ((p_1, p_2), \alpha, (p'_1, p'_2)) \mid p_1 \xrightarrow{\alpha}_1 p'_1, \alpha \in \Sigma_\tau \setminus \text{Synch}(P_1, P_2) \right\} \\ \cup \left\{ ((p_1, p_2), \alpha, (p_1, p'_2)) \mid p_2 \xrightarrow{\alpha}_2 p'_2, \alpha \in \Sigma_\tau \setminus \text{Synch}(P_1, P_2) \right\} \\ \cup \left\{ ((p_1, p_2), \alpha, (p'_1, p'_2)) \mid p_1 \xrightarrow{\alpha}_1 p'_1, p_2 \xrightarrow{\alpha}_2 p'_2, \alpha \in \text{Synch}(P_1, P_2) \right\},$
- $\xrightarrow{\quad}_{12} = \left\{ ((p_1, p_2), \alpha, (p'_1, p'_2)) \mid p_1 \xrightarrow{\alpha}_1 p'_1, \alpha \in \Sigma_\tau \setminus \text{Synch}(P_1, P_2) \right\} \\ \cup \left\{ ((p_1, p_2), \alpha, (p_1, p'_2)) \mid p_2 \xrightarrow{\alpha}_2 p'_2, \alpha \in \Sigma_\tau \setminus \text{Synch}(P_1, P_2) \right\} \\ \cup \left\{ ((p_1, p_2), \alpha, (p'_1, p'_2)) \mid p_1 \xrightarrow{\alpha}_1 p'_1, p_2 \xrightarrow{\alpha}_2 p'_2, \alpha \in \text{Synch}(P_1, P_2) \right\},$
- $E = (P_1 \times E_2) \cup (E_1 \times P_2)$  *geerbte Fehler*  
 $\left. \begin{array}{l} \cup \left\{ (p_1, p_2) \mid \exists a \in O_1 \cap I_2 : p_1 \xrightarrow{a}_1 \wedge p_2 \xrightarrow{a}_2 \right\} \\ \cup \left\{ (p_1, p_2) \mid \exists a \in I_1 \cap O_2 : p_1 \xrightarrow{a}_1 \wedge p_2 \xrightarrow{a}_2 \right\} \end{array} \right\}$  *neue Kommunikationsfehler.*

Dabei bezeichnet  $\text{Synch}(P_1, P_2) = (I_1 \cap O_2) \cup (O_1 \cap I_2) \cup (I_1 \cap I_2)$  die Menge der zu synchronisierenden Aktionen. Die synchronisierten Aktionen werden als Output bzw. Input der Komposition beibehalten.

$P_1$  wird Partner von  $P_2$  genannt, wenn die Parallelkomposition von  $P_1$  und  $P_2$  geschlossen ist, d.h. sie duale Signaturen  $\text{Sig}(P_1) = (I, O)$  und  $\text{Sig}(P_2) = (O, I)$  haben.

Ein neuer Kommunikationsfehler entsteht, wenn eines der MEIOs die Möglichkeit für einen Output hat (may-Transition) und das andere MEIO den passenden Input nicht erzwingt (keine must-Transition vorhanden). Es muss also in möglichen Implementierungen nicht wirklich zu diesem Kommunikationsfehler kommen, da die Output-Transition nicht zwingendermaßen implementiert werden muss und die Input-Transition durch eine may-Transition trotzdem erlaubt sein kann.

Wie bereits in [Sch16] kann es durch die Synchronisation von Inputs zu keinen neuen Kommunikationsfehler kommen, da dies in beiden Transitionssystemen keine lokal kontrollierte Aktion ist. Falls jedoch nur eines der Transitionssysteme die Möglichkeit für einen Input hat, der synchronisiert wird, besteht diese Möglichkeit in der Parallelkomposition nicht mehr. Es kann also in der Kommunikation mit einem weiteren MEIO dort zu einen neuen Kommunikationsfehler kommen.

**Definition 1.3 ((starke) Simulation).** Eine (starke) alternierende Simulation ist eine Relation  $R \subseteq P \times Q$  auf zwei MEIOs  $P$  und  $Q$ , falls für alle  $(p, q) \in R$  gilt:

1.  $q \xrightarrow{\alpha}_Q q'$  impliziert  $p \xrightarrow{\alpha}_P p'$  für ein  $p'$  mit  $p' R q'$ ,
2.  $p \xrightarrow{\alpha}_P p'$  impliziert  $q \xrightarrow{\alpha}_Q q'$  für ein  $q'$  mit  $p' R q'$ ,
3.  $p \in E_P \Rightarrow q \in E_Q$ .

Die Vereinigung  $\sqsubseteq_{\text{as}}$  aller dieser Relationen wird als (starke) as-Verfeinerung(-s Relation) (auch modal Verfeinerung) bezeichnet. Es wird  $P \sqsubseteq_{\text{as}} Q$  geschrieben, falls  $p_0 \sqsubseteq_{\text{as}} q_0$  gilt, und  $P$  as-verfeinert  $Q$  (stark) oder  $P$  ist eine (starke) as-Verfeinerung von  $Q$ .

Für ein MEIO  $Q$  und eine Implementierung  $P$  mit  $P \sqsubseteq_{\text{as}} Q$ , ist  $P$  eine as-Implementierung von  $Q$  und es wird  $\text{as-impl}(Q)$  für die Menge aller as-Implementierungen von  $Q$  verwendet.

**Definition 1.4 (schwache Simulation).** Eine schwache alternierende Simulation ist eine Relation  $R \subseteq P \times Q$  auf zwei MEIOs  $P$  und  $Q$ , falls für alle  $(p, q) \in R$  gilt:

1.  $q \xrightarrow{i}_Q q'$  impliziert  $p \xrightarrow{i}_P \xRightarrow{\varepsilon}_P p'$  für ein  $p'$  mit  $p' R q'$ ,
2.  $q \xrightarrow{\omega}_Q q'$  impliziert  $p \xRightarrow{\hat{\omega}}_P p'$  für ein  $p'$  mit  $p' R q'$ ,
3.  $p \xrightarrow{i}_P p'$  impliziert  $q \xrightarrow{i}_Q \xRightarrow{\varepsilon}_Q q'$  für ein  $q'$  mit  $p' R q'$ ,
4.  $p \xrightarrow{\omega}_P p'$  impliziert  $q \xRightarrow{\hat{\omega}}_Q q'$  für ein  $q'$  mit  $p' R q'$ ,
5.  $p \in E_P \Rightarrow q \in E_Q$ .

Wobei  $i \in I$  und  $\omega \in O \cup \{\tau\}$ .

Analog zur starken alternierenden Simulation, wird hier  $\sqsubseteq_{\text{w-as}}$  als Relationssymbol verwendet und man kann auch entsprechend schwache as-Verfeinerung betrachten.

Ebenso kann  $\sqsubseteq_{\text{w-as}}$  für ein MEIO  $Q$  und eine Implementierung  $P$  definiert werden mit

## 1 Definitionen

$P \sqsubseteq_{w\text{-as}} Q$ , ist  $P$  eine  $w$ -as-Implementierung von  $Q$  und es wird  $w\text{-as-impl}(Q)$  für die Menge aller  $w$ -as-Implementierungen von  $Q$  verwendet.

Die schwache Simulation erlaubt interne Aktionen beim MEIO, das die entsprechende Aktion matchen muss. Jedoch ist es zwingen notwendig, dass ein Input sofort aufgeführt wird und erst dann interne Aktinen möglich sind. Da ein Input die Reaktion auf eine Aktion ist, die die Umwelt auslöst und die nicht auf das Transitionssystem warten kann. Outputs hingegen können auch verzögert werden, da die Umgebung dies dann als Inputs aufnimmt und für diese somit nicht lokal kontrolliert ist.

Die Parallelkomposition von Wörtern und Mengen kann aus [Sch16] übernommen werden.

**Definition 1.5 (*Parallelkomposition auf Traces*).**

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

Ebenso können die Definitionen der Funktionen `prune` und `cont` zum Abschneiden und Verlängern von Traces aus [Sch16] übernommen werden. Hierbei ist zu beachten, dass in dieser Arbeit  $\varepsilon$  das leere Wort und  $\mathfrak{P}(M)$  die Potenzmenge der Menge  $M$  bezeichnet.

**Definition 1.6 (*Pruning- und Fortsetzungs-Funktion*).**

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

**Definition 1.7 (*Sprache*).** Die Sprache eines MEIOs  $P$  ist  $L(P) := \{w \in \Sigma^* \mid p_0 \xRightarrow{w}_P\}$ .

## 2 allgemeine Folgerungen

**Proposition 2.1 (*Sprache und Implementierung*).** Die (maximale) Sprache eines MEIOs  $P$  ist  $L(P) = \{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'}\}$ .

*Beweis.* Für ein  $w \in L(P)$  gilt nach Definition 1.7 und den Definitionen der Transitions-Notation  $\exists p_1, p_2, \dots, p_{n-1}, p' \exists w' \in \Sigma^* : \hat{w}' = w \wedge w' = \alpha_1 \alpha_2 \dots \alpha_n \wedge p_0 \xrightarrow{\alpha_1}_P p_1 \xrightarrow{\alpha_2}_P \dots p_{n-1} \xrightarrow{\alpha_n}_P p'$ . Für ein  $w$  aus  $\{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'}\}$  gilt, für ein  $P' \in \text{as-impl}(P)$  das analoge nur mit must- anstatt may-Transitionen.

Aufgrund von Definition 1.3 2. kann jedes Element aus  $\text{as-impl}(P)$  nur die bereits in  $P$  vorhandenen may-Transitionen implementieren. Somit gibt es für jedes  $w$ , dass in der Sprache einer as-Implementierung von  $P$  enthalten ist auch ein entsprechendes  $w \in L(P)$  mit einem Trace wie oben.

Da in  $\{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'}\}$  alle Wörter enthalten sind, für dies es eine as-Implementierung gibt, die dieses Wort ausführen kann, werden somit alle möglichen as-Implementierungen betrachtet. Jede may-Transition aus  $P$  wird von mindestens einem  $P' \in \text{as-impl}(P)$  als must-Transition implementiert. Deshalb sind auch alle Wörter, die in  $L(P)$  enthalten sind in der Menge der Wörter alle as-Implementierungen von  $P$  enthalten.

Da für Implementierungen die must-Transitions-Relations-Menge die gleiche ist, wie die Menge der may-Transitions-Relationen könnte man auch für die as-Implementierungen die Definition 1.7 anwenden um die jeweilige Sprache zu bestimmen. Die Sprache eines MEIO entspricht dann der Vereinigung der Sprachen seiner as-Implementierungen.  $\square$

**Proposition 2.2 (*Sprache der Parallelkomposition*).** Für zwei komponierbare MEIOs  $P_1$  und  $P_2$  gilt:  $L_{12} := L(P_{12}) = L_1 \parallel L_2$ .

*Beweis.* Jedes Wort, dass in  $L_{12}$  enthalten ist, kann auf  $P_1$  und  $P_2$  projiziert werden und die Projektionen sind dann in  $L_1$  und  $L_2$  enthalten. In einer Parallelkomposition werden die Wörter der beiden MEIOs gemeinsam ausgeführt, falls es sich um synchronisierte Aktionen handelt, und verschränkt sequenziell, wenn es sich um unsynchronisierte Aktionen handelt. Somit sind alle Wörter aus  $L_1 \parallel L_2$  auch Wörter der Parallelkomposition  $L(P_{12})$ .  $\square$

**Lemma 2.3 (*as-Implementierungen und Parallelkomposition*).**

$P'_1 \in \text{as-impl}(P_1) \wedge P'_2 \in \text{as-impl}(P_2) \Rightarrow (P'_1 \parallel P'_2) \in \text{as-impl}(P_1 \parallel P_2)$ .

## 2 allgemeine Folgerungen

*Beweis.* Es gelte  $j \in \{1, 2\}$ . Da  $P'_j \in \text{as-impl}(P_j)$  gilt, gibt es nach Definition 1.3 eine as-Verfeinerungs-Relation  $\mathcal{R}_j$ , die beschreibt, wie  $P'_j$   $P_j$  verfeinert. Die Parallelkomposition werden auf Basis von Definition 1.2 gebildet. Die Zustände sind also Tupel der Zustände der Komponenten. In dem man aus den Zuständen, die die  $\mathcal{R}_j$  in Relation setzt auch solche Tupel zusammensetzt, kann man auch eine neue as-Verfeinerungs-Relation für die Verfeinerung von  $P_1 \parallel P_2$  durch  $P'_1 \parallel P'_2$  erstellen. Die neue as-Verfeinerungs-Relation soll  $\mathcal{R}_{12}$  heißen und wie folgt definiert sein:  $\forall p'_1, p'_2, p_1, p_2 : ((p'_1, p'_2), (p_1, p_2)) \in \mathcal{R}_{12} \Leftrightarrow (p'_1, p_1) \in \mathcal{R}_1 \wedge (p'_2, p_2) \in \mathcal{R}_2$ . Es bleibt nun zu zeigen, dass  $\mathcal{R}_{12}$  eine zulässige Verfeinerungs-Relation nach Definition 1.3 ist, da die Parallelkomposition von zwei Implementierungen auch immer eine Implementierung ist (1.2). Für die folgenden Fälle soll  $(p_1, p_2)\mathcal{R}_{12}(p'_1, p'_2)$  vorausgesetzt sein.

1. Für diesen Punkt der Simulations-Definition 1.3 ist folgendes zu zeigen:  $(p_1, p_2) \xrightarrow{\alpha}_{12} (q_1, q_2)$  impliziert  $(p'_1, p'_2) \xrightarrow{\alpha}_{P'_1 \parallel P'_2} (q'_1, q'_2)$  für ein  $(q'_1, q'_2)$  mit  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$ .

$(p_1, p_2) \xrightarrow{\alpha}_{12} (q_1, q_2)$  kann in  $P_1 \parallel P_2$  für ein  $\alpha$  aus  $\text{Synch}(P_1, P_2)$  nur gelten, wenn in  $P_1$  die  $\alpha$ -Transition zwischen  $p_1$  und  $q_1$  auch bereits eine must-Transition war und analog für  $p_2$  und  $q_2$  in  $P_2$ . Somit erzwingen die  $\mathcal{R}_j$  für die Komponenten bereits die Implementierung der must-Transitions, so dass es dann entsprechende  $(q_j, q'_j) \in \mathcal{R}_j$  gibt. Die Parallelkomposition der implementierten must-Transitionen aus  $P'_1$  und  $P'_2$  führt in  $P'_1 \parallel P'_2$  zu der geforderten Transition. Falls  $\alpha$  keine synchronisierte Aktion ist, enthält die Parallelkomposition die Transition nur, da eine Komponente diese Transition alleine ausführen kann (ersten beiden Zeilen der  $\xrightarrow{\alpha}_{12}$  Definition in 1.2). OBdA  $p_1 \xrightarrow{\alpha}_1 q_1$  und somit gilt  $p_2 = q_2$ . Da  $\mathcal{R}_1$  eine as-Verfeinerungs-Relation ist, gibt es in  $P'_1$  zwischen  $p'_1$  und  $q'_1$  eine must-Transition und es gilt  $(q_1, q'_1) \in \mathcal{R}_1$ .  $\alpha$  ist auch in der Parallelkomposition der as-Implementierungen eine unsynchronisierte Aktion und somit entsteht die Transition dort auch nur aus der Transition von  $P'_1$  und es gilt  $p'_2 = q'_2$ .

Da in beiden Fällen  $(q_j, q'_j) \in \mathcal{R}_j$  für beide Werte von  $i$  gilt, gilt nach der Definition von  $\mathcal{R}_{12}$  auch  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$ .  $p_2 \mathcal{R}_2 p'_2$  muss nach Voraussetzung gelten und somit gilt wegen der Gleichheiten der Zustände auch  $q_2 \mathcal{R}_2 q'_2$ .

2. Es ist  $(p'_1, p'_2) \xrightarrow{\alpha}_{P'_1 \parallel P'_2} (q'_1, q'_2)$  impliziert  $(p_1, p_2) \xrightarrow{\alpha}_{12} (q_1, q_2)$  für ein  $(q_1, q_2)$  mit  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$  für diesen Punkt zu zeigen.

Die Argumentation könnte wie in Punkt 1. erneut in unsynchronisierte und synchronisierte Aktionen gesplittet werden. Jedoch würde man dadurch nur auf das Ergebnis kommen, dass die eine Komponente oder beide die entsprechende may-Transition ausführen können müssen, damit die Parallelkomposition dies auch kann. Es gilt also mindestens für eine der Komponenten  $p'_j \xrightarrow{\alpha}_{P'_j} q'_j$  und durch die Definition 1.3, die für die entsprechende Relation  $\mathcal{R}_j$  gilt, muss es in  $P_j$  die Transition  $p_j \xrightarrow{\alpha}_j q_j$  geben, so dass dann  $(q_j, q'_j) \in \mathcal{R}_j$  gilt. Durch die Definition von  $\xrightarrow{\alpha}_{12}$  in 1.2 werden die may-Transitions der Komponenten entsprechend in die Parallelkomposition  $P_1 \parallel P_2$  aufgenommen und die Relation  $\mathcal{R}_{12}$  gilt, mit den analogen Begründungen wie in 1., auch für die entsprechenden Zustands-Tupel.

## 2 allgemeine Folgerungen

3. Hierfür muss gezeigt werden, wenn  $(p'_1, p'_2) \in E_{P'_1 \parallel P'_2}$  gilt, dann ist auch  $(p_1, p_2)$  in  $P_1 \parallel P_2$  ein Zustand, der in der Menge der Fehler-Zustände  $E_{12}$  enthalten ist. Falls  $(p'_1, p'_2)$  ein geerbter Fehler ist, dann ist oBdA  $p'_1 \in E_{P'_1}$  und  $p'_2 \in P'_2$ . Aufgrund von  $\mathcal{R}_1$  und Definition 1.3 3. muss dann auch  $p_1 \in E_1$  gelten. Für  $p_2$  gilt durch die Signatur von  $\mathcal{R}_2$   $p_2 \in P_2$ . Zusammen in  $P_1 \parallel P_2$  ergibt das wieder einen geerbten Fehler, also  $(p_1, p_2) \in E_{12}$ .  $(p'_1, p'_2)$  kann jedoch auch ein neuer Kommunikationsfehler sein, dann gilt oBdA  $p'_1 \not\rightarrow_{P'_1}^a$  und  $p'_2 \not\rightarrow_{P'_2}^a$  für ein  $a$  aus  $I_1 \cap O_2$ . Aufgrund von Definition 1.3 2. muss dann auch  $p_2 \not\rightarrow_2^a$  gelten. Für  $p_1$  kann nicht  $p_1 \rightarrow_1^a$  gelten, da sonst die Simulations Relation  $\mathcal{R}_1$  die Implementierung dieser Transition in  $P'_1$  fordern würde (1.3 1.). Es gilt also  $p_1 \not\rightarrow_1^a$  und in der Parallelkomposition  $P_1 \parallel P_2$  ergibt sich daraus ebenfalls ein neuer Kommunikationsfehler mit  $(p_1, p_2) \in E_{12}$ .

$\Rightarrow (P'_1 \parallel P'_2) \in \text{as-impl}(P_1 \parallel P_2)$ . □

Die entgegengesetzte Richtung von Lemma 2.3 gilt im allgemeinen nicht, d.h. es muss zu einer as-Implementierung einer Parallelkomposition  $P' \in \text{as-impl}(P_1 \parallel P_2)$  keine as-Implementierungen  $P'_1$  bzw.  $P'_2$  der einzelnen Komponenten  $P_1$  bzw.  $P_2$  geben, deren Parallelkomposition  $P'_1 \parallel P'_2$  der as-Implementierung der Parallelkomposition  $P$  entsprechen. Die Problematik wird in Abbildung 2.1 an einem Beispiel dargestellt. In der Parallelkomposition wird die may-Transition von  $P_2$  zu zwei may-Transitionen, für die in einer as-Implementierung unabhängig entschieden werden kann, ob sie implementiert werden oder nicht. Somit kommt es in  $P'$  zu dem Problem, dass keine as-Implementierung von  $P_2$  (entweder keine Transition implementiert oder die  $o'$  Transition ist implementiert) in Parallelkomposition mit der Implementierung  $P_1$   $P'$  ergeben würde.



Abbildung 2.1: Gegenbeispiel für Umkehrung von Lemma 2.3

**Lemma 2.4 (*w-as-Implementierungen und Parallelkomposition*).**

$P'_1 \in \text{w-as-impl}(P_1) \wedge P'_2 \in \text{w-as-impl}(P_2) \Rightarrow (P'_1 \parallel P'_2) \in \text{w-as-impl}(P_1 \parallel P_2)$ .

## 2 allgemeine Folgerungen

*Beweis.* Es gelte  $j \in \{1, 2\}$ . Da  $P'_j \in \text{w-as-impl}(P_j)$  gilt, gibt es nach Definition 1.4 eine schwache as-Verfeinerungs-Relation  $\mathcal{R}_j$ , die beschreibt, wie  $P'_j$   $P_j$  verfeinert. Analog zum Beweis von Lemma 2.3 kann auch hier die neue schwache as-Verfeinerungs-Relation der Parallelkompositionen  $\mathcal{R}_{12}$  auf Basis der  $\mathcal{R}_j$  definiert werden:  $\forall p'_1, p'_2, p_1, p_2 : ((p'_1, p'_2), (p_1, p_2)) \in \mathcal{R}_{12} \Leftrightarrow (p'_1, p_1) \in \mathcal{R}_1 \wedge (p'_2, p_2) \in \mathcal{R}_2$ . Es bleibt nun zu zeigen, dass  $\mathcal{R}_{12}$  eine zulässige schwache Verfeinerungs-Relation nach Definition 1.4 ist, da die Parallelkomposition von zwei Implementierungen auch immer eine Implementierung ist (1.2). Für alle folgenden Fälle wird  $((p_1, p_2), (p'_1, p'_2)) \in \mathcal{R}_{12}$  vorausgesetzt.

1. Aus der schwachen Simulations-Definition 1.4 folgt, dass für den ersten Punkt folgendes zu zeigen ist:  $(p_1, p_2) \xrightarrow{i}_{12} (q_1, q_2)$  impliziert  $(p'_1, p'_2) \xrightarrow{i}_{P'_1 \| P'_2} \xRightarrow{\varepsilon}_{P'_1 \| P'_2} (q'_1, q'_2)$  für ein  $(q'_1, q'_2)$  mit  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$ .

Falls die Transition  $(p_1, p_2) \xrightarrow{i}_{12} (q_1, q_2)$  in der Parallelkomposition  $P_1 \| P_2$  für ein  $i \in \text{Synch}(P_1, P_2)$  möglich ist, dann gab es diese Transition auch bereits in  $P_1$  und  $P_2$  als must-Transition. Somit verlangen bereits die schwachen as-Verfeinerungs-Relation  $\mathcal{R}_1$  und  $\mathcal{R}_2$ , dass  $p'_j \xrightarrow{i}_{P'_j} \xRightarrow{\varepsilon}_{P'_j} q'_j$  für  $j = 1$  und  $j = 2$  gilt. Wenn auf die MEIOs mit diesen Transitionen die Parallelkomposition angewendet wird, entstehen der Ablauf  $(p'_1, p'_2) \xrightarrow{i}_{P'_1 \| P'_2} \xRightarrow{\varepsilon}_{12} (q'_1, q'_2)$ . Die  $q'_j$  müssen in der Relation  $\mathcal{R}_j$  mit dem jeweiligen  $q_j$  ein Tupel bilden. Somit gilt auch  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$ . Falls  $i \notin \text{Synch}(P_1, P_2)$ , ist die  $i$  Transition in  $P_1 \| P_2$  nur aufgrund einer entsprechenden Transition in einer Komponente möglich. OBdA gilt  $p_1 \xrightarrow{i}_1 q_1$  und mit Definition 1.4 1. gilt dann  $p'_1 \xrightarrow{i}_{P'_1} q'_1$  und  $(q_1, q'_1) \in \mathcal{R}_1$ . Da es sich um eine unsynchronisierte Aktion von  $P_1$  bzw.  $P'_1$  handelt, muss für die zweite Komponente  $(p_2, q_2) = (p'_2, q'_2) \in \mathcal{R}_2$  gelten. Die interne Aktion ist immer unsynchronisiert und  $i$  in diesem Fall auch, deshalb ist  $(p'_1, p'_2) \xrightarrow{i}_{P'_1 \| P'_2} \xRightarrow{\varepsilon}_{P'_1 \| P'_2} (q'_1, q'_2)$  die in der Parallelkomposition entstehende Transitionsfolge.  $(q_1, q_2) \mathcal{R}_{12} (q'_1, q'_2)$  gilt für den Fall des unsynchronisierten  $i$ 's ebenfalls.

2. Analog zu 1. kann für diesen Punkt  $(p_1, p_2) \xrightarrow{\omega}_{12} (q_1, q_2)$  impliziert  $(p'_1, p'_2) \xRightarrow{\hat{\omega}}_{P'_1 \| P'_2} (q'_1, q'_2)$  für ein  $(q'_1, q'_2)$  mit  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$  gezeigt werden.

Die  $\omega$  Transition in  $P_1 \| P_2$  ist entweder aus einem synchronisierten oder aus einem unsynchronisierten  $\omega$  entstanden. Entsprechend ist dann in einem oder beiden  $P_j$  die Transition möglich und durch die Relationen  $\mathcal{R}_j$  folgen die Transitionen  $p'_j \xRightarrow{\omega}_{P'_j} q'_j$  mit  $(q_j, q'_j) \in \mathcal{R}_j$ . Durch die Parallelkomposition von  $P'_1$  mit  $P'_2$  folgt dann das zu zeigende.

3.  $(p'_1, p'_2) \xrightarrow{-i}_{P'_1 \| P'_2} (q'_1, q'_2)$  impliziert  $(p_1, p_2) \xrightarrow{-i}_{12} \xRightarrow{\varepsilon}_{12} (q_1, q_2)$  für ein  $(q_1, q_2)$  mit  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$  ist die Voraussetzung des 3. Punktes, um zu beweisen, dass  $\mathcal{R}_{12}$  eine schwache as-Verfeinerungs-Relation ist. Die Transition  $i$  kann wiederum durch Synchronisation von zwei Transitionen entstanden sein oder durch eine Transition aus einer Komponenten und  $i \notin \text{Synch}(P'_1 \| P'_2)$ . In beiden Fällen gilt für ein  $j$  oder beide  $p'_j \xrightarrow{-i}_{P'_j} q'_j$ . Daraus folgt mit  $\mathcal{R}_j$   $p_j \xrightarrow{-i}_j \xRightarrow{\varepsilon}_j q_j$  mit  $(q_j, q'_j) \in \mathcal{R}_j$ . Durch Synchronisation oder durch übernehmen der Transitionen



## 2 allgemeine Folgerungen

folgt dann  $(p_1, p_2) \xrightarrow{i} {}_{12} \xRightarrow{\varepsilon} (q_1, q_2)$  und mit Hilfe der Definition von  $\mathcal{R}_{12}$  auch  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$ .

4. Analog zu 3. kann für diesen Punkt  $(p'_1, p'_2) \xrightarrow{\omega} {}_{P'_1 \parallel P'_2} (q'_1, q'_2)$  impliziert  $(p_1, p_2) \xRightarrow{\omega} {}_{12} (q_1, q_2)$  für ein  $(q_1, q_2)$  mit  $((q_1, q_2), (q'_1, q'_2)) \in \mathcal{R}_{12}$  gezeigt werden.  
Die  $\omega$  Transition in  $P'_1 \parallel P'_2$  ist entweder aus einem synchronisierten oder aus einem unsynchronisierten  $\omega$  entstanden. Entsprechend ist dann in einem oder beiden  $P'_j$  die Transitionen möglich und durch die Relationen  $\mathcal{R}_j$  folgen die Transitionen  $p_j \xRightarrow{\omega} {}_j q_j$  mit  $(q_j, q'_j) \in \mathcal{R}_j$ . Durch die Parallelkomposition von  $P_1$  mit  $P_2$  folgt dann das zu zeigende.
5. Für diesen Punkt ist zu zeigen, dass aus  $(p'_1, p'_2) \in E_{P'_1 \parallel P'_2}$   $(p_1, p_2) \in E_{12}$  folgt.  
Falls  $(p'_1, p'_2) \in E_{P'_1 \parallel P'_2}$  ein geerbter Fehler ist, gilt oBdA  $p'_1 \in E_{P'_1}$  und  $p'_2 \in P'_2$ . Mit Definition 1.4 5., die für  $\mathcal{R}_1$  gilt, folgt  $p_1 \in E_1$ .  $p_2$  steht mit  $p'_2$  in der Relation  $\mathcal{R}_2$ . Durch die Signatur folgt  $p_2 \in P_2$ . Es gilt also in der Parallelkomposition  $P_1 \parallel P_2$   $(p_1, p_2) \in E_{12}$  ist ein von  $P_1$  geerbter Fehler.  $(p'_1, p'_2)$  kann jedoch in  $P'_1 \parallel P'_2$  auch ein neuer Kommunikationsfehler sein. Dann gilt oBdA  $p'_1 \not\xrightarrow{a} {}_{P'_1}$  und  $p'_2 \xrightarrow{a} {}_{P'_2}$  für ein  $a \in I_0 \cap O_2$ . Mit  $\mathcal{R}_2$  und 1.4 4. folgt die Gültigkeit von  $p_2 \xRightarrow{a} {}_2$  in  $P_2$ . Da es für  $p'_1$  keine ausgehende  $a$  must-Transition geben darf, darf es auch in  $P_1$  keine  $a$  must-Transition von  $p_1$  ausgehend geben, ansonsten wäre 1.4 1. für  $\mathcal{R}_1$  verletzt.  $(q_1, q_2)$  ist also auch ein neuer Kommunikationsfehler und somit in  $E_{12}$  enthalten.

$\Rightarrow (P'_1 \parallel P'_2) \in \text{w-as-impl}(P_1 \parallel P_2)$ . □

Ein neuer Kommunikationsfehler in einer Parallelkomposition muss in einer Implementierung (as oder w-as) nicht auftauchen, auch nicht in der Parallelkomposition von Implementierungen der einzelnen Komponenten. Dies liegt daran, dass für den Input nur gesagt wird, dass keine must-Transition für die Synchronisation der Aktion vorhanden ist. Es kann trotzdem eine may-Transition für den Input geben, die auch implementiert werden kann. Falls es aber in der Parallelkomposition zweier MEIO zu einem neuen Kommunikationsfehler kommt, dann gibt es auch immer mindestens eine Implementierung, die diesen Kommunikationsfehler enthält und es gibt auch immer mindestens ein Implementierungs-Paar der Komponenten, in deren Parallelkomposition sich dieser Kommunikationsfehler ebenfalls zeigt.

### 3 Verfeinerungen für Kommunikationsfehler-Freiheit

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

**Definition 3.1 (fehler-freie Kommunikation).** *Ein Fehler-Zustand ist lokal erreichbar in einem MEIO  $P$ , wenn ein  $w \in O^*$  existiert mit  $p_0 \xRightarrow{w}_P p \in E$ . Zwei MEIOs  $P_1$  und  $P_2$  kommunizieren fehler-frei, wenn keine as-Implementierungen ihrer Parallelkomposition  $P_{12}$  einen Fehler-Zustände lokal erreichen kann.*

**Definition 3.2 (Kommunikationsfehler-Verfeinerungs-Basirelation).** *Für zwei MEIOs  $P_1$  und  $P_2$  mit der gleichen Signatur wird  $P_1 \sqsubseteq_E^B P_2$  geschrieben, wenn ein Fehler-Zustand in einer as-Implementierung von  $P_1$  nur dann lokal erreichbar ist, wenn es auch eine as-Implementierung von  $P_2$  gibt, in der dieser Fehler-Zustand auch lokal erreichbar ist. Die Basisrelation stellt eine Verfeinerung bezüglich Kommunikationsfehlern dar.  $\sqsubseteq_E^C$  bezeichnet die vollständig abstrakte Präkongruenz von  $\sqsubseteq_E^B$  bezüglich  $\cdot\|\cdot$ , d.h. die größte Präkongruenz bezüglich  $\cdot\|\cdot$ , die in  $\sqsubseteq_E^B$  enthalten ist.*

Für as-Implementierungen  $P_1$  und  $P_2$  entspricht  $\sqsubseteq_E^B$  der Relation  $\sqsubseteq_E^B$  aus [Sch16].

Wie in [Sch16] werden die Fehler hier Trace-basiert betrachtet.

**Definition 3.3 (Kommunikationsfehler-Traces).** *Für ein MEIO  $P$  wird definiert:*

- strikte Kommunikationsfehler-Traces:  $StET(P) := \{w \in \Sigma^* \mid p_0 \xRightarrow{w}_P p \in E\}$ ,
- gekürzte Kommunikationsfehler-Traces:  $PrET(P) := \{\text{prune}(w) \mid w \in StET(P)\}$ ,
- Input-kritische-Traces:  $MIT(P) := \{wa \in \Sigma^* \mid p_0 \xRightarrow{w}_P p \wedge a \in I \wedge p \not\xrightarrow{a}_P\}$ .

Da die Basisrelation über as-Implementierungen spricht, ist es wichtig bereits in den Trace-Mengen eine Beziehung zwischen der allgemeinen Definition für MEIOs und deren as-Implementierungen herzustellen. Deshalb wird in der folgenden Proposition eine alternative Sichtweise auf die Trace-Definitionen dargestellt. Die Traces eines MEIOs entsprechen somit der Vereinigung der Traces aller seiner as-Implementierungen.

**Proposition 3.4 (Kommunikationsfehler-Traces und Implementierung).** *Für ein MEIO  $P$  gilt:*

1. strikte Kommunikationsfehler-Traces:  $StET(P) = \{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'} p' \in E\}$
2. Input-kritische-Traces:  $MIT(P) = \{wa \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'} p' \wedge a \in I \wedge p' \not\xrightarrow{a}_{P'}\}$

*Beweis.*

1. Wie schon in Beweis zu 2.1 festgestellt, sind alle Abläufe, die in  $P$  via may-Transitionen möglich sind in mindestens einer as-Implementierung von  $P$  via must-Transitionen möglich. Umgekehrt ist auch jeder Ablauf, der in einer as-Implementierung von  $P$  möglich ist auch in  $P$  durch may-Transitionen möglich.  
Aufgrund des 3. Punktes der Definition 1.3 kann jede as-Implementierung von  $P$  nur Fehler-Zustände enthalten, die auch  $P$  enthält. Da alle möglichen Implementierungen von  $P$  in  $\{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'} p' \in E\}$  betrachtet werden, ist auch jeder in  $P$  durch may-Transitionen erreichbare Fehler-Zustand auch in mindestens einer as-Implementierung ebenfalls erreichbar, jedoch durch must-Transitionen.
2. Für jedes  $w$  in  $L(P)$  gibt es mindestens eine as-Implementierung von  $P$ , die dieses  $w$  auch ausführen kann und umgekehrt (Beweis von 2.1). Falls in  $MIT(P)$  mindestens ein Element gibt, gibt es in  $P$  einen Trace von  $w$ , nach dem ein Input  $a$  nicht zwingendermaßen folgen muss. Die  $a$  Transition also entweder eine may-Transition ist oder gar nicht existiert in  $P$ . Es muss also auch einen  $w$ -Trace in einer as-Implementierung geben, die in einem Zustand endet, der mit dem Zustand aus  $P$  in Relation steht, in dem das  $a$  nicht erzwungen wird. Falls die Transition in  $P$  nicht vorhanden ist, muss jede as-Implementierung, die so einen  $w$ -Trace enthält auch  $wa$  als Input-kritischen-Trace haben. Andernfalls handelt es sich bei  $a$  in  $P$  um eine may-Transition, die von mindestens einer as-Implementierung, die den  $w$ -Trace enthält, nicht implementiert wird und somit  $wa$  auch als Input-kritischen-Trace enthält.  
Für ein  $wa \in \{wa \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'} p' \wedge a \in I \wedge p' \not\xrightarrow{a}_{P'}\}$  muss es eine as-Implementierung von  $P$  geben, die diesen Input-kritischen-Trace implementiert.  $P$  muss also auch das  $w$  ausführen können zu einem Zustand, in dem  $P$   $a$  nicht als must-Transition enthalten. Falls  $P$   $a$  nach  $w$  nur als must-Transition enthalten würde, würde 1.3 1. die Implementierung von  $a$  erzwingen und somit könnte für keine as-Implementierung von  $P$   $wa$  ein Input-kritischer-Trace sein. Es gilt also auch  $wa \in MIT(P)$ .

□

**Definition 3.5 (Kommunikationsfehler-Semantik).** Sei  $P$  ein MEIO.

- Die Menge der Kommunikationsfehler-Traces von  $P$  ist  $ET(P) := \text{cont}(PrET(P)) \cup \text{cont}(MIT(P))$ .

- Die Kommunikationsfehler-geflutete Sprache von  $P$  ist  $EL(P) := L(P) \cup ET(P)$ .

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

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

**Satz 3.6 (Kommunikationsfehler-Semantik für Parallelkompositionen).** Für zwei komponierbare MEIOs  $P_1, P_2$  und ihre Komposition  $P_{12}$  gilt:

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

*Beweis.*

1. „ $\subseteq$ “:

Da beide Seiten der Gleichung unter der Fortsetzung  $\text{cont}$  abgeschlossen sind, genügt es ein präfix-minimales Element  $w$  von  $ET_{12}$  zu betrachten. Diese Element ist aufgrund der Definition der Menge der Kommunikationsfehler-Traces 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  $(p_{01}, p_{02}) \xRightarrow{x}_{12} (p_1, p_2) \wedge a \in I_{12} \wedge (p_1, p_2) \not\xrightarrow{a}_{12}$ . Da  $I_{12} = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$  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  $a \in (I_1 \cup I_2) \setminus (I_1 \cap I_2)$  ist.
  - Fall 1a) ( $a \in (I_1 \cap I_2)$ ): Durch Projektion des Ablaufes auf die einzelnen Transitionssysteme erhält man oBdA  $p_{01} \xRightarrow{x_1}_1 p_1 \not\xrightarrow{a}_1$  und  $p_{02} \xRightarrow{x_2}_2 p_2 \not\xrightarrow{a}_2$  oder  $p_{02} \xRightarrow{x_2}_2 p_2 \xrightarrow{a}_2$  mit  $x \in x_1 \parallel x_2$ . 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  $w \in (x_1 \parallel x_2) \cdot \{a\} \subseteq (x_1 a) \parallel (x_2 a) \subseteq ET_1 \parallel EL_2$ , und somit ist  $w$  in der rechten Seite der Gleichung enthalten.
  - Fall 1b) ( $a \in (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ ): OBdA gilt  $a \in I_1$ . Durch die Projektion auf die einzelnen Komponenten erhält man:  $p_{01} \xRightarrow{x_1}_1 p_1 \not\xrightarrow{a}_1$  und  $p_{02} \xRightarrow{x_2}_2 p_2$  mit  $x \in x_1 \parallel x_2$ . Daraus folgt  $x_1 a \in \text{cont}(MIT_1) \subseteq ET_1$  und  $x_2 \in L_2 \subseteq EL_2$ . Somit gilt  $w \in (x_1 \parallel x_2) \cdot \{a\} \subseteq (x_1 a) \parallel x_2 \subseteq ET_1 \parallel EL_2$ . Dies ist eine Teilmenge der rechten Seite der Gleichung.
- Fall 2 ( $w \in PrET_{12}$ ): Aus der Definition von  $PrET$  und  $\text{prune}$  folgt, dass ein  $v \in O_{12}^*$  gibt, so dass  $(p_{01}, p_{02}) \xRightarrow{w}_{12} (p_1, p_2) \xRightarrow{v}_{12} (p'_1, p'_2)$  gilt mit  $(p'_1, p'_2) \in E_{12}$  und  $w = \text{prune}(wv)$ . Durch Projektion auf die Komponenten erhält man  $p_{01} \xRightarrow{w_1}_1 p_1 \xRightarrow{v_1}_1 p'_1$  und  $p_{02} \xRightarrow{w_2}_2 p_2 \xRightarrow{v_2}_2 p'_2$  mit  $w \in w_1 \parallel w_2$  und  $v \in v_1 \parallel v_2$ . Aus  $(p'_1, p'_2) \in ET_{12}$  folgt, dass es sich entweder um einen geerbten oder einen neuen Fehler handelt. Bei einem geerbten wäre bereits einer der beiden Zustände  $p'_1$  bzw.  $p'_2$  ein Fehler-Zustand gewesen. Ein neuer Kommunikationsfehler hingegen

wäre durch das fehlen der Synchronisations-Erzwingung (fehlende must-Transition) in einer der Komponenten entstanden.

- Fall 2a) (geerbter Fehler): OBdA gilt  $p'_1 \in E_1$ . Daraus folgt,  $w_1v_1 \in StET_1 \subseteq \text{cont}(PrET_1) \subseteq ET_1$ . Da  $p_{02} \xrightarrow{w_2v_2}_2$  gilt, erhält man  $w_2v_2 \in L_2 \subseteq EL_2$ . Dadurch ergibt sich  $wv \in ET_1 \parallel EL_2$  mit  $w = \text{prune}(wv)$  und somit ist  $w$  in der rechten Seite der Gleichung enthalten.
- Fall 2b) (neuer Kommunikationsfehler): OBdA gilt  $a \in I_1 \cap O_2$  mit  $p'_1 \xrightarrow{a}_1 \wedge p'_2 \xrightarrow{a}_1$ . Daraus folgt  $w_1v_1a \in MIT_1 \subseteq ET_1$  und  $w_2v_2a \in L_2 \subseteq EL_2$ . Damit ergibt sich  $wva \in ET_1 \parallel EL_2$ , da  $a \in O_1 \subseteq O_{12}$  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  $\text{cont}$  wird auch in diesem Fall nur ein präfix-minimales Element  $x \in \text{prune}((ET_1 \parallel EL_2) \cup (EL_1 \parallel ET_2))$  betrachtet. Da  $x$  durch die Anwendung der  $\text{prune}$ -Funktion entstanden ist, existiert ein  $y \in O_{12}^*$  mit  $xy \in (ET_1 \parallel EL_2) \cup (EL_1 \parallel ET_2)$ . OBdA wird davon ausgegangen, dass  $xy \in ET_1 \parallel EL_2$  gilt, d.h. es gibt  $w_1 \in ET_1$  und  $w_2 \in EL_2$  mit  $xy \in w_1 \parallel w_2$ .

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  $I_{12}$  endet oder  $v = \varepsilon$ . Damit muss  $v$  ein Präfix von  $x$  sein.  $\varepsilon$  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  $P_1$  ein Fehler-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.  $v_1 = v'_1a$ .  $v'$  sei das Präfix von  $xy$ , das mit der letzten Aktion von  $v_1$  endet, d.h. mit  $a$  und  $v'_2 = v'|_{\Sigma_2}$ . Falls  $v_2 = w_2 \in L_2$ , 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  $v_2 \in MIT_2$  weiß man zusätzlich, dass  $v_2$  auf  $b \in I_2$  endet. Es kann jedoch  $a = b$  gelten.

In allen Fällen erhält man  $v'_2 = v'|_{\Sigma_2}$  ist ein Präfix von  $v_2$  und  $v' \in v_1 \parallel v'_2$  ist ein Präfix von  $xy$ . Es kann nur für die Fälle  $a \notin I_2$  gefolgert werden, dass  $p_{02} \xrightarrow{v'_2}_2$  gilt.

- Fall I ( $v_1 \in MIT_1$  und  $v_1 \neq \varepsilon$ ): Es gibt einen Ablauf der Form  $p_{01} \xRightarrow{v'_1}_1 p_1 \xrightarrow{a}_1$  und es gilt  $v' = v''a$ .
  - Fall Ia) ( $a \notin \Sigma_2$ ): Es gilt  $p_{02} \xRightarrow{v'_2}_2 p_2$  mit  $v'' \in v'_1 \| v'_2$ . Dadurch erhält man  $(p_{01}, p_{02}) \xRightarrow{v''}_{12} (p_1, p_2) \xrightarrow{a}_{12}$  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  $p_{02} \xRightarrow{v''_2}_2 \xrightarrow{a}_2$  und  $v'' \in v'_1 \| v''_2$ .  $a$  ist für  $P_2$ , ebenso wie für  $P_1$ , ein nicht erzwungener Input. Daraus folgt, dass  $(p_1, p_2) \xrightarrow{a}_{12}$  gilt. Es wird ebenfalls  $v := v''a = v' \in MIT_{12}$  gewählt.
  - Fall Ic) ( $a \in I_2$  und  $v'_2 \in L_2 \setminus MIT_2$ ): Es gilt  $p_{02} \xRightarrow{v''_2}_2 p_2 \xrightarrow{a}_2$  mit  $v'_2 = v''_2 a$ . Da die gemeinsamen Inputs synchronisiert werden, folgt  $(p_1, p_2) \xrightarrow{a}_{12}$  bereits aus  $q_1 \xrightarrow{a}_1$ . 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  $p_{02} \xRightarrow{v'_2}_2$ . Man erhält also  $p_{02} \xRightarrow{v''_2}_2 p_2 \xrightarrow{a}_2$  mit  $v'' \in v'_1 \| v''_2$ . Daraus ergibt sich  $(p_{01}, p_{02}) \xRightarrow{v''}_{12} (p_1, p_2)$  mit  $p_2 \xrightarrow{a}_2$ ,  $p_1 \xrightarrow{a}_1$ ,  $a \in I_1$  und  $a \in O_2$ , somit gilt  $(p_1, p_2) \in E_{12}$ . Es wird  $v := \text{prune}(v'') \in PrET_{12}$  gewählt.
- Fall II ( $v_1 \in PrET_1$ ):  $\exists u_1 \in O_1^* : p_{01} \xRightarrow{v_1}_1 p_1 \xRightarrow{u_1}_1 p'_1$  mit  $p'_1 \in E_1$ . Im Fall  $v'_1 \neq \varepsilon$  kann das  $a$ , auf das  $v_1$  endet, ebenfalls der letzte Buchstabe von  $v_2$  sein. Im Fall von  $v_2 \in MIT_2$  kann somit  $a = b$  gelten, 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  $p_{02} \xRightarrow{v'_2}_2 p_2$  mit  $(p_{01}, p_{02}) \xRightarrow{v'}_{12} (p_1, p_2)$ .
  - Fall IIa) ( $u_2 \in (O_1 \cap I_2)^*, c \in (O_1 \cap I_2)$ , sodass  $u_2 c$  Präfix von  $u_1|_{I_2}$  mit  $p_2 \xRightarrow{u_2}_2 p'_2 \xrightarrow{c}_2$ ): Für das Präfix  $u'_1 c$  von  $u_1$  mit  $(u'_1 c)|_{I_2} = u_2 c$  weiß man, dass  $q_1 \xRightarrow{u'_1}_1 q'_1 \xrightarrow{c}_1$ . Somit gilt  $u'_1 \in v'_1 \| u_2$  und  $(p_1, p_2) \xRightarrow{u'_1}_{12} (q'_1, q'_2) \in E_{12}$ , der für  $P_2$  der entsprechende Input nicht erzwungen wird, der mit dem  $c$  Output von  $P_1$  zu koppeln wäre. Es handelt sich also um einen neuen Kommunikationsfehler. 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) ( $p_2 \xRightarrow{u_2}_2 p'_2$  mit  $u_2 = u_1|_{I_2}$ ): Es gilt  $u_1 \in v'_1 \| u_2$  und  $(p_1, p_2) \xRightarrow{u_1}_{12} (p'_1, p'_2) \in E_{12}$ , da  $p'_1 \in E_1$  und somit handelt es sich in  $P_{12}$  um einen geerbten Fehler. Nun wird  $v := \text{prune}(v' u_1) \in PrET_{12}$  gewählt, das wiederum ein Präfix von  $v'$  ist.

2.:

Durch die Definitionen ist klar, dass  $L_i \subseteq EL_i$  und  $ET_i \subseteq EL_i$  gilt. Die Argumentation startet auf den rechten Seite der Gleichung:

$$\begin{aligned}
 (EL_1 \| EL_2) \cup ET_{12} &\stackrel{3.5}{=} ((L_1 \cup ET_1) \| (L_2 \cup ET_2)) \cup ET_{12} \\
 &= (L_1 \| L_2) \cup \underbrace{(L_1 \| ET_2)}_{\substack{\subseteq (EL_1 \| ET_2) \\ \stackrel{1.}{\subseteq} ET_{12}}} \cup \underbrace{(ET_1 \| L_2)}_{\substack{\subseteq (ET_1 \| EL_2) \\ \stackrel{1.}{\subseteq} ET_{12}}} \cup \underbrace{(ET_1 \| ET_2)}_{\substack{\subseteq (EL_1 \| ET_2) \\ \stackrel{1.}{\subseteq} ET_{12}}} \cup ET_{12} \\
 &= (L_1 \| L_2) \cup ET_{12} \\
 &\stackrel{2.2}{=} L_{12} \cup ET_{12} \\
 &\stackrel{3.5}{=} EL_{12}.
 \end{aligned}$$

□

**Korollar 3.7 (Kommunikationsfehler-Präkongruenz).** Die Relation  $\sqsubseteq_E$  ist eine Präkongruenz bezüglich  $\cdot \| \cdot$ .

*Beweis.* Es muss gezeigt werden: Wenn  $P_1 \sqsubseteq_E P_2$  gilt, dann für jedes komponierbare  $P_3$  auch  $P_{31} \sqsubseteq_E P_{32}$ . 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  $\text{cont}$ ,  $\text{prune}$  und  $\cdot \| \cdot$  auf Sprachen wie folgt:

- $ET_{31} \stackrel{3.6}{=}^1 \text{cont}(\text{prune}((ET_3 \| EL_1) \cup (EL_3 \| ET_1)))$   
 $\begin{array}{l} ET_1 \subseteq ET_2 \\ \text{und} \\ EL_1 \subseteq EL_2 \end{array} \subseteq \text{cont}(\text{prune}((ET_3 \| EL_2) \cup (EL_3 \| ET_2)))$   
 $\stackrel{3.6}{=}^1 ET_{32},$
- $EL_{31} \stackrel{3.6}{=}^2 (EL_3 \| EL_1) \cup E_{31}$   
 $\begin{array}{l} EL_1 \subseteq EL_2 \\ \text{und} \\ ET_{31} \subseteq ET_{32} \end{array} \subseteq (EL_3 \| EL_2) \cup ET_{32}$   
 $\stackrel{3.6}{=}^2 EL_{32}.$

□

**Lemma 3.8 (Verfeinerung mit Kommunikationsfehlern).** Gegeben sind zwei MEIOs  $P_1$  und  $P_2$  mit der gleichen Signatur. Wenn  $U \| P_1 \sqsubseteq_E^B U \| P_2$  für alle Partner  $U$  gilt, dann folgt daraus die Gültigkeit von  $P_1 \sqsubseteq_E P_2$ .

*Beweis.* Da  $P_1$  und  $P_2$  die gleiche Signaturen 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  $P_1 \sqsubseteq_E P_2$  zu zeigen, wird nachgeprüft, ob folgendes gilt:

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

Für ein gewähltes präfix-minimales Element  $w \in ET_1$  wir gezeigt, dass dieses  $w$  oder eines seiner Präfixe in  $ET_2$  enthalten ist. Dies ist möglich, da die beiden Mengen  $ET_1$  und  $ET_2$  durch  $\text{cont}$  abgeschlossen sind.

- Fall 1 ( $w = \varepsilon$ ): Es handelt sich um einen lokal erreichbaren Fehler-Zustand in  $P_1$ . Für  $U$  wird ein Transitionssystem verwendet, das nur aus dem Startzustand und einer must-Schleife für alle Inputs  $x \in I_U$  besteht. Somit kann  $P_1$  die im Prinzip gleichen Fehler-Zustände lokal erreichen wie alle möglichen as-Implementierungen von  $U \parallel P_1$  zusammen. Daraus folgt, dass auch mindestens eine as-Implementierung von  $U \parallel P_2$  einen lokal erreichbaren Fehler-Zustand haben muss. Durch die Definition von  $U$  kann dieser Fehler nur von  $P_2$  geerbt sein. Es muss also in  $P_2$  ein Fehler-Zustand durch interne Aktionen und Outputs erreichbar sein, d.h. es gilt  $\varepsilon \in \text{Pr}ET_2$ .
- Fall 2 ( $w = x_1 \dots x_n x_{n+1} \in \Sigma^+$  mit  $n \geq 0$  und  $x_{n+1} \in I = O_U$ ): Es wird der folgende Partner  $U$  betrachtet (siehe auch Abbildung 3.1):

- $U = \{p_0, p_1, \dots, p_{n+1}\}$ ,
- $p_0 U = p_0$ ,
- $\rightarrow_U = \{(p_i, x_{i+1}, p_{i+1}) \mid 0 \leq i \leq n\}$   
 $\cup \{(p_i, x, p_{n+1}) \mid x \in I_U \setminus \{x_{i+1}\}, 0 \leq i \leq n\}$   
 $\cup \{(p_{n+1}, x, p_{n+1}) \mid x \in I_U\}$ .
- $E_U = \emptyset$ ,

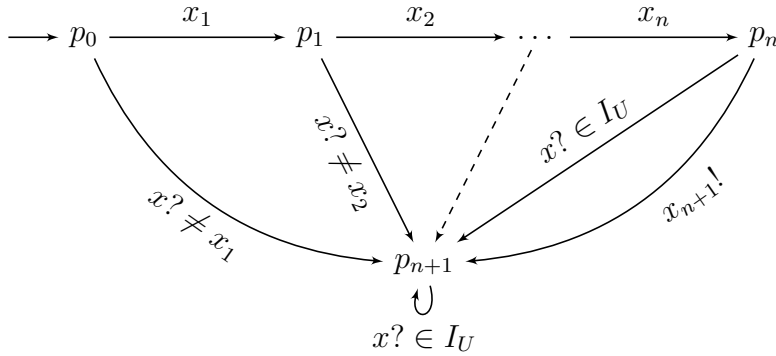


Abbildung 3.1:  $x? \neq x_i$  steht für alle  $x \in I_U \setminus \{x_i\}$

Für  $w$  können nun zwei Fälle unterschieden werden. Aus beiden wird folgen, dass für mindestens eine as-Implementierung  $P'$  von  $U \parallel P_1$   $\varepsilon \in \text{Pr}ET(P')$ .



- Fall 2a) ( $w \in MIT_1$ ): In  $U \parallel P_1$  erhält man  $(p_1, p_{01}) \xrightarrow{x_1 \dots x_n} U \parallel P_1 (p_n, p')$  mit  $p' \xrightarrow{x_{n+1}}_1$  und  $p_n \xrightarrow{x_{n+1}}_U$ . Deshalb gilt  $(p_1, p') \in E_{U \parallel P_1}$ . Da alle Aktionen aus  $w$  bis auf  $x_{n+1}$  synchronisiert werden und  $I \cap I_U = \emptyset$ , gilt  $x_1, \dots, x_n \in O_{U \parallel P_1}$ . Da  $(p_1, p') \in E_{U \parallel P_1}$  gibt es mindestens ein  $P'$  in  $\text{as-impl}(U \parallel P_1)$ , die diesen Fehler-Zustand ebenfalls enthält. Daraus ergibt sich dann  $\varepsilon \in \text{PrET}(P')$ .
- Fall 2b) ( $w \in \text{PrET}_1$ ): In  $U \parallel P_1$  erhält man  $(p_0, p_{01}) \xrightarrow{w} U \parallel P_1 (p_{n+1}, p'') \xrightarrow{u} U \parallel P_1 (p_{n+1}, p')$  für  $u \in O^*$  und  $p' \in E_1$ . Daraus folgt  $(p_{n+1}, p') \in E_{U \parallel P_1}$  und somit  $wu \in \text{StET}(U \parallel P_1)$ . Da alle Aktionen in  $w$  synchronisiert werden und  $I \cap I_U = \emptyset$ , gilt  $x_1, \dots, x_n, x_{n+1} \in O_{U \parallel P_1}$  und, da  $u \in O^*$ , folgt  $u \in O_{U \parallel P_1}^*$ . Somit ergibt sich für eine as-Implementierung  $P'$  von  $U \parallel P_1$   $\varepsilon \in \text{PrET}(P')$ .

Da  $\varepsilon \in \text{PrET}(P')$  für ein  $P'$  aus  $\text{as-impl}(U \parallel P_1)$  gilt, kann durch  $U \parallel P_1 \sqsubseteq_E^B U \parallel P_2$  geschlossen werden, dass auch in mindestens einer as-Implementierung von  $U \parallel P_2$  ein Fehler-Zustand lokal erreichbar sein muss. Da as-Implementierungen die Definition 1.3 erfüllen müssen, muss jeder in einer as-Implementierung von  $U \parallel P_2$  lokal erreichbare Fehler auch in  $U \parallel P_2$  lokal erreichbar sein.

Dieser Fehler kann geerbt oder neu sein.

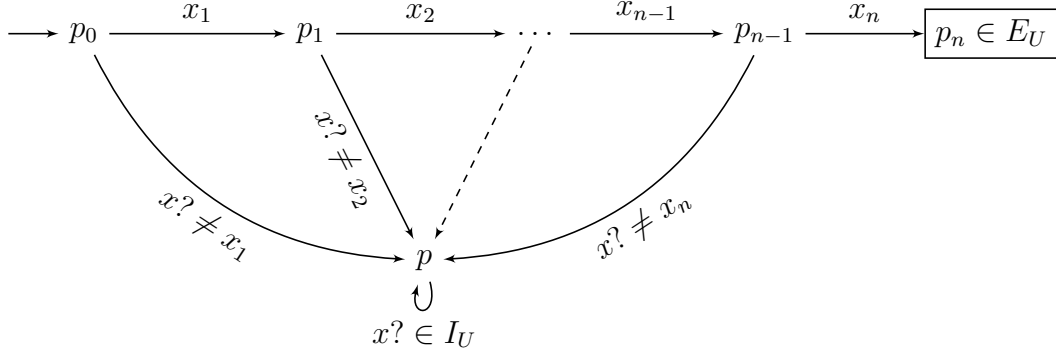
- Fall 2i) (neuer Fehler): Da jeder Zustand von  $U$  alle Inputs  $x \in O = I_U$  durch must-Transitionen erzwingt, muss ein lokal erreichbarer Fehler-Zustand der Form sein, dass ein Output  $a \in O_U$  von  $U$  möglich ist, der nicht mit einem passenden Input aus  $P_2$  synchronisiert werden muss ( $P_2$  enthält die entsprechende  $a$  Transitionen nicht als must-Transition). Durch die Konstruktion von  $U$  sind in  $p_{n+1}$  keine Outputs möglich. Ein neuer Kommunikationsfehler muss also die Form  $(p_i, p')$  haben mit  $i \leq n$ ,  $p' \xrightarrow{x_{i+1}}_2$  und  $x_{i+1} \in O_U = I$ . Durch Projektion erhält man dann  $p_{02} \xrightarrow{x_1 \dots x_i} p' \xrightarrow{x_{i+1}}_2$  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 Fehler):  $U$  hat  $x_1 \dots x_i u$  mit  $u \in I_U^* = O^*$  ausgeführt und ebenso hat  $P_2$  dieses Wort abgearbeitet. Durch dies hat  $P_2$  einen Zustand  $E_2$  erreicht, da von  $U$  kleine Fehler geerbt werden können. Es gilt dann  $\text{prune}(x_1 \dots x_i u) = \text{prune}(x_1 \dots x_i) \in \text{PrET}_2 \subseteq ET_2$ . 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 Fehler-Zustand. Da  $ET$  der Menge aller Verlängerungen von gekürzten Kommunikationsfehler-Traces 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  $L_1 \setminus ET_1 \subseteq EL_2$  gilt.

Es wird dafür ein beliebiges  $w \in L_1 \setminus ET_1$  gewählt und gezeigt, dass es in  $EL_2$  enthalten ist.

- Fall 1 ( $w = \varepsilon$ ): Da  $\varepsilon$  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)

- $U = \{p_0, p_1, \dots, p_n, p\}$ ,
- $p_{0U} = p_0$ ,
- $\rightarrow_U = \{(p_i, x_{i+1}, p_{i+1}) \mid 0 \leq i < n\}$   
 $\cup \{(p_i, x, p) \mid x \in I_U \setminus \{x_{i+1}\}, 0 \leq i < n\}$   
 $\cup \{(p, x, p) \mid x \in I_U\}$ .
- $E_U = \{p_n\}$ ,


 Abbildung 3.2:  $x? \neq x_i$  steht für alle  $x \in I_U \setminus \{x_i\}$ ,  $p_n$  ist der einzige Fehler-Zustand

Da  $p_{01} \xRightarrow{w}_1 p'$  gilt, kann man schließen, dass  $U \parallel P_1$  einen lokal erreichbaren geerbten Fehler hat. Es gibt also auch mindestens eine Implementierung in  $\text{as-impl}(U \parallel P_1)$ , die diesen lokal erreichbaren Fehler implementiert. Somit muss es eine as-Implementierung von  $U \parallel P_2$  geben, die ebenfalls einen lokal erreichbaren Fehler-Zustand hat. Aufgrund von Definition 1.3 3. muss dieser Fehler-Zustand auch in  $U \parallel P_2$  lokal erreichbar sein.

- Fall 2a) (neuer Fehler aufgrund von  $x_i \in O_U$  und  $p_{02} \xRightarrow{x_1 \dots x_{i-1}}_2 q'' \not\xrightarrow{x_i}_2$ ): Es gilt  $x_1 \dots x_i \in MIT_2$  und somit  $w \in EL_2$ . Anzumerken ist, dass es nur auf diesem Weg Outputs von  $U$  möglich sind, deshalb gibt es keine anderen Outputs von  $U$ , die zu einem neuen Fehler führen können.
- Fall 2b) (neuer Fehler aufgrund von  $a \in O = I_U$ ): Der einzige Zustand, in dem  $U$  nicht alle Input erlaubt sind, ist  $p_n$ , der bereits ein Fehler-Zustand ist. Da in diesem Fall dieser Zustand in  $U \parallel P_2$  erreichbar ist, besitzt das komponierte MEIO einen geerbten Fehler und es gilt  $w \in L_2 \subseteq EL_2$ , wegen dem folgenden Fall 2c).
- Fall 2c) (geerbter Fehler von  $U$ ): Da  $p_n$  der einzige Fehler-Zustand in  $U$  ist und alle Aktionen synchronisiert sind, ist dies nur möglich, wenn  $p_{02} \xRightarrow{x_1 \dots x_n}_2$  gilt. In diesem Fall gilt  $w \in L_2 \subseteq EL_2$ .

- Fall 2d) (geerbter Fehler von  $P_2$ ): Es gilt dann  $p_{02} \stackrel{x_1 \dots x_i u}{\equiv} p' \in E_2$  für  $i \geq 0$  und  $u \in O^*$ . Somit ist  $x_1 \dots x_i u \in StET_2$  und damit  $\text{prune}(x_1 \dots x_i u) = \text{prune}(x_1 \dots x_i) \in PrET_2 \subseteq EL_2$ . Somit gilt  $w \in EL_2$ .

□

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

**Satz 3.9 (Vollständige Abstraktheit für Kommunikationsfehler-Semantik).** Für zwei MEIOs  $P_1$  und  $P_2$  mit derselben Signatur gilt  $P_1 \sqsubseteq_E^C P_2 \Leftrightarrow P_1 \sqsubseteq_E P_2$ .

*Beweis.*

„ $\Leftarrow$ “: Nach Definition gilt, genau dann wenn  $\varepsilon \in ET(P)$ , ist ein Fehler-Zustand lokal erreichbar in  $P$ .  $P_1 \sqsubseteq_E P_2$  impliziert, dass  $\varepsilon \in ET_2$  gilt, wenn  $\varepsilon \in ET_1$ . Somit ist ein Fehler-Zustand in  $P_1$  nur dann lokal erreichbar, wenn dieser auch in  $P_2$  lokal erreichbar ist. Falls es also eine as-Implementierung von  $P_1$  gibt, in der ein Fehler-Zustand lokal erreichbar ist, dann gibt es auch mindestens eine as-Implementierung von  $P_2$ , die einen Fehler-Zustand lokal erreichen kann. Dadurch folgt, dass  $P_1 \sqsubseteq_E^B P_2$  gilt, da  $\sqsubseteq_E^B$  in Definition 3.2 über die lokale Erregbarkeit der Fehler-Zustände in den as-Implementierungen definiert wurde und die  $ET$ -Mengen von  $P_1$  und  $P_2$  auch durch die Vereinigung der Traces ihrer as-Implementierungen, wie in Proposition 3.4, ausgedrückt werden können. Es ist also  $\sqsubseteq_E$  in  $\sqsubseteq_E^B$  enthalten. Wie in Korollar 3.7 gezeigt, ist  $\sqsubseteq_E$  eine Präkongruenz bezüglich  $\cdot\|\cdot$ . Da  $\sqsubseteq_E^C$  die größte Präkongruenz bezüglich  $\cdot\|\cdot$  ist, die in  $\sqsubseteq_E^B$  enthalten ist, muss  $\sqsubseteq_E$  in  $\sqsubseteq_E^C$  enthalten sein. Es folgt also aus  $P_1 \sqsubseteq_E P_2$ , dass auch  $P_1 \sqsubseteq_E^C P_2$  gilt.

„ $\Rightarrow$ “: Durch die Definition von  $\sqsubseteq_E^C$  als Präkongruenz in 3.2 folgt aus  $P_1 \sqsubseteq_E^C P_2$ , dass  $U\|P_1 \sqsubseteq_E^C U\|P_2$  für alle MEIOs  $U$  gilt, die mit  $P_1$  komponierbar sind. Da  $\sqsubseteq_E^C$  nach Definition auch in  $\sqsubseteq_E^B$  enthalten sein soll, folgt aus  $U\|P_1 \sqsubseteq_E^C U\|P_2$  auch die Gültigkeit von  $U\|P_1 \sqsubseteq_E^B U\|P_2$  für alle diese MEIOs  $U$ . Mit Lemma 3.8 folgt dann  $P_1 \sqsubseteq_E P_2$ . □

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

Angenommen man definiert, dass  $P_1 P_2$  verfeinern soll, genau dann wenn für alle Partner MEIOs  $U$ , für die  $P_2$  fehler-frei mit  $U$  kommuniziert, folgt, dass  $P_1$  ebenfalls fehler-frei mit  $U$  kommuniziert. Dann wird auch diese Verfeinerung durch  $\sqsubseteq_E$  charakterisiert.

**Korollar 3.10.** Es gilt:  $P_1 \sqsubseteq_E P_2 \Leftrightarrow U\|P_1 \sqsubseteq_E^B U\|P_2$  für alle Partner  $U$ .

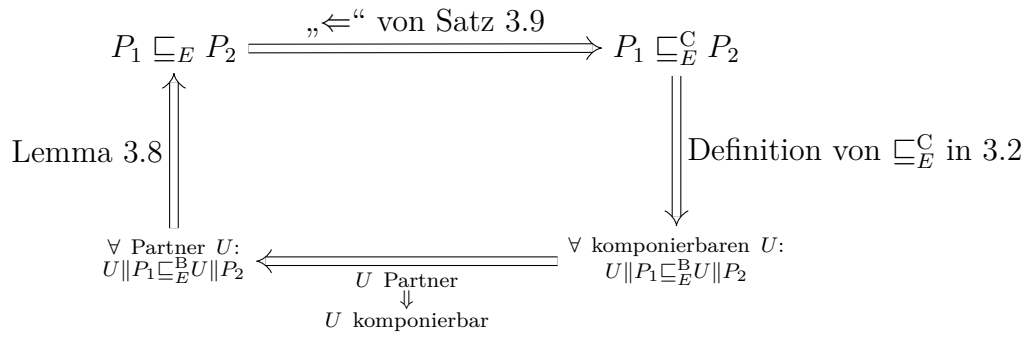


Abbildung 3.3: Folgerungskette der Kommunikationsfehler-Relationen

## 4 Verfeinerungen für Kommunikationsfehler- und Ruhe-Freiheit

In diesem Kapitel wird die Menge der betrachteten Zustandsmengen von den Kommunikations-basierten Fehlern im letzten Kapitel erweitert um Ruhe-Zustände.

Zustände, die keine Outputs ohne einen Input ausführen können in Form einer must-Transition, werden als in einer Art Verklemmung angesehen, da sie ohne Zutun von Außen den Zustand nicht mehr verlassen können, falls ein möglicherweise vorhandener may-Output nicht implementiert wird. So ein Zustand hat also keine must-Transitions-Möglichkeiten für einen Output. Falls dieser Zustand die Möglichkeit für eine interne Aktion via einer must-Transition hat, darf durch die  $\tau$ s niemals ein Zustand erreicht werden, von dem aus ein Output in Implementierungen erzwungen wird. Ein Zustand, der keine Outputs und  $\tau$ s via must-Transitionen ausführen kann, ist also ein Deadlock-Zustand, in denen das System nichts mehr tun können muss ohne einen Input. Wenn man eine Erweiterung um  $\tau$ s zu Zuständen ohne must-Outputs zulässt, hat man zusätzlich 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 müssen. Die Menge der Zustände, die sich in einer Verklemmung befinden, würde also durch  $\{p \in P \mid \forall a \in O : p \not\stackrel{a}{\rightarrow}_P\}$  beschrieben werden. Somit wären dies alle Zustände, die keine Möglichkeit haben ohne einen Input von Außen oder eine implementierte may-Output-Transition je wieder einen Output machen zu können. Falls man diese Definition verwenden würde, müsste man immer alle Zustände betrachten, die durch  $\tau$ s 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  $\tau$ s ausführen können.

**Definition 4.1 (*Ruhe*).** Ein Ruhe-Zustand ist ein Zustand in einem MEIO  $P$ , der keine Outputs und kein  $\tau$  zulässt via must-Transitionen.

Somit ist die Menge der Ruhe-Zustände in einem MEIO  $P$  wie folgt formal definiert:  $Qui(P) := \{p \in P \mid \forall \alpha \in (O \cup \{\tau\}) : p \not\stackrel{\alpha}{\rightarrow}_P\}$ .

**Proposition 4.2 (*Ruhe und Implementierung*).** Für ein MEIO  $P$  gilt:  $Qui(P) = \{p' \in P' \mid P' \in \text{as-impl}(P) \wedge \forall \alpha \in (O \cup \{\tau\}) : p' \not\stackrel{\alpha}{\rightarrow}_{P'}\}$ .

*Beweis.* Da für alle  $P' \in \text{as-impl}(P) \longrightarrow_{P'} \text{---} \rightarrow_{P'}$  gilt, dürfen alle  $p'$  keine ausgehenden Transitionen haben. Nach Definition 1.3 dürfen zu  $p'$  in Relation stehenden Zustände  $p$  in  $P$  keine ausgehenden must-Transitionen für lokale Aktionen haben, da  $p'$  diese sonst implementieren müsste. Somit sind alle  $p'$ , die die Forderung dieser Proposition erfüllen auch in  $\text{Qui}(P)$  enthalten. Für alle  $p$ , die in  $\text{Qui}(P)$  enthalten sind, gibt es keine ausgehenden must-Transitionen für Outputs oder die interne Aktion  $\tau$ , somit können diese Aktionen entweder keine ausgehende Transition von  $p$  sein oder nur als may-Transition von  $p$  zu einem möglicherweise anderen Zustand führen. Nicht vorhandenen Transitionen in  $P$  sind auch in keiner as-Implementierung von  $P$  enthalten und es gibt mindestens eine Implementierung in  $\text{as-impl}(P)$ , die alle ausgehenden may-Transitionen der lokalen Aktionen von  $p$  nicht implementiert, somit sind alle  $p$  aus  $\text{Qui}(P)$  auch in  $\{p' \in P' \mid P' \in \text{as-impl}(P) \wedge \forall \alpha \in (O \cup \{\tau\}) : p' \not\rightarrow_{P'}^\alpha\}$  enthalten.  $\square$

Für die Erreichbarkeit wird wie im letzten Kapitel ein optimistischer Anzahl der lokalen Erreichbarkeit für die Fehler-Zustände, der Kommunikationsfehler, verwendet. Ruhe ist kein unabwendbarer Fehler, sondern kann durch einen Input repariert werden oder im Fall von vorhandenen may-Output-Transitionen oder may- $\tau$ -Transitionen, durch eine Implementierung dieses Outputs oder des  $\tau$ s. Daraus ergibt sich, dass Ruhe im Vergleich zu Kommunikationsfehler als weniger „schlimmer Fehler“ anzusehen ist. Somit ist ein Ruhe-Zustand ebenso wie ein Fehler-Zustand erreichbar, sobald er durch Outputs und  $\tau$ s erreicht werden kann, jedoch ist nicht jede beliebige Fortsetzung eines Traces, das durch lokale Aktionen zu einem Ruhe-Zustand führt ein Ruhe-Trace.

**Definition 4.3 (fehler- und ruhe-freie Kommunikation).** Zwei MEIOs  $P_1$  und  $P_2$  kommunizieren fehler- und ruhe-frei, wenn keine as-Implementierung ihrer Parallelkomposition  $P_{12}$  einen Fehler- oder Ruhe-Zustand lokal erreichen kann.

**Definition 4.4 (Ruhe-Verfeinerungs-Basisrelation).** Für MEIOs  $P_1$  und  $P_2$  mit der gleichen Signatur wird  $P_1 \sqsubseteq_{\text{Qui}}^B P_2$  geschrieben, wenn ein Fehler- oder Ruhe-Zustand in einer as-Implementierung von  $P_1$  nur dann lokal erreichbar ist, wenn es auch eine as-Implementierung von  $P_2$  gibt, in der ein solcher lokal erreichbar ist. Diese Basisrelation stellt eine Verfeinerung bezüglich Kommunikationsfehlern und Ruhe dar.

$\sqsubseteq_{\text{Qui}}^C$  bezeichnet die vollständig abstrakte Präkongruenz von  $\sqsubseteq_{\text{Qui}}^B$  bezüglich  $\cdot \parallel \cdot$ .

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 grösste Präkongruenz charakterisieren zu können. Wie bereits oben erwähnt, ist Ruhe ein reparierbarer Fehler im Gegensatz zu Kommunikationsfehlern. Es genügt deshalb für Ruhe die strikten Traces ohne Kürzung zu betrachten.

**Definition 4.5 (Ruhe-Traces).** Sei  $P$  ein MEIO und definiere:

- strikte Ruhe-Traces:  $\text{StQT}(P) := \{w \in \Sigma^* \mid p_0 \xRightarrow{w}_P p \in \text{Qui}(P)\}.$

**Proposition 4.6 (*Ruhe-Traces und Implementierung*).** Für ein MEIO  $P$  gilt  $StQT(P) = \{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'} p' \in Qui(P')\}$ .

*Beweis.* Da  $P'$  einen mit  $w$  beschrifteten must-Trace enthält und  $P'$  eine as-Implementierung von  $P$  ist, muss  $w$  als may-Trace bereits in  $P$  möglich gewesen sein (Definition 1.3). Der Zustand  $p'$  kann in  $P'$  auch nur ruhig sein, wenn der entsprechende in Relation stehende Zustand aus  $P$  auch bereits ruhig war. Alle  $w$  aus  $\{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'} p' \in Qui(P')\}$  sind somit also auch in  $StQT(P)$  enthalten. Jeder may-Trace aus  $P$  kann implementiert werden durch eine as-Implementierung. Die Menge  $\text{as-impl}(P)$  enthält alle as-Implementierungen von  $P$ , also auch eine, die den  $w$  may-Trace aus  $StQT(P)$  zu einem mit in  $p$  Relation stehenden Zustand implementiert.  $p$  besitzt keine ausgehenden must-Transitionen für lokale Aktionen, also gibt es unter den as-Implementierungen, die  $w$  entsprechend als must-Trace enthalten auch mindestens eine, die keine ausgehende lokale Aktion an dem zu  $p$  in Relation stehenden Zustand implementiert. Die Behauptung der hier zu beweisenden Proposition gilt.  $\square$

Für  $ET$  und  $EL$  gelten die Definitionen aus dem letzten Kapitel. Es wird nur für Ruhe eine neue Semantik definiert.

**Definition 4.7 (*Ruhe-Semantik*).** Sei  $P$  ein MEIO.

- Die Menge der fehler-gefluteten Ruhe-Traces von  $P$  ist  $QET(P) := StQT(P) \cup ET(P)$ .

Für zwei MEIOs  $P_1, P_2$  mit der gleichen Signatur wird  $P_1 \sqsubseteq_{Qui} P_2$  geschrieben, wenn  $P_1 \sqsubseteq_E P_2$  und  $QET_1 \subseteq QET_2$  gilt.

**Lemma 4.8 (*Ruhe-Zustände unter Parallelkomposition*).**

1. Ein Zustand  $(p_1, p_2)$  aus der Parallelkomposition  $P_{12}$  ist ruhig, wenn es auch die Zustände  $p_1$  und  $p_2$  in  $P_1$  bzw.  $P_2$  sind.
2. Wenn der Zustand  $(p_1, p_2)$  ruhig ist und nicht in  $E_{12}$  enthalten ist, dann sind auch die auf die Teilsysteme projizierten Zustände  $p_1$  und  $p_2$  ruhig.

*Beweis.*

1. Da  $p_1 \in Qui_1$  und  $p_2 \in Qui_2$  gilt, haben diese beiden Zustände jeweils höchstens die Möglichkeit für Input-Transitionen oder Output- und  $\tau$ -may-Transitionen, jedoch keine Möglichkeit für Outputs oder  $\tau$ s als must-Transitionen. Angenommen der Zustand, der durch die Parallelkomposition aus den Zuständen  $p_1$  und  $p_2$  entsteht, ist nicht ruhig, d.h. er hat eine ausgehende must-Transition für einen Output oder ein  $\tau$ .

- Fall 1  $((p_1, p_2) \xrightarrow{\tau}_{12})$ : Ein  $\tau$  ist eine interne Aktion und kann in der Parallelkomposition nicht durch das Verbergen von Aktionen bei der Synchronisation entstehen. Ein  $\tau$  in der Parallelkomposition ist also auch nur möglich, wenn dies bereits für einen der beiden Zustände als must-Transition im der einzelnen Komponente möglich war für einen der Zustände, aus denen  $(p_1, p_2)$  zusammensetzt ist. Jedoch verbietet die Voraussetzung, dass  $p_1$  oder  $p_2$  eine ausgehende  $\tau$  must-Transition haben, deshalb kann auch  $(p_1, p_2)$  keine solche Transition besitzen.
- Fall 2  $((p_1, p_2) \xrightarrow{a}_{12}$  mit  $a \in O_{12} \setminus \text{Synch}(P_1, P_2)$ ): Da es sich bei  $a$  um einen Output handelt, der nicht in  $\text{Synch}(P_1, P_2)$  enthalten ist, kann dieser nicht aus der Synchronisation von zwei Aktionen entstanden sein, sondern muss bereits für  $P_1$  oder  $P_2$  als must-Transition ausführbar gewesen sein. Es gilt also  $\text{oBdA } p_1 \xrightarrow{a}_1$  mit  $a \in O_1$ . Dies ist jedoch aufgrund der Voraussetzung nicht möglich. Somit kann die Parallelkomposition diese Transition für  $(p_1, p_2)$  ebenfalls nicht als must-Transition enthalten.
- Fall 3  $((p_1, p_2) \xrightarrow{a}_{12}$  mit  $a \in O_{12} \cap \text{Synch}(P_1, P_2)$ ): Der Output  $a$  ist in diesem Fall durch Synchronisation von einem Output mit einem Input entstanden. OBdA gilt  $a \in O_1 \cap I_2$ . Für die einzelnen Systeme muss also gelten, dass  $p_1 \xrightarrow{a}_1$  und  $p_2 \xrightarrow{a}_2$ . Die Transition für das System  $P_1$  ist jedoch in der Voraussetzung ausgeschlossen worden. Somit ist es nicht möglich, dass  $P_{12}$  diese in diesem Fall angenommene must-Transition für den Zustand  $(p_1, p_2)$  ausführen kann.

Da alle diese Fälle zu einem Widerspruch mit der Voraussetzung führen folgt, dass bereits die Annahme, dass der Zustand  $(p_1, p_2)$  nicht ruhig ist, falsch war. Es gilt also, dass aus  $p_i \in \text{Qui}_i$  für  $i \in \{1, 2\}$   $(p_1, p_2) \in \text{Qui}_{12}$  folgt.

2. Es gilt  $(p_1, p_2) \in \text{Qui}_{12} \setminus E_{12}$ , somit hat dieser Zustand allenfalls die Möglichkeit für must-Transitionen, die mit Inputs beschriftet sind.

Angenommen  $p_1 \notin \text{Qui}_1$ , dann ist für  $p_1$  entweder eine  $\tau$ -must-Transition oder eine Output-must-Transition möglich.

- Fall 1  $(p_1 \xrightarrow{\tau}_1)$ : Da die Transition für  $P_1$  möglich ist, hat auch  $P_{12}$  die Möglichkeit für eine  $\tau$ -must-Transition. Dies ist jedoch durch die Voraussetzung verboten und somit kann dieser Fall nicht eintreten.
- Fall 2  $(p_1 \xrightarrow{a}_1$  mit  $a \in O_1 \setminus \text{Synch}(P_1, P_2)$ ): Da es sich bei  $a$  um einen must-Output handelt, der nicht zu synchronisieren ist, wird dieser einfach in die Parallelkomposition übernommen. Es müsste also  $(p_1, p_2) \xrightarrow{a}_{12}$  mit  $a \in O_{12}$  gelten, was jedoch verboten ist. Somit kann die Transition für  $P_1$  in diesem Fall nicht möglich sein.
- Fall 3  $(p_1 \xrightarrow{a}_1$  mit  $a \in O_1 \cap \text{Synch}(P_1, P_2)$  und  $p_2 \xrightarrow{a}_2$ ): In diesem Fall ist die Synchronisation des Outputs  $a$  von  $P_1$  mit dem Input  $a$  von  $P_2$  möglich, so dass in der Parallelkomposition der Output  $a$  als must-Transition für  $(p_1, p_2)$



entsteht. Diese must-Transition ist jedoch für  $P_{12}$  nach Voraussetzung nicht erlaubt. Es folgt also auch, dass dieser Fall nicht eintreten kann.

- Fall 4 ( $p_1 \xrightarrow{a}_1$  mit  $a \in O_1 \cap \text{Synch}(P_1, P_2)$  und  $p_2 \not\xrightarrow{a}_2$ ): Da  $P_2$  die  $a$  Transition nicht als must-Transition enthält, handelt es sich hier um einen neuen Kommunikationsfehler. Das  $a$  kann für  $P_2$  kein Output sein, da sonst  $P_1$  und  $P_2$  nicht komponierbar wäre. Der neue Kommunikationsfehler kann dadurch entstehen, dass die Synchronisation des Outputs  $a$  von  $P_1$  mit dem Input  $a$  von  $P_2$  an dieser Stelle nicht möglich ist, oder da der Input  $a$  für  $p_2$  nur als may-Transition vorliegt und somit die Gefahr besteht, dass dieser in einer Implementierung nicht vorhanden ist. Im zweiten Fall synchronisieren die beiden Transitionen zu einer  $a$  Output-may-Transition, die in  $P_{12}$  zulässig wäre. Jedoch wird der Zustand  $(p_1, p_2)$  in beiden Fällen in die Menge  $E_{12}$  der Parallelkomposition eingefügt (Definition 1.2). Dies wurde in der Voraussetzung für den Zustand ausgeschlossen und dieser Fall ist somit nicht möglich.

Alle aufgeführten Fälle führen zu einem Widerspruch mit der Voraussetzung, somit folgt, dass die Annahme bereits falsch war und  $p_1 \in Qui_1$  gelten muss. Analog kann für  $p_2$  argumentiert werden, so dass dann auch  $p_2 \in Qui_2$  folgt.

□

In dem folgenden Satz sind die Punkte 1. und 3. nur zur Vollständigkeit aufgeführt. Sie entsprechen Punkt 1. und 2. aus Satz 3.6.

**Satz 4.9 (Kommunikationsfehler- und Ruhe-Semantik für Parallelkompositionen).** Für zwei komponierbare MEIOs  $P_1, P_2$  und ihre Komposition  $P_{12}$  gilt:

1.  $ET_{12} = \text{cont}(\text{prune}((ET_1 \parallel EL_2) \cup (EL_1 \parallel ET_2)))$ ,
2.  $QET_{12} = (QET_1 \parallel QET_2) \cup ET_{12}$ ,
3.  $EL_{12} = (EL_1 \parallel EL_2) \cup ET_{12}$ .

*Beweis.* Es wird nur der 2. Punkt beweisen.

„ $\subseteq$ “:

Hier muss unterschieden werden, ob ein  $w \in StQT_{12} \setminus ET_{12}$  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  $w \in StQT_{12} \setminus ET_{12}$  betrachtet und es wird versucht dessen Zugehörigkeit zur rechten Menge zu zeigen. Aufgrund von Definition 4.5 weiß man, dass  $(p_{01}, p_{02}) \xRightarrow{w}_{12} (p_1, p_2)$  gilt mit  $(p_1, p_2) \in Qui_{12} \setminus E_{12}$ . Durch Projektion erhält man  $p_{01} \xRightarrow{w_1}_1 p_1$  und  $p_{02} \xRightarrow{w_2}_2 p_2$  mit  $w \in w_1 \parallel w_2$ . Aus  $(p_1, p_2) \in Qui_{12} \setminus E_{12}$  kann mit dem zweiten Punkt von Lemma 4.8 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  $w \in QET_1 \parallel QET_2$  und somit ist  $w$  in der rechten Seite der Gleichung enthalten.

„ $\supseteq$ “:

Es muss wieder danach unterscheiden 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  $w \in QET_1 \parallel QET_2$  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  $w \in w_1 \parallel w_2$  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 3.6 gefolgert werden, dass  $w \in ET_{12}$  gilt und damit ist  $w$  in der linken Seite der Gleichung enthalten.
- Fall 2 ( $w_1 \in StQT_1 \setminus ET_1 \wedge w_2 \in StQT_2 \setminus ET_2$ ): Es gilt in diesem Fall  $p_{01} \xRightarrow{w_1}_1 p_1 \in Qui_1$  und  $p_{02} \xRightarrow{w_2}_2 p_2 \in Qui_2$ . Da  $p_1$  und  $p_2$  in der jeweiligen Ruhe-Menge enthalten sind, ist auch der Zustand, der aus ihnen zusammengesetzt ist, in der Parallelkomposition ruhig, wie bereits im ersten Punkt von Lemma 4.8 gezeigt. Es gilt also für die Komposition  $(p_{01}, p_{02}) \xRightarrow{w}_{12} (p_1, p_2) \in Qui_{12}$  und dadurch ist  $w$  in der linken Seite der Gleichung enthalten, da  $w \in StQT_{12} \subseteq QET_{12}$  gilt.

□

**Korollar 4.10 (Ruhe-Präkongruenz).** Die Relation  $\sqsubseteq_{Qui}$  ist eine Präkongruenz bezüglich  $\cdot \parallel \cdot$ .

*Beweis.* Es muss gezeigt werden: Wenn  $P_1 \sqsubseteq_{Qui} P_2$  gilt, so auch  $P_{31} \sqsubseteq_{Qui} P_{32}$  für jedes komponierbare System  $P_3$ . D.h. es ist zu zeigen, dass aus  $P_1 \sqsubseteq_E P_2$  und  $QET_1 \subseteq QET_2$  sowohl  $P_{31} \sqsubseteq_E P_{32}$  als auch  $QET_{31} \subseteq QET_{32}$  folgt. Dies ergibt sich, wie im Beweis zu Korollar 3.7, aus der Monotonie von  $\cdot \parallel \cdot$  auf Sprachen wie folgt:

- $P_{31} \xrightarrow[\sqsubseteq_E]{\substack{\text{Korollar 3.7} \\ \text{und} \\ P_1 \sqsubseteq_E P_2}} P_{32},$
- $QET_{31} \xrightarrow[4.9.2.]{\substack{ET_{31} \subseteq ET_{32} \\ \text{und} \\ QET_1 \subseteq QET_2}} (QET_3 \parallel QET_1) \cup ET_{31} \\ \xrightarrow[4.9.2.]{\subseteq} (QET_3 \parallel QET_2) \cup ET_{32} \\ \xrightarrow[4.9.2.]{=} QET_{32}.$

□

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

**Lemma 4.11 (Verfeinerung mit Ruhe-Zuständen).** Gegeben sind zwei MEIOs  $P_1$  und  $P_2$  mit der gleichen Signatur. Wenn  $U \parallel P_1 \sqsubseteq_{Qui}^B U \parallel P_2$  für alle Partner  $U$  gilt, dann folgt daraus  $P_1 \sqsubseteq_{Qui} P_2$ .

*Beweis.* Da  $P_1$  und  $P_2$  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  $P_1 \sqsubseteq_{Qui} P_2$  gilt, müssen die folgenden Punkte nachgewiesen werden:

- $P_1 \sqsubseteq_E P_2$ ,
- $QET_1 \subseteq QET_2$ .

In Lemma 3.8 wurde bereits etwas Ähnliches gezeigt, jedoch wurde dort als Voraussetzung  $U \parallel P_1 \sqsubseteq_E^B U \parallel P_2$  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 der Komponente aussagt, kann der Beweis aus Lemma 3.8 nicht verwendet werden. Es kann also aus der lokalen Erreichbarkeit eines Fehler-Zustands der Art Kommunikationsfehler in  $P'_1$  und dem relationalen Zusammenhang von  $P'_1 \sqsubseteq_{Qui}^B P'_2$  nur geschlossen werden, dass in  $P'_2$  auch ein Fehler lokal erreichbar ist, jedoch kann dieser Fehler ein Kommunikationsfehler oder Ruhe sein. Analog verhält es sich, wenn in  $P'_1$  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  $U \parallel P_1 \sqsubseteq_{Qui}^B U \parallel P_2$  gezeigt, dass  $ET_1 \subseteq ET_2$  gilt. Da beide  $ET$ -Mengen unter  $\text{cont}$  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 Fehler-Zustand in  $P_1$ . Für  $U$  wird ein Transitionssystem verwendet, das nur aus dem Startzustand, einer must-Schleife für alle Inputs  $x \in I_U$  und einer must-Schlinge für  $\tau$  besteht. Somit kann  $P_1$  die im Prinzip gleichen Fehler-Zustände der Art Kommunikationsfehler lokal erreichen wie  $U \parallel P_1$ . Es gibt also einen lokal erreichbaren Zustand von  $U \parallel P_1$ , der in  $E_{U \parallel P_1}$  enthalten ist. Mindestens eine as-Implementierung von  $U \parallel P_1$  wird diesen Kommunikationsfehler implementieren und somit muss es auch mindestens eine as-Implementierung von  $U \parallel P_2$  geben, die ebenfalls einen lokal erreichbaren Fehler hat. Dieser Fehler kann jedoch sowohl an der fehlenden Möglichkeit für lokale Aktionen in einem Zustand liegen, wie an einem Zustand, der in  $E_{U \parallel P_2}$  enthalten ist. Falls es sich in der as-Implementierung um lokal erreichbare Ruhe handelt, dürfen in dem in Relation stehenden Zustand in  $U \parallel P_2$  die lokalen Aktionen keine must-Transitionen gewesen sein, da dies sonst einen Widerspruch zu Definition 1.3 1. darstellen würde. Ein lokal erreichbarer Kommunikationsfehler in der as-Implementierung würde hingegen durch Definition 1.3 ebenfalls zu einem lokal erreichbaren Kommunikationsfehler in  $U \parallel P_2$  führen. Es muss also in allen Fällen ein Fehler in  $U \parallel P_2$  lokal erreichbar sein. Hierbei kann es sich jedoch nur um

einen lokal erreichbaren Kommunikationsfehler handeln, da es in der Komposition mit  $U$  keine Ruhe-Zustände geben kann. Da  $U$  keinen Fehler-Zustand und auch keine fehlenden Input-Möglichkeiten enthält, kann der Kommunikationsfehler nur von  $P_2$  geerbt sein. Somit muss in  $S_2$  ein Fehler-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):

- $U = \{p_0, p_1, \dots, p_{n+1}\}$ ,
- $p_0 U = p_0$ ,
- $\rightarrow_U = \{(p_i, x_{i+1}, p_{i+1}) \mid 0 \leq i \leq n\}$   
 $\cup \{(p_i, x, p_{n+1}) \mid x \in I_U \setminus \{x_{i+1}\}, 0 \leq i \leq n\}$   
 $\cup \{(p_{n+1}, x, p_{n+1}) \mid x \in I_U\}$   
 $\cup \{(p_i, \tau, p_i) \mid 0 \leq i \leq n+1\}$ ,
- $E_U = \emptyset$ .

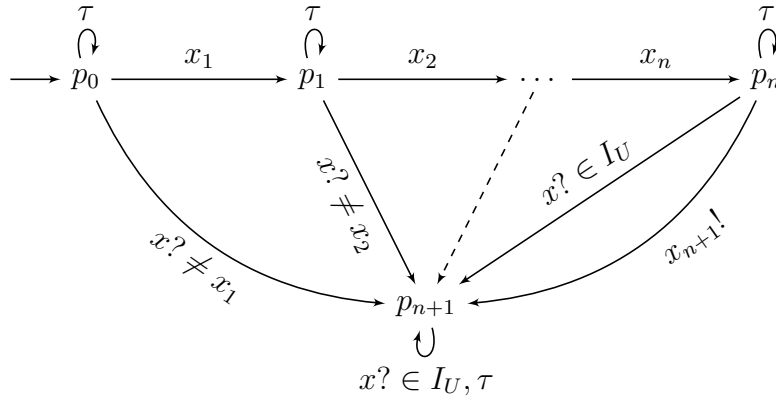


Abbildung 4.1:  $x? \neq x_i$  steht für alle  $x \in I_U \setminus \{x_i\}$

Die Menge der Ruhe-Zustände der hier betrachteten  $U$ s ist leer. Da im Vergleich zum Transitionssystem in Abbildung 3.1 nur die  $\tau$ -Schlingen ergänzt wurden, ändert sich nichts an den Fällen 2a) und 2b). Die Begründungen, wieso in den beiden Fällen  $\varepsilon \in PrET(P')$  für ein  $P'$  aus  $as\text{-}impl(U \parallel P_1)$  gilt, bleibt also analog zum Beweis des ersten Punktes von Lemma 3.8. Durch die must- $\tau$ -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 Kommunikationsfehler in einer as-Implementierung von  $U \parallel P_1$  auf die lokale Erreichbarkeit eines Fehler-Zustandes der Art Kommunikationsfehler in einer as-Implementierung von  $U \parallel P_2$  und somit auch in  $U \parallel P_2$  geschlossen werden. Die weitere Argumentation verläuft dann analog zu Fall 2, derselben Inklusion im Beweis zu Lemma 3.8. Da  $\tau$ s nur interne Aktionen einer einzelnen Komponente sind,

verändert sich auch nichts an den Traces über die argumentiert wird. Es können zwar möglicherweise  $\tau$ -Transitionen ausgeführt werden, diese können jedoch weder zu einem Fehler führen noch beeinflussen, dass ein anderer Trace nicht ausgeführt werden kann.

Nun wird mit dem zweiten Unterpunkt des ersten Beweispunktes begonnen. Genau wie im Beweis zu 3.8 ist hier jedoch aufgrund des bereits geführten Beweisteils nur noch  $L_1 \setminus ET_1 \subseteq EL_2$  zu zeigen. Es wird also für ein beliebig gewähltes  $w \in L_1 \setminus ET_1$  gezeigt, dass dieses auch in  $EL_2$  enthalten ist.

- Fall 1 ( $w = \varepsilon$ ): Ebenso wie in 3.8 gilt auch hier, dass  $\varepsilon$  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  $\tau$ -must-Schleifen an den Zuständen des Transitionssystems vom Beweis des zweiten Punktes aus Lemma 3.8 ab. Somit ist der Partner  $U$  dann wie folgt definiert (siehe dazu auch Abbildung 4.2):

- $U = \{p_0, p_1, \dots, p_n, p\}$ ,
- $p_0 U = p_0$ ,
- $\rightarrow_U = \{(p_i, x_{i+1}, p_{i+1}) \mid 0 \leq i < n\}$   
 $\cup \{(p_i, x, p) \mid x \in I_U \setminus \{x_{i+1}\}, 0 \leq i < n\}$   
 $\cup \{(p_i, \tau, p_i) \mid 0 \leq i \leq n\}$   
 $\cup \{(p, \alpha, p) \mid \alpha \in I_U \cup \{\tau\}\},$
- $E_U = \{p_n\}$ .

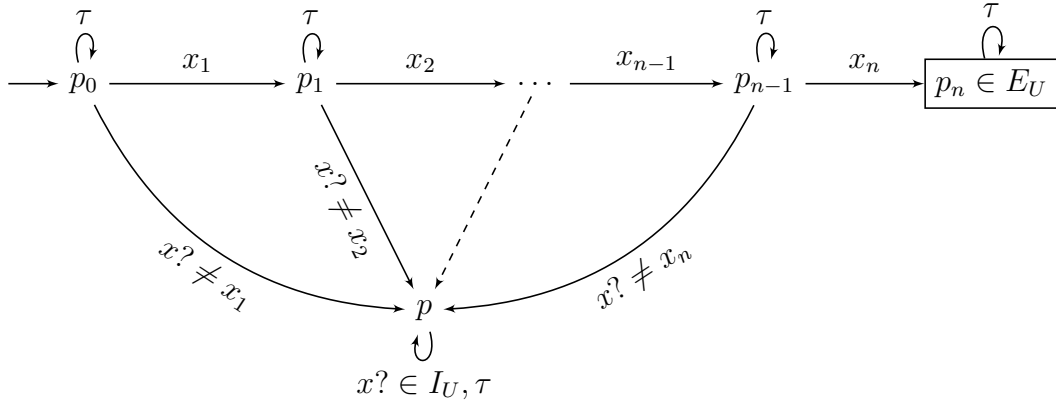


Abbildung 4.2:  $x? \neq x_i$  steht für alle  $x \in I_U \setminus \{x_i\}$ ,  $p_n$  ist der einzige Error-Zustand

Da durch die  $\tau$ -must-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 3.8. 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  $QET_1 \subseteq QET_2$ . Die Inklusion kann jedoch, analog zum Beweis der Inklusion der Kommunikationsfehlergefluteten Sprache, noch weiter eingeschränkt werden. Da bereits bekannt ist, dass  $ET_1 \subseteq ET_2$  gilt, muss nur noch  $StQT_1 \setminus ET_1 \subseteq QET_2$  gezeigt werden.

Es wird ein  $w \in StQT_1 \setminus ET_1$  gewählt und gezeigt, dass dieses auch in  $QET_2$  enthalten ist.

Durch die Wahl des  $w$ s wird vom Startzustand von  $P_1$  durch das Wort  $w$  ein ruhiger Zustand erreichbar. Dies hat nur Auswirkungen auf die Parallelkomposition  $U \parallel P_1$ , 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):

- $U = \{p_0, p_1, \dots, p_n, p\}$ ,
- $p_{0U} = p_0$ ,
- $\rightarrow_U = \{(p_i, x_{i+1}, p_{i+1}) \mid 0 \leq i < n\}$   
 $\cup \{(p_i, x, p) \mid x \in I_U \setminus \{x_{i+1}\}, 0 \leq i < n\}$   
 $\cup \{(p_i, \tau, p_i) \mid 0 \leq i < n\}$   
 $\cup \{(p_n, x, p) \mid x \in I_U\}$   
 $\cup \{(p, \alpha, p) \mid \alpha \in I_U \cup \{\tau\}\},$
- $E_U = \emptyset$ .

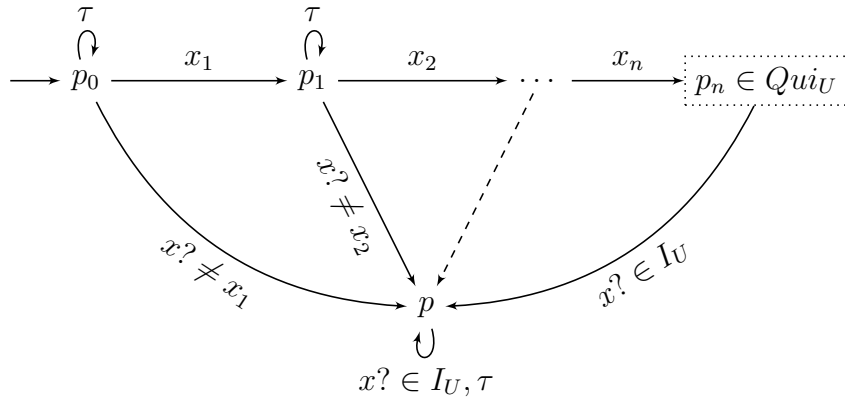


Abbildung 4.3:  $x? \neq x_i$  steht für alle  $x \in I_U \setminus \{x_i\}$ ,  $p_n$  ist der einzige Ruhe-Zustand

Falls für das betrachtete  $w = \varepsilon$  gilt, reduziert sich der Partner  $U$  auf den Zustand  $p_n = p_0$  und den Zustand  $p$ . Es ist also in diesem Fall der Startzustand gleich dem ruhigen Zustand.

Allgemein ist der Zustand  $p_n$  aus  $U$  der einzig ruhige Zustand in  $U$ . Es gilt wegen des

ersten Punktes von Lemma 4.8, dass auch in der Parallelkomposition  $U \parallel P_1$  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  $w \in O_{U \parallel P_1}$  und  $w \in StQT(U \parallel P_1)$ . 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  $P_1$  zurückgelegt wird, kein Kommunikationsfehler lokal erreicht werden. Es kann also weder von  $P_1$  noch von  $U$  ein Kommunikationsfehler auf diesem Weg geerbt werden und durch den Aufbau von  $U$  kann auch kein neuer Kommunikationsfehler in der Parallelkomposition beider Systeme entstehen. Da ein Ruhe-Zustand in  $U \parallel P_1$  lokal erreichbar ist, muss dies auch in einer as-Implementierung von  $U \parallel P_1$  der Fall sein. Daraus folgt mit der Relation  $\sqsubseteq_{Qui}^B$ , dass es auch ein Element aus  $as\text{-}impl(U \parallel P_2)$  geben muss bei dem ein Fehler lokal erreichbar ist. Mit der Begründung aus der ersten Inklusion in diesem Beweis muss also der entsprechende Fehler aus der as-Implementierung auf einen in  $U \parallel P_2$  zurückführbar sein und somit auch dort lokal erreichbar sein. Jedoch kann zunächst keine Aussage getroffen werden, ob das  $w$  in  $U \parallel P_2$  ausführbar ist und ob es sich bei dem Fehler um Ruhe oder einen Kommunikationsfehler handelt.

- Fall a) ( $\varepsilon \in ET(U \parallel P_2)$ ): Es handelt sich bei dem lokal erreichbaren Fehler um einen Kommunikationsfehler. Es ist somit nicht relevant, ob  $w$  ausführbar ist. Der Fehler-Zustand kann sowohl von  $P_2$  geerbt sein, wie auch durch fehlende Input-must-Transitionen als neuer Kommunikationsfehler in der Parallelkomposition entstanden sein. Es gilt also, dass bereits in  $P_2$  ein Präfix von  $w$  in  $ET_2$  enthalten ist, wegen des Beweises des ersten Punktes aus Lemma 3.8 und das  $U$  nur neue Kommunikationsfehler auf dem Trace  $w$  zulässt. Da die Menge  $ET$  unter  $\text{cont}$  abgeschlossen ist, gilt also auch  $w \in ET_2 \subseteq QET_2$ .
- Fall b) (Ruhe-Zustand lokal erreichbar in  $U \parallel P_2$  und  $\varepsilon \notin ET(U \parallel P_2)$ ): Da in  $U$  nur durch  $w$  ein ruhiger Zustand erreicht werden kann, muss es sich bei dem lokal erreichbaren Ruhe-Zustand in  $U \parallel P_2$  um einen handeln, der mit  $w$  erreicht werden kann. Mit Lemma 4.8 kann somit gefolgert werden, dass auch in  $P_2$  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 größte Präkongruenz bezüglich  $\cdot \parallel \cdot$  charakterisiert wurde, die in  $\sqsubseteq_{Qui}^B$  enthalten ist

**Satz 4.12 (Vollständige Abstraktheit für Ruhe-Semantik).** Seien  $P_1$  und  $P_2$  zwei MEIOs mit der selben Signatur. Dann gilt  $P_1 \sqsubseteq_{Qui}^C P_2 \Leftrightarrow P_1 \sqsubseteq_{Qui} P_2$ .

*Beweis.*

„ $\Leftarrow$ “: Nach Definition gilt  $w \in QET(P)$  mit  $w \in O(P)^*$  genau dann, wenn in  $P$  ein Ruhe-Zustand oder ein Fehler-Zustand der Art Kommunikationsfehler lokal erreichbar ist.  $P_1 \sqsubseteq_{Qui} P_2$  impliziert, dass  $w \in QET_2$  gilt, wenn  $w \in QET_1$  gilt. Somit ist ein Ruhe-Zustand oder ein Fehler-Zustand nur dann in  $P_1$  lokal erreichbar, wenn auch ein

solcher in  $P_2$  lokal erreichbar ist. Daraus folgt, dass es as-Implementierungen von  $P_1$  und  $P_2$  gibt, die die analogen Fehler lokal erreichen. Somit gilt  $P_1 \sqsubseteq_{Qui}^B P_2$ . Es ist also  $\sqsubseteq_{Qui}$  in  $\sqsubseteq_{Qui}^B$  enthalten. Im Korollar 4.10 wurde festgestellt, dass  $\sqsubseteq_{Qui}$  eine Präkongruenz ist. Da jedoch  $\sqsubseteq_{Qui}^C$  nach Definition 4.4 die größte Präkongruenz bezüglich  $\cdot\|\cdot$  ist, die in  $\sqsubseteq_{Qui}^B$  enthalten ist, muss  $\sqsubseteq_{Qui}$  in  $\sqsubseteq_{Qui}^C$  enthalten sein. Es folgt also aus  $P_1 \sqsubseteq_{Qui} P_2$ , dass auch  $P_1 \sqsubseteq_{Qui}^C P_2$  gilt.

„ $\Rightarrow$ “: Durch die Definition von  $\sqsubseteq_{Qui}^C$  als Präkongruenz in 4.4 folgt aus  $P_1 \sqsubseteq_{Qui}^C P_2$ , dass  $U\|P_1 \sqsubseteq_{Qui}^C U\|P_2$  für alle MEIOs  $U$  gilt, die mit  $P_1$  komponierbar sind. Da  $\sqsubseteq_{Qui}^C$  nach Definition in  $\sqsubseteq_{Qui}^B$  enthalten ist, folgt auch die Gültigkeit von  $U\|P_1 \sqsubseteq_{Qui}^B U\|P_2$  für alle diese MEIOs  $U$ . Mit Lemma 4.11 folgt dann  $P_1 \sqsubseteq_{Qui} P_2$ .  $\square$

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.

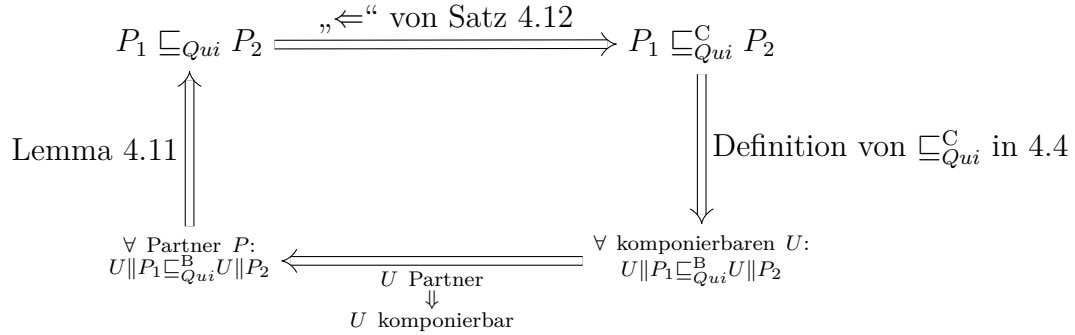


Abbildung 4.4: Folgerungskette

Angenommen man definiert, dass  $P_1$   $P_2$  verfeinern soll genau dann, wenn für alle Partner MEIOs  $U$  für die  $P_2$  fehler- und ruhe-frei mit  $U$  kommuniziert, folgt, dass  $P_2$  ebenfalls fehler- und ruhe-frei mit  $U$  kommuniziert. Dann wird auch diese Verfeinerung durch  $\sqsubseteq_{Qui}$  charakterisiert.

Aus Satz 4.12 und Lemma 4.11 erhält man das folgende Korollar.

**Korollar 4.13.** *Es gilt:  $P_1 \sqsubseteq_{Qui} P_2 \Leftrightarrow U\|P_1 \sqsubseteq_{Qui}^B U\|P_2$  für alle Partner  $U$ .*



## 5 Verfeinerungen für Kommunikationsfehler-, Ruhe- und Divergenz-Freiheit

In diesem Kapitel soll die Menge der betrachteten Zustände noch einmal erweitert werden. Somit werden dann Fehler-, Ruhe- und Divergenz-Zustände betrachtet.

**Definition 5.1 (*Divergenz*).** Ein Divergenz-Zustand ist ein Zustand in einem MEIO  $P$ , der eine unendliche Folge an  $\tau$ s ausführen kann via may-Transitionen.

Die Menge  $Div(P)$  besteht aus all diesen divergenten Zuständen des MEIOs  $P$ .

**Proposition 5.2 (*Divergenz und Implementierung*).** Für ein MEIO  $P$  gilt  $Div(P) = \{p' \in P' \mid P' \in \text{as-impl}(P) \wedge P' \text{ kann von } p' \text{ aus eine unendliche Folge von } \tau\text{s ausführen}\}$ .

*Beweis.* Falls von einem  $p'$  in einem  $P' \in \text{as-impl}(P)$  eine unendliche  $\tau$ -Folge ausführbar ist, dann ist dies via must-Transitionen möglich. Diese Transitionen müssen jedoch aus möglichen may-Transitionen in  $P$  folgen. Deshalb war auch der Zustand, der zu  $p'$  in Simulations-Relation nach Definition 1.3 steht, auch bereits divergent und in  $Div(P)$  enthalten. Für jedes  $p$  aus  $Div(P)$  gibt es eine unendliche Folge von  $\tau$ s, die via may-Transitionen möglich ist in  $P$ . Also gibt es auch in  $\text{as-impl}(P)$  mindestens eine Implementierung, die alle diese beteiligten may-Transitionen als must-Transitionen implementiert und einen zu  $p$  analogen Zustand als divergent enthält und somit in die Menge  $\{p' \in P' \mid P' \in \text{as-impl}(P) \wedge P' \text{ kann von } p' \text{ aus eine unendliche Folge von } \tau\text{s ausführen}\}$  einfügt.  $\square$

Die unendliche Folge an  $\tau$ s kann durch eine Schleife an einem durch  $\tau$ s erreichbaren Zustand ausführbar sein oder durch einen Weg, der mit  $\tau$ s ausführbar ist, mit dem unendliche viele Zustände durchlaufen werden. Es ist jedoch zu beachten, dass ein Zustand, von dem aus unendlich viele Zustände durch  $\tau$ s erreichbar sind, nicht divergent sein muss. Es ist auch möglich, dass dieser Zustand eine unendliche Verzweigung hat und somit keine unendlichen Folgen an  $\tau$ s 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 Fehler-Zustand des Types Kommunikationsfehler.

**Definition 5.3 (fehler-, ruhe- und divergenz-freie Kommunikation).** Zwei MEIOs  $P_1$  und  $P_2$  kommunizieren fehler-, ruhe- und divergenz-frei, wenn keine as-Implementierung ihrer Parallelkomposition  $P_{12}$  einen Fehler-, Ruhe- oder Divergenz-Zustand lokal erreichen kann.

**Definition 5.4 (Divergenz-Verfeinerungs-Basisrelation).** Für MEIOs  $P_1$  und  $P_2$  mit der gleichen Signatur wird  $P_1 \sqsubseteq_{Div}^B P_2$  geschrieben, wenn ein Fehler-, Ruhe- oder Divergenz-Zustand in einer as-Implementierung von  $P_1$  nur dann lokal erreichbar ist, wenn es auch eine as-Implementierung von  $P_2$  gibt, in der ein solcher lokal erreichbar ist. Die Basisrelation stellt eine Verfeinerung bezüglich Fehler, Ruhe und Divergenz dar.  $\sqsubseteq_{Div}^C$  bezeichnet die vollständig abstrakte Präkongruenz von  $\sqsubseteq_{Div}^B$  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 Kommunikationsfehler und dass das Divergieren eines Systems nicht mehr verhinderbar ist, sobald ein divergenter Zustand lokal erreichbar ist, kommt für die Divergenz-Traces wieder die prune-Funktion zum Einsatz. Ein System, das unendliche viele  $\tau$ s ausführen kann, ist von außen nicht von so einem System zu unterscheiden, das einen Fehler-Zustand der Art Kommunikationsfehler erreicht. Somit wird in den Trace-Mengen auch nicht zwischen Kommunikationsfehler-Traces und Divergenz-Traces explizit unterschieden. Dadurch genügt es nicht mehr nur mit den Kommunikationsfehler-Traces die Sprache zu fluten, sondern es muss sowohl mit den Kommunikationsfehler-Traces wie auch den Divergenz-Traces geflutet werden. Ebenso werden die strikten Ruhe-Traces mit diesen beiden Trace-Mengen geflutet.

**Definition 5.5 (Divergenz-Traces).** Sei  $P$  ein MEIO und definiere:

- strikte Divergenz-Traces:  $StDT(P) := \{w \in \Sigma^* \mid p_0 \xRightarrow{w}_P p \in Div(P)\}$ ,
- gekürzte Divergenz-Traces:  $PrDT(P) := \bigcup \{\text{prune}(w) \mid w \in StDT(P)\}$ .

Analog zu den Propositionen 3.4 und 4.6 gibt es hier auch eine Proposition, die die Divergenz-Traces eines MEIOs mit den Divergenz-Traces seiner as-Implementierungen verbindet. Die Begründung würde analog wie die der beiden Propositionen der vorangegangenen Kapitel laufen, in Kombination mit den Argumenten des Beweises zur Proposition 5.2 in diesem Kapitel.

**Proposition 5.6 (Divergenz-Traces und Implementierung).** Für ein MEIO  $P$  gilt  $StDT(P) = \{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \xRightarrow{w}_{P'} p' \in Div(P')\}$ .

Da die Ruhe-Traces mit den Kommunikationsfehler- und Divergenz-Traces geflutet werden sollen, kann die Ruhe-Semantik nicht aus dem letzten Kapitel übernommen werden auch die geflutete Sprache aus dem Kommunikationsfehler-Kapitel kann nicht übernommen werden. Nur die Kommunikationsfehler-Traces  $ET$  können ohne Veränderung auch in diesem Kapitel verwendet werden. Jedoch werden diese Traces im weiteren Verlauf nur innerhalb der größeren Trace-Menge  $EDT$  relevant sein.

**Definition 5.7 (Kommunikationsfehler-, Ruhe- und Divergenz-Semantik).**

Sei  $P$  ein MEIO.

- Die Menge der Divergenz-Traces von  $P$  ist  $DT(P) := \text{cont}(\text{PrDT}(P))$ .
- Die Menge der Fehler-Divergenz-Traces von  $P$  ist  $EDT(P) := ET(P) \cup DT(P)$ .
- Die Menge der Kommunikationsfehler-divergenz-gefluteten Ruhe-Traces von  $P$  ist  $QDT(P) := \text{StQT}(P) \cup EDT(P)$ .
- Die Menge der Kommunikationsfehler-divergenz-gefluteten Sprache von  $P$  ist  $EDL(P) := L(P) \cup EDT(P)$ .

Für zwei MEIOs  $P_1, P_2$  mit der gleichen Signatur schreibt man  $P_1 \sqsubseteq_{Div} P_2$ , wenn  $EDT_1 \subseteq EDT_2, QDT_1 \subseteq QDT_2$  und  $EDL_1 \subseteq EDL_2$  gilt.

$\sqsubseteq_{Div}$  ist somit keine Einschränkung von  $\sqsubseteq_E$  so wie  $\sqsubseteq_{Qui}$ . Es können Systeme mit einem Kommunikationsfehler nicht von Systemen mit Divergenz unterschieden werden. Da die Basisrelation zwischen diesen Fehler-Arten auch keine Unterscheidung kennt, muss eine sinnvolle Relation diese Eigenschaft auch übernehmen, so wie  $\sqsubseteq_{Div}$  dies tut.

**Satz 5.8 (Kommunikationsfehler-, Ruhe- und Divergenz-Semantik für Parallelkompositionen).** Für zwei komponierbare MEIOs  $P_1, P_2$  und ihre Komposition  $P_{12}$  gilt:

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

*Beweis.*

1. „ $\subseteq$ “:

Da beide Seiten der Gleichung unter  $\text{cont}$  abgeschlossen sind, genügt es ein präfix-minimales Element  $w$  zu betrachten. Es muss hier unterschieden werden, ob  $w \in ET_{12}$  oder  $w \in DT_{12} \setminus ET_{12}$  betrachtet wird. Im ersten Fall ist das  $w$  in der rechten Seite der Gleichung enthalten wegen des Beweises des ersten Punktes von Satz 3.6 und da  $ET(P) \subseteq EDT(P)$  und  $EL(P) \subseteq EDL(P)$  gilt. Deshalb wird im weiteren Verlauf dieses Beweises davon ausgegangen, dass  $w \in DT_{12} \setminus ET_{12}$  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  $w \in \text{PrDT}_{12} \setminus ET_{12}$ . Aus der Definition 5.7 weiß man, dass ein  $v \in O_{12}^*$  existiert, sodass  $(p_{01}, p_{02}) \xRightarrow{w}_{12} (p_1, p_2) \xRightarrow{v}_{12} (p'_1, p'_2)$  gilt mit  $(p'_1, p'_2) \in \text{Div}_{12}$ . Durch die Projektion auf die Transitionssysteme  $P_1$  und  $P_2$  erhält man  $p_{01} \xRightarrow{w_1}_1 p_1 \xRightarrow{v_1}_1 p'_1$  und  $p_{02} \xRightarrow{w_2}_2 p_2 \xRightarrow{v_2}_2 p'_2$  mit  $w \in w_1 \parallel w_2$  und  $v \in v_1 \parallel v_2$ . Aus  $(p'_1, p'_2) \in \text{Div}_{12}$  folgt, dass oBdA  $p'_1 \in \text{Div}_1$  gilt, d.h.  $w_1 v_2 \in \text{StDT}_1 \subseteq EDT_1$ . Da  $p_{02} \xRightarrow{w_2 v_2}_2$  gilt, erhält man  $w_2 v_2 \in EDL_2$ . Somit gilt insgesamt  $wv \in EDT_1 \parallel EDL_2$  und da  $v \in O_{12}^*$ , ist  $w$  in der rechten Seite der Gleichung enthalten und es folgt insgesamt  $\text{prune}(wv) = \text{prune}(w)$ .

1. „ $\supseteq$ “:

Es wird ebenso wie oben nur ein präfix-minimales  $x$  betrachtet wegen des Abschlusses beider Seiten der Gleichung unter  $\text{cont}$ . Es wird also für ein beliebiges  $x \in \text{prune}((EDT_1 \parallel EDL_2) \cup (EDL_1 \parallel EDT_2))$  gezeigt, dass dieses oder eines seiner Präfixe auch in  $EDT_{12}$  enthalten ist. Da das  $x$  aus der  $\text{prune}$ -Funktion entstanden ist, lässt sich ein  $y$  aus  $O_{12}^*$  finden, sodass  $xy \in (EDT_1 \parallel EDL_2) \cup (EDL_1 \parallel EDT_2)$ . Es wird nun noch vorausgesetzt, dass  $\text{oBdA } xy \in EDT_1 \parallel EDL_2$  gilt, d.h. es existiert  $w_1 \in EDT_1$  und  $w_2 \in EDL_2$  mit  $xy \in w_1 \parallel w_2$ .

Die folgende Argumentation läuft analog zu der im Beweis der Inklusion  $ET_{12} \supseteq \text{cont}(\text{prune}((ET_1 \parallel EL_2) \cup (EL_1 \parallel ET_2)))$  aus Satz 3.6. Es muss dazu nur jeweils an den Stellen, an denen  $\text{PrET}(P) \cup \text{MIT}(P)$  steht auch noch eine Vereinigung mit  $\text{PrDT}(P)$  vorgenommen werden. Für Fall I und II aus dem Beweis der oben genannten Inklusion von Satz 3.6 ist jeweils kein weiterer Unterfall für  $v'_2$  notwendig da, wenn  $v'_2$  nicht ausführbar ist, bereits ein Fehler-Zustand der Art Kommunikationsfehler 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 nicht an der Ausführbarkeit ändert. Am Ende ist ein zusätzlicher Fall für  $v_1 \in \text{PrDT}_1$  zu ergänzen:

- Fall III ( $v_1 \in \text{PrDT}_1$ ): Es existiert ein  $u_1$  aus  $O_1^*$ , sodass  $p_{01} \xRightarrow{v_1}_1 p_1 \xRightarrow{u_1}_1 p'_1$  mit  $p'_1 \in \text{Div}_1$  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  $v_2 \in \text{MIT}_2$  kann somit  $a = b$  gelten und damit wäre  $v_2 = v'_2$ . Dieser Fall verläuft jedoch analog zu Fall Ic) aus dem Beweis der oben genannten Inklusion von Satz 3.6 und wird somit hier nicht weiter betrachtet. Deshalb gilt für alle im folgenden betrachteten Fälle  $p_{02} \xRightarrow{v'_2}_2 p_2$  mit  $(p_{01}, p_{02}) \xRightarrow{v'}_2$ .
  - Fall IIIa) ( $u_2 \in (O_1 \cap I_2)^*, c \in (O_1 \cap I_2)$ , sodass  $u_2c$  ein Präfix von  $u_1|_{I_2}$  mit  $p_2 \xRightarrow{u_2}_2 p'_2 \not\xrightarrow{c}_2$ ): Für ein Präfix von  $u'_1c$  von  $u_1$  mit  $(u'_1c)|_{I_2} = u_2c$  weiß man, dass  $p_1 \xRightarrow{u'_1}_1 p''_1 \xrightarrow{c}_1$ . Somit gilt  $u'_1 \in u_1 \parallel u_2$  und  $(p_1, p_2) \xRightarrow{u'_1}_{12} (p''_1, p'_2) \in E_{12}$ , da für  $P_2$  der entsprechende Input fehlt, der mit dem Output von  $c$  von  $P_1$  zu koppeln wäre. Es handelt sich also um einen neuen Kommunikationsfehler. Es wird  $v := \text{prune}(v'u'_1) \in \text{PrET}_{12}$  gewählt, dies ist ein Präfix von  $v'$ , da  $u_1 \in O_1^*$ .
  - Fall IIIb) ( $p_2 \xRightarrow{u_2}_2 p'_2$  mit  $u_2 = u_1|_{I_2}$ ): Somit ist  $u_1 \in u_1 \parallel u_2$  und  $(p_1, p_2) \xRightarrow{u_1}_{12} (p'_1, p'_2) \in \text{Div}_{12}$ , da  $p_1 \in \text{Div}_1$ .  $P_{12}$  hat also die Divergenz von  $P_1$  geerbt. Es wird nun  $v := \text{prune}(v'u_1) \in \text{PrDT}_{12}$  gewählt, das wiederum ein Präfix von  $v'$  ist.

2. „ $\subseteq$ “:

Diese Inklusionsrichtung kann analog zum Beweis derselben Inklusionsrichtung des zweiten Punktes von Satz 4.9 gezeigt werden. Es muss dabei nur in der Argumentation die Menge  $ET_{12}$  durch die Menge  $EDT_{12}$  und die Mengen  $QET(P)$  durch die Mengen

$QDT(P)$  für die entsprechenden Transitionssysteme  $P$  ersetzt werden. Dadurch kann ebenso gefolgert werden, dass im Fall  $w \in StQT_{12} \setminus EDT_{12}$  der erreichte Zustand  $(p_1, p_2)$  kein Kommunikationsfehler sein kann, da  $ET_{12} \subseteq EDT_{12}$  gilt und somit lässt sich auch hier der zweite Punkt von Lemma 4.8 anwenden.

2. „ $\supseteq$ “:

Es muss wieder danach unterschieden werden, aus welcher Menge das betrachtete Element stammt. Falls  $w$  ein Element von  $EDT_{12}$  ist, folgt die Zugehörigkeit zur linken Seite der Gleichung direkt. Somit wird für den weiteren Verlauf dieses Beweises davon ausgegangen, dass  $w \in QDT_1 \parallel QDT_2$  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, existierten für  $w_1$  und  $w_2$  mit  $w \in w_1 \parallel w_2$  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 ( $w_1 \in StQT_1 \setminus EDT_1 \wedge w_2 \in StQT_2 \setminus EDT_2$ ): Dieser Fall läuft analog zu Fall 2 derselben Inklusionsrichtung des Beweises von Satz 4.9. Hierfür muss die Menge  $QET_{12}$  durch  $QDT_{12}$  ersetzt werden.

3.:

Durch die Definition 5.7 ist klar, dass  $L_i \subseteq EDL_i$  und  $EDT_i \subseteq EDL_i$  gilt. Die Argumentation wird von der rechten Seite der Gleichung aus begonnen:

$$\begin{aligned}
 (EDL_1 \parallel EDL_2) \cup EDT_{12} &\stackrel{5.7}{=} ((L_1 \cup EDT_1) \parallel (L_2 \cup EDT_2)) \cup EDT_{12} \\
 &= (L_1 \parallel L_2) \cup \underbrace{(L_1 \parallel EDT_2)}_{\substack{\subseteq (EDL_1 \parallel EDT_2) \\ \stackrel{1.}{\subseteq} EDT_{12}}} \cup \underbrace{(EDT_1 \parallel L_2)}_{\substack{\subseteq (EDT_1 \parallel EDL_2) \\ \stackrel{1.}{\subseteq} EDT_{12}}} \\
 &\quad \cup \underbrace{(EDT_1 \parallel EDT_2)}_{\substack{\subseteq (EDL_1 \parallel EDT_2) \\ \stackrel{1.}{\subseteq} EDT_{12}}} \cup EDT_{12} \\
 &= (L_1 \parallel L_2) \cup EDT_{12} \\
 &\stackrel{2.2}{=} L_{12} \cup EDT_{12} \\
 &\stackrel{5.7}{=} EDL_{12}.
 \end{aligned}$$

□

**Korollar 5.9 (Divergenz-Präkongruenz).** Die Relation  $\sqsubseteq_{Div}$  ist eine Präkongruenz bezüglich  $\cdot \parallel \cdot$ .

*Beweis.* Um zu zeigen, dass es sich bei  $\sqsubseteq_{Div}$  um eine Präkongruenz handelt, muss nachgewiesen werden, dass aus  $P_1 \sqsubseteq_{Div} P_2$  auch  $P_{31} \sqsubseteq_{Div} P_{32}$  für jedes komponierbare

System  $P_3$  folgt. 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 3.7 und 4.10, aus der Monotonie von  $\text{cont}$ ,  $\text{prune}$  und  $\cdot\|\cdot$  auf Sprachen wie folgt:

- $EDT_{31} \stackrel{5.8}{=}^1 \text{cont}(\text{prune}((EDT_3\|EDL_1) \cup (EDL_3\|EDT_1)))$   
 $\stackrel{EDT_1 \subseteq EDT_2}{\text{und}} \stackrel{EDL_1 \subseteq EDL_2}{\subseteq} \text{cont}(\text{prune}((EDT_3\|EDL_2) \cup (EDL_3\|EDT_2)))$   
 $\stackrel{5.8}{=}^1 EDT_{32},$
- $QDT_{31} \stackrel{5.8}{=}^2 (QDT_3\|QDT_1) \cup EDT_{31}$   
 $\stackrel{EDT_{31} \subseteq EDT_{32}}{\text{und}} \stackrel{QDT_1 \subseteq QDT_2}{\subseteq} (QDT_3\|QDT_2) \cup EDT_{32}$   
 $\stackrel{5.8}{=}^2 QDT_{32}.$
- $EDL_{31} \stackrel{5.8}{=}^3 (EDL_3\|EDL_1) \cup EDT_{31}$   
 $\stackrel{EDT_{31} \subseteq EDT_{32}}{\text{und}} \stackrel{EDL_1 \subseteq EDL_2}{\subseteq} (EDL_3\|EDL_2) \cup EDT_{32}$   
 $\stackrel{5.8}{=}^3 EDL_{32}.$

□

**Definition 5.10 ( $\omega$ -Partner).** Ein MEIO  $P_1$  ist ein  $\omega$ -Partner von einem MEIO  $P_2$ , wenn  $I_1 = O_2$  und  $O_1 = I_2 \cup \{\omega\}$  mit  $\omega \notin I_2 \cup O_2$  gilt.

**Lemma 5.11 (Verfeinerung mit Divergenz-Zuständen).** Gegeben sind zwei MEIOs  $P_1$  und  $P_2$  mit der gleichen Signatur. Wenn  $U\|P_1 \sqsubseteq_{Div}^B U\|P_2$  für alle  $\omega$ -Partner  $U$  gilt, dann folgt daraus  $P_1 \sqsubseteq_{Div} P_2$ .

*Beweis.* **TODO zu beweisen**

□

# Literaturverzeichnis

- [BFLV16] Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen, und Walter Vogler, *Non-deterministic Modal Interfaces*, Theor. Comput. Sci. **642** (2016), 24–53.
- [BV15] Ferenc Bujtor und Walter Vogler, *Failure Semantics for Modal Transition Systems*, ACM Trans. Embedded Comput. Syst. **14** (2015), no. 4, 67:1–67:30.
- [Sch16] Ayleen Schinko, *Kommunikationsfehler, Verklemmung und Divergenz bei Interface-Automaten*, Bachelorarbeit, Universität Augsburg, 2016.