# 1 Definitionen

## Stand: 16. August 2017

Das Definitions-Kapitel wurde auf Grundlage von [BV15b] und [Sch16] zusammengestellt und teilweise von [BFLV16] beeinflusst.

**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* $\mathrm{Sig}(P) = (I, O)$*.*
*Falls* $\longrightarrow = \dashrightarrow$ *gilt, wird* $P$ *auch* Implementierung *genannt.*

Implementierungen entsprechen den z.B. in [Sch16] behandelten EIOs.
Must-Transitionen sind Transitionen, die von einer Verfeinerung implementiert werden müssen. Die may-Transitionen sind hingegen die zulässigen Transitionen für eine Verfeinerung. Alle nicht vorhandenen Transitionen dürfen auch in keiner Verfeinerung einer Spezifikation in MEIO-Form auftauchen.
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 $p \overset{\alpha}{\dashrightarrow} p'$ für $(p, \alpha, p') \in \dashrightarrow$ und $p \overset{\alpha}{\dashrightarrow}$ für $\exists p' : (p, \alpha, p') \in \dashrightarrow$ verwendet. Dies kann entsprechend auf Buchstaben-Sequenzen $w \in \Sigma_\tau^*$ erweitert werden:

für $p \overset{w}{\dashrightarrow} p'$ ($p \overset{w}{\dashrightarrow}$) steht für die Existenz eines Laufes $p \overset{\alpha_1}{\dashrightarrow} p_1 \overset{\alpha_2}{\dashrightarrow} \dots p_{n-1} \overset{\alpha_n}{\dashrightarrow} p'$ ($p \overset{\alpha_1}{\dashrightarrow} p_1 \overset{\alpha_2}{\dashrightarrow} \dots p_{n-1} \overset{\alpha_n}{\dashrightarrow}$) mit $w = \alpha_1 \dots \alpha_n$.
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. $\widehat{w}$ steht für $w|_\Sigma$. Es wird $p \overset{w}{\Longrightarrow} p'$ für ein $w \in \Sigma^*$ geschrieben, falls $\exists w' \in \Sigma_\tau^* : \widehat{w'} = w \wedge p \overset{w'}{\dashrightarrow} p'$, und $p \overset{w}{\Longrightarrow}$, falls $p \overset{w}{\Longrightarrow} p'$ für ein beliebiges $p'$ gilt. Falls $p_0 \overset{w}{\Longrightarrow} p$ gilt, dann wird $w$ *Trace* genannt und $p$ ist ein *erreichbarer Zustand*.
Analog zu $\dashrightarrow$ und $\Longrightarrow$ 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 $p \overset{a}{\not\longrightarrow}$ und $p \overset{a}{\not\dashrightarrow}$ für $\neg\exists p' : p \overset{a}{\longrightarrow} p'$ und $\neg\exists p' : p \overset{a}{\dashrightarrow} p'$ stehen soll. $p \overset{a}{\longrightarrow}\overset{\varepsilon}{\Longrightarrow} p'$ wird geschrieben, wenn $p''$ existiert, so dass $p \overset{a}{\longrightarrow} p'' \overset{\varepsilon}{\Longrightarrow} p'$ gilt. Diese Transition wird auch als *schwach-nachlaufende must-Transition* bezeichnet. Entsprechend steht $\overset{a}{\dashrightarrow}\overset{\varepsilon}{\Longrightarrow}$ für die *schwach-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* $P_1 = (P_1, I_1, O_1, \longrightarrow_1, \dashrightarrow_1, p_{01}, E_1)$ *und* $P_2 = (P_2, I_2, O_2, \longrightarrow_2, \dashrightarrow_2, p_{02}, E_2)$ *sind* komponierbar*, falls* $O_1 \cap O_2 = \emptyset$. *Für solche MEIOs ist die* Parallelkomposition $P_{12} := P_1 \| P_2 = ((P_1 \times P_2), I, O, \longrightarrow_{12}, \dashrightarrow_{12}, (p_{01}, p_{02}), E)$ *definiert mit:* TODO: erzwungenen Zeilenumbrüche kontrollieren

- $I = (I_1 \cup I_2) \backslash (O_1 \cup O_2)$,
- $O = (O_1 \cup O_2)$,
- $$\begin{aligned} \longrightarrow_{12} = &\left\{((p_1,p_2),\alpha,(p_1',p_2)) \mid p_1 \overset{\alpha}{\longrightarrow}_1 p_1', \alpha \in \Sigma_\tau \backslash \mathrm{Synch}(P_1,P_2)\right\} \\ &\cup \left\{((p_1,p_2),\alpha,(p_1,p_2')) \mid p_2 \overset{\alpha}{\longrightarrow}_2 p_2', \alpha \in \Sigma_\tau \backslash \mathrm{Synch}(P_1,P_2)\right\} \\ &\cup \left\{((p_1,p_2),\alpha,(p_1',p_2')) \mid p_1 \overset{\alpha}{\longrightarrow}_1 p_1', p_2 \overset{\alpha}{\longrightarrow}_2 p_2', \alpha \in \mathrm{Synch}(P_1,P_2)\right\}, \end{aligned}$$
- $$\begin{aligned} \dashrightarrow_{12} = &\left\{((p_1,p_2),\alpha,(p_1',p_2)) \mid p_1 \overset{\alpha}{\dashrightarrow}_1 p_1', \alpha \in \Sigma_\tau \backslash \mathrm{Synch}(P_1,P_2)\right\} \\ &\cup \left\{((p_1,p_2),\alpha,(p_1,p_2')) \mid p_2 \overset{\alpha}{\dashrightarrow}_2 p_2', \alpha \in \Sigma_\tau \backslash \mathrm{Synch}(P_1,P_2)\right\} \\ &\cup \left\{((p_1,p_2),\alpha,(p_1',p_2')) \mid p_1 \overset{\alpha}{\dashrightarrow}_1 p_1', p_2 \overset{\alpha}{\dashrightarrow}_2 p_2', \alpha \in \mathrm{Synch}(P_1,P_2)\right\}, \end{aligned}$$
- $$\left.\begin{aligned} E = &(P_1 \times E_2) \cup (E_1 \times P_2) \\ &\cup \left\{(p_1,p_2) \mid \exists a \in O_1 \cap I_2 : p_1 \overset{a}{\dashrightarrow}_1 \wedge p_2 \overset{a}{\not\longrightarrow}_2\right\} \\ &\cup \left\{(p_1,p_2) \mid \exists a \in I_1 \cap O_2 : p_1 \overset{a}{\not\longrightarrow}_1 \wedge p_2 \overset{a}{\dashrightarrow}_2\right\} \end{aligned}\right\} \begin{aligned} &\textit{geerbte Fehler} \\ &\textit{neue Kommunikationsfehler.} \end{aligned}$$

*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 Inputs, in dem Fall* $(I_1 \cap I_2)$*, bzw. Outputs, in allen anderen Fällen, der Komposition beibehalten.*
$P_1$ *wird* Partner *von* $P_2$ *genannt, wenn die Parallelkomposition von* $P_1$ *und* $P_2$ *geschlossen ist. Eine Parallelkomposition von zwei MEIOs* $P_1$ *und* $P_2$ *ist* geschlossen*, wenn* $P_1$ *und* $P_2$ *duale Signaturen* $\text{Sig}(P_1) = (I, O)$ *und* $\text{Sig}(P_2) = (O, I)$ *haben.*

Ein neuer Fehler entsteht, wenn eines der MEIOs die Möglichkeit für einen Output hat (may-Transition) und das andere MEIO den passenden Input nicht sicher stellt (keine must-Transition vorhanden). Es muss also in möglichen Implementierungen nicht wirklich zu diesem Fehler kommen, da die Output-Transition nicht zwingendermaßen implementiert werden muss und die may-Input-Transition trotzdem erlaubt sein kann.
Wie bereits unter anderem in [Sch16] gesehen werden konnte, kann es durch die Synchronisation von Inputs zu keinen neuen Fehler kommen, da die Inputs in beiden Transitionssystemen keine lokal kontrollierten Aktionen sind. 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 einem neuen Fehler kommen.

**Definition 1.3 (*alternierende Simulation*).** *Eine* (starke) alternierende Simulation *ist eine Relation* $\mathcal{R} \subseteq P \times Q$ *auf zwei MEIOs* $P$ *und* $Q$*, wenn für alle* $(p, q) \in \mathcal{R}$ *mit* $q \notin E_Q$ *und* $\alpha \in \Sigma_\tau$ *gilt:*

1. $p \notin E_P$,
2. $q \xrightarrow{\alpha}_Q q'$ *impliziert* $p \xrightarrow{\alpha}_P p'$ *für ein* $p'$ *mit* $p' \mathcal{R} q'$,
3. $p \dashrightarrow^{\alpha}_P p'$ *impliziert* $q \dashrightarrow^{\alpha}_Q q'$ *für ein* $q'$ *mit* $p' \mathcal{R} q'$.

*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.* $P \sqsubseteq_{\text{as}} Q$ *steht dabei dafür, dass* $P$ $Q$ (stark) as-verfeinert *oder dass* $P$ *eine* (starke) as-Verfeinerung *von* $Q$ *ist.*
*Für ein MEIO* $Q$ *und eine Implementierung* $P$ *mit* $P \sqsubseteq_{\text{as}} Q$*, ist* $P$ *eine* as-Implementierung *von* $Q$ *und es wird* as-impl$(Q)$ *für die Menge aller as-Implementierungen von* $Q$ *verwendet.*

Da für zwei MEIOs $P$ und $Q$ und alle möglichen Zustands-Tupel $(p, q)$ in einer alternierenden Simulations-Relation $\mathcal{R}$ gelten muss, dass aus $q \notin E_Q$ folgt, dass auch $p$ nicht in $E_P$ enthalten ist, gilt auch die Implikation $p \in E_P \Rightarrow q \in E_Q$.
Für LTS, die nach Definition keine Modalitäten und keine Fehler-Zustände enthalten, entspricht die as-Verfeinerung einer Bisimulation. Dafür müssen die Transitionen eines LTS als must-Transitionen aufgefasst werden. Man kann also auf LTS mit einer as-Verfeinerungs-Relation zwischen zwei Systemen deren Äquivalenz beweisen.

TODO: falls möglich passendes Quellen Beispiel einfügen

Auf den EIO, die z.B. in [Sch16] betrachtet wurden, lässt die as-Verfeinerungs-Relation

zu, dass es in einer Verfeinerung möglicherweise weniger Fehler gibt und zusätzliches Verhalten, dass die Spezifikation nicht hatte kann auftreten. Die EIO entsprechen Implementierungen von MEIOs, es ist also möglich, eine Implementierung mit Fehler durch eine andere as zu verfeinern, die keine Fehler enthält, aber potentiell zusätzliches Verhalten aufweist. Falls eine Implementierung bereits fehler-frei ist, entspricht sie einem LTS und kann somit nur noch duch äquivalente Implementierungen „verfeiner“ werden.

**Definition 1.4 (*schwache alternierende Simulation*).** *Eine* schwache alternierende Simulation *ist eine Relation $\mathcal{R} \subseteq P \times Q$ auf zwei MEIOs $P$ und $Q$, wenn für alle $(p, q) \in \mathcal{R}$ mit $q \notin E_Q$ und $i \in I$ und $\omega \in O \cup \{\tau\}$ gilt:*

1. $p \notin E_P$,
2. $q \xrightarrow{i}_Q q'$ *impliziert* $p \xrightarrow{i}_P \overset{\varepsilon}{\Longrightarrow}_P p'$ *für ein* $p'$ *mit* $p' \mathcal{R} q'$,
3. $q \xrightarrow{\omega}_Q q'$ *impliziert* $p \overset{\hat{\omega}}{\Longrightarrow}_P p'$ *für ein* $p'$ *mit* $p' \mathcal{R} q'$,
4. $p \overset{i}{\dashrightarrow}_P p'$ *impliziert* $q \overset{i}{\dashrightarrow}_Q \overset{\varepsilon}{\Longrightarrow}_Q q'$ *für ein* $q'$ *mit* $p' \mathcal{R} q'$,
5. $p \overset{\omega}{\dashrightarrow}_P p'$ *impliziert* $q \overset{\hat{\omega}}{\Longrightarrow}_Q q'$ *für ein* $q'$ *mit* $p' \mathcal{R} q'$.

*Analog zur starken alternierenden Simulation, wird hier $\sqsubseteq_{\text{w-as}}$ als Relationssymbol verwendet und man kann auch entsprechend* schwache as-Verfeinerungen *betrachten.*
*Ebenso kann $\sqsubseteq_{\text{w-as}}$ für ein MEIO $Q$ und eine Implementierung $P$ definiert werden mit $P \sqsubseteq_{\text{w-as}} Q$, ist $P$ eine* w-as-Implementierung *von $Q$ und es wird* w-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 zwingend 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.
Auch für alle Tupel $(p, q)$ in einer schwach alternierenden Simulations-Relation $\mathcal{R}$ gilt $p \in E_P \Rightarrow q \in E_Q$.

Die Parallelkomposition von Wörtern und Mengen kann z.B. aus [BV15a] 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:* $w_1 \| w_2 := \{w \in (\Sigma_1 \cup \Sigma_2)^* \mid w|_{\Sigma_1} = w_1 \wedge w|_{\Sigma_2} = w_2\}$.
- *Für zwei Mengen von Wörtern bzw. Sprachen $W_1 \subseteq \Sigma_1^*$ und $W_2 \subseteq \Sigma_2^*$ ist deren Parallelkomposition definiert als:* $W_1 \| W_2 := \bigcup \{w_1 \| w_2 \mid w_1 \in W_1 \wedge w_2 \in W_2\}$.

Ebenso können die Definitionen der Funktionen prune und cont zum Abschneiden und Verlängern von Traces z.B. aus [BV15a] 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*).**

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

**Definition 1.7 (*Sprache*).** *Die* Sprache *eines MEIOs* $P$ *ist definiert als:* $L(P) := \left\{w \in \Sigma^* \mid p_0 \overset{w}{\Rightarrow}_P\right\}$.

Somit entspricht die Sprache eines MEIOs, das eine Implementierung ist der Definition aus [Sch16] für EIOs. Jedoch muss die Sprache einer as-Verfeinerung eine MEIOs nicht mehr Teilmenge der Sprache des MEIOs sein, da Definition 1.3 beliebiges Verhalten nach einem Fehler-Zustand in dem zu verfeinernden MEIO zulässt. Falls jedoch der MEIO bereits fehler-frei ist, ist seine Sprache die Vereinigung der Sprachen all seine möglichen as-Implementierungen.
Von der Sprache einer as-Verfeinerung eines MEIOs kann man keine Rückschlüsse mehr auf die ursprüngliche Sprache ziehen, da man nicht weiß, welche Fehler-Zustände in der Verfeinerung übernommen wurden und welche als normale Zustände mit beliebigen Verhalten umgesetzt wurden.
Man hätte alternativ die Sprache eines MEIOs anders definieren können um weiterhin einen eindeutigen Zusammenhang zwischen dieser und den Sprachen der as-Implementierungen zu haben, jedoch wäre dann die Äquivalenz zur EIO Sprach-Definition in [Sch16] verloren gegangen. Eine Implementierung mit Fehler-Zuständen hätte dann eine Sprache mit Wörtern, die sie nicht mal ausführen können muss.

# 2 allgemeine Folgerungen

**Proposition 2.1 (*Sprache und Implementierungen*).** *Für die Sprache eines MEIOs $P$ gilt* $L(P) \subseteq \left\{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \stackrel{w}{\Longrightarrow}_{P'}\right\} = \bigcup_{P' \in \text{as-impl}(P)} L(P')$.

*Beweis.*
Sei $P'$ die as-Implementierung von $P$, die alle may- und must-Transitionen von $P$ implementiert. Die Definition von $P'$ lautet dann:

- $P' = P$,
- $p'_0 = p_0$,
- $I_{P'} = I_P$ und $O_{P'} = O_P$,
- $\longrightarrow_{P'} = \dashrightarrow_{P'} = \dashrightarrow_P$,
- $E_{P'} = \emptyset$.

Die entsprechende starke as-Verfeinerungs-Relation $\mathcal{R}$, die zwischen $P'$ und $P$ gilt, ist die Identitäts-Relation zwischen den Zuständen der Transitionssysteme. Für 1.3 3. betrachten wir $(p, p) \in \mathcal{R}$ und $p \stackrel{\alpha}{\dashrightarrow}_{P'} p'$ in $P'$. Diese Transition muss auch in $P$ existieren und da $\mathcal{R}$ die Identitäts-Relation ist auch $(p', p') \in \mathcal{R}$ gelten. Alle must-Transitionen in $P$ haben zugrunde liegende may-Transitionen. Es gibt also zu jeder must-Transition $p \stackrel{\alpha}{\longrightarrow}_P p'$ in $P$ auch die Transition $p \stackrel{\alpha}{\longrightarrow}_{P'} p'$, wenn $(p, p)$ in $\mathcal{R}$ enthalten ist. Es folgt dann auch $(p', p') \in \mathcal{R}$ und dadurch ist auch 1.3 2. erfüllt. Der erste Punkt von 1.3 gilt, da $E_{P'}$ leer ist.
Für alle $w \in L(P) = \left\{w \in \Sigma^* \mid p_0 \stackrel{w}{\Rightarrow}_P\right\}$ gilt $w \in L(P') = \left\{w \in \Sigma^* \mid p'_0 \stackrel{w}{\Longrightarrow}_{P'}\right\}$, da alle Transitionen von $P$ in $P'$ implementiert werden. □

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

*Beweis.* Jedes Wort, dass in $L_{12}$ enthalten ist, hat einen einen entsprechenden Ablauf, der in $P_{12}$ ausführbar ist. Dieser Ablauf kann auf Abläufe von $P_1$ 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 $L_1 \| L_2$ auch Wörter der Parallelkomposition $L(P_{12})$. □

**Lemma 2.3 (*w-as-Verfeinerung und Parallelkomposition*).** *Für zwei komponierbar MEIOs $P_1$ und $P_2$ gilt, falls $P_1'$ und $P_2'$ schwache as-Verfeinerungen von $P_1$ bzw. $P_2$ sind, dann muss $P_1'\|P_2'$ eine keine schwache as-Verfeinerung von $P_1\|P_2$ sein. Jedoch für eine Relation $\mathcal{R}_{12}$ für die $((p_1', p_2'), (p_1, p_2)) \in \mathcal{R}_{12}$ genau dann gilt, wenn $(p_1', p_1) \in \mathcal{R}_1$ und $(p_2', p_2) \in \mathcal{R}_2$ für die schwachen as-Verfeinerungs-Relationen von $P_j'$ auf $P_j$ für $j \in \{1, 2\}$, dann sind für $\mathcal{R}_{12}$ die Punkte 2. bis 5. aus der Definition 1.4 erfüllt.*

*TODO: falls Zeit: in Lemma mit nur einer Verfeinerung umformulieren (andere folgt wegen Kommutativität)*

*Beweis.* Für alle folgenden Fälle wird $((p_1', p_2'), (p_1, p_2)) \in \mathcal{R}_{12}$ mit $(p_1, p_2) \notin E_{12}$ vorausgesetzt.

1. Es gilt für diesen Punkt nicht unbedingt, dass $(p_1', p_2')$ kein Element von $E_{P_1'\|P_2'}$ ist. Deshalb ist dieser Punkt der Definition nicht erfüllt und $P_1'\|P_2'$ muss keine schwache as-Verfeinerung von $P_1\|P_2$ sein.
   Dies folgt direkt aus der Voraussetzung, dass $(p_1, p_2) \notin E_{P_1\|P_2}$ für das Tupel $((p_1', p_2'), (p_1, p_2))$ aus $\mathcal{R}_{12}$ gilt. Indem man auf $\mathcal{R}_{12}$ die Definition von oben anwendet, erhält man $(p_j', p_j) \in \mathcal{R}_j$ für beide $j$ Werte. Die $p_j$ dürfen beide keine Fehler-Zustände sein, da sonst auch $(p_1, p_2)$ ein solcher wäre. Somit folgt mit Definition 1.4 1. $p_j' \notin E_j$ für beide $j$ Werte. Die beiden gestrichenen Zustände in Parallelkomposition können also keinen geerbten Fehler produzieren. Jedoch könnte $(p_1', p_2')$ aufgrund eines nicht erzwungenen Inputs ein neuer Fehler-Zustand sein. Dafür müsste oBdA $p_1' \overset{a}{\not\rightarrow}_{P_1'}$ und $p_2' \overset{a}{\dashrightarrow}_{P_2'}$ für ein $a$ aus $I_1 \cap O_2$ gelten. $\mathcal{R}_2$ erzwingt mit 1.4 5. die schwache Ausführbarkeit des Outputs $a$ in $P_2$, d.h. $p_2 \overset{a}{\Rightarrow}_2$. Da dieser Output nur schwach ausführbar sein muss, kann es in der Parallelkomposition von $P_1'\|P_2'$ zu einem neuen Kommunikationsfehler kommen, der in $P_1\|P_2$ keiner ist.
   Ein Gegenbeispiel dafür ist in Abbildung 2.1 dargestellt. Hierfür soll $a$ in Schnitt der Inputs $I_1$ von $P_1$ bzw. $P_1'$ und der Outputs $O_2$ von $P_2$ bzw. $P_2'$ enthalten sein. Die Relation $\mathcal{R}_1$ enthält das Zustands-Tupel $(p_{01}', p_{01})$ und die $\mathcal{R}_2$ die Tupel $(p_{02}', p_{02})$ und $(p_2', p_2)$. Somit ist $((p_{01}', p_{02}'), (p_{01}, p_{02}))$ in $\mathcal{R}_{12}$ enthalten und es gilt $(p_{01}, p_{02}) \notin E_{12}$. Jedoch ist $(p_{01}', p_{02}')$ trotzdem ein Fehler-Zustand in der Parallelkomposition von $P_1'$ und $P_2'$.
   Es wäre auch möglich, dass $P_2$ ebenfalls eine Implementierung ist. Die must-$\tau$-Transition würde dann eine schwache Implementierung in $P_2'$ fordern, jedoch muss es dafür keine echte Transition in $P_2'$ geben, da $\tau$ die interne Aktion ist. Die Implementierung von $a$ würde ebenfalls schwach gefordert werden, ist jedoch bereits stark vorhanden. $P_1\|P_2$ würde mit der must-$\tau$-Transition dann ebenfalls zu einer Implementierung werden. Die must-$a$-Transition in $P_2'$ könnte auch eine may-Transition sein, solange die $a$-Transition in $P_2$ keine must-Transition ist.
   Um dieses Problem zu lösen könnte man auch die Definition der Parallelkomposition verändern. Es wäre denkbar, dass alle Zustände, die lokal Fehler-Zustände erreichen können bereits als solche angesehen werden. Jedoch stellt dies eine stärkere Forderung. Auf Implementierungen ist die Forderung stärkere, wie die auf

EIOs in z.B. [Sch16]. Dieser Ansatz käme jedoch dem Vorgehen des Abschneidens der Fehler-Zustände mit ihren lokalen Vorgängern in [BFLV16] näher.

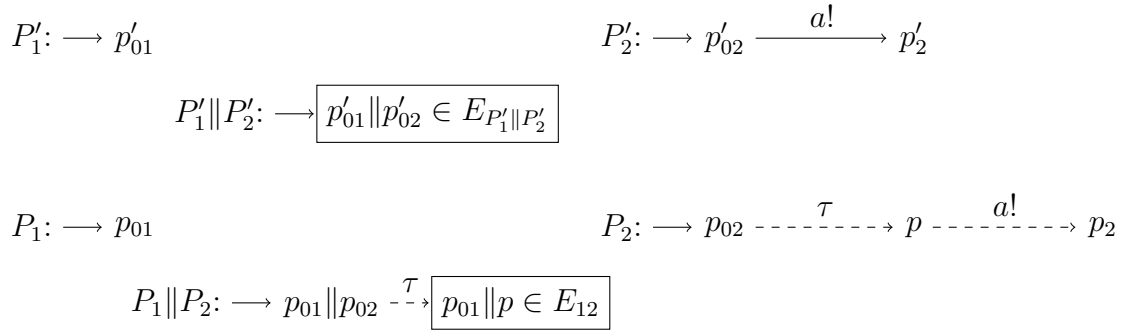


Abbildung 2.1: Gegenbeispiel für 1. von $\mathcal{R}_{12}$ bzgl. Definition 1.3

2. Aus der Definition der schwachen alternierenden Simulation in 1.4 folgt, dass für diesen Punkt 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 $((q_1', q_2'), (q_1, q_2)) \in \mathcal{R}_{12}$.
Die $i$-must-Transition in $P_1\|P_2$ kann entweder aus der Synchronisation von zwei must-Inputs entstanden sein oder als unsynchronisierte Aktion aus einem $P_1$ übernommen worden sein.

   - Fall 1 ($i \notin \text{Synch}(P_1\|P_2)$): OBdA ist $i$ in $I_1$ enthalten. Es muss also in $P_1$ die $i$-Transition als must-Transition von $p_1$ ausgehen, es gilt $p_1 \xrightarrow{i}_1 q_1$. Mit der Relation $\mathcal{R}_1$ und 1.4 2. folgt, dass in $P_1'$ $i$ als schwache Transition in der Form $p_1' \xrightarrow{i}_{P_1'} \xRightarrow{\varepsilon}_{P_1'} q_1'$ ausführbar sein musst und $q_1' \mathcal{R}_1 q_1$ gelten muss. $p_2 = q_2$ muss erfüllt sein, da $i$ nicht in $\Sigma_2$ enthalten ist. Da $i$ wenn es kein Element der Aktionen von $P_2$ ist auch keine Aktion der schwachen as-Verfeinerung $P_2'$ sein kann entsteht mit $q_2' = p_2'$ die Transitionsfolge $(p_1', p_2') \xrightarrow{i}_{P_1'\|P_2'} \xRightarrow{\varepsilon}_{P_1'\|P_2'} (q_1', q_2')$. Aus der Voraussetzung folgt $(p_2', p_2) = (q_2', q_2) \in \mathcal{R}_2$. Mit der Definition von $\mathcal{R}_{12}$ kann dann daraus $((q_1', q_2'), (q_1, q_2)) \in \mathcal{R}_{12}$ gefolgert werden.
   - Fall 2 ($i \in \text{Synch}(P_1\|P_2)$): Damit $i$ auch in $P_1\|P_2$ ein Input ist, muss $i \in I_1 \cap I_2$ gelten. Um die Transition $(p_1, p_2) \xrightarrow{i}_{12} (q_1, q_2)$ in der Komposition möglich zu machen, muss in beiden $P_j$'s $p_j \xrightarrow{i}_j q_j$ gelten. Durch $\mathcal{R}_j$ und die Definition 1.4 2. folgt für beide $j$-Werte $p_j' \xrightarrow{i}_{P_j'} \xRightarrow{\varepsilon}_{P_j'} q_j'$ mit $(q_j', q_j) \in \mathcal{R}_j$. Daraus ergibt sich $((q_1', q_2'), (q_1, q_2)) \in \mathcal{R}_{12}$ mit der Definition von $\mathcal{R}_{12}$. Durch die Synchronisation der $i$-Inputs und Verschränkung der $\tau$-Transitionen in der Komposition von $P_1'$ und $P_2'$ gilt $(p_1', p_2') \xrightarrow{i}_{P_1'\|P_2'} \xRightarrow{\varepsilon}_{P_1'\|P_2'} (q_1', q_2')$.

3. Analog zu 2. 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.

- Fall 1 ($\omega \notin \text{Synch}(P_1 \| P_2)$): OBdA sei $\omega$ eine Output- oder interen Aktion von $P_1$. Um in der Komposition $P_1 \| P_2$ die must-Transition zu erhalten muss bereits für die Transition in $P_1$ $p_1 \xrightarrow{\omega}_1 q_1$ gelten sowie $p_2 = q_2$ in $P_2$. Mit 1.4 3. kann für $\mathcal{R}_1$ gefolgert werden, dass $p_1' \overset{\hat{\omega}}{\Longrightarrow}_{P_1'} q_1'$ mit $(q_1', q_1) \in \mathcal{R}_1$ gilt. In der Komposition folgt dann mit $p_2' = q_2'$ $(p_1', p_2') \overset{\hat{\omega}}{\Longrightarrow}_{P_1' \| P_2'} (q_1', q_2')$, da $\omega \notin \Sigma_2$ ist und somit $(p_2', p_2) = (q_2', q_2) \in \mathcal{R}_2$ gilt. Es folgt insgesamt auch noch die Zugehörigkeit des Zustands-Tupels $((q_1', q_2'), (q_1, q_2))$ zur Relation $\mathcal{R}_{12}$.
- Fall 2 ($\omega \in \text{Synch}(P_1 \| P_2)$): Da in der Menge $\text{Synch}(P_1 \| P_2)$ nur Inputs und Outputs enthalten sein können, muss in diesem Fall $\omega \neq \tau$ gelten. Um einen Output $\omega$ in der Parallelkomposition von $P_1$ und $P_2$ zu erhalten, muss oBdA $\omega \in I_1 \cap O_2$ gelten. Es folgt also $p_1 \xrightarrow{\omega}_1 q_1$ mit $\omega \in I_1$ und $p_2 \xrightarrow{\omega}_2 q_2$ mit $\omega \in O_2$ für die einzeln Transitionssysteme. Mit $\mathcal{R}_1$ und 1.4 2. folgt $p_1' \xrightarrow{\omega}_{P_1'} \overset{\varepsilon}{\Longrightarrow}_{P_1'} q_1'$ und $q_1' \mathcal{R}_1 q_1$. Für $\mathcal{R}_2$ gilt 1.4 3. man erhält dadurch $p_2' \overset{\omega}{\Longrightarrow}_{P_2'} q_2'$ mit $q_2' \mathcal{R}_2 q_2$. Da $\omega$ in $P_2'$ ein Output ist, gilt $\omega = \hat{\omega}$. In der Parallelkomposition von $P_1'$ und $P_2'$ werden zuerst die internen Aktionen von $P_2'$ ausgeführt, bis dort der Output erreicht ist, dann wird $\omega$ synchronisiert und danach werden die internen Aktionen beider Komponenten verschränkt ausgeführt, bis man bei den Zuständen $q_1'$ und $q_2'$ angekommen ist. Es folgt also die Transitionsfolge $(p_1', p_2') \overset{\hat{\omega}}{\Longrightarrow}_{P_1' \| P_2'} (q_1', q_2')$ und das Tupel $((q_1', q_2'), (q_1, q_2))$ in der Relation $\mathcal{R}_{12}$.

4. $(p_1', p_2') \overset{i}{\dashrightarrow}_{P_1' \| P_2'} (q_1', q_2')$ impliziert $(p_1, p_2) \overset{i}{\dashrightarrow}_{12} \overset{\varepsilon}{\Longrightarrow}_{12} (q_1, q_2)$ für ein $(q_1, q_2)$ mit $((q_1', q_2'), (q_1, q_2)) \in \mathcal{R}_{12}$ ist die Voraussetzung des 4. Punktes, um zu beweisen, dass $\mathcal{R}_{12}$ eine schwache as-Verfeinerungs-Relation, bis auf die Erfüllung von 1. aus der Definition 1.4, ist.
   Die Transition $i$ kann wiederum durch Synchronisation von zwei Transitionen entstanden sein oder durch eine Transition aus einer der beiden Komponenten mit der Voraussetzung $i \notin \text{Synch}(P_1' \| P_2')$.
   - Fall 1 ($i \notin \text{Synch}(P_1' \| P_2')$): OBdA ist $i$ in $I_1$ enthalten. Es muss also in $P_1'$ eine ausgehende $i$-Transition von Zustand $p_1'$ geben, so dass $p_1' \overset{i}{\dashrightarrow}_1 q_1'$ gilt. Mit der Relation $\mathcal{R}_1$ und 1.4 4. folgt, dass in $P_1$ $i$ als schwache Transition in der Form $p_1 \overset{i}{\dashrightarrow}_1 \overset{\varepsilon}{\Longrightarrow}_1 q_1$ ausführbar sein musst und $q_1' \mathcal{R}_1 q_1$ gelten muss. $p_2' = q_2'$ muss gelten, da $i$ nicht in $\Sigma_2$ enthalten ist. Aus der Voraussetzung folgt $(p_2', p_2) = (q_2', q_2) \in \mathcal{R}_2$, da $i$ wenn es kein Element der Aktionen von $P_2'$ ist auch keine Aktion der Spezifikation $P_2$ sein kann. Mit der Definition von $\mathcal{R}_{12}$ kann dann daraus $((q_1', q_2'), (q_1, q_2)) \in \mathcal{R}_{12}$ gefolgert werden. In der Parallelkomposition von $P_1$ und $P_2$ entsteht die Transitionsfolge $(p_1, p_2) \overset{i}{\dashrightarrow}_{12} \overset{\varepsilon}{\Longrightarrow}_{12} (q_1, q_2)$.
   - Fall 2 ($i \in \text{Synch}(P_1' \| P_2')$): Damit $i$ auch in $P_1' \| P_2'$ ein Input ist, muss $i \in I_1 \cap I_2$ gelten. Um die Transition $(p_1', p_2') \overset{i}{\dashrightarrow}_{P_1' \| P_2'} (q_1', q_2')$ in der Komposition möglich zu machen, muss in beiden Transitionssystemen $P_j'$ $p_j \overset{i}{\dashrightarrow}_{P_j'} q_j'$ gelten. Durch $\mathcal{R}_j$ und die Definition 1.4 4., die für diese Relationen gilt, folgt für beide $j$

Werte $p_j \overset{i}{\dashrightarrow}_j \overset{\varepsilon}{\Rightarrow}_j q_j$ mit $(q'_j, q_j) \in \mathcal{R}_j$. Es folgt $((q'_1, q'_2), (q_1, q_2)) \in \mathcal{R}_{12}$ mit der Definition von $\mathcal{R}_{12}$. Durch die Synchronisation des $i$'s in der Komposition von $P_1$ und $P_2$ gilt $(p_1, p_2) \overset{i}{\dashrightarrow}_{12} \overset{\varepsilon}{\Rightarrow}_{12} (q_1, q_2)$.

5. Analog zu 3. und 4. kann für diesen Punkt $(p'_1, p'_2) \overset{\omega}{\dashrightarrow}_{P'_1 \| P'_2} (q'_1, q'_2)$ impliziert $(p_1, p_2) \overset{\hat{\omega}}{\Rightarrow}_{12} (q_1, q_2)$ 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 $P'_1 \| P'_2$ ist entweder aus einem synchronisierten oder aus einem unsynchronisierten $\omega$ entstanden.

   - Fall 1 ($\omega \notin \text{Synch}(P'_1 \| P'_2)$): OBdA ist $\omega$ in $O_1$ enthalten oder eine interne Aktion, die von $P'_1$ geerbt wurde. Um in $P'_1 \| P'_2$ die may-Transition zu erhalten muss bereits in $P'_1$ die Transition $p'_1 \overset{\omega}{\dashrightarrow}_{P'_1} q'_1$ möglich gewesen sein. Mit 1.4 5. kann für $\mathcal{R}_1$ gefolgert werden, dass $p_1 \overset{\hat{\omega}}{\Rightarrow}_1 q_1$ mit $(q'_1, q_1) \in \mathcal{R}_1$ gilt. In der Komposition folgt dann $(p_1, p_2) \overset{\hat{\omega}}{\Rightarrow}_{12} (q_1, q_2)$, da $\omega \notin \Sigma_2$ ist und somit $(p'_2, p_2) = (q'_2, q_2) \in \mathcal{R}_2$ gilt. Es folgt insgesamt auch noch die Zugehörigkeit des Zustands-Tupels $((q'_1, q'_2), (q_1, q_2))$ zur Relation $\mathcal{R}_{12}$.
   - Fall 2 ($\omega \in \text{Synch}(P'_1 \| P'_2)$): Es muss $\omega \neq \tau$ gelten und somit muss oBdA $\omega \in I_1 \cap O_2$. Es folgt also $p'_1 \overset{\omega}{\dashrightarrow}_{P'_1} q'_1$ mit $\omega \in I_1$ und $p'_2 \overset{\omega}{\dashrightarrow}_{P'_2} q'_2$ mit $\omega \in O_2$ für die einzeln Transitionssysteme. Mit $\mathcal{R}_1$ und 1.4 4. folgt $p_1 \overset{\omega}{\dashrightarrow}_1 \overset{\varepsilon}{\Rightarrow}_1 q_1$ und $q'_1 \mathcal{R}_1 q_1$. Wenn man $\mathcal{R}_2$ mit 1.4 5. angewendet erhält man $p_2 \overset{\omega}{\Rightarrow}_2 q_2$ mit $q'_2 \mathcal{R}_2 q_2$. Da $\omega$ in $P_2$ ein Output ist, gilt $\omega = \hat{\omega}$. In der Parallelkomposition von $P_1$ und $P_2$ werden zuerst die internen Aktionen von $P_2$ ausgeführt, bis dort der Output erreicht ist, dann wird $\omega$ synchronisiert und danach werden die internen Aktionen beider Komponenten verschränkt ausgeführt, bis man bei den Zuständen $q_1$ und $q_2$ angekommen ist. Es folgt also die Transitionsfolge $(p_1, p_2) \overset{\hat{\omega}}{\Rightarrow}_{12} (q_1, q_2)$ und das Tupel $((q'_1, q'_2), (q_1, q_2))$ in der Relation $\mathcal{R}_{12}$.

□

**Korollar 2.4 (*as-Verfeinerungen und Parallelkomposition*).** *Für zwei komponierbar MEIOs $P_1$ und $P_2$ gilt, falls $P'_1$ und $P'_2$ as-Verfeinerungen von $P_1$ bzw. $P_2$ sind, dann ist auch $P'_1 \| P'_2$ eine as-Verfeinerung von $P_1 \| P_2$.*

*Beweis.* Falls die Relationen $\mathcal{R}_1$ und $\mathcal{R}_2$ aus dem Lemma 2.3 keine schwachen as-Verfeinerungs-Relationen sondern starke as-Verfeinerungs-Relation sind, ist $\mathcal{R}_{12}$ eine as-Verfeinerungs-Relation zwischen $P'_1 \| P'_2$ und $P_1 \| P_2$. Es ist also nur zu zeigen, wie aus den einzelnen Beweispunkten des Beweises von 2.3 folgt, dass $\mathcal{R}_{12}$ eine starke as-Verfeinerungs-Relation ist und zusätzlich, dass hier der erste Punkt erfüllt ist. Es wird hier ebenso für alle Punkte vorausgesetzt, dass $((p'_1, p'_2), (p_1, p_2)) \in \mathcal{R}_{12}$ mit $(p_1, p_2) \notin E_{12}$ gilt.

1. Dieser Punkt kann kann im Gegensatz zu 1. aus dem Beweis von Lemma 2.3 bewiesen werden. Dies ist möglich, da für $p_2$ der Output $a$ nicht nur schwach sondern direkt ausführbar ist. Es ist also zu zeigen, dass $(p'_1, p'_2)$ kein Element von $E_{P'_1 \| P'_2}$

ist.
Dies folgt direkt aus der Voraussetzung, dass $(p_1, p_2) \notin E_{P_1\|P_2}$ für das Tupel $((p'_1, p'_2), (p_1, p_2))$ aus $\mathcal{R}_{12}$ gilt. In dem man auf das $\mathcal{R}_{12}$ die Definition anwendet, erhält man $(p'_j, p_j) \in \mathcal{R}_j$ für beide $j$ Werte. Die $p_j$ dürfen beide keine Fehler-Zustände sein, da sonst auch $(p_1, p_2)$ ein solcher wäre. Somit folgt mit Definition 1.3 1. $p'_j \notin E_j$ für beide $j$ Werte. Die beiden gestrichenen Zustände in Parallelkomposition können also keinen geerbten Fehler produzieren. Jedoch könnte $(p'_1, p'_2)$ aufgrund eines nicht sichergestellten Inputs ein neuer Fehler-Zustand sein. Dafür müsste oBdA $p'_1 \overset{a}{\not\longrightarrow}_{P'_1}$ und $p'_2 \overset{a}{\dashrightarrow}_{P'_2}$ für ein $a$ aus $I_1 \cap O_2$ gelten. $\mathcal{R}_2$ erzwingt mit 1.3 3. die Ausführbarkeit des Outputs $a$ in $P_2$, d.h. $p_2 \overset{a}{\dashrightarrow}_2$. Mit 1.3 2. von $\mathcal{R}_1$ folgt $p_1 \overset{a}{\not\longrightarrow}_1$. Somit müsste auch $(p_1, p_2) \in E_{12}$ gelten, was ein Widerspruch zur Voraussetzung wäre. $(p'_1, p'_2)$ kann also weder ein geerbter noch ein neuer Fehler in $P'_1\|P'_2$ sein und deshalb gilt $(p'_1, p'_2) \notin E_{P'_1\|P'_2}$.

2. $\alpha$ kann sowohl Input, Output wie auch interen Aktion sein. Um diesen Punkt zu beweisen muss man 2. und 3. aus dem Beweis von Lemma 2.3 kombinieren. Da die $\mathcal{R}_j$ für $j \in \{1, 2\}$ jedoch die Transition in den $P'_j$ ohne zusätzliche $\tau$-Transitionen fordern, entstehen in den einzelnen Komponenten keine schwachen Transitionen für die $\alpha$s und somit ist $\alpha$ auch in der Parallelkomposition $P'_1\|P'_2$ eine direkte Transition ohne zusätzliche $\tau$s. Es folgt das $\mathcal{R}_{12}$ die Forderungen für die strake as-Verfeinerungs-Relation dieses Punktes erfüllt.

3. Hierfür werden die Punkte 3. und 4. aus dem Beweis des Lemmas 2.3 kombiniert. Analog wie bei 2. diese Beweises fallen die zusätzlichen $\tau$-Transitionen durch die stärkere Forderung an $\mathcal{R}_1$ und $\mathcal{R}_2$ weg. Dieser Punkt gilt also ebenfalls.

□

**Korollar 2.5 (*as-Implementierungen und Parallelkomposition*).** *Für zwei komponierbare MEIOs $P_1$ und $P_2$ gilt:* $P'_1 \in \text{as-impl}(P_1) \wedge P'_2 \in \text{as-impl}(P_2) \Rightarrow (P'_1\|P'_2) \in \text{as-impl}(P_1\|P_2)$.

*Beweis.* $P'_1$ und $P'_2$ sind Aufgrund der Definition 1.3 auch starke as-Verfeinerungen von $P_1$ bzw. $P_2$. Somit ist die Parallelkomposition $P'_1\|P'_2$ auch eine starke as-Verfeinerung von $P_1\|P_2$, wegen Korollar 2.4. Für Implementierungen gilt $\longrightarrow = \dashrightarrow$. Durch die Definition der Parallelkomposition in 1.2 können aus zwei komponierbaren Implementierungen in der Komposition keine may-Transitionen ohne zugehörige must-Transitionen entstehen. Es gilt also auch $\longrightarrow_{P'_1\|P'_2} = \dashrightarrow_{P'_1\|P'_2}$ und somit ist $P'_1\|P'_2$ eine Implementierung und eine as-Verfeinerung von $P_1\|P_2$. Dies entspricht der Definition der starken as-Implementierung, sodass $(P'_1\|P'_2) \in \text{as-impl}(P_1\|P_2)$ gilt. □

Für schwache as-Implementierungen kann kein analoges Korollar zur 2.5 gegen, da die Verfeinerung bereits scheitert (Lemma 2.3).
Die entgegengesetzte Richtung von Korollar 2.4 gilt im allgemeinen nicht, d.h. es muss zu einer as-Verfeinerung $P'$ einer Parallelkomposition $P_1\|P_2$ keine as-Verfeinerungen $P'_1$

bzw. $P_2'$ der einzelnen Komponenten $P_1$ bzw. $P_2$ geben, deren Parallelkomposition $P_1' \| P_2'$ der as-Verfeinerung der Parallelkomposition $P'$ entsprechen. Die Problematik wird in Abbildung 2.2 an einem Beispiel dargestellt. In der Parallelkomposition wird die may-Transition von $P_2$ zu zwei may-Transitionen, für die in einer as-Verfeinerung unabhängig entschieden werden kann, ob sie übernommen, implementiert oder weggelassen werden. Somit kommt es in $P'$ zu dem Problem, dass keine as-Verfeinerung von $P_2$ (entweder keine Transition oder die $o'$ Transition wird als may- oder must-Transition ausgeführt) in Parallelkomposition mit der Implementierung $P_1$ $P'$ ergeben würde.
Auch im Spezialfall von as-Implementierungen kann das Gegenbeispiel angewendet werden, da $P'$ auch eine Implementierung von $P_1 \| P_2$ ist und es auch keine passende as-Implementierung von $P_2$ geben kann, wenn es schon keine passende Verfeinerung gibt.

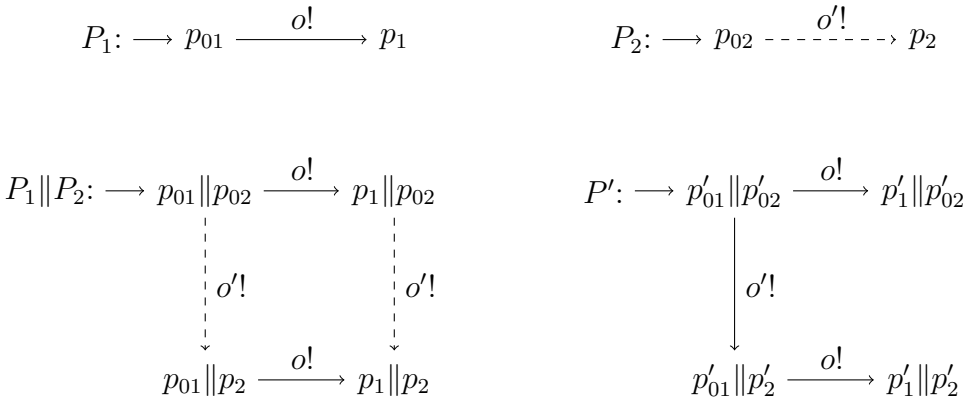


Abbildung 2.2: Gegenbeispiel für Umkehrung von Lemma 2.4

Ein neuer Fehler in einer Parallelkomposition zweier MEIOs muss in einer Implementierung (as oder w-as) dieser Parallelkomposition nicht auftauchen, auch nicht in der Parallelkomposition von Implementierungen der einzelnen Komponenten. Dies liegt daran, dass für den Input nur vorausgesetzt 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 Fehler kommt, dann gibt es auch immer mindestens eine mögliche Implementierung, die diesen Fehler enthält und es gibt auch immer mindestens ein Implementierungs-Paar der Komponenten, in deren Parallelkomposition sich dieser Fehler ebenfalls zeigt.

# 3 Verfeinerungen für Kommunikationsfehler-Freiheit

## 3.1 gröbster Präkongruenz-Ansatz

Dieses Kapitel versucht die Präkongruenz für Error bei EIOs aus z.B. [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 $p_0 \stackrel{w}{\Rightarrow}_P p \in E$.*
*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 $P_1 \sqsubseteq_E^{\mathrm{B}} P_2$ geschrieben, wenn nur dann ein Fehler-Zustand in einer as-Implementierung von $P_1$ lokal erreichbar ist, wenn es auch eine as-Implementierung von $P_2$ gibt, in der ein Fehler-Zustand ebenfalls lokal erreichbar ist. Die Basisrelation stellt eine* Verfeinerungs-Relation *bezüglich* Kommunikationsfehler-Freiheit *dar.*
*$\sqsubseteq_E^{\mathrm{C}}$ bezeichnet die* vollständig abstrakte Präkongruenz *von $\sqsubseteq_E^{\mathrm{B}}$ bezüglich $\cdot\|\cdot$, d.h. die gröbste Präkongruenz bezüglich $\cdot\|\cdot$, die in $\sqsubseteq_E^{\mathrm{B}}$ enthalten ist.*

Für as-Implementierungen $P_1$ und $P_2$ entspricht $\sqsubseteq_E^{\mathrm{B}}$ der Basisrelation $\sqsubseteq_E^{\mathrm{B}}$ aus [Sch16].

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

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

- strikte Fehler-Traces*:* $StET(P) := \left\{ w \in \Sigma^* \mid p_0 \stackrel{w}{\Rightarrow}_P p \in E \right\}$,
- gekürzte Fehler-Traces*:* $PrET(P) := \{\mathrm{prune}(w) \mid w \in StET(P)\}$,
- Input-kritische-Traces*:* $MIT(P) := \left\{ wa \in \Sigma^* \mid p_0 \stackrel{w}{\Rightarrow}_P p \wedge a \in I \wedge p \stackrel{a}{\not\rightarrow}_P \right\}$.

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. Die nächste Proposition stellt eine Teilmengen-Beziehung zwischen den Traces eines MEIOs und den Traces seiner as-Implementierungen dar.

**Proposition 3.4 (*Kommunikationsfehler-Traces und Implementierungen*).** *Sei $P$ ein MEIO.*

1. *Für die strikten Fehler-Traces von $P$ gilt:* $StET(P) \subseteq \left\{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \overset{w}{\Longrightarrow}_{P'} p' \in E\right\} = \bigcup_{P' \in \text{as-impl}(P)} StET(P')$. TODO: erzwungen Zeilenumbruch kontrollieren

2. *Für die gekürzten Fehler-Traces von $P$ gilt:* $PrET(P) \subseteq \{\text{prune}(w) \mid \exists P' \in \text{as-impl}(P) : w \in StET(P')\} = \bigcup_{P' \in \text{as-impl}(P)} PrET(P')$, TODO: erzwungen Zeilenumbruch kontrollieren

3. *Für Input-kritischen-Traces von $P$ gilt:* $MIT(P) \subseteq \left\{wa \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \overset{w}{\Longrightarrow}_{P'} p' \wedge a \in I \wedge p' \overset{a}{\not\longrightarrow}_{P'}\right\} = \bigcup_{P' \in \text{as-impl}(P)} MIT(P')$. TODO: erzwungen Zeilenumbruch kontrollieren

*Beweis.*

1. Um die Inklusion zu zeigen wird eine Implementierung $P'$ angegeben, die die strikten Fehler-Traces von $P$ implementiert und zusätzlich auch noch eine passende as-Verfeinerungs-Relation $\mathcal{R}$ zwischen den beiden Transitionssystemen. $P'$ implementiert wie im Beweis zu Proposition 2.1 alle Transitionen von $P$. Das $P'$ wird hier jedoch im Gegensatz zu Beweis von 2.1 auch noch alle Fehler-Zustände aus $P$ implementieren. Die entsprechende as-Verfeinerungs-Relation $\mathcal{R}$ ist hier ebenfalls wieder die Identitäts-Relation zwischen den Zuständen der Transitionssysteme. Die Definition von $P'$ lautet:

   - $P' = P$,
   - $p'_0 = p_0$,
   - $I_{P'} = I_P$ und $O_{P'} = O_P$,
   - $\longrightarrow_{P'} = \dashrightarrow_{P'} = \dashrightarrow_P$,
   - $E_{P'} = E_P$.

   Die Tupel, die von 1.3 2. und 3. als Elemente der as-Verfeinerungs-Relation $\mathcal{R}$ gefordert werden, sind bereits durch die Identitäts-Relation garantiert, wie im Beweis von 2.1. Für 1.3 1. muss für jedes in der as-Verfeinerungs-Relation $\mathcal{R}$ enthaltene Zustands-Paar gelten, wenn der Zustand aus $P$ kein Fehler-Zustand ist, dann ist auch der Zustand aus $P'$ keiner. Dies folgt aus der Gleichheit der Mengen $E_P$ und $E_{P'}$. Jeder Trace aus $StET(P)$ ist via may-Transitionen in $P$ ausführbar und führt dort zu einem Fehler-Zustand. Der analoge Trace ist auch in $P'$ möglich, da alle may-Transitionen aus $P$ in $P'$ als must-Transitionen implementiert wurden. Der dabei erreichte Zustand steht mit dem Fehler-Zustand in $P$ in der Identitäts-Relation $\mathcal{R}$, die Zustände entsprechen sich also. Es gilt mit $E_{P'} = E_P$, dass auch der in $P'$ erreichte Zustand ein Fehler-Zustand ist. Für die as-Implementierung $P'$ von $P$ und der Identitäts-Relation $\mathcal{R}$ als starke as-Verfeinerungs-Relation zwischen den Transitionssystemen gilt also $StET(P) = StET(P')$.

2. Da der erste Punkt dieser Proposition bereits beweisen wurde, gilt bereits, dass alle strikten Fehler-Traces von $P$ in der Vereinigung aller strikten Fehler-Traces der as-Implementierungen von $P$ enthalten sind. Wenn auf alle Wörter in beiden Mengen die prune-Funktion angewendet wird, gilt die Inklusion der daraus entstanden Mengen weiterhin. Dies entspricht der Behauptung dies Punktes.

3. Auch für diese Inklusion wird eine starke as-Verfeinerungs-Relation $\mathcal{R}$ und eine Implementierung $P'$ angegeben. Jedoch werden nicht wie bei 1. alle Transitionen von $P$ in $P'$ implementiert. Es wird auch für jedes $wa$ aus $MIT(P)$ eine eigene Implementierung $P'$ geben und nicht eine für alle. Es werden alle must-Transitionen aus $P$ in $P'$ implementiert und zusätzlich die may-Transitionen, die zum Ausführen von $w$ benötigt werden, so dass das $a$ danach in $P$ nicht gefordert wird. Es kann aufgrund möglicher Schleifen in $P$ auch nicht mehr die Identitäts-Relation als as-Verfeinerungs-Relation gewählt werden. Der Trace $w$ wird entsprechend seiner Länge abgewickelt, so dass sicher gestellt wird, dass der Zustand am Ende dieses Traces wirklich einen fehlenden Input aufweist. Für das Abwickeln werden die Zustände entsprechend ihrer Position im Ablauf, auf dem $w$ ausgeführt wird, durchnummeriert. Für ein $w$, für das $wa \in MIT(P)$, gilt: $\exists w' \in \Sigma_\tau^*, \exists \alpha_1, \alpha_2, \ldots, \alpha_n, \exists p_1, p_2, \ldots, p_n : \hat{w'} = w \wedge w' = \alpha_1\alpha_2 \ldots \alpha_n \wedge p_0 \overset{\alpha_1}{\dashrightarrow}_P p_1 \overset{\alpha_2}{\dashrightarrow}_P \ldots p_{n-1} \overset{\alpha_n}{\dashrightarrow}_P p_n \overset{a}{\not\longrightarrow}_P$. Die starke as-Verfeinerungs-Relation $\mathcal{R}$ enthält in diesem Fall Tupel $((p, j), p)$ für alle $0 \leq j \leq n$. Die entsprechende Definition für das $P'$, das $wa$ als Input-kritischen-Trace enthalten soll lautet:

   - $P' = P \times \{0, 1, \ldots n\}$,
   - $p'_0 = (p_0, 0)$,
   - $I'_P = I_P$ und $O'_P = O_P$,
   - $$\longrightarrow'_P = \dashrightarrow'_P = \left\{((p, j), \alpha, (p', j+1)) \mid p \overset{\alpha}{\dashrightarrow}_P p', 0 \leq j < n\right\} \cup \left\{((p, j), \alpha, (p', j)) \mid p \overset{\alpha}{\longrightarrow}_P p', 0 \leq j \leq n\right\},$$
   - $E_{P'} = \emptyset$.

   Der Ablauf von $w$ aus $P$ wird in $P'$ durch $(p_0, 0) \overset{\alpha_1}{\longrightarrow}_{P'} (p_1, 1) \overset{\alpha_2}{\longrightarrow}_{P'} \ldots (p_{n-1}, n-1) \overset{\alpha_n}{\longrightarrow}_{P'} (p_n, n)$ simuliert. $w$ ist also in $P'$ ausführbar. $a$ ist für $(p_n, n)$ nicht ausführbar, da in $P$ für $p_n$ $p_n \overset{a}{\not\longrightarrow}_P$ gilt und für die Zustände mit der Nummer $n$ in $P'$ nur die must-Transitionen implementiert werden. Die Transitionen $p \overset{\alpha}{\longrightarrow}_P p'$ aus $P$ werden in $P'$ durch die Transitionen $(p, j) \overset{\alpha}{\longrightarrow}_{P'} (p', j)$ gematched. Für die may-Transitionen $(p, j) \overset{\alpha}{\dashrightarrow}_{P'} (p', j+1)$ (bzw. $(p', j)$) in $P'$ sind die entsprechenden matchenden Transitionen in $P$ die Transition $p \overset{\alpha}{\dashrightarrow}_P p'$. Da die Menge $E_{P'}$ leer ist, gelten alle Bedingungen, damit $\mathcal{R}$ eine as-Verfeinerungs-Relation zwischen $P'$ und $P$ ist. Mit der Begründung von oben folgt auch $wa \in MIT(P')$. Da für alle $wa$ aus $MIT(P)$ eine entsprechende Implementierung mit as-Verfeinerungs-Relation $\mathcal{R}$ angegeben, werden kann, gilt die Inklusion.

□

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

- *Die Menge der* Fehler-Traces *von $P$ ist $ET(P) := \text{cont}(PrET(P)) \cup \text{cont}(MIT(P))$.*
- *Die* Fehler-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.
Aus den Propositionen für die Sprache und die Traces konnte für die Vereinigung der gleichen Mengen über die Implementierungen immer nur eine Inklusionsrichtung gefolgert werden, da die Definition 1.3 nach einen Fehler-Zustand in $P$ beliebiges Verhalten in dessen as-Implementierungen zulässt. Mit dem Einsatz der cont-Funktion zum beliebigen fortsetzten der Traces kann dies ausgeglichen werden. Somit gilt wie die nächsten Proposition behauptet für die Fehler-Traces und die Fehler-geflutete Sprache Gleichheit und nicht nur die Inklusion, die aus den Propositionen vorher bereits folgt.

**Proposition 3.6 (*Kommunikationsfehler-Semantik und Implementierungen*).** *Sie $P$ ein MEIO.*

1. *Für die Menge der Fehler-Traces von $P$ gilt $ET(P) = \bigcup_{P' \in \text{as-impl}(P)} ET(P')$.*
2. *Für die Fehler-geflutete Sprache von $P$ gilt $EL(P) = \bigcup_{P' \in \text{as-impl}(P)} EL(P')$.*

*Beweis.*
1. „$\subseteq$“:

$$\begin{aligned} ET(P) &\overset{3.5}{=} \text{cont}(PrET(P)) \cup \text{cont}(MIT(P)) \\ &\overset{3.4}{\subseteq} \text{cont}\left(\bigcup_{P' \in \text{as-impl}(P)} PrET(P')\right) \cup \text{cont}\left(\bigcup_{P' \in \text{as-impl}(P)} MIT(P')\right) \\ &\overset{\substack{\text{cont} \\ \text{monoton}}}{=} \bigcup_{P' \in \text{as-impl}(P)} \text{cont}(PrET(P')) \cup \text{cont}(MIT(P')) \\ &\overset{3.5}{=} \bigcup_{P' \in \text{as-impl}(P)} ET(P'). \end{aligned}$$

1. „$\supseteq$“:
Da für $P$ $ET(P) = \text{cont}(PrET(P)) \cup \text{cont}(MIT(P))$ gilt und für alle as-Implementierungen $P'$ von $P$ die analogen Gleichungen für $ET(P')$ gelten, genügt es ein präfixminimales $w$ aus $ET(P')$ für eine as-Implementierung $P'$ von $P$ zu betrachten. Da $w$

präfix-minimal ist für $P'$ ist $w$ entweder vollständig oder bis auf den letzten Buchstaben in $P'$ ausführbar. Dies hängt davon ab, ob $w \in PrET(P')$ oder $w \in MIT(P')$ gilt. Für $P$ muss jedoch nicht mal das Präfix von $w$ ohne den letzten Buchstaben von $w$ ausführbar sein.

- Fall 1 ($w \in PrET(P')$): In $P'$ existiert eine Verlängerung $v \in O^*$ von $w$, so dass $wv \in StET(P')$ gilt. Es gibt also Zustände $p'_1, p'_2, \ldots, p'_n$ in $P'$ und ein Wort $w'$ für das $\hat{w'} = wv$ und $w' = \alpha_1\alpha_2 \ldots \alpha_n$ gilt. Aus diesen Bausteinen ist der strikte Fehler-Trace $p'_0 \xrightarrow{\alpha_1}_{P'} p'_1 \xrightarrow{\alpha_2}_{P'} \ldots p'_{n-1} \xrightarrow{\alpha_n}_{P'} p'_n \in E_{P'}$ aufgebaut. Es kann oBdA davon ausgegangen werden, dass für alle Zustände außer $p'_n$ gilt, dass sie keine Fehler-Zustände sind. Da $P'$ eine as-Implementierung von $P$ ist, muss für die Startzustände der beiden Transitionssysteme $p'_0 \sqsubseteq_{\text{as}} p_0$ gelten. Es wird nun versucht via Induktion zu zeigen, dass man den strikten Fehler-Trace in $P$ nachmachen kann. Es soll dafür beweisen werden, falls $p_j \notin E_P$ gilt und $p'_j \sqsubseteq_{\text{as}} p_j$ bereits nachgewiesen wurde, dass auch $p'_{j+1} \sqsubseteq_{\text{as}} p_{j+1}$ gelten muss für ein $p_{j+1} \in P$ und $0 \leq j < n$. Für $j = 0$ gilt bereits $p'_j \sqsubseteq_{\text{as}} p_j$ wie oben beschrieben wurde. Falls ein $p_j$ vor $p_n$ in $E_P$ enthalten ist, dann ist ein Präfix von $wv$ ein strikter Fehler-Trace in $P$ und mit $\text{prune}(w) = \text{prune}(wv)$ ist somit ein Präfix von $w$ in $ET(P)$ enthalten. Da $ET$ unter Fortsetzung der Traces abgeschlossen ist, gilt auch $w \in ET(P)$. Es wird im Folgenden also davon ausgegangen, dass für alle $p_j$ mit $j < n$ gilt $p_j \notin E_P$. Es gelten somit alle Voraussetzungen für die Induktion und in $P'$ gibt es die Transition $p'_j \overset{\alpha_j}{\dashrightarrow}_{P'} p'_{j+1}$. Mit Definition 1.3 3. folgt draus, dass es auch eine Transition $p_j \overset{\alpha_j}{\dashrightarrow}_{P} p_{j+1}$ in $P$ geben muss, so dass $p'_{j+1} \sqsubseteq_{\text{as}} p_{j+1}$ gilt. Aus der Induktion kann nun gefolgert werden, dass $p'_n \sqsubseteq_{\text{as}} p_n$ gelten muss. Da $p'_n \in E_{P'}$ gilt, folgt draus mit 1.3 1., dass bereits $p_n$ in $E_P$ enthalten sein muss. Es gilt also $wv \in StET(P)$ und mit der Argumentation von oben folgt daraus $w \in ET(P)$.

- Fall 2 ($w \in MIT(P') \backslash PrET(P')$): Da $w$ in $MIT(P')$ enthalten ist, gibt es ein $a \in I$ für das $w = va$ gilt mit $v \in \Sigma^*$. Analog zu Fall 1 kann man auch den Input-kritischen Trace in $P'$ darstellen als $p'_0 \xrightarrow{\alpha_1}_{P'_1} p'_1 \xrightarrow{\alpha_2}_{P'_1} \ldots p'_{n-1} \xrightarrow{\alpha_n \not}_{P'_1}$ wobei $w = (\alpha_1\alpha_2 \ldots \alpha_n)|_\Sigma$ und $a = \alpha_n$ gilt. Mit einer analogen Induktion wie im letzten Fall für $0 \leq j < n-1$ kann $p'_{n-1} \sqsubseteq_{\text{as}} p_{n-1}$ für einen Zustand $p_{n-1}$ in $P$ gefolgert werden, falls nicht davor ein $p_j$ in der Induktion auftaucht, für das $p_j \in E_P$ gilt. Falls eines der $p_j$ mit $0 \leq j \leq n-1$ ein Fehler-Zustand ist, gilt $w \in ET(P)$ mit der analogen Argumentation wie oben. Es wird im folgenden davon ausgegangen, dass $p_{n-1}$ mit $p'_{n-1}$ in Relation steht und keine Fehler-Zustände aufgetreten sind. Falls $a$ für $p_{n-1}$ eine ausgehende must-Transition wäre, würde 1.3 2. auch für $p'_{n-1}$ die Implementierung der $a$ Transition fordern und somit würde nicht $va \in MIT(P')$ gelten. Es gilt also $p \overset{a}{\not\dashrightarrow}$ und $va \in MIT(P)$. Daraus ergibt sich direkt $w \in ET(P)$.

2. „$\subseteq$“:

$$EL(P) \overset{3.5}{=} L(P) \cup ET(P)$$

$$
\begin{aligned}
&\overset{2.1}{\subseteq} \left( \bigcup_{P' \in \text{as-impl}(P)} L(P') \right) \cup ET(P) \\
&\overset{1.}{=} \left( \bigcup_{P' \in \text{as-impl}(P)} L(P') \right) \cup \left( \bigcup_{P' \in \text{as-impl}(P)} ET(P') \right) \\
&= \bigcup_{P' \in \text{as-impl}(P)} L(P') \cup ET(P') \\
&\overset{3.5}{=} \bigcup_{P' \in \text{as-impl}(P)} EL(P').
\end{aligned}
$$

2. „$\supseteq$“:
Da der erste Punkt dieser Proposition bereits beweisen ist, reicht es aus für diesen Punkt zu zeigen, dass $\bigcup_{P' \in \text{as-impl}(P)} EL(P') \backslash ET(P')$ eine Teilmenge von $EL(P)$ ist. Die Menge $EL(P') \backslash ET(P')$ entspricht $L(P') \backslash ET(P')$. Es musst also ein Wort $w$ aus der Sprache einer as-Implementierung von $P$ betrachtet werden, dass nicht in den Fehler-Traces dieser as-Implementierung enthalten ist. Das Wort $w$ ist also in $P'$ ausführbar. Falls das $w$ in $P$ jedoch nicht ausführbar ist, folgt wie zuvor $w \in ET(P) \subseteq EL(P)$, da ein Präfix von $w$ in $P$ zu einem Fehler-Zustand führen musst. Falls $w$ ausführbar ist in $P$ gilt $w \in L(P) \subseteq EL(P)$. □

**Korollar 3.7 (*lokale Fehler Erreichbarkeit*).**

*(i) Es ist ein Fehler lokal erreichbar in einem MEIO $P \Leftrightarrow \exists$ as-Implementierung von $P$, in der ein Fehler lokal erreichbar ist.*

*(ii) Falls für zwei MEIOs $P_1 \sqsubseteq_E^{\mathrm{B}} P_2$ gilt und in $P_1$ ein Fehler lokal erreichbar ist, dann ist auch in $P_2$ ein Fehler lokal erreichbar.*

*Beweis.*
$(i) \Rightarrow$:
Da ein Fehler in $P$ lokal erreichbar ist, gilt $\varepsilon \in PrET_P \subseteq ET_P$. Es muss aufgrund von Proposition 3.6 1. mindestens ein $P' \in \text{as-impl}(P)$ geben, für dass $\varepsilon \in ET_{P'}$ gilt. Dies kann nur der Fall sein, wenn durch lokale Aktionen in $P'$ ein Fehler-Zustand erreicht werden kann. Es ist also auch ein der as-Implikation $P'$ ein Fehler lokal erreichbar, da $ET$ die Fortsetzungen von $PrET$ und $MIT$ enthält und Elemente aus $MIT$ mindestens die Länge 1 haben müssen.

$(i) \Leftarrow$:
Sei $P'$ die as-Implementierung von $P$, in der ein Fehler lokal erreichbar ist. Es gilt dann $\varepsilon \in PrET(P') \subseteq ET_{P'}$. Mit Proposition 3.6 1. folgt daraus $\varepsilon \in ET_P$. Es muss also auch in $P$ ein Fehler-Zustand lokal erreichbar sein.

(*ii*):
Da für $P_1$ ein Fehler lokale erreichbar ist folgt mit 1. dass es auch eine as-Implikation $P_1'$ von $P_1$ gibt, für die ein Fehler lokal erreichbar ist. Mit $P_1 \sqsubseteq_E^{\mathrm{B}} P_2$ ergibt sich daraus, dass es auch eine as-Implementierung $P_2'$ von $P_2$ geben muss, die lokal einen Fehler-Zustand erreichen kann. $P_2$ muss aufgrund von 1. ebenfalls einen Fehler lokal erreichen können, da es eine as-Implementierung gibt, die dies kann. □

**Satz 3.8 (*Kommunikationsfehler-Semantik für Parallelkompositionen*).** *Für zwei komponierbare MEIOs $P_1, P_2$ und ihre Komposition $P_{12}$ gilt:*

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

*Beweis.*
1. „$\subseteq$“:
Da beide Seiten der Gleichung unter der Fortsetzung cont abgeschlossen sind, genügt es ein präfix-minimales Element $w$ aus $ET_{12}$ zu betrachten. Diese Element ist aufgrund der Definition der Menge der Fehler-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 $(p_{01}, p_{02}) \stackrel{x}{\Rightarrow}_{12} (p_1, p_2) \wedge a \in I_{12} \wedge (p_1, p_2) \stackrel{a}{\not\rightarrow}_{12}$. Da $I_{12} = (I_1 \cup I_2) \backslash (O_1 \cup O_2)$ ist, folgt $a \in (I_1 \cup I_2)$ und $a \notin (O_1 \cup O_2)$. Es wird unterschieden, ob $a \in (I_1 \cap I_2)$ oder $a \in (I_1 \cup I_2) \backslash (I_1 \cap I_2)$ ist.
  - Fall 1a) ($a \in (I_1 \cap I_2)$): Durch Projektion des Ablaufes auf die einzelnen Transitionssysteme erhält man oBdA $p_{01} \stackrel{x_1}{\Rightarrow}_1 p_1 \stackrel{a}{\not\rightarrow}_1$ und $p_{02} \stackrel{x_2}{\Rightarrow}_2 p_2 \stackrel{a}{\not\rightarrow}_2$ oder $p_{02} \stackrel{x_2}{\Rightarrow}_2 p_2 \stackrel{a}{\rightarrow}_2$ mit $x \in x_1 \| x_2$. Daraus kann $x_1 a \in MIT_1 \subseteq ET_1$ und $x_2 a \in EL_2$ ($x_2 a \in MIT_2$ oder $x_2 a \in L_2$) gefolgert werden. Damit folgt $w \in (x_1 \| x_2) \cdot \{a\} = (x_1 a) \| (x_2 a) \subseteq ET_1 \| EL_2$, und somit ist $w$ in der rechten Seite der Gleichung enthalten.
  - Fall 1b) ($a \in (I_1 \cup I_2) \backslash (I_1 \cap I_2)$): OBdA gilt $a \in I_1$. Durch die Projektion auf die einzelnen Komponenten erhält man: $p_{01} \stackrel{x_1}{\Rightarrow}_1 p_1 \stackrel{a}{\not\rightarrow}_1$ und $p_{02} \stackrel{x_2}{\Rightarrow}_2 p_2$ mit $x \in x_1 \| x_2$. Daraus folgt $x_1 a \in MIT_1 \subseteq ET_1$ und $x_2 \in L_2 \subseteq EL_2$. Somit gilt $w \in (x_1 \| x_2) \cdot \{a\} \subseteq (x_1 a) \| x_2 \subseteq ET_1 \| EL_2$. Dies ist eine Teilmenge der rechten Seite der Gleichung.
- Fall 2 ($w \in PrET_{12}$): Aus der Definition von $PrET$ und prune folgt, dass ein $v \in O_{12}^*$ gibt, so dass $(p_{01}, p_{02}) \stackrel{w}{\Rightarrow}_{12} (p_1, p_2) \stackrel{v}{\Rightarrow}_{12} (p_1', p_2')$ gilt mit $(p_1', p_2') \in E_{12}$ und $w = \mathrm{prune}(wv)$. Durch Projektion auf die Komponenten erhält man $p_{01} \stackrel{w_1}{\Rightarrow}_1 p_1 \stackrel{v_1}{\Rightarrow}_1 p_1'$ und $p_{02} \stackrel{w_2}{\Rightarrow}_2 p_2 \stackrel{v_2}{\Rightarrow}_2 p_2'$ mit $w \in w_1 \| w_2$ und $v \in v_1 \| v_2$. 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 Fehler hingegen wäre durch das fehlende Sicherstellen der Synchronisation (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 $p_{02} \overset{w_2v_2}{\Longrightarrow}_2$ gilt, erhält man $w_2v_2 \in L_2 \subseteq EL_2$. Dadurch ergibt sich $wv \in ET_1 \| EL_2$ mit $w = \text{prune}(wv)$ und somit ist $w$ in der rechten Seite der Gleichung enthalten.

- Fall 2b) (neuer Fehler): OBdA gilt $a \in I_1 \cap O_2$ mit $p'_1 \overset{a}{\not\rightarrow}_1$ und $p'_2 \overset{a}{\dashrightarrow}_2$. Daraus folgt $w_1v_1a \in MIT_1 \subseteq ET_1$ und $w_2v_2a \in L_2 \subseteq EL_2$. Damit ergibt sich $wva \in ET_1 \| EL_2$, 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 cont wird auch in diesem Fall nur ein präfix-minimales Element $x \in \text{prune}((ET_1 \| EL_2) \cup (EL_1 \| ET_2))$ betrachtet. Da $x$ durch die Anwendung der prune-Funktion entstanden ist, existiert ein $y \in O_{12}^*$ mit $xy \in (ET_1 \| EL_2) \cup (EL_1 \| ET_2)$. OBdA wird davon ausgegangen, dass $xy \in ET_1 \| EL_2$ gilt, d.h. es gibt $w_1 \in ET_1$ und $w_2 \in EL_2$ mit $xy \in w_1 \| w_2$.
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. Also $v$ entweder auf einen Input $I_{12}$ endet oder $v = \varepsilon$. Damit muss $v$ ein Präfix von $x$ sein, denn $\varepsilon$ 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 $v_1$. Es gilt dann $w_2 \in ET_2$ 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 den beiden vorangegangen Fällen erhält man $v'_2 = v'|_{\Sigma_2}$ ist ein Präfix von $v_2$ und $v' \in v_1 \| v'_2$ ist ein Präfix von $xy$. Es kann nur für die Fälle $a \notin I_2$ gefolgert werden, dass $p_{02} \overset{v'_2}{\Longrightarrow}_2$ gilt. Falls $p_{02} \overset{v'_2}{\Longrightarrow}$ nicht gilt, ist $v'_2 = v_2 \in MIT_2$ und $v'_2$ endet auf $a \in I_2$.

- Fall I ($v_1 \in MIT_1$): Da $v_1 \in MIT_1$ gilt, muss $v_1$ ungleich $\varepsilon$ sein. Es gibt einen Ablauf der Form $p_{01} \stackrel{v_1'}{\Longrightarrow}_1 p_1 \stackrel{a}{\not\longrightarrow}_1$ und es gilt $v' = v''a$.

  - Fall Ia) ($a \notin \Sigma_2$): Es gilt $p_{02} \stackrel{v_2'}{\Longrightarrow}_2 p_2$ mit $v'' \in v_1' \| v_2'$. Dadurch erhält man $(p_{01}, p_{02}) \stackrel{v''}{\Longrightarrow}_{12} (p_1, p_2) \stackrel{a}{\not\longrightarrow}_{12}$ 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 $p_{02} \stackrel{v_2''}{\Longrightarrow}_2 p_2 \stackrel{a}{\not\longrightarrow}_2$ und $v'' \in v_1' \| v_2''$. $a$ ist für $P_2$, ebenso wie für $P_1$, ein nicht sichergestellter Input. Daraus folgt, dass $(p_1, p_2) \stackrel{a}{\not\longrightarrow}_{12}$ gilt. Es wird ebenfalls $v := v''a = v' \in MIT_{12}$ gewählt.

  - Fall Ic) ($a \in I_2$ und $v_2' \notin MIT_2$): Es gilt $p_{02} \stackrel{v_2''}{\Longrightarrow}_2 p_2 \stackrel{a}{\longrightarrow}_2$ mit $v_2' = v_2''a \in L_2$. Da die gemeinsamen Inputs synchronisiert werden, folgt $(p_1, p_2) \stackrel{a}{\not\longrightarrow}_{12}$ bereits aus $q_1 \stackrel{a}{\not\longrightarrow}_1$. 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 $p_{02} \stackrel{v_2'}{\Longrightarrow}_2$. Man erhält also $p_{02} \stackrel{v_2''}{\Longrightarrow}_2 p_2 \stackrel{a}{\dashrightarrow}_2$ mit $v'' \in v_1' \| v_2''$. Daraus ergibt sich $(p_{01}, p_{02}) \stackrel{v''}{\Longrightarrow}_{12} (p_1, p_2)$ mit $p_2 \stackrel{a}{\dashrightarrow}_1$, $p_1 \stackrel{a}{\not\longrightarrow}_1$, $a \in I_1$ 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 ($v_1 \in PrET_1$): $\exists u_1 \in O_1^* : p_{01} \stackrel{v_1}{\Longrightarrow}_1 p_1 \stackrel{u_1}{\Longrightarrow}_1 p_1'$ 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 $p_{02} \stackrel{v_2'}{\Longrightarrow}_2 p_2$ mit $(p_{01}, p_{02}) \stackrel{v'}{\Longrightarrow}_{12} (p_1, p_2)$.

  - Fall IIa) $\Big(u_2 \in (O_1 \cap I_2)^*, c \in (O_1 \cap I_2)$, sodass $u_2c$ Präfix von $u_1|_{I_2}$ mit $p_2 \stackrel{u_2}{\Longrightarrow}_2 p_2' \stackrel{c}{\not\longrightarrow}_2\Big)$: Für das Präfix $u_1'c$ von $u_1$ mit $(u_1'c)|_{I_2} = u_2c$ weiß man, dass $p_1 \stackrel{u_1'}{\Longrightarrow}_1 p_1'' \stackrel{c}{\dashrightarrow}_1$. Somit gilt $u_1' \in u_1' \| u_2$ und $(p_1, p_2) \stackrel{u_1'}{\Longrightarrow}_{12} (p_1'', p_2') \in E_{12}$, da für $P_2$ der entsprechende Input nicht sichergestellt wird, der mit dem $c$ Output von $P_1$ zu koppeln wäre. Es handelt sich also um einen neuen Fehler. Es wird $v := \text{prune}(v'u_1') \in PrET_{12}$ gewählt, dies ist ein Präfix von $v'$, da $u_1 \in O_1^*$.

  - Fall IIb) $\Big(p_2 \stackrel{u_2}{\Longrightarrow}_2 p_2'$ mit $u_2 = u_1|_{I_2}\Big)$: Es gilt $u_1 \in u_1 \| u_2$ und $(p_1, p_2) \stackrel{u_1}{\Longrightarrow}_{12} (p_1', p_2') \in E_{12}$, 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 $L_j \subseteq EL_j$ und $ET_j \subseteq EL_j$ gilt. Die Argumentation startet auf den rechten Seite der Gleichung:

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

□

**Korollar 3.9 (*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 cont, prune und $\cdot \| \cdot$ auf Sprachen wie folgt:

- $$\begin{aligned}ET_{31} &\overset{3.8\ 1.}{=} \operatorname{cont}(\operatorname{prune}((ET_3 \| EL_1) \cup (EL_3 \| ET_1))) \\ &\overset{\substack{ET_1 \subseteq ET_2 \\ \text{und} \\ EL_1 \subseteq EL_2}}{\subseteq} \operatorname{cont}(\operatorname{prune}((ET_3 \| EL_2) \cup (EL_3 \| ET_2))) \\ &\overset{3.8\ 1.}{=} ET_{32},\end{aligned}$$
- $$\begin{aligned}EL_{31} &\overset{3.8\ 2.}{=} (EL_3 \| EL_1) \cup E_{31} \\ &\overset{\substack{EL_1 \subseteq EL_2 \\ \text{und} \\ ET_{31} \subseteq ET_{32}}}{\subseteq} (EL_3 \| EL_2) \cup ET_{32} \\ &\overset{3.8\ 2.}{=} EL_{32}.\end{aligned}$$

□

**Lemma 3.10 (*Verfeinerung mit Kommunikationsfehlern*).** *Gegeben sind zwei MEIOs $P_1$ und $P_2$ mit der gleichen Signatur. Wenn $U \| P_1 \sqsubseteq_E^{\mathrm{B}} U \| P_2$ 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 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 $U \| P_1$. Wegen 3.7 (ii) erreicht auch $U \| P_2$ lokal einen Fehler-Zustand. 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 $\varepsilon \in PrET_2$.
- Fall 2 ($w = x_1 \dots x_n x_{n+1} \in \Sigma^+$ mit $n \geq 0$ und $x_{n+1} \in I = O_U$): Es wird der folgende Partner $U$ betrachtet (siehe auch Abbildung 3.1):
  - $U = \{p_0, p_1, \dots, p_{n+1}\}$,
  - $p_{0U} = p_0$,
  - $\dashrightarrow_U = \longrightarrow_U = \{(p_j, x_{j+1}, p_{j+1}) \mid 0 \leq j \leq n\}$
    $\cup \{(p_j, x, p_{n+1}) \mid x \in I_U \backslash \{x_{j+1}\}, 0 \leq j \leq n\}$
    $\cup \{(p_{n+1}, x, p_{n+1}) \mid x \in I_U\}$,
  - $E_U = \emptyset$.

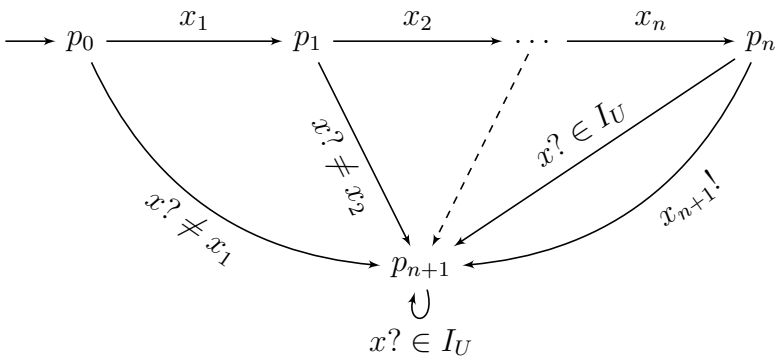


Abbildung 3.1: $x? \neq x_j$ steht für alle $x \in I_U \backslash \{x_j\}$

Für $w$ können nun zwei Fälle unterschieden werden. Aus beiden wird folgen, dass für $U \| P_1$ $\varepsilon \in PrET(U \| P_1)$ gilt.

- Fall 2a) ($w \in MIT_1$): In $U \| P_1$ erhält man $(p_0, p_{01}) \stackrel{x_1 \dots x_n}{\Longrightarrow}_{U \| P_1} (p_n, p')$ mit $p' \stackrel{x_{n+1}}{\not\longrightarrow}_1$ und $p_n \stackrel{x_{n+1}}{\longrightarrow}_U$. Deshalb gilt $(p_n, p') \in E_{U \| P_1}$. Da alle Aktionen aus $w$

bis auf $x_{n+1}$ synchronisiert werden und $I \cap I_U = \emptyset$, gilt $x_1, \ldots, x_n \in O_{U\|P_1}$. Da $(p_n, p') \in E_{U\|P_1}$ in $U\|P_1$ lokal erreichbar ist gilt $\varepsilon \in PrET(U\|P_1)$.

- Fall 2b) ($w \in PrET_1$): In der Parallelkomposition von $U$ und $P_1$ erhält man $(p_0, p_{01}) \stackrel{w}{\Rightarrow}_{U\|P_1} (p_{n+1}, p'') \stackrel{u}{\Rightarrow}_{U\|P_1} (p_{n+1}, p')$ für $u \in O^*$ und $p' \in E_1$. Daraus folgt $(p_{n+1}, p') \in E_{U\|P_1}$ und somit $wu \in StET(U\|P_1)$. Da alle Aktionen in $w$ synchronisiert werden und $I \cap I_U = \emptyset$, gilt $x_1, \ldots, x_n, x_{n+1} \in O_{U\|P_1}$ und, da $u \in O^*$, folgt $u \in O^*_{U\|P_1}$. Somit ergibt sich $\varepsilon \in PrET(U\|P_1)$.

Da $\varepsilon \in PrET(U\|P_1)$ gilt, kann durch $U\|P_1 \sqsubseteq^{\mathrm{B}}_E U\|P_2$ unter zuhilfenahme von 3.7 (ii) geschlossen werden, dass auch in $U\|P_2$ ein Fehler-Zustand lokal erreichbar sein muss.
Der in $U\|P_2$ erreichbare 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 sicherstellt, muss ein lokal erreichbarer Fehler-Zustand der Form sein, dass ein Output $a \in O_U$ von $U$ möglich ist, dessen Synchronisation in $P_2$ mit einem passenden Input nicht sichergestellt ist ($P_2$ enthält die entsprechende $a$ Transitionen nicht als must-Transition). Durch die Konstruktion von $U$ sind in $p_{n+1}$ keine Outputs möglich. Ein neuer Fehler muss also die Form $(p_i, p')$ haben mit $i \leq n, p' \stackrel{x_{i+1}}{\not\rightarrow}_2$ und $x_{i+1} \in O_U = I$. Durch Projektion erhält man dann $p_{02} \stackrel{x_1 \ldots x_i}{\Rightarrow}_2 p' \stackrel{x_{i+1}}{\not\rightarrow}_2$ und damit gilt $x_1 \ldots x_{i+1} \in MIT_2 \subseteq ET_2$. Somit ist ein Präfix von $w$ in $ET_2$ enthalten.

- Fall 2ii) (geerbter Fehler): $U$ hat $x_1 \ldots x_i u$ mit $u \in I_U^* = O^*$ ausgeführt und ebenso hat $P_2$ dieses Wort abgearbeitet. Durch dies hat $P_2$ einen Zustand in $E_2$ erreicht, da von $U$ keine Fehler geerbt werden können. Es gilt dann $\mathrm{prune}(x_1 \ldots x_i u) = \mathrm{prune}(x_1 \ldots x_i) \in PrET_2 \subseteq ET_2$. Da $x_1 \ldots x_i$ ein Präfix von $w$ ist, führt in diesem Fall eine Verlängerung um lokale Aktionen von einem Präfix von $w$ zu einem Fehler-Zustand. Da $ET$ der Menge aller Verlängerungen von gekürzten Fehler-Traces entspricht, ist $x_1 \ldots x_i$ in $ET_2$ enthalten und somit auch $w$ in $ET_2$ enthalten.

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

- Fall 1 ($w = \varepsilon$): Da $\varepsilon$ immer in $EL_2$ enthalten ist, muss hier nichts gezeigt werden.
- Fall 2 ($w = x_1 \ldots x_n$ mit $n \geq 1$): Es wird ein Partner $U$ wie folgt konstruiert (siehe dazu auch Abbildung 3.2):
  - $U = \{p_0, p_1, \ldots, p_n, p\}$,
  - $p_{0U} = p_0$,

- $\dashrightarrow_U = \longrightarrow_U = \{(p_j, x_{j+1}, p_{j+1}) \mid 0 \leq j < n\}$
  $\cup \{(p_j, x, p) \mid x \in I_U \backslash \{x_{j+1}\}, 0 \leq j < n\}$
  $\cup \{(p, x, p) \mid x \in I_U\},$
- $E_U = \{p_n\}$.

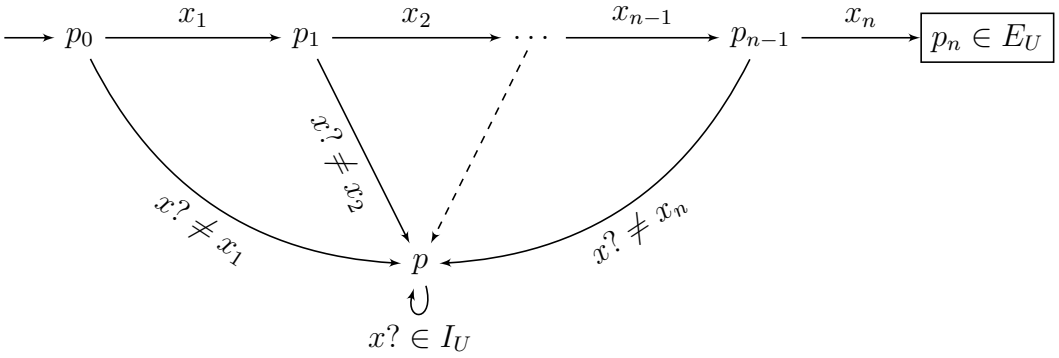


Abbildung 3.2: $x? \neq x_j$ steht für alle $x \in I_U \backslash \{x_j\}$, $p_n$ ist der einzige Fehler-Zustand

Da $p_{01} \overset{w}{\Longrightarrow}_1 p'$ gilt, kann man schließen, dass $U \| P_1$ einen lokal erreichbaren geerbten Fehler hat. Aufgrund von Korollar 3.7 (ii) muss ebenfalls ein Fehler-Zustand in $U \| P_2$ lokal erreichbar sein.

- Fall 2a) (neuer Fehler aufgrund von $x_i \in O_U$ und $p_{02} \overset{x_1 \ldots x_{i-1}}{\Longrightarrow}_2 p'' \overset{x_i}{\not\longrightarrow}_2$): Es gilt $x_1 \ldots x_i \in MIT_2$ 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 könnten.
- Fall 2b) (neuer Fehler aufgrund von $a \in O = I_U$): Der einzige Zustand, in dem $U$ nicht alle Inputs erlaubt sind, ist $p_n$, der bereits ein Fehler-Zustand ist. Da in diesem Fall der Fehler-Zustand in $U \| P_2$ 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 $p_{02} \overset{x_1 \ldots x_n}{\Longrightarrow}_2$ gilt. In diesem Fall ist $w$ ausführbar und es gilt $w \in L_2 \subseteq EL_2$.
- Fall 2d) (geerbter Fehler von $P_2$): Es gilt $p_{02} \overset{x_1 \ldots x_i u}{\Longrightarrow}_2 p' \in E_2$ für ein $i \geq 0$ und $u \in O^*$. Somit ist $x_1 \ldots x_i u \in StET_2$ und damit $\text{prune}(x_1 \ldots x_i u) = \text{prune}(x_1 \ldots x_i) \in PrET_2 \subseteq EL_2$. Es gilt also $w \in EL_2$.

□

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

**Satz 3.11 (*Vollständige Abstraktheit für Kommunikationsfehler-Semantik*).** *Für zwei MEIOs $P_1$ und $P_2$ mit derselben Signatur gilt $P_1 \sqsubseteq_E^{\mathrm{C}} P_2 \Leftrightarrow P_1 \sqsubseteq_E P_2$.*

*Beweis.*
„⇐“: 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. Diese as-Implementierung muss es geben, wenn $P_1$ und $P_2$ lokal erreichbare Fehler enthalten, wegen Korollar 3.7 (i). Dadurch folgt, dass $P_1 \sqsubseteq_E^{\mathrm{B}} P_2$ gilt, da $\sqsubseteq_E^{\mathrm{B}}$ in Definition 3.2 über die lokale Erreichbarkeit der Fehler-Zustande in den as-Implementierungen definiert wurde. Es ist also $\sqsubseteq_E$ in $\sqsubseteq_E^{\mathrm{B}}$ enthalten. Wie in Korollar 3.9 gezeigt, ist $\sqsubseteq_E$ eine Präkongruenz bezüglich $\cdot\|\cdot$. Da $\sqsubseteq_E^{\mathrm{C}}$ die gröbste Präkongruenz bezüglich $\cdot\|\cdot$ ist, die in $\sqsubseteq_E^{\mathrm{B}}$ enthalten ist, muss $\sqsubseteq_E$ in $\sqsubseteq_E^{\mathrm{C}}$ enthalten sein. Es folgt also aus $P_1 \sqsubseteq_E P_2$, dass auch $P_1 \sqsubseteq_E^{\mathrm{C}} P_2$ gilt.

„⇒“: Durch die Definition von $\sqsubseteq_E^{\mathrm{C}}$ als Präkongruenz in 3.2 folgt aus $P_1 \sqsubseteq_E^{\mathrm{C}} P_2$, dass $U\|P_1 \sqsubseteq_E^{\mathrm{C}} U\|P_2$ für alle MEIOs $U$ gilt, die mit $P_1$ komponierbar sind. Da $\sqsubseteq_E^{\mathrm{C}}$ nach Definition auch in $\sqsubseteq_E^{\mathrm{B}}$ enthalten sein soll, folgt aus $U\|P_1 \sqsubseteq_E^{\mathrm{C}} U\|P_2$ auch die Gültigkeit von $U\|P_1 \sqsubseteq_E^{\mathrm{B}} U\|P_2$ für alle diese MEIOs $U$. Mit Lemma 3.10 folgt dann $P_1 \sqsubseteq_E P_2$. □

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

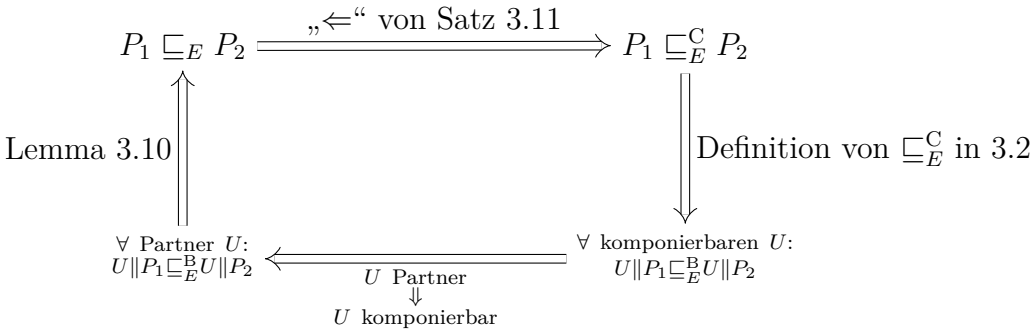


Abbildung 3.3: Folgerungskette der Fehler-Relationen

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 3.12.** *Es gilt: $P_1 \sqsubseteq_E P_2 \Leftrightarrow U\|P_1 \sqsubseteq_E^{\mathrm{B}} U\|P_2$ für alle Partner $U$.*

## 3.2 Testing-Ansatz

Der gröbste Präkongruenz-Ansatz aus dem letzten Teil stützt sich auf die Parallelkomposition von MEIOs, die wie im bereits in 2.3 1. erwähnt auch anders gestaltet hätte werden können. Spezielle bei neuen Fehler ist dort gezeigt worden, dass es zu einem ungewöhnlichen Verfeinerungs-Verhalten kommen kann, das möglicherweise gar nicht so gewollt ist. Deshalb ist es sinnvoll sich nur auf die Definition der Parallelkomposition von EIO zu stützen, die hier für die Implementierungen gilt. Dies führt zu dem in diesem Teil behandelten Testing-Ansatz. Dieser lässt jedoch wiederum keine Aussage zu gröbste Präkongruenz zu bestimmen, wie das im letzen Teil möglich war. Der Test ersetzt also die Basisrelation als grundlegende Definition für den Ansatz.
Der Testing-Ansatz stützt sich auf das Vorgehen, das in [BV15b] angewendet wurde. Jedoch sind Tests dort Tupel aus einer Implementierung und einer Menge an Aktionen, über denen synchronisiert werden soll. Die MEIOs, die hier komponiert werden bringen die Menge Synch an Aktionen, die synchronisiert werden bereits mit. Es wird im Gegensatz zu [BV15b] mit Inputs und Outputs gearbeitet und dadurch scheint der Ansatz die gemeinsamen Aktionen zu synchronisieren natürlicher, wie eine Menge an Aktionen beim Test vorzugeben.
Die Definition von lokaler Erreichbarkeit eines Fehler-Zustandes soll aus dem Erweiterungs-Ansatz übernommen werden. Es wird also trotzdem noch optimistisch davon ausgegangen, dass Fehler erst zu Problemen führen, wenn eine hilfreiche Umgebung dies nicht mehr verhindern kann.

**Definition 3.13 (*Test und Verfeinerung für Kommunikationsfehler*).** *Ein* Test $T$ *ist eine Implementierung. Ein MEIO* $P$ as-erfüllt *einen Kommunikationsfehler-Test* $T$*, falls* $S\|T$ *fehler-frei ist für alle* $S \in \text{as-impl}(P)$*. Es wird dann* $P \, sat^E_{\text{as}} \, T$ *geschrieben. Die Parallelkomposition* $S\|T$ *ist* fehler-frei*, wenn kein Fehler lokal erreichbar ist. Ein MEIO* $P$ Fehler-verfeinert $P'$*, falls für alle Tests* $T$*:* $P' \, sat^E_{\text{as}} \, T \Rightarrow P \, sat^E_{\text{as}} \, T$*.*

Abgesehen von Korollar 3.7 (ii) erweisen sich Definition 3.3 bis Korollar 3.9 auch hier als nützlich.
Die Basisrelation aus dem gröbsten Präkongruenz-Ansatz gibt es in diesem Teil nicht, somit kann das Lemma 3.10 nicht in dieser Art formuliert werden. Jedoch kann hier mit Tests gearbeitet werden und unter deren zuhilfenahme ein ähnliches Lemma formuliert werden.

**Lemma 3.14 (*Testing-Verfeinerung mit Kommunikationsfehlern*).** *Gegeben sind zwei MEIOs* $P_1$ *und* $P_2$ *mit der gleichen Signatur. Wenn für alle Tests* $T$*, die Partner von* $P_1$ *bzw.* $P_2$ *sind,* $P_2 \, sat^E_{\text{as}} \, T \Rightarrow P_1 \, sat^E_{\text{as}} \, T$ *gilt, dann folgt daraus die Gültigkeit von* $P_1 \sqsubseteq_E P_2$*.*

*Beweis.* Da $P_1$ und $P_2$ die gleichen Signaturen haben wird $I := I_1 = I_2$ und $O := O_1 = O_2$ definiert. Für jeden Test Partner $T$ gilt $I_T = O$ und $O_T = I$.
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 $w \in ET_1$ wird gezeigt, dass dies $w$ oder eines seiner Präfixe in $ET_2$ enthalten ist. Dies ist möglich, da die beiden Mengen $ET_1$ und $ET_2$ durch cont abgeschlossen sind.
Mit Proposition 3.6 folgt aus $w \in ET_1$, dass es auch eine as-Implementierung $P_1'$ von $P_1$ geben muss, für die $w$ ebenfalls in $ET_{P_1'}$ enthalten ist.

- Fall 1 ($w = \varepsilon$): Es ist ein Fehler-Zustand in $P_1'$ lokal erreichbar. Für $T$ wird ein Transitionssystem verwendet, das nur aus dem Startzustand und einer must-Schleife für alle Inputs $x \in I_T$ besteht. Somit kann $P_1'$ die im Prinzip gleichen Fehler-Zustände lokal erreichen wie $P_1' \| T$. $P_1'$ ist in Parallelkomposition mit $T$ nicht fehler-frei somit gilt $P_1 \, sat_{\text{as}}^E \, T$ nicht. Es darf also auch $P_2$ den Test $T$ nicht as-erfüllen, wegen der Implikation $P_2 \, sat_{\text{as}}^E \, T \Rightarrow P_1 \, sat_{\text{as}}^E \, T$. Damit $P_2$ $T$ nicht as-erfüllt muss es eine as-Implementierungen $P_2'$ geben, die in Parallelkomposition mit $T$ zu einem nicht fehler-freien System führt. Es muss also in $P_2' \| T$ ein Fehler lokal erreichbar sein. Durch die Definition von $T$ kann dieser Fehler nur von $P_2'$ geerbt sein. In $P_2'$ kann dieser Fehler-Zustand nur durch interen Aktionen und Outputs erreichbar sein, da $T$ keine Outputs besitzt, die man mit Inputs aus $P_2'$ synchronisieren könnte und unsynchronisierte Aktionen sind in einer Parallelkomposition von Partner nicht möglich. Somit gilt $\varepsilon \in PrET_{P_2'} \subseteq ET_{P_2'}$. Mit Proposition 3.6 folgt daraus $\varepsilon \in ET_2$.

- Fall 2 ($w = x_1 \ldots x_n x_{n+1} \in \Sigma^+$ mit $n \geq 0$ und $x_{n+1} \in I = O_T$): Es wird der folgende Partner $T$ betrachtet (dieser entspricht bis auf die Benennung der Mengen dem Transitionssystem $U$ aus Abbildung 3.1):

  - $T = \{p_0, p_1, \ldots, p_{n+1}\}$,
  - $p_{0T} = p_0$,
  - $\dashrightarrow_T = \longrightarrow_T = \{(p_j, x_{j+1}, p_{j+1}) \mid 0 \leq j \leq n\}$
    $\cup \{(p_j, x, p_{n+1}) \mid x \in I_T \backslash \{x_{j+1}\}, 0 \leq j \leq n\}$
    $\cup \{(p_{n+1}, x, p_{n+1}) \mid x \in I_T\}$,
  - $E_T = \emptyset$.

  Für $w$ können nun zwei Fälle unterschieden werden, für die beide $\varepsilon \in PrET(P_1' \| T)$ folgen wird.

  - Fall 2a) ($w \in MIT_{P_1'}$): In $P_1' \| T$ erhält man $(p_{01}', p_0) \stackrel{x_1 \ldots x_n}{\Longrightarrow}_{P_1' \| T} (p', p_n)$ mit $p' \stackrel{x_{n+1}}{\not\longrightarrow}_{P_1'}$ und $p_n \stackrel{x_{n+1}}{\longrightarrow}_T$. Deshalb gilt $(p', p_n) \in E_{P_1' \| T}$. Da alle Aktionen aus $w$ bis auf $x_{n+1}$ synchronisiert werden und $I \cap I_T = \emptyset$, gilt $x_1, \ldots, x_n \in O_{P_1' \| T}$. Es gilt also $\varepsilon \in PrET(P_1' \| T)$.

– Fall 2b) ($w \in PrET_{P_1'}$): In der Parallelkomposition $P_1'\|T$ erhält man die Transitionsfolge $(p_{01}, p_0) \stackrel{w}{\Longrightarrow}_{P_1'\|T} (p'', p_{n+1}) \stackrel{u}{\Longrightarrow}_{P_1'\|T} (p', p_{n+1})$ für $u \in O^*$ und $p' \in E_{P_1'}$. Daraus folgt $(p', p_{n+1}) \in E_{P_1'\|T}$ und somit $wu \in StET(P_1'\|T)$. Da alle Aktionen in $w$ synchronisiert werden und $I \cap I_T = \emptyset$, gilt $x_1, \ldots, x_n, x_{n+1} \in O_{P_1'\|T}$ und, da $u \in O^*$, folgt $u \in O^*_{P_1'\|T}$. Somit ergibt sich $\varepsilon \in PrET(P_1'\|T)$.

Da $\varepsilon \in PrET(P_1'\|T)$ gilt, kann durch $P_2\, sat^E_{\text{as}}\, T \Rightarrow P_1\, sat^E_{\text{as}}\, T$ geschlossen werden, dass auch $P_2\, sat^E_{\text{as}}\, T$ nicht gelten kann. Die Relation gilt für $P_2$ nicht, da es eine as-Implementierung $P_2'$ von $P_2$ gibt, so dass $P_2'\|T$ nicht fehler-frei ist.
Der lokal erreichbar Fehler in $P_2'\|T$ kann geerbt oder neu sein.

– Fall 2i) (neuer Fehler): Da jeder Zustand von $T$ alle Inputs $x \in O = I_T$ zulässt, muss ein lokal erreichbarer Fehler-Zustand der Form sein, dass ein Outputs $a \in O_T$ von $T$ möglich ist, der nicht mit einem passenden Input aus $P_2'$ synchronisiert werden werden kann. Durch die Konstruktion von $T$ sind in $p_{n+1}$ keine Outputs möglich. Ein neuer Fehler muss also die Form $(p', p_i)$ haben mit $i \leq n$, $p' \stackrel{x_{i+1}}{\not\longrightarrow}_{P_2'}$ und $x_{i+1} \in O_T = I$. Durch Projektion erhält man dann $p_{02} \stackrel{x_1 \ldots x_i}{\Longrightarrow}_{P_2'} p' \stackrel{x_{i+1}}{\not\longrightarrow}_{P_2'}$ und damit gilt $x_1 \ldots x_{i+1} \in MIT_{P_2'} \subseteq ET_{P_2'}$. Es ist also ein Präfix von $w$ in $ET_{P_2'}$ enthalten und mit Proposition 3.6 auch in $ET_2$.

– Fall 2ii) (geerbter Fehler): $T$ hat $x_1 \ldots x_i u$ mit $u \in I_T^* = O^*$ ausgeführt und ebenso hat $P_2'$ dieses Wort abgearbeitet. Durch dies hat $P_2'$ einen Zustand in $E_{P_2'}$ erreicht, da von $T$ keine Fehler geerbt werden können. Es gilt dann $\text{prune}(x_1 \ldots x_i u) = \text{prune}(x_1 \ldots x_i) \in PrET_{P_2'} \subseteq ET_{P_2'}$. Da $x_1 \ldots x_i$ ein Präfix von $w$ ist, führt in diesem Fall eine Verlängerung um lokale Aktionen von einem Präfix von $w$ zu einem Fehler-Zustand. Da $ET$ der Menge aller Verlängerungen von gekürzten Fehler-Traces entspricht, ist $x_1 \ldots x_i$ in $ET_{P_2'}$ enthalten und somit ist mit Proposition 3.6 ein Präfix von $w$ in $ET_2$ enthalten.

Um die andere Inklusion zu beweisen, reich es aufgrund der ersten Inklusion und der Definition von $EL$ aus zu zeigen, dass $L_1 \backslash ET_1 \subseteq EL_2$ gilt.
Es wird dafür ein beliebiges $w \in L_1 \backslash ET_1$ gewählt und gezeigt, dass es in $EL_2$ enthalten ist. Das $w$ ist wegen der Propositionen 2.1 und 3.6 auch für eine as-Implementierung $P_1'$ von $P_1$ in $L_{P_1'} \backslash ET_{P_1'}$ enthalten.

- Fall 1 ($w = \varepsilon$): Da $\varepsilon$ immer in $EL_2$ enthalten ist, muss hier nichts gezeigt werden.
- Fall 2 ($w = x_1 \ldots x_n$ mit $n \geq 1$): Es wird ein Partner $T$ wie folgt konstruiert ($T$ entspricht dabei $U$ aus Abbildung 3.2 bis auf die Benennung der Mengen):
  - $T = \{p_0, p_1, \ldots, p_n, p\}$,
  - $p_{0T} = p_0$,
  - $\dashrightarrow_T = \longrightarrow_T = \{(p_j, x_{j+1}, p_{j+1}) \mid 0 \leq j < n\}$
    $\cup \{(p_j, x, p) \mid x \in I_T \backslash \{x_{j+1}\}, 0 \leq j < n\}$
    $\cup \{(p, x, p) \mid x \in I_T\}$,

– $E_T = \{p_n\}$.

Da $p_{01} \stackrel{w}{\Longrightarrow}_{P_1'} p'$ gilt, kann man schließen, dass $P_1'\|T$ ein lokal erreichbaren geerbten Fehler hat. Es muss also auch eine as-Implementierung $P_2'$ von $P_2$ geben, für die $P_2'\|T$ einen lokal erreichbaren Fehler-Zustand hat.

– Fall 2a) (neuer Fehler aufgrund von $x_i \in O_T$ und $p_{02} \stackrel{x_1 \dots x_{i-1}}{\Longrightarrow}_{P_2'} p'' \stackrel{x_i}{\not\longrightarrow}_{P_2'}$): Es gilt $x_1 \dots x_i \in MIT_{P_2'}$ und somit $w \in EL_{P_2'}$. Anzumerken ist, dass es nur auf diesem Weg Outputs von $T$ möglich sind, deshalb gibt es keine anderen Outputs von $T$, die zu einem neuen Fehler führen können. Es gilt $w \in EL_2$ wegen Proposition 3.6.

– Fall 2b) (neuer Fehler aufgrund von $a \in O = I_T$): Der einzige Zustand, in dem $T$ nicht alle Inputs erlaubt sind, ist $p_n$, der bereits ein Fehler-Zustand ist. Da in diesem Fall der Fehler-Zustand in $P_2'\|T$ erreichbar ist, besitzt der komponierte MEIO einen geerbten Fehler und es gilt $w \in L_{P_2'} \subseteq EL_2$, wegen dem folgenden Fall 2c).

– Fall 2c) (geerbter Fehler von $T$): Da $p_n$ der einzige Fehler-Zustand in $T$ ist und alle Aktionen synchronisiert sind, ist dies nur möglich, wenn $p_{02} \stackrel{x_1 \dots x_n}{\Longrightarrow}_{P_2'}$ gilt. In diesem Fall ist $w \in L_{P_2'} \subseteq EL_{P_2'}$. Daraus folgt mit Proposition 3.6 $w \in EL_2$.

– Fall 2d) (geerbter Fehler von $P_2'$): Es gilt $p_{02} \stackrel{x_1 \dots x_n}{\Longrightarrow}_{P_2'} p' \in E_{P_2'}$ für ein $i \geq 0$ und $u \in O^*$. Somit ist $x_1 \dots x_i u \in StET_{P_2'}$ und damit $\text{prune}(x_1 \dots x_i u) = \text{prune}(x_1 \dots x_i) \in PrET_{P_2'} \subseteq EL_{P_2'}$. Mit Hilfe von Proposition 3.6 folgt $w \in EL_{P_2'} \subseteq EL_2$.

□

**Satz 3.15.** *Falls $P_1 \sqsubseteq_E P_2$ gilt folgt draus, dass $P_1$ $P_2$ Fehler-verfeinert.*

*Beweis.* 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. Aufgrund von Proposition 3.7 (i) gibt es dann auch entsprechende as-Implementierungen $P_1'$ und $P_2'$ für $P_1$ bzw. $P_2$, die ebenfalls Fehler-Zustände lokal erreichen können, wenn dies in $P_1$ bzw. $P_2$ möglich ist. Für alle Tests $T$ gilt, dass $P_j'\|T$ für $j \in \{1, 2\}$ einen lokal erreichbaren Fehler hat, wenn $P_j'$ einen solchen hat. Dies Schlussfolgerung ist möglich, da Satz 3.8 1. aussagt, dass die Parallelkomposition von einem Fehler-Trace aus $ET_{P_j'}$ mit einem Wort aus $EL_T$ in den Fehler-Traces von $ET_{P_j'\|T}$ enthalten ist. $\varepsilon$ ist ein Element von $ET_{P_j'}$, da ein Fehler lokal erreichbar ist und $\varepsilon \in EL_T$ gilt für alle Test $T$ und somit folgt $\varepsilon \in ET_{P_j'\|T}$. Dies impliziert die lokal Fehler Erreichbarkeit in $P_j'\|T$. Falls also $P_1$ einen lokal erreichbaren Fehler-Zustand enthält, dann zeigt sich dies Verhalten auch in einer as-Implementierungen und ebenfalls in der Parallelkomposition der as-Implementierung mit allen Tests $T$. Aus einem lokal erreichbaren Fehler in $P_1$ folgt auch

ein lokal erreichbarer Fehler-Zustand in $P_2$ und somit auch ein lokal erreichbarer Fehler-Zustand in der Parallelkomposition einer as-Implementierung von $P_2$ mit allen Tests $T$. Aus $P_1 \sqsubseteq_E P_2$ folgt also die Implikation $\neg P_1\, sat^E_{\text{as}}\, T \Rightarrow \neg P_2\, sat^E_{\text{as}}\, T$ für alle Test $T$. Wenn man die Negationen entfernt, ergibt sich für alle Tests $T$: $P_2\, sat^E_{\text{as}}\, T \Rightarrow P_1\, sat^E_{\text{as}}\, T$. Dies entspricht der Definition von $P_1$ Fehler-verfeinert $P_2$. □

Auch die in diesem Abschnitt gezeigten Folgerungen schließen sich zu einem Ring. Dies ist in Abbildung 3.4 dargestellt.

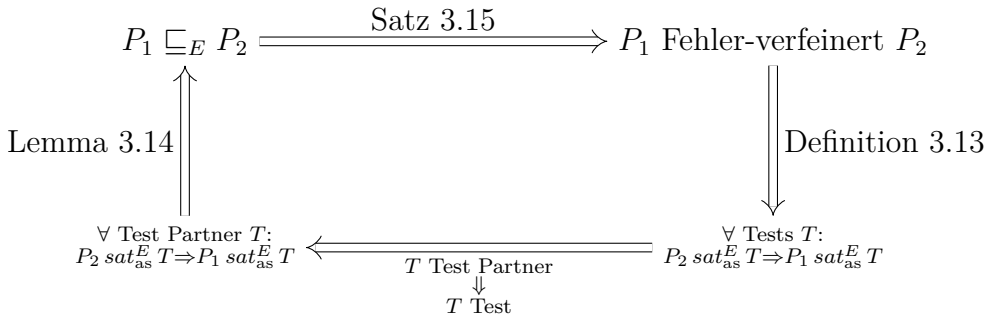


Abbildung 3.4: Folgerungskette der Testing-Verfeinerung und Fehler-Relation

## 3.3 Zusammenhänge

**Satz 3.16 (*Zusammenhang der Verfeinerungs-Relationen mit der Fehler-Relation*).** *Für MEIOs $P$ und $Q$ gilt $P \sqsubseteq_{\text{as}} Q \Rightarrow P \sqsubseteq_{\text{w-as}} Q \Rightarrow P \sqsubseteq_E Q$. Die Implikationen in die andere Richtung gelten jedoch nicht.*

*Beweis.*
$P \sqsubseteq_{\text{as}} Q \Rightarrow P \sqsubseteq_{\text{w-as}} Q$:
Um diese Implikation zu zeigen, muss man nachweisen, dass jede starke as-Verfeinerungs-Relation auch die Definition 1.4 der schwachen as-Verfeinerungs-Relation erfüllt. In beiden Simulations-Definitionen (1.3 und 1.4) müssen die Punkte für alle $(p, q) \in \mathcal{R}$ mit $q \notin E_Q$ gelten. Sei $\mathcal{R}$ nun eine as-Verfeinerungs-Relation. Es gilt also mit 1.3 1. dass $p$ kein Fehler-Zustand von $P$ ist. Somit ist auch 1. von 1.4 erfüllt. Für alle $\alpha \in \Sigma_\tau$ impliziert 1.3 2. $q \xrightarrow{\alpha}_Q q'$ $p \xrightarrow{\alpha}_P p'$ für ein $p'$ mit $p' \mathcal{R} q'$. Da $\Sigma_\tau = I \cup O \cup \{\tau\}$ gilt, sind dadurch 2. und 3. der Definition 1.4 erfüllt. Die schwache $\varepsilon$-Transition aus 2. führt keine echten Transitionen aus, sondern bleibt beim Zustand $p'$. Die schwache $\hat{\omega}$-Transition aus 3. entspricht in $\mathcal{R}$ nur einer einzigen Transition für $\omega$. Die Punkte 4. und 5. aus Definition 1.4 werden durch 1.3 3. erfüllt. Es gilt für $\mathcal{R}$ $p \overset{\alpha}{\dashrightarrow}_P p'$ impliziert $q \overset{\alpha}{\dashrightarrow}_Q q'$ für ein $q'$ mit $p' \mathcal{R} q'$. Die in 1.4 geforderten schwachen may-Transitionen werden hier jeweils stark

durch eine einzige Transition umgesetzt. $\mathcal{R}$ ist also auch eine schwache as-Verfeinerungs-Relation.

$P \sqsubseteq_{\text{w-as}} Q \Rightarrow P \sqsubseteq_E Q$:
Um diese Implikation zu beweisen wird gezeigt, dass eine beliebige schwache as-Verfeinerungs-Relation $\mathcal{R}$ zwischen $P$ und $Q$ auch die Eigenschaften der Relation $\sqsubseteq_E$ erfüllt. Da $\mathcal{R}$ eine schwache as-Verfeinerungs-Relation zwischen $P$ und $Q$ ist, muss $p_0 \mathcal{R} q_0$ gelten. Es sind die folgenden Punkte nachzuweisen:

- $ET_P \subseteq ET_Q$,
- $EL_P \subseteq EL_Q$.

Für den ersten Punkt wird ein beliebiges $w$ aus $ET_P$ betrachtet und gezeigt, dass dieses auch in $ET_Q$ enthalten ist. Es kann davon ausgegangen werden, dass $w$ präfix-minimal ist, da beide $ET$-Mengen unter cont abgeschlossen sind. $w$ kann ein Element aus $PrET(P)$ sein oder ein Element aus $MIT(P)$.

- Fall 1 ($w \in PrET(P)$): Es existiert ein $v \in O_P$, sodass das Wort $wv$ in $P$ einen Fehler-Zustand erreicht. Für die entsprechende Transitionsfolge existieren Zustände $p_1, p_2, \ldots, p_n$ in $P$ und Aktionen $\alpha_1, \alpha_2, \ldots, \alpha_n$, die zusammengesetzt ein Wort $w' = \alpha_1 \alpha_2 \ldots \alpha_n$ bilden, dass ohne die internen Aktionen $wv$ entspricht. Die entsprechende Transitionsfolge in $P$ ist dann $p_0 \overset{\alpha_1}{\dashrightarrow}_P p_1 \overset{\alpha_2}{\dashrightarrow}_P \ldots p_{n-1} \overset{\alpha_n}{\dashrightarrow}_P p_n \in E_P$. Abhängig davon, ob $\alpha_j$ ein Input oder eine lokale Aktion ist, kann mit 1.4 4. bzw. 5. argumentiert werden, dass das jeweilige $\alpha_j$ auch in $Q$ schwach ausführbar ist, solange der entsprechende Zustand $q_{j-1}$ kein Fehler-Zustand ist für $j \in \{1, 2, \ldots n\}$. Falls ein $q_j$ für $j < n$ in $E_Q$ enthalten ist, wurde bis dort ein Präfix von $wv$ ausgeführt. Dieses Präfix ist in $StET_Q$ enthalten. Es gilt also mit $w = \text{prune}(wv)$ und dem Abschluss von $ET$ unter cont $w \in ET_Q$. Ansonsten gibt es in $Q$ einen Trace $q_0 \overset{\widehat{\alpha_1}}{\Longrightarrow}_Q q_1 \overset{\widehat{\alpha_2}}{\Longrightarrow}_Q \ldots q_{n-1} \overset{\widehat{\alpha_n}}{\Longrightarrow}_Q q_n$, wobei $p_j \mathcal{R} q_j$ für alle $0 \leq j \leq n$ gilt. Mit 1.4 1. folgt, dass $q_n \in E_Q$ gelten muss und somit auch $w \in ET_Q$ mit der Begründung von oben.

- Fall 2 ($w \in MIT(P)$): $w$ ist in $P$ ein Input-kritischer Trace. Es existiert also eine Aufteilung von $w$ in $va$ mit $v \in \Sigma^*$ und $a \in I$. Der Trace in $P$ kann wie folgt dargestellt werden: $\exists v' \in \Sigma_\tau^*, \exists p_1, p_2, \ldots, p_n, \exists \alpha_1, \alpha_2, \ldots, \alpha_n : \hat{v'} = v \wedge v' = \alpha_1 \alpha_2 \ldots \alpha_n \wedge p_0 \overset{\alpha_1}{\dashrightarrow}_P p_1 \overset{\alpha_2}{\dashrightarrow}_P \ldots p_{n-1} \overset{\alpha_n}{\dashrightarrow}_P p_n \overset{a}{\not\rightarrow}_P$. Basierend auf 1.4 4. bzw. 5. kann daraus gefolgert werden, dass die $\alpha_j$ auch schwach ausführbar sind in $Q$, falls kein $q_{j-1}$ angetroffen wird, dass in $E_Q$ enthalten ist für $j \in \{1, 2, \ldots n\}$. Falls eines der $q_j$ mit $0 \leq j \leq n$ ein Fehler-Zustand ist, dann ist $w$ in $ET_Q$ enthalten, da ein Präfix von $w$ ein strikter Fehler-Trace in $Q$ ist. Ansonsten gilt $q_0 \overset{\widehat{\alpha_1}}{\Longrightarrow}_Q q_1 \overset{\widehat{\alpha_2}}{\Longrightarrow}_Q \ldots q_{n-1} \overset{\widehat{\alpha_n}}{\Longrightarrow}_Q q_n$ mit $p_j \mathcal{R} q_j$ für alle $j \in \{0, 1, \ldots, n\}$. Da $a$ für $p_n$ in $P$ keine ausgehende must-Transition sein kann, gilt mit 1.4 2. auch $q_n \overset{a}{\not\rightarrow}$. $w$ ist in $MIT_Q \subseteq ET_Q$ enthalten.

Für den zweiten Punkt kann man sich auf die Inklusion $EL_P \backslash ET_P \subseteq EL_Q$ einschränken, da der erste Punkt bereits vorausgesetzt werden kann. $EL_P \backslash ET_P$ ist eine Teilmenge der Sprache $L_P$. Somit ist ein $w$ in $P$ ausführbar. Es gibt also einen Trace $p_0 \overset{\alpha_1}{\dashrightarrow}_P p_1 \overset{\alpha_2}{\dashrightarrow}_P \dots p_{n-1} \overset{\alpha_n}{\dashrightarrow}_P p_n$ in $P$, wobei $w$ $\alpha_1\alpha_2\dots\alpha_n$ entspricht bis auf die internen Aktionen. Erneut können hier 1.4 4. und 5. dazu verwendet werden einen analogen Trace in $Q$ zu finden. Falls ein $q_j$ für $0 \leq j \leq n$ in $E_Q$ enthalten ist, gilt $w \in ET_Q \subseteq EL_Q$. Es wird also im Folgenden davon ausgegangen, dass kein $q_j$ ein Fehler-Zustand ist. Es gilt dann $q_0 \overset{\widehat{\alpha_1}}{\Longrightarrow}_Q q_1 \overset{\widehat{\alpha_2}}{\Longrightarrow}_Q \dots q_{n-1} \overset{\widehat{\alpha_n}}{\Longrightarrow}_Q q_n$ in $Q$ mit $p_j \mathcal{R} q_j$ für $0 \leq j \leq n$ und somit $w \in L_Q \subseteq EL_Q$.

$P \sqsubseteq_{\text{as}} Q \not\Leftarrow P \sqsubseteq_{\text{w-as}} Q$:
Im Abbildung 3.5 wird ein Gegenbeispiel dargestellt mit einem MEIO $Q$ und einer schwachen as-Verfeinerung $P$ von $Q$, die jedoch keine starke as-Verfeinerung von $Q$ ist. Die schwache as-Verfeinerungs-Relationen $\mathcal{R}$ zwischen $P$ und $Q$ enthält die Tupel $(p_0, q_0)$ und $(p_1, q_{12})$. Damit $\mathcal{R}$ eine schwache Simulations-Relation zwischen $P$ und $Q$ sein kann müssen die Startzustände in Relation stehen. Dies ist durch das $(p_0, q_0) \in \mathcal{R}$ erfüllt. Es sind keine Fehler-Zustände in $Q$ und $P$ enthalten, somit ist 1. der Definition 1.4 bereits für beide Zustands-Tupel erfüllt. Für das Tupel $(p_1, q_{12})$ sind auch 2.-5. von 1.4 erfüllt, da weder $p_1$ noch $q_{12}$ ausgehende Transitionen besitzen. Für $(p_0, q_0) \in \mathcal{R}$ gibt es keine ausgehende must-Transitionen. Also ist 2. und 3. von 1.4 bereits erfüllt. Falls $\alpha$ ein Input ist, fordert 1.4 4., dass die Transition $p_0 \overset{\alpha}{\dashrightarrow}_P p_1$ in $Q$ schwach ausführbar ist in der Form $q_0 \overset{\alpha}{\dashrightarrow}_Q \overset{\varepsilon}{\Longrightarrow}_Q q$. Ein entsprechendes $q$ ist in diesem Fall $q_{12}$ und es gilt $p_1 \mathcal{R} q_{12}$. Falls $\alpha$ eine lokale Aktion ist, lautet die Forderung $q_0 \overset{\hat{\alpha}}{\Longrightarrow}_Q q$ für $Q$ und $q_{12}$ ist wieder der passende Zustand für $q$, der mit $p_1$ in Relation stehen. $\mathcal{R}$ ist also eine schwache as-Verfeinerungs-Relation zwischen $P$ und $Q$.
Angenommen es gibt auch eine starke as-Verfeinerungs-Relation $\mathcal{R}'$ zwischen $P$ und $Q$, dann muss $p_0 \mathcal{R}' Q_0$ gelten. Mit 1.3 3. wird gefordert, dass die Transition $p_0 \overset{\alpha}{\dashrightarrow}_P p_1$ durch eine Transition der Form $q_0 \overset{\alpha}{\dashrightarrow}_Q q$ in $Q$ gematched werden muss. Für den Zustand $q$ kommt dieses mal nur $q_{11}$ in Frage. Es muss also $(p_1, q_{11}) \in \mathcal{R}$ gelten. Der zweite Punkt der Definition 1.3 fordert, dass die $\tau$-must-Transition aus $Q$ auch in $P$ auftauchen muss. Es müsste also ein $p$ geben, für dass $p_1 \xrightarrow{\tau}_P p$ gilt und das Tupel $(p, q_{12})$ müsste in $\mathcal{R}$ enthalten sein. Da es keine solche Transition gibt, tritt ein Widerspruch zur Annahme auf. Es kann also keine starke as-Verfeinerungs-Relation zwischen $P$ und $Q$ geben.


Abbildung 3.5: Gegenbeispiel zu $\sqsubseteq_{\text{as}} \not\Leftarrow \sqsubseteq_{\text{w-as}}$

$P \sqsubseteq_{\text{w-as}} Q \not\Leftarrow P \sqsubseteq_E Q$:
Die nicht Gültigkeit dieser Implikation beruht darauf, dass Simulationen strenger sind als Sprach Inklusionen. Das Gegenbeispiel hier ist also so aufgebaut, dass $ET(P) = ET(Q) = \emptyset$ und $L(P) \subseteq L(Q)$ gilt, jedoch keine schwache as-Verfeinerungs-Relation zwischen $P$ und $Q$ existieren kann. $Q$ und $P$ sind in der Abbildung 3.6 dargestellt.

Damit $ET(P) = ET(Q) = \emptyset$ gilt, dürfen keine der Zustände Fehler-Zustände sein und es muss gefordert werden, dass die Menge $I$ der Inputs für die MEIOs leer ist, ansonsten würde es Input-kritische Traces gegen. $P$ kann keine Aktionen ausführen und $Q$ nur die Output Aktion $o$ somit gilt für die Sprachen $\{\varepsilon\} = L(P) \subset L(Q) = \{\varepsilon, o\}$.
Angenommen es gibt eine schwache as-Verfeinerungs-Relation $\mathcal{R}$ zwischen $P$ und $Q$. Dafür muss $(p_0, q_0) \in \mathcal{R}$ gelten. Da es keine Fehler-Zustände in $P$ gibt, ist 1.4 1. erfüllt. Die Punkte 2., 4. und 5. der Definition 1.4 stellen keine Forderungen an die Relation $\mathcal{R}$. Da jedoch die Transition $q_0 \xrightarrow{o}_Q q_1$ in $Q$ vorhanden ist, wird die Verfeinerung dieser in $P$ gefordert es müsste also auch eine must-Output-Transition in $P$ geben. Da diese nicht vorhanden ist, stellt dies einen Widerspruch zur Annahme dar und es folgt, dass es keine schwache as-Verfeinerungs-Relation zwischen $P$ und $Q$ geben kann.


Abbildung 3.6: Gegenbeispiel zu $\sqsubseteq_{\text{w-as}} \Leftarrow \sqsubseteq_E$

□

In dieser Arbeit werden im Gegensatz zu [BFLV16] die Fehler bei einer Parallelkomposition beibehalten bzw. aus dieser entstehend angesehen. Es werden dabei alle Transitionen, die nicht durch fehlende Synchronisations-Möglichkeiten wegfallen, übernommen. In [BFLV16] hingegen wird der Ansatz verfolgt alle Fehler zu entfernen und durch einen universal Zustand zu ersetzten, der nur eingehende may-Input-Transitionen zulässt. Auf diese Normierung wurde hier mit Absicht verzichtete um sehen zu können, dass die Fehler einen Ursprung haben, den man später auch noch einsehen kann. Jedoch gibt es trotzdem einen Zusammenhang zwischen diesen beiden Ansätzen.
Die Unterscheidung, die die Basisrelation $\sqsubseteq_E^{\text{B}}$ hier herbei führt, würde dort der Unterscheidung zwischen Transitionssystemen, die den universal Zustand $e$ als Startzustand haben und denen, die einen Startzustand ungleich $e$ besitzen, entsprechen. Falls hier in einer as-Implementierung von $P$ ein Fehler lokal erreichbar ist, dann muss auch in $P$ ein Fehler-Zustand lokal erreichbar sein, wegen 3.7 (i). Dies entspricht $\varepsilon \in PrET(P)$. Das Abschneiden den lokalen Aktionen wird hier nur in der Trace Menge praktiziert, in [BFLV16] jedoch direkt auf den Transitionssystemen. Im Fall der lokalen Fehler-Erreichbarkeit bleibt also nur noch der Zustand $e$ als universal Zustand übrig, für den jedes Verhalten zulässig ist. Falls man hier diese Pruning mit beliebigem Verhalten nachmachen wollen würde, müsste man an den Zustand $e$ noch eine may-Schleife für alle Aktionen aus $I$ und $O$ hinzufügen.
Da auch der Testing-Ansatz die lokale Fehler-Erreichbarkeit verwendet, existiert der Zusammenhang auch dort für die Parallelkomposition mit dem entsprechenden Test.
Falls es keine Input-kritischen Traces gibt, entspricht $\sqsubseteq_E$ der Relation $\sqsubseteq$ aus [BFLV16], falls auf die MEIOs das entsprechende Abschneiden der Fehler-Traces angewendet würde und diese dann durch einen universal Zustand mit may-Schleife für alle Inputs und

Outputs ersetzt würden. Falle es jedoch in $Q$ Input-kritische Traces gibt, würde $P \sqsubseteq_E Q$ zulassen, dass $P$ einen gekürzten Fehler-Trace anstatt dessen besitzt. Diese Art der Verfeinerung lässt $\sqsubseteq$ nicht zu. Jedoch sind die Input-kritischen Traces auch dort die potentiellen Auslöser für neue Fehler-Zustände in einer Parallelkomposition. Diese Problem basiert auf dem Problem, dass $\sqsubseteq_E$ keine schwache as-Verfeinerungs-Relation sein muss, die in $\sqsubseteq_{\text{w-as}}$ enthalten ist.

# 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. Es wird nur noch der Testing-Ansatz des letzten Kapitels fortgeführt, da dieser sich auf die Parallelkomposition von EIOs stützt und nicht auf die Definition der Parallelkomposition von MEIOs, die auch anders gestaltet hätte werden können.
Zustände, die keine must-Outputs ohne einen Input ausführen können, werden als in einer Art Verklemmung angesehen, da sie ohne Zutun von Außen den Zustand nicht mehr verlassen können, 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 sicher gestellt 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 $\left\{p \in P \mid \forall a \in O : p \overset{a}{\Rightarrow\!\!\!\!\!/}_P\right\}$ 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 via must-Transitionen 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:* $Qui(P) := \left\{p \in P \mid \forall \alpha \in (O \cup \{\tau\}) : p \overset{\alpha}{\not\rightarrow}_P\right\}$.

Für die Erreichbarkeit wird wie im letzten Kapitel ein optimistischer Anzahl der lokalen Erreichbarkeit für die Fehler-Zustände verwendet. Ruhe ist kein unabwendbare „Fehler-Art“, sondern kann durch einen Input repariert werden oder im Fall von vorhandenen may-Output-Transitionen oder may-$\tau$-Transitionen, durch eine Implementierung dieser lokalen Aktionen . Daraus ergibt sich, dass Ruhe im Vergleich zu den Fehlern aus dem letzten Kapitel 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 4.2 (*Test und Verfeinerung für Ruhe*).** *Ein* Test $T$ *ist eine Implementierung. Ein MEIO* $P$ as-erfüllt *einen Ruhe-Test* $T$*, falls* $S \| T$ *fehler- und ruhe-frei ist für alle* $S \in$ as-impl$(P)$*. Es wird dann* $P\, sat_{\text{as}}^{Qui}\, T$ *geschrieben. Die Parallelkomposition* $S \| T$ *ist* fehler- *und* ruhe-frei*, wenn kein Fehler- und kein Ruhe-Zustand lokal erreichbar ist.*
*Ein MEIO* $P$ Ruhe-verfeinert $P'$*, falls für alle Tests* $T$*:* $P'\, sat_{\text{as}}^{Qui}\, T \Rightarrow P\, sat_{\text{as}}^{Qui}\, T$*.*

Um eine genauere Auseinandersetzung mit den Präkongruenzen zu ermöglichen, benötigt man wie im letzten Kapitel die Definition von Traces auf der Struktur. Wie bereits oben erwähnt, ist Ruhe ein reparierbares „Fehlverhalten“ im Gegensatz zu Fehlern. Es genügt deshalb für Ruhe die strikten Traces ohne Kürzung zu betrachten.

**Definition 4.3 (*Ruhe-Traces*).** *Sei* $P$ *ein MEIO und definiere:*

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

**Proposition 4.4 (*Ruhe-Traces und Implementierungen*).** *Für ein MEIO* $P$ *gilt für die strikte Ruhe-Traces:* $StQT(P) \subseteq \left\{ w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p_0' \overset{w}{\Longrightarrow}_{P'} p' \in Qui(P') \right\} = \bigcup_{P' \in \text{as-impl}(P)} StQT(P')$.

*Beweis.* Analog zu den Propositionen 2.1 und 3.4 ist die Inklusion am Besten mit einer as-Implementierung zu zeigen und der entsprechenden as-Verfeinerungs-Relation $\mathcal{R}$. Falls der Startzustand $p_0$ von $P$ ein Ruhe-Zustand ist, muss man zwei as-Implementierungen betrachten, ansonsten genügt es eine für alle $w$ aus $StQT(P)$ anzugeben, wobei $w$ möglicherweise nicht $\varepsilon$ entsprechend darf.
Die as-Implementierung $P'$ für den Fall $p_0 \in Qui(P)$ implementiert alle must-Transitionen, keine may-Transitionen und keine Fehler-Zustände von $P$ und hat die Identitäts-Relation als starke as-Verfeinerungs-Relation $\mathcal{R}$. In diesem Fall gilt $\varepsilon \in StQT(P)$. Für alle $a \in \Sigma$ folgt, wenn in $P$ für einen Zustand $p$ $p \overset{a}{\not\rightarrow}_P$, gilt auch in $P'$ $p' \overset{a}{\not\rightarrow}_{P'}$ für den Zustand, der mit $p$ in Relation steht. Der Startzustand von $P'$ ist mit $\varepsilon$ erreichbar und ebenfalls ruhig. Es gilt also $p_0' \in Qui(P')$ und $\varepsilon \in StQT(P')$. Der 2. Punkt der Definition 1.3 ist für die Identitäts-Relation als as-Verfeinerungs-Relation $\mathcal{R}$ erfüllt, da alle must-Transitionen aus $P$ entsprechend in $P'$ umgesetzt wurden. Alle must-Transitionen in $P$ müssen zugrundeliegende may-Transitionen haben, somit gilt auch 3. von 1.3. Der

1. Punkt der Definition ist auch erfüllt, da $E_{P'} = \emptyset$ gilt, wenn keine Fehler-Zustände implementiert werden.
Für alle $w \neq \varepsilon$ und für $w = \varepsilon$, dass zu einem anderen ruhigen Zustand führt wie $p_0$, mit $w \in StQT(P)$ kann $P'$ als die folgende as-Implementierung gewählt werden:

- $P' = \{q \mid p \in P\} \cup \{q' \mid p \in P\}$,
- $p'_0 = q_0$,
- $I_{P'} = I_P$ und $O_{P'} = O_P$,
- $$\begin{aligned} \longrightarrow_{P'} = \dashrightarrow_{P'} = &\left\{(q_0, \alpha, q_j) \mid p_0 \overset{\alpha}{\dashrightarrow} p_j\right\} \\ &\cup \left\{(q_0, \alpha, q'_j) \mid p_0 \overset{\alpha}{\dashrightarrow} p_j\right\} \\ &\cup \left\{(q_j, \alpha, q_k) \mid p_j \overset{\alpha}{\longrightarrow} p_k\right\} \\ &\cup \left\{(q'_j, \alpha, q'_k) \mid p_j \overset{\alpha}{\longrightarrow} p_k\right\} \\ &\cup \left\{(q'_j, \alpha, q_k) \mid j \neq 0, p_j \overset{\alpha}{\dashrightarrow} p_k, p_j \overset{\alpha}{\not\longrightarrow} p_k\right\} \\ &\cup \left\{(q'_j, \alpha, q'_k) \mid j \neq 0, p_j \overset{\alpha}{\dashrightarrow} p_k, p_j \overset{\alpha}{\not\longrightarrow} p_k\right\}, \end{aligned}$$
- $E_{P'} = \emptyset$.

Als as-Verfeinerungs-Relation zwischen $P$ und $P'$ wird die Relation $\mathcal{R} = \{(q_j, p_j) \mid p_j \in P\} \cup \{(q'_j, p_j) \mid p_j \in P\}$ verwendet. Es werden in $P'$ für die ungestrichenen Zustände $q$ nur die must-Transitionen und für die gestrichenen Zustände $q'$ werden die must- und may-Transitionen implementiert. Die must-Transitionen werden nur zu den Zuständen der „gleichen Sorte“ umgesetzt, wohingegen die may-Transitionen, zu denen es keine entsprechende must-Transition in $P$ gibt, von den Zuständen $q'$ zu dem entsprechenden ungestrichenen und gestrichenen Zustand implementiert wird. Da die Menge der Fehler-Zustände leer ist, gilt 1.3 1. für $\mathcal{R}$. Die must-Transitionen werden für die ungestrichenen und gestrichenen Zustände umgesetzt, dies erfüllt zusammen mit $\mathcal{R}$ die Definition 1.3 2. Ebenso wird der dritte Punkt der Definition 1.3 erfüllt, da sowohl die gestrichenen wie auch die ungestrichenen Zustände mit den entsprechenden Zuständen aus $P$ in der Relation $\mathcal{R}$ stehen. $\mathcal{R}$ ist also eine starke alternierende Simulations-Relation auf $P'$ und $P$. Falls ein Zustand $p_j$ in $P$ ruhig war, ist es auch der entsprechenden Zustand $q_j$ in $P'$, da für $q_j$ alle ausgehenden must-Transitionen von $p_j$ implementiert wurden, aber keine einzige may-Transition, die keine der must-Transitionen entspricht. Wenn also für $p_j$ keine Outputs und kein $\tau$ möglich waren via must-Transitionen, dann ist es dies auch für $q_j$ nicht. $q_j$ ist in $P'$ mit den selben Traces erreichbar wie $p_j$ in $P$, da jeder ungestrichene und gestrichene Zustand in $P'$ die selben eingehenden Transitionen hat wie der entsprechende Zustand in $P$. Falls der Trace zu $p_j$ may-Transitionen ohne entsprechende must-Transitionen enthält, kann der Trace in $P'$ ausgeführt werden, in dem von $q_0$ aus der Trace über die gestrichenen Zustände genommen wird bis zur letzten Transition, die zu einem ungestrichenen Zuständen führt in dem auszuführenden Wort. Ab da hat der Trace in $P$ nur must-Transitionen genommen und kann somit in den ungestrichenen Zuständen in $P'$ nachgefolgt werden. Falls der Trace in $P$ insgesamt nur

aus must-Transitionen bestanden hat, ist direkt von $q_0$ aus der Weg über ungestrichene Zustände zu $q_j$ möglich. Es gilt also $StQT(P)\backslash\{\varepsilon\} = StQT(P')\backslash\{\varepsilon\}$. Falls $\varepsilon$ zu einem Ruhe-Zustand $p \neq p_0$ in $P$ geführt hat, gilt sogar $StQT(P) = StQT(P')$, da ein Trace aus internen Aktionen in $P$ und $P'$ zu dem entsprechenden ruhigen Zustand $p$ bzw. $q$ führt. □

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

**Definition 4.5 (*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.*

**Proposition 4.6 (*Ruhe-Semanik und Implementierungen*).** *Für die Menge der fehler-gefluteten Ruhe-Traces von $P$ gilt $QET(P) = \bigcup_{P' \in \text{as-impl}(P)} QET(P')$.*

*Beweis.*
$\subseteq$:

$$\begin{aligned}
QET(P) &\overset{4.5}{=} StQT(P) \cup ET(P) \\
&\overset{4.4}{\subseteq} \left( \bigcup_{P' \in \text{as-impl}(P)} StET(P') \right) \cup ET(P) \\
&\overset{3.6}{=} \left( \bigcup_{P' \in \text{as-impl}(P)} StET(P') \right) \cup \left( \bigcup_{P' \in \text{as-impl}(P)} ET(P') \right) \\
&= \bigcup_{P' \in \text{as-impl}(P)} StQT(P') \cup ET(P') \\
&\overset{4.5}{=} \bigcup_{P' \in \text{as-impl}(P)} QET(P').
\end{aligned}$$

$\supseteq$:
Es wird hier für ein $w \in QET(P')$ einer beliebigen as-Implementierung $P'$ von $P$ gezeigt, dass das Wort $w$ auch in $QET(P)$ enthalten ist. Es kann danach unterschieden werden, ob $w$ aus $StQT(P')\backslash ET(P')$ stammt oder aus $ET(P')$. Falls $w \in ET(P')$ gilt, folgt mit Proposition 3.6 bereits, dass $w \in ET(P) \subseteq QET(P)$ gilt. Somit wird für den Rest des Beweises davon ausgegangen, dass $w \in StQT(P')\backslash ET(P')$ ist. $w$ führt in $P'$ also nur zu einem ruhigen Zustand und hat nichts mit Fehler-Zuständen in $P'$ zu tun. Der Trace, der durch $w$ in $P'$ beschrieben hat, hat die folgende Form:

$\exists w' \in \Sigma_\tau, \exists \alpha_1, \alpha_2, \ldots, \alpha_n, \exists p'_1, p'_2, \ldots p'_n : \hat{w'} = w \wedge w' = \alpha_1\alpha_2 \ldots \alpha_n \wedge p'_0 \xrightarrow{\alpha_1}_{P'} p'_1 \xrightarrow{\alpha_2}_{P'} \ldots p'_{n-1} \xrightarrow{\alpha_n}_{P'} p'_n \in Qui_{P'}$. Da es eine as-Verfeinerungs-Relation $\mathcal{R}$ geben muss, die beweist, dass $P'$ $P$ as-verfeinert, muss es ein Präfix von $w$ geben, dass auch in $P$ ausführbar ist. Falls $w$ nicht vollständig ausführbar ist in $P$, muss auf dem Weg, auf dem das Präfix von $w$ ausgeführt wird ein Zustand $p_j$ mit $0 \leq j \leq n$ liegen, der ein Fehler-Zustand ist. Es gilt dann $w \in ET(P) \subseteq QET(P)$. Falls jedoch $w$ in $P$ ausführbar ist ohne einen Fehler-Zustand zu erreichen, gibt es einen analogen Trace zu dem in $P'$, der Form: $p_0 \xrightarrow{\alpha_1}_P p_1 \xrightarrow{\alpha_2}_P \ldots p_{n-1} \xrightarrow{\alpha_n}_P p_n$, wobei $p'_j \mathcal{R} p_j$ für alle $j$ aus $\{0, 1, \ldots, n\}$ gilt. Es wird also durch $w$ ein Zustand $p_n$ erreicht, der mit dem Zustand $p'_n$ in der starken as-Verfeinerungs-Relation $\mathcal{R}$ stehen. $p'_n$ ist ruhig, nach Voraussetzung. Es gilt also für alle $\omega \in O \cup \{\tau\}$ $p'_n \not\xrightarrow{\omega}$. Da $(p'_n, p_n) \in \mathcal{R}$ gilt und beide Zustände keine Fehler-Zustände sind, muss auch $p_n \not\xrightarrow{\omega}$ für alle $\omega \in O \cup \{\tau\}$ gelten, da sonst 1.3 2. verletzt würde. Es gilt also in diesem Fall $w \in StQT(P) \subseteq QET(P)$. □

Wie im letzten Kapitel kann aus der vorangegangen Proposition über die Gleichheit der betrachteten Trace Mengen in der Relation $\sqsubseteq_{Qui}$ auch eine Aussage über die lokale Erreichbarkeit „fehlerhafter Zustände“ in einer Spezifikation und den zugehörigen Implementierungen getroffen werden.

**Korollar 4.7 (*lokale Ruhe Erreichbarkeit*).**

*(i) Falls in einem MEIO $P$ ein Fehler lokal erreichbar ist, dann existiert auch eine as-Implementierung, in der ein Fehler lokal erreichbar ist.*

*(ii) Falls in einem MEIO $P$ einen lokal erreichbaren Ruhe-Zustand besitzt, jedoch keinen lokal erreichbaren Fehler, dann existiert auch eine as-Implementierung, in der ein Ruhe-Zustand und kein Fehler lokal erreichbar ist.*

*(iii) Falls es eine as-Implementierung von $P$ gibt, die einen Fehler oder Ruhe lokal erreicht, dann ist auch ein Fehler oder Ruhe in $P$ lokal erreichbar.*

*Beweis.*

(i) Dieser Punkt folgt direkt aus Proposition 3.6.

(ii) Da ein Ruhe-Zustand in $P$ lokal erreichbar ist, gilt $w \in StQT_P \backslash \text{cont}(PrET)_P \subseteq QET_P$ für $w \in O$. Es muss wegen Proposition 4.6 mindestens ein $P' \in \text{as-impl}(P)$ geben, für dass $w \in QET_{P'}$ gilt. Das $w$ kann also in $ET_{P'}$ oder in $StQT_{P'}$ enthalten sein. Da $w$ nur aus lokalen Aktion bestehen kann, kann $w$ in $P$ kein Input-kritisch Trace sein und auch keine Verlängerung von einem solchen. Es gilt also $w \notin ET_P$. Mit 3.6 folgt daraus, dass auch in keiner as-Implementierung von $P$ $w$ in der Menge $ET$ enthalten ist. Es gilt also $w \in QET_{P'} \backslash ET_{P'} \subseteq StQT_{P'}$. Es ist also auch in $P'$ ein Ruhe-Zustand lokal erreichbar.

(iii) Sei $P'$ die as-Implementierung von $P$, in der ein Fehler- oder Ruhe-Zustand lokal erreichbar ist. Es gilt dann $w \in QET'_P$ für $w \in O$. Mit Proposition 4.6 folgt draus

$w \in QET_P$. Es muss also auch in $P$ ein Fehler- oder Ruhe-Zustand lokal erreichbar sein.

□

Für spätere Beweise werden noch Zusammenhänge zwischen Ruhe-Zuständen in den einzelnen Komponenten und in einer Parallelkomposition dieses Komponenten benötigt.

**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 $\Big((p_1, p_2) \xrightarrow{\tau}_{12}\Big)$: 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 als must-Transition in einer 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 $\Big((p_1, p_2) \xrightarrow{a}_{12}$ mit $a \in O_{12} \backslash \text{Synch}(P_1, P_2)\Big)$: 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 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 sein.
   - Fall 3 $\Big((p_1, p_2) \xrightarrow{a}_{12}$ mit $a \in O_{12} \cap \text{Synch}(P_1, P_2)\Big)$: 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 $p_j \in Qui_j$ für $j \in \{1, 2\}$ $(p_1, p_2) \in Qui_{12}$ folgt.

2. Es gilt $(p_1, p_2) \in Qui_{12} \backslash E_{12}$, somit hat dieser Zustand allenfalls die Möglichkeit für must-Transitionen, die mit Inputs beschriftet sind.
Angenommen $p_1 \notin Qui_1$, dann ist für $p_1$ entweder eine $\tau$-must-Transition oder eine Output-must-Transition möglich.

   - Fall 1 $\left(p_1 \xrightarrow{\tau}_1\right)$: 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 $\left(p_1 \xrightarrow{a}_1 \text{ mit } a \in O_1 \backslash \text{Synch}(P_1, P_2)\right)$: 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 $\left(p_1 \xrightarrow{a}_1 \text{ mit } a \in O_1 \cap \text{Synch}(P_1, P_2) \text{ und } p_2 \xrightarrow{a}_2\right)$: 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 $\left(p_1 \xrightarrow{a}_1 \text{ mit } a \in O_1 \cap \text{Synch}(P_1, P_2) \text{ und } p_2 \not\xrightarrow{a}_2\right)$: Da $P_2$ die $a$ Transition nicht als must-Transition enthält, handelt es sich hier um einen neuen Fehler. Das $a$ kann für $P_2$ kein Output sein, da sonst $P_1$ und $P_2$ nicht komponierbar wäre. Der neue Fehler 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 3.8.

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

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

*Beweis.* Es wird nur der 2. Punkt beweisen.
„$\subseteq$“:
Hier muss unterschieden werden, ob ein $w \in StQT_{12} \backslash ET_{12}$ oder ein $w \in ET_{12}$ betrachtet wird. Im zweiten Fall ist das $w$ offensichtlich in der rechten Seite enthalten. Somit wird im Folgenden ein $w \in StQT_{12} \backslash ET_{12}$ betrachtet und es wird versucht dessen Zugehörigkeit zur rechten Menge zu zeigen. Aufgrund von Definition 4.3 weiß man, dass $(p_{01}, p_{02}) \overset{w}{\Longrightarrow}_{12} (p_1, p_2)$ gilt mit $(p_1, p_2) \in Qui_{12} \backslash E_{12}$. Durch Projektion erhält man $p_{01} \overset{w_1}{\Longrightarrow}_1 p_1$ und $p_{02} \overset{w_2}{\Longrightarrow}_2 p_2$ mit $w \in w_1 \| w_2$. Aus $(p_1, p_2) \in Qui_{12} \backslash E_{12}$ kann mit dem zweiten Punkt von Lemma 4.8 gefolgert werden, dass $q_1 \in Qui_1$ 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 $w \in QET_1 \| QET_2$ und somit ist $w$ in der rechten Seite der Gleichung enthalten.

„$\supseteq$“:
Es muss wieder danach unterschieden werden aus welcher Menge das betrachtete Element stammt. Falls $w \in ET_{12}$ gilt, so kann die Zugehörigkeit zur linken Seite direkt gefolgert werden. Somit wird für den weiteren Beweis dieser Inklusionsrichtung ein Element $w \in QET_1 \| QET_2$ betrachtet und gezeigt, dass es in der linken Menge enthalten ist. Da $QET_i = StQT_i \cup ET_i$ gilt, existieren für $w_1$ und $w_2$ mit $w \in w_1 \| w_2$ unterschiedliche Möglichkeiten:

- Fall 1 ($w_1 \in ET_1 \vee w_2 \in ET_2$): OBdA gilt $w_1 \in ET_1$. Nun kann $w_2 \in StQT_2 \subseteq L_2$ oder $w_2 \in ET_2$ gelten und somit ist auf jeden Fall $w_2$ in $EL_2$ enthalten. Daraus kann dann mit dem ersten Punkt von Satz 3.8 gefolgert werden, dass $w \in ET_{12}$ gilt und damit ist $w$ in der linken Seite der Gleichung enthalten.
- Fall 2 ($w_1 \in StQT_1 \backslash ET_1 \wedge w_2 \in StQT_2 \backslash ET_2$): Es gilt in diesem Fall $p_{01} \overset{w_1}{\Longrightarrow}_1 p_1 \in Qui_1$ und $p_{02} \overset{w_2}{\Longrightarrow}_2 p_2 \in Qui_2$. 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 $(p_{01}, p_{02}) \overset{w}{\Longrightarrow}_{12} (p_1, p_2) \in Qui_{12}$ 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 $\cdot \| \cdot$.*

*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 3.9, aus der Monotonie von $\cdot\|\cdot$ auf Sprachen wie folgt:

- $P_{31} \overset{\substack{\text{Korollar 3.9}\\ \text{und}\\ P_1 \sqsubseteq_E P_2}}{\sqsubseteq_E} P_{32}$,
- $QET_{31} \overset{4.9\ 2.}{=} (QET_3 \| QET_1) \cup ET_{31}$
  $\overset{\substack{ET_{31} \subseteq ET_{32}\\ \text{und}\\ QET_1 \subseteq QET_2}}{\subseteq} (QET_3 \| QET_2) \cup ET_{32}$
  $\overset{4.9\ 2.}{=} QET_{32}$. □

Im nächsten Lemma soll eine Verfeinerung bezüglich guter Kommunikation mit Partnern betrachtet werden. Die gute Kommunikation stützt sich dabei auf die Definition von Tests und der daraus resultierenden Verfeinerung in 4.2.

**Lemma 4.11 (*Testing-Verfeinerung mit Ruhe*).** *Gegeben sind zwei MEIOs $P_1$ und $P_2$ mit der gleichen Signatur. Wenn für alle Tests $T$, die Partner von $P_1$ bzw. $P_2$ sind, $P_2\, sat_{\text{as}}^{Qui}\, T \Rightarrow P_1\, sat_{\text{as}}^{Qui}\, T$ gilt, dann folgt daraus die Gültigkeit von $P_1 \sqsubseteq_{Qui} P_2$.*

*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 Test Partner $T$ gilt $I_T = O$ und $O_T = I$.
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 3.14 wurde bereits etwas Ähnliches gezeigt, jedoch wurde dort als Voraussetzung $P_2\, sat_{\text{as}}^{E}\, T \Rightarrow P_1\, sat_{\text{as}}^{E}\, T$ für alle Test Partner $T$ verwendet und hier dieselbe Aussage mit der Test Erfüllung für Ruhe. Die hier verwendeten Tests sagen nichts darüber aus, welche Art von „fehlerhaftem Zustand“ enthalten ist. Die Aussage des Lemmas 3.14 kann hier also nicht verwendet werden. Aus der lokalen Erreichbarkeit eines Fehler-Zustandes in der Parallelkomposition einer as-Implenetierung von von $P_1$ mit $T$ lässt sich nur schließen, dass $P_2$ den Test $T$ ebenfalls nicht as-erfüllt. Dies kann aber aufgrund einer as-Implementierung von $P_2$ sein, die in Parallelkomposition mit $T$ einen Fehler- oder Ruhe-Zustand lokal erreicht. Analoges gilt auch für die lokale Erreichbarkeit eines Ruhe-Zustandes in der Komposition einer as-Implementierung von $P_1$ mit einem Test $T$. 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 $P_2\, sat_{\text{as}}^{Qui}\, T \Rightarrow P_1\, sat_{\text{as}}^{Qui}\, T$ gezeigt, dass $ET_1 \subseteq ET_2$ gilt. Da beide $ET$-Mengen unter cont abgeschlossen sind, reicht es ein präfix-minimales Element $w \in ET_1$ zu betrachten und zu zeigen, dass dieses $w$ oder eines seiner Präfixe in $ET_2$ enthalten ist. $w$ muss, wegen Proposition 3.6, in einer as-Implementierung $P_1'$ von $P_1$ ebenfalls ein präfix-minimales Element in $ET_{P_1'}$ sein.

- Fall 1 ($w = \varepsilon$): Es handelt sich um einen lokal erreichbaren Fehler-Zustand in $P_1'$. Für $T$ 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 lokal erreichen wie $P_1' \| T$. Es gibt also einen lokal erreichbaren Zustand von $P_1' \| T$, der in $E_{P_1' \| T}$ enthalten ist. Somit erfüllt $P_1$ nicht alle Tests $T$ und es muss somit auch mindestens eine as-Implementierung $P_2'$ von $P_2$ geben, die den Test $T$ ebenfalls nicht erfüllt. Da eine Implementierung den Test $T$ erfüllt, wenn die Parallelkomposition der Implementierung mit dem Test fehler- und ruhe-frei ist, kann die nicht Erfüllung eines Testes sowohl an einem Fehler- wie auch einem Ruhe-Zustand liegen. Bei dem lokal erreichbaren „fehlerhaften Zustand“ kann es sich nur um einen Fehler handeln, da es in der Komposition mit $T$ keine Ruhe-Zustände geben kann. Da $T$ keinen Fehler-Zustand und auch keine fehlenden Input-Möglichkeiten enthält, kann der Fehler nur von $P_2'$ geerbt sein. Somit muss in $P_2'$ ein Fehler-Zustand lokal erreichbar sein. Es gilt also $\varepsilon \in PrET_{P_2'} \subseteq ET_{P_2'}$ und mit Proposition 3.6 auch $\varepsilon \in ET_2$.

- 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 $T$ betrachtet (siehe auch Abbildung 4.1):

  - $T = \{p_0, p_1, \dots, p_{n+1}\}$,
  - $p_{0T} = p_0$,
  - $\dashrightarrow_T = \longrightarrow_T = \{(p_j, x_{j+1}, p_{j+1}) \mid 0 \leq j \leq n\}$
    $\cup \{(p_j, x, p_{n+1}) \mid x \in I_T \backslash \{x_{j+1}\}, 0 \leq j \leq n\}$
    $\cup \{(p_{n+1}, x, p_{n+1}) \mid x \in I_T\}$
    $\cup \{(p_j, \tau, p_j) \mid 0 \leq j \leq n+1\}$,
  - $E_T = \emptyset$.

  Die Menge der Ruhe-Zustände des hier betrachteten $T$s ist leer. Da im Vergleich zum Transitionssystem in Abbildung 3.1 nur die $\tau$-Schlingen ergänzt wurden und die Umbennennung der Mengen, ändert sich nichts an den Fällen 2a) und 2b) aus dem Beweis der selben Inklusion von Lemma 3.14. Die Begründungen, wieso in den beiden Fällen $\varepsilon \in PrET(P_1' \| T)$ gilt, bleibt also analog zum Beweis des ersten Punktes des Lemmas aus dem vorangegangnen Kapitel. Durch die must-$\tau$-Schlingen wurde, genau wie im letzten Fall nur erreicht, das in einer Parallelkomposition mit $T$ keine Ruhe-Zustände möglich sind. Es kann also auch hier aus

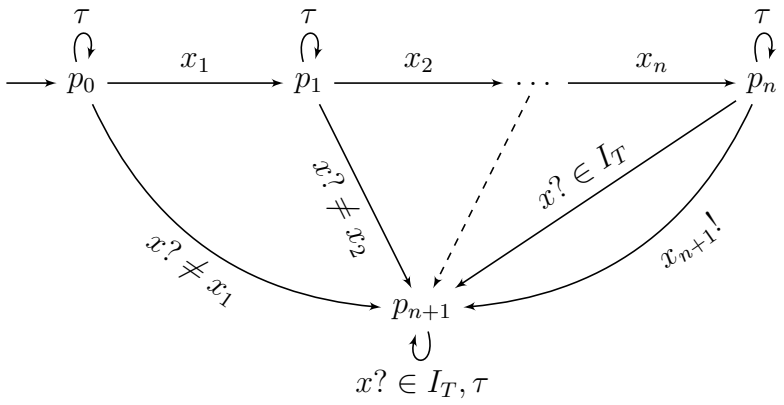


Abbildung 4.1: $x? \neq x_j$ steht für alle $x \in I_T \backslash \{x_j\}$

der lokalen Erreichbarkeit eines Fehler in $P_1' \| T$ auf die lokale Erreichbarkeit eines Fehler-Zustandes in $P_2' \| T$ für eine as-Implementierung $P_2'$ von $P_2$ geschlossen werden. Die weitere Argumentation verläuft analog zu Fall 2, derselben Inklusion im Beweis von Lemma 3.14. 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 3.14 ist hier jedoch aufgrund des bereits geführten Beweisteils nur noch $L_1 \backslash ET_1 \subseteq EL_2$ zu zeigen. Es wird also für ein beliebig gewähltes $w \in L_1 \backslash ET_1$ gezeigt, dass dieses auch in $EL_2$ enthalten ist. Aufgrund der Propositionen 2.1 und 3.6 gibt es auch eine as-Implementierung $P_1'$ von $P_1$ für die $w \in L_{P_1'} \backslash ET_{P_1'}$ gilt.

- Fall 1 ($w = \varepsilon$): Ebenso wie in 3.14 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 $T$ weicht wie im letzten Beweisteil nur durch die $\tau$-must-Schleifen an den Zuständen des Transitionssystems vom Beweis des zweiten Punktes aus Lemma 3.14 ab. Somit ist der Partner $T$ dann wie folgt definiert (siehe dazu auch Abbildung 4.2):
  - $T = \{p_0, p_1, \dots, p_n, p\}$,
  - $p_{0T} = p_0$,
  - $\dashrightarrow_T = \longrightarrow_T = \{(p_j, x_{j+1}, p_{j+1}) \mid 0 \leq j < n\}$
    $\cup \{(p_j, x, p) \mid x \in I_T \backslash \{x_{j+1}\}, 0 \leq j < n\}$
    $\cup \{(p_j, \tau, p_j) \mid 0 \leq j \leq n\}$
    $\cup \{(p, \alpha, p) \mid \alpha \in I_T \cup \{\tau\}\}$,

– $E_T = \{p_n\}$.

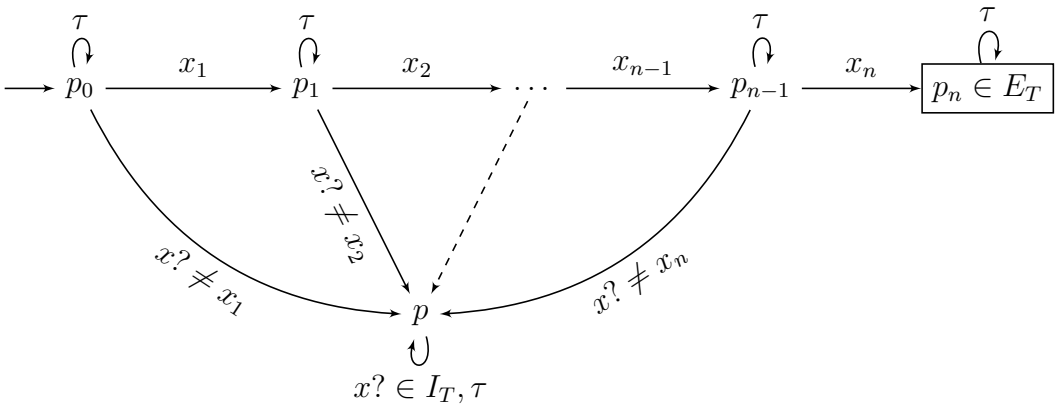


Abbildung 4.2: $x? \neq x_j$ steht für alle $x \in I_T \backslash \{x_j\}$, $p_n$ ist der einzige Fehler-Zustand

Da durch die $\tau$-must-Schlingen an den Zuständen wie oben vermieden wird, dass es in einer Komposition mit $T$ und auch in $T$ selbst Ruhe-Zustände gibt, verläuft der Rest des Beweises dieses Punktes analog zum Beweis der selben Inklusionsrichtung von Lemma 3.14. Und somit gilt für alle Fälle (2a) bis 2d)), dass $w$ in $EL_2$ enthalten ist.

So bleibt nur noch der letzt Beweispunkt zu zeigen, d.h. die Inklusion $QET_1 \subseteq QET_2$. Die Inklusion kann jedoch, analog zum Beweis der Inklusion der Fehler-gefluteten Sprache, noch weiter eingeschränkt werden. Da bereits bekannt ist, dass $ET_1 \subseteq ET_2$ gilt, muss nur noch $StQT_1 \backslash ET_1 \subseteq QET_2$ gezeigt werden.
Es wird ein $w \in StQT_1 \backslash ET_1$ gewählt und gezeigt, dass dieses auch in $QET_2$ enthalten ist. Mit den Propositionen 3.6 und 4.4 kann gefolgert werden, dass es auch eine as-Implementierung $P_1'$ von $P_1$ gibt, für die $w \in StQT_{P_1'} \backslash ET_{P_1'}$ gilt.
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 $P_1' \| T$, wenn in $T$ 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 $T$ betrachtet (siehe auch Abbildung 4.3):

- $T = \{p_0, p_1, \dots, p_n, p\}$,
- $p_{0T} = p_0$,
- $\dashrightarrow_T = \longrightarrow_T = \{(p_j, x_{j+1}, p_{j+1}) \mid 0 \leq j < n\}$
  $\cup \{(p_j, x, p) \mid x \in I_T \backslash \{x_{j+1}\}, 0 \leq j < n\}$
  $\cup \{(p_j, \tau, p_j) \mid 0 \leq j < n\}$
  $\cup \{(p_n, x, p) \mid x \in I_T\}$
  $\cup \{(p, \alpha, p) \mid \alpha \in I_T \cup \{\tau\}\}$,
- $E_T = \emptyset$.

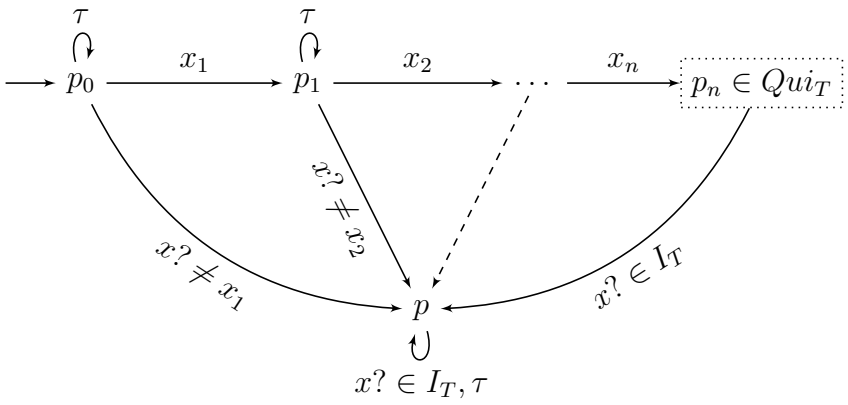


Abbildung 4.3: $x? \neq x_j$ steht für alle $x \in I_T \backslash \{x_j\}$, $p_n$ ist der einzige Ruhe-Zustand

Falls für das betrachtete $w = \varepsilon$ gilt, reduziert sich der Partner $T$ 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 $T$ der einzig ruhige Zustand in $T$. Es gilt wegen des ersten Punktes von Lemma 4.8, dass auch in der Parallelkomposition $P_1' \| T$ ein Ruhe-Zustand mit $w$ erreicht wird. Bei allen in $w$ befindlichen Aktionen handelt es sich um synchronisierte Aktionen und es gilt $I_T \cap I = \emptyset$. Daraus folgt $w \in O_{P_1' \| T}$ und $w \in StQT(P_1' \| T)$. Es kann also in der Parallelkomposition durch $w$ ein Ruhe-Zustand lokal erreicht werden. Da $w \notin ET_{P_1'}$ gilt, kann auf dem Weg, der mit $w$ im Transitionssystem $P_1'$ zurückgelegt wird, kein Fehler lokal erreicht werden. Es kann also weder von $P_1'$ noch von $T$ ein Fehler auf diesem Weg geerbt werden und durch den Aufbau von $T$ kann auch kein neuer Fehler in der Parallelkomposition der beiden Systeme entstehen. Da ein Ruhe-Zustand in $P_1' \| T$ lokal erreichbar ist, muss auch in $P_2' \| T$ für eine as-Implementierung $P_2'$ von $P_2$ ein „fehlerhafter Zustand“ lokal erreichbar sein. Es kann zunächst keine Aussage getroffen werden, ob das $w$ in $P_2' \| T$ ausführbar ist und ob es sich bei dem „fehlerhaften Zustand“ um Ruhe oder einen Fehler handelt.

- Fall a) ($\varepsilon \in ET(P_2' \| T)$): Es handelt sich bei dem lokal erreichbaren „fehlerhaften Zustand“ um einen Fehler. 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 Fehler in der Parallelkomposition entstanden sein. Es gilt, dass bereits in $P_2'$ ein Präfix von $w$ in $ET_{P_2'}$ enthalten ist, wegen des Beweises des ersten Punktes aus Lemma 3.14 und da $T$ nur neue Fehler auf dem Trace $w$ zulässt. Die Menge $ET$ ist unter cont abgeschlossen, somit gilt $w \in ET_{P_1'} \subseteq QET_{P_2'}$. Mit Proposition 4.6 folgt daraus $w \in QET_2$.

- Fall b) (Ruhe-Zustand lokal erreichbar in $P_2' \| T$ und $\varepsilon \notin ET(P_2' \| T)$): Da in $T$ nur durch $w$ ein ruhiger Zustand erreicht werden kann, muss es sich bei dem lokal erreichbaren Ruhe-Zustand in $P_2' \| T$ 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 $w \in StQT_{P_2'} \subseteq QET_{P_2'} \subseteq QET_2$, aufgrund von Proposition 4.6.

□

**Satz 4.12.** *Falls $P_1 \sqsubseteq_{Qui} P_2$ gilt folgt draus auch, dass $P_1$ $P_2$ Ruhe-verfeinert.*

*Beweis.* Nach Definition gilt $w \in QET(P)$ mit $w \in O(P)^*$ genau dann, wenn in $P$ ein Ruhe-Zustand oder ein Fehler-Zustand 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- oder Fehler-Zustand nur dann in $P_1$ lokal erreichbar, wenn auch ein Ruhe- oder Fehler-Zustand in $P_2$ lokal erreichbar ist. Daraus folgt, dass es as-Implementierungen $P_1'$ und $P_2'$ von $P_1$ bzw. $P_2$ gibt, die analoge Fehler lokal erreichen wegen 4.7. Es gilt dann auch, dass $P_j' \| T$ einen lokal erreichbaren Fehler oder lokal erreichbare Ruhe hat, wenn $P_j'$ dies hat, für $j \in \{1, 2\}$ und einen Test $T$. Falls $P_1$ einen „fehlerhaften Zustand“ lokal erreicht, dann zeigt sich auch in einer as-Implementierung von $P_1$ dieser durch einen Fehler-Zustand oder einen Ruhe-Zustand. In der Parallelkomposition der as-Implementierung mit einem Test $T$ tritt der „fehlerhafte Zustand“ auf. Falls es sich um einen Fehler handelt, dann tritt mit allen Tests $T$ ein Fehler in der Parallelkomposition auf. Bei Ruhe hingegen zeigt sich das „Fehlverhalten“ nur, mit Tests $T$, die einen analogen Ruhe-Zustand enthalten. Mit der Relation $\sqsubseteq_{Qui}$ gibt es auch in $P_2$ einen lokal erreichbaren „fehlerhaften Zustand“, wenn es in $P_1$ einen solchen gibt. Es tritt dann auch in einer as-Implementierung Ruhe oder ein Fehler auf, dies zeigt sich dann auch in der Parallelkomposition mit Tests $T$. Im Fall von Fehlern, zeigt sich sowohl in der Parallelkomposition einer as-Implementierung von $P_1$ wie auch einer as-Implementierung von $P_2$ mit beliebigen Tests $T$, der Fehler, da $\sqsubseteq_E$ gilt und somit die Argumentationen aus Satz 3.15 anwendbar sind. Ruhe tritt in $P_2$ nur ohne einen Fehler auf, wenn in $P_1$ auch nur Ruhe und kein Fehler auf dem Trace vorhanden war. Ansonsten hätte $\sqsubseteq_E$ auch einen Fehler in $P_2$ gefordert. In der Parallelkomposition einer as-Implementierung von $P_1$ mit $T$ hat sich die Ruhe bereits gezeigt, also tut sie dies auch in der Parallelkomposition einer as-Implementierung von $P_2$ mit $T$. Es gilt also $\neg P_1\, sat^E_{\text{as}}\, T \Rightarrow \neg P_2\, sat^E_{\text{as}}\, T$ für alle Tests $T$. Daraus ergibt sich für alle Tests $T$ die Implikation $P_2\, sat^E_{\text{as}}\, T \Rightarrow P_1\, sat^E_{\text{as}}\, T$ und somit Ruhe-verfeinert $P_1$ $P_2$. □

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

**Satz 4.13 (*Zusammenhang der Verfeineruns-Relationenen mit der Ruhe-Relation*).** *Für MEIOs $P$ und $Q$ gilt $P \sqsubseteq_{\text{w-as}} Q \Rightarrow P \sqsubseteq_{Qui} Q \Rightarrow P \sqsubseteq_E Q$. Die Implikationen in die andere Richtung gelten jedoch nicht.*

*Beweis.*
$P \sqsubseteq_{\text{w-as}} Q \Rightarrow P \sqsubseteq_{Qui} Q$:
Im Beweis des Satzes 3.16 wurde bereits bewiesen, dass eine schwache as-Verfeinerungs-Relation $\mathcal{R}$ zwischen $P$ und $Q$ die Eigenschaften der Fehler-Relation $\sqsubseteq_E$ erfüllt. Es fehlt für diese Implikation also nur noch der Beweis der Inklusion $QET_P \subseteq QET_Q$. Da bereits

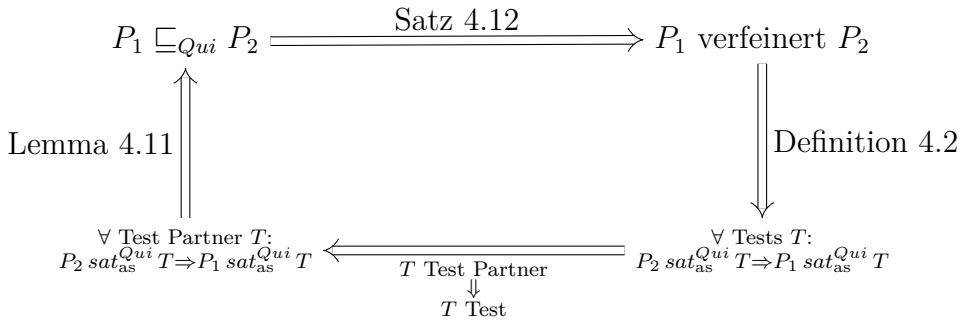


Abbildung 4.4: Folgerungskette der Testing-Verfeinerung und Ruhe-Relation

$ET_P \subseteq ET_Q$ beweisen wurde, reicht es aus zu beweißen, dass $StQT_P \backslash ET_P \subseteq QET_Q$ gilt. Für ein Wort $w$ aus $StQT_P \backslash ET_P$ gilt: $\exists w' \in \Sigma_\tau^*, \exists p_1, p_2, \ldots, p_n, \exists \alpha_1, \alpha_2, \ldots, \alpha_n : \hat{w'} = w \wedge w' = \alpha_1\alpha_2 \ldots \alpha_n \wedge p_0 \xrightarrow{\alpha_1}_P p_1 \xrightarrow{\alpha_2}_P \ldots p_{n-1} \xrightarrow{\alpha_n}_P p_n \in Qui_P$. Aufgrund von 1.4 4. bzw. 5. gibt es einen analogen Trace in $Q$, falls ein $q_j$ für $0 \leq j \leq n$ in $E_Q$ angetroffen wird. Es gilt $w \in StET_Q \subset ET_Q \subset QET_Q$, falls ein Fehler-Zustand in $Q$ auf einem Präfix-Trace von $w$ auftritt. Im folgenden wird davon ausgegangen, das $w$ in $Q$ ohne das erreichben eines Fehler-Zustandes ausführbar ist. Es gibt also einen Trace der Form $q_0 \stackrel{\widehat{\alpha_1}}{\Longrightarrow}_Q q_1 \stackrel{\widehat{\alpha_2}}{\Longrightarrow}_Q \ldots q_{n-1} \stackrel{\widehat{\alpha_n}}{\Longrightarrow}_Q q_n$ in $Q$ mit $p_j \mathcal{R} q_j$ für alle $j$ aus $\{0, 1, \ldots n\}$. $p_n$ ist für $P$ ein Ruhe-Zustand, es gilt also für alle $\alpha \in (O \cup \{\tau\})$ $p_n \not\xrightarrow{\alpha}_P$. Da $\mathcal{R}$ eine schwache as-Verfeinerungs-Relation ist, muss wegen 1.4 3. auch $q_n \not\xrightarrow{\alpha}_Q$ gelten für alle $\alpha \in (O \cup \{\tau\})$. $q_n$ ist also auch ein ruhiger Zustand. Somit ist $w$ in $StQT_Q \subset QET_Q$ enthalten.

$P \sqsubseteq_{Qui} Q \Rightarrow P \sqsubseteq_E Q$:
Diese Implikation folgt direkt aus der Definition von $\sqsubseteq_{Qui}$ in 4.5. Da $P \sqsubseteq_{Qui} Q$ dort definiert wurde als Relation, die $P \sqsubseteq_E Q$ un $QET_P \subset QET_Q$ erfüllt. Es gilt also $P \sqsubseteq_E Q$.

$P \sqsubseteq_{\text{w-as}} Q \not\Leftarrow P \sqsubseteq_{Qui} Q$:
Wie im Gegenbeispiel für die analoge Implikation aus der Relation $\sqsubseteq_E$ beruht der Grund für die nicht Gültigkeit hier auf darauf, dass Simulationen strenger sind als Sprach Inklusionen. Jedoch funktioniert hier nicht das gleiche Gegenbeispiel, da es zu Problemen mit den Ruhe-Traces führen würde. Um diese zu vermeiden, wird wieder die Technik angewendet an alle Zustände eine $\tau$-Schleife anzufügen. Das daraus entstehende Gegenbeipiel ist in Abbildung 4.5 dargestellt. Die Menge der Inputs $I$ der beiden MEIOs ist leer. Es gilt also $ET_P = ET_Q = QET_P = QET_Q = \emptyset$ und $\{\varepsilon\} = L(P) \subset L(Q) = \{\varepsilon, o\}$. Angenommen es gib eine schache as-Verfeinerungs-Relation $\mathcal{R}$ zwischen $P$ und $Q$. Dann stehen die Startzustände in dieser Relation, es gilt also $p_0 \amalg_0$. Da es keine Fehler-Zustände in $P$ gibt, ist 1.4 1. erfüllt. Die $\tau$-must-Schlingen der Startzustände werden bezüglich der Punkt 3. und 5. der Definition 1.4 gematched. Der 2. Punkt von 1.4 stellt keine Forderungen an die Relation $\mathcal{R}$. Die Transition $q_0 \xrightarrow{o}_Q q_1$ fordert durch 1.4 3. ihre Verfeinerung in $P$. Da es jedoch keine mit $o$ beschriftete Transition in $P$ gibt, kann $\mathcal{R}$ keine as-Verfeinerungs-Relation zwischen $P$ und $Q$ sein.

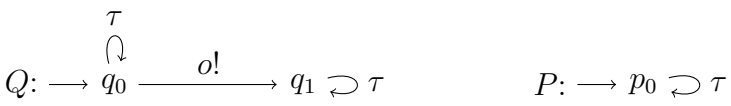


Abbildung 4.5: Gegenbeispiel zu $\sqsubseteq_{\text{w-as}} \Leftarrow \sqsubseteq_{Qui}$

$P \sqsubseteq_{Qui} Q \not\Leftarrow P \sqsubseteq_E Q$:
Die Relation $\sqsubseteq_{Qui}$ stützt sich auf die Definition der Relation $\sqsubseteq_E$. Jedoch erweiter sich die Definition noch um eine weiter Voraussetzung. Es muss also in einem entsprechenden Gegenbeispiel die Inklustion $QET_P \subseteq QET_Q$ verletzt sein. Das Gegenbeispiel ist in Abbildung 4.6 dargestellt. Es wird $I = \emptyset$ vorausgesetzt, damit keine Input-kritischen Traces auftretten. Es gilt also $ET_P = ET_Q = \emptyset$ und $L(P) = L(Q) = \{\varepsilon\}$. Es gilt also $P \sqsubseteq_E Q$.
Jedoch gilt für die strickten Ruhe-Traces $StQT_P = \{\varepsilon\}$ und $StQT_Q = \emptyset$. Es folgt also $QET_P \not\subseteq QET_Q$ und somit ist die Relation $\sqsubseteq_{Qui}$ zwischen $P$ und $Q$ auch nicht erfüllt.


Abbildung 4.6: Gegenbeispiel zu $\sqsubseteq_{Qui} \Leftarrow \sqsubseteq_E$

□

# 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. Wie im letzten Kapitel wird auch hier nur der Testing-Ansatz 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 $P$.*

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 und somit eine optimistische Betrachtungsweise. 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.

**Definition 5.2 (*Test und Verfeinerung für Divergenz*).** *Ein* Test *$T$ ist eine Implementierung. Ein MEIO $P$* as-erfüllt *einen Divergenz-Test $T$, falls $S\|T$ fehler-, ruhe- und divergenz-frei ist für alle $S \in$ as-impl$(P)$. Es wird dann $P\, sat_{\text{as}}^{Div}\, T$ geschrieben.*
*Die Parallelkomposition $S\|T$ ist* fehler-, ruhe- *und* divergenz-frei*, wenn kein Fehler-, Ruhe- oder Divergenz-Zustand lokal erreichbar ist.*
*Ein MEIO $P$* Divergenz-verfeinert *$P'$, falls für alle Tests $T$: $P'\, sat_{\text{as}}^{Div}\, T \Rightarrow P\, sat_{\text{as}}^{Div}\, T$.*

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 „schlimmes Fehlverhalten“ anzusehen ist wie Fehler 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 erreicht.

Somit wird in den Trace-Mengen auch nicht zwischen Fehler-Traces und Divergenz-Traces explizit unterschieden. Dadurch genügt es nicht mehr nur mit den Fehler-Traces die Sprache zu fluten, sondern es muss sowohl mit den Fehler-Traces wie auch den Divergenz-Traces geflutet werden. Ebenso werden die strikten Ruhe-Traces mit diesen beiden Trace-Mengen geflutet.

**Definition 5.3 (*Divergenz-Traces*).** *Sei $P$ ein MEIO und definiere:*

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

Analog zu den Propositionen 3.4 und 4.4 gibt es hier auch eine Proposition, die die Divergenz-Traces eines MEIOs mit den Divergenz-Traces seiner as-Implementierungen verbindet. Die Begründung verläuft analog zu den Propositionen der vorangegangenen Kapitel.

**Proposition 5.4 (*Divergenz-Traces und Implementierungen*).** *Für ein MEIO $P$ gilt:*

1. $StDT(P) \subseteq \left\{w \in \Sigma^* \mid \exists P' \in \text{as-impl}(P) : p'_0 \overset{w}{\Longrightarrow}_{P'} p' \in Div(P')\right\}$,
2. TODO: PrDT Aussage

*Beweis.*

1. Um diese Inklusion beweisen zu können wird wieder eine as-Implementierung $P'$ von $P$ und eine passende as-Verfeinerungs-Relation $\mathcal{R}$ angegeben, so dass alle strikten Divergenz-Traces von $P$ auch in $P'$ enthalten sind. In diesem Fall funktioniert der Ansatz, alle Traces aus $P$ in $P'$ zu implementieren und keine Fehler-Zustände zu übernehmen. Die Definition von $P'$ lautet also:
   - $P' = P$,
   - $p'_0 = p_0$,
   - $I_{P'} = I_P$ und $O_{P'} = O_P$,
   - $\longrightarrow_{P'} = \dashrightarrow_{P'} = \dashrightarrow_P$,
   - $E_{P'} = \emptyset$.

   Die passende as-Verfeinerungs-Relation $\mathcal{R}$ ist die Identität-Relation. Wie bereits im Beweis zu Proposition 2.1 begründet erfüllt $\mathcal{R}$ alle Punkte der Definition 1.3 um eine as-Verfeinerungs-Relation zu sein. Es wird ein $w$ aus $StDT(P)$ betrachtet. Es gibt also einen Trace in $P$ auf dem das Wort $w$ ausgeführt wird und der einen divergenten Zustand erreicht. Es gilt also $\exists w' \in \Sigma_\tau^*, \exists \alpha_1, \alpha_2, \ldots, \alpha_n, \exists p_1, p_2, \ldots, p_n : \hat{w'} = w \wedge w' = \alpha_1\alpha_2\ldots\alpha_n \wedge p_0 \overset{\alpha_1}{\dashrightarrow}_P p_1 \overset{\alpha_2}{\dashrightarrow}_P \ldots p_{n-1} \overset{\alpha_n}{\dashrightarrow}_P p_n \in Div_P$. Die Identitäts-Relation $\mathcal{R}$ setzt die Zustände des Traces mit den analogen Zuständen aus $P'$ in Relation. Zusätzlich mit der Implementierung aller Transitionen aus $P$

in $P'$ ergibt sich der selbe Trace in $P'$. Es gilt also $p'_0 \overset{\alpha_1}{\dashrightarrow}_{P'} p'_1 \overset{\alpha_2}{\dashrightarrow}_{P'} \dots p'_{n-1} \overset{\alpha_n}{\dashrightarrow}_{P'} p'_n$ mit $(p'_j, p_j) \in \mathcal{R}$ für $0 \leq j \leq n$. Da $p_n$ in $Div_P$ enthalten ist, gibt es von diesem Zustand in $P$ aus die Möglichkeit eine unendliche Folge an $\tau$s auszuführen. Die Ausführbarkeit muss sich dabei auf mit $\tau$ beschriftete may-Transitionen in $P$ stützen. Da diese Transitionen in $P'$ alle übernommen wurden, ist auch für $p'_n$ eine unendliche Folge an $\tau$s ausführbar. Es gilt also $p'_n \in Div_{P'}$ und somit $w \in StDT(P')$. Insgesamt folgt also für diese $P'$ $StDT(P) = StDT(P')$.

2. TODO: zu beweisen

□

Da die Ruhe-Traces mit den Fehler- und Divergenz-Traces geflutet werden sollen, kann die Ruhe-Semantik nicht aus dem letzten Kapitel übernommen werden. Auch die geflutete Sprache aus dem Fehler-Kapitel kann nicht beibehalten werden. Nur die Fehler-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 5.5 (*Kommunikationsfehler-, Ruhe- und Divergenz-Semantik*).** *Sei $P$ ein MEIO.*

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

**Proposition 5.6 (*Kommunikationsfehler-, Ruhe-, Divergenz-Semantik und Implementierungen*).** *Sie $P$ ein MEIO.*

1. *Für die Menge der Divergenz-Traces von $P$ gilt $DT(P) \subseteq \bigcup_{P' \in \text{as-impl}(P)} DT(P')$.*
2. *Für die Menge der Fehler-Divergenz-Traces von $P$ gilt die folgende Gleichheit $EDT(P) = \bigcup_{P' \in \text{as-impl}(P)} EDT(P')$.*
3. *Für die Menge der Fehler-divergenz-gefluteten Ruhe-Traces von $P$ gilt $QDT(P) = \bigcup_{P' \in \text{as-impl}(P)} QDT(P')$.*

4. *Für die Menge der Fehler-divergenz-gefluteten Sprache von $P$ gilt* $EDL(P) = \bigcup_{P' \in \text{as-impl}(P)} EDL(P')$.

*Beweis.* TODO: zu beweisen □

Aus der so eben bewiesenen Proposition über die Gleichheit der betrachteten Traces, lässt sich wie in den letzten beiden Kapiteln eine Aussage über die lokale Erreichbarkeit der „fehlerhaften Zustände“ in einer Spezifikation und den zugehörigen Implementierungen treffen.

**Korollar 5.7 (*lokale Divergenz Erreichbarkeit*).** *(i) Falls in einem MEIO $P$ ein Fehler lokal erreichbar ist, dann existiert auch eine as-Implementierung, in der ein Fehler lokal erreichbar ist.*

*(ii) Falls in einem MEIO $P$ Divergenz lokal erreichbar ist, dann existiert auch eine as-Implementierung, in der Divergenz lokal erreichbar ist.*

*(iii) Falls ein MEIO $P$ einen lokal erreichbaren Ruhe-Zustand besitzt, jedoch keinen lokal erreichbaren Fehler, dann existiert auch eine as-Implementierung, in der ein Ruhe-Zustand und kein Fehler lokal erreichbar ist.*

*(iv) Falls es eine as-Implementierung von $P$ gibt, die einen Fehler, Ruhe oder Divergenz lokal erreicht, dann ist auch ein Fehler, Ruhe oder Divergenz in $P$ lokal erreichbar.*

*Beweis.* TODO: zu beweisen □

$\sqsubseteq_{Div}$ ist somit keine Einschränkung von $\sqsubseteq_E$ so wie $\sqsubseteq_{Qui}$. Es können Systeme mit einem Fehler nicht von Systemen mit Divergenz unterschieden werden. Da die Divergenz-Test zwischen diesen Fehler-Arten auch keine Unterscheidung machen, muss eine sinnvolle Relation dies 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. $EDT_{12} = \text{cont}(\text{prune}((EDT_1 \| EDL_2) \cup (EDL_1 \| EDT_2)))$,
2. $QDT_{12} = (QDT_1 \| QDT_2) \cup EDT_{12}$,
3. $EDL_{12} = (EDL_1 \| EDL_2) \cup EDT_{12}$.

*Beweis.*
1. „$\subseteq$“:
Da beide Seiten der Gleichung unter cont abgeschlossen sind, genügt es ein präfix-minimales Element $w$ zu betrachten. Es muss hier unterschieden werden, ob $w \in ET_{12}$ oder $w \in DT_{12} \backslash ET_{12}$ betrachtet wird. Im ersten Fall ist das $w$ in der rechten Seite der Gleichung enthalten wegen des Beweises des ersten Punktes von Satz 3.8 und da

$ET(P) \subseteq EDT(P)$ und $EL(P) \subseteq EDL(P)$ gilt. Deshalb wird im weiteren Verlauf dieses Beweises davon ausgegangen, dass $w \in DT_{12} \backslash ET_{12}$ gilt und es wird versuch zu zeigen, dass dieses $w$ ebenfalls in der rechten Seite enthalten ist. Da das betrachtete $w$ präfix-minimal ist, gilt $w \in PrDT_{12} \backslash ET_{12}$. Aus der Definition 5.5 weiß man, dass ein $v \in O_{12}^*$ existiert, sodass $(p_{01}, p_{02}) \overset{w}{\Rightarrow}_{12} (p_1, p_2) \overset{v}{\Rightarrow}_{12} (p_1', p_2')$ gilt mit $(p_1', p_2') \in Div_{12}$. Durch die Projektion auf die Transitionssysteme $P_1$ und $P_2$ erhält man $p_{01} \overset{w_1}{\Rightarrow}_1 p_1 \overset{v_1}{\Rightarrow}_1 p_1'$ und $p_{02} \overset{w_2}{\Rightarrow}_2 p_2 \overset{v_2}{\Rightarrow}_2 p_2'$ mit $w \in w_1 \| w_2$ und $v \in v_1 \| v_2$. Aus $(p_1', p_2') \in Div_{12}$ folgt, dass oBdA $p_1' \in Div_1$ gilt, d.h. $w_1 v_2 \in StDT_1 \subseteq EDT_1$. Da $p_{02} \overset{w_2 v_2}{\Rightarrow}_2$ gilt, erhält man $w_2 v_2 \in EDL_2$. Somit gilt insgesamt $wv \in EDT_1 \| EDL_2$ 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 cont. Es wird also für ein beliebiges $x \in \text{prune}((EDT_1 \| EDL_2) \cup (EDL_1 \| EDT_2))$ gezeigt, dass dieses oder eines seiner Präfixe auch in $EDT_{12}$ enthalten ist. Da das $x$ aus der prune-Funktion entstanden ist, lässt sich ein $y$ aus $O_{12}^*$ finden, sodass $xy \in (EDT_1 \| EDL_2) \cup (EDL_1 \| EDT_2)$. Es wird nun noch vorausgesetzt, dass oBdA $xy \in EDT_1 \| EDL_2$ gilt, d.h. es existiert $w_1 \in EDT_1$ und $w_2 \in EDL_2$ mit $xy \in w_1 \| w_2$.

TODO: erzwungenen Zeilenumbruch kontrollieren

Die folgende Argumentation läuft analog zu der im Beweis der Inklusion $ET_{12} \supseteq \text{cont}(\text{prune}((ET_1 \| EL_2) \cup (EL_1 \| ET_2)))$ aus Satz 3.8. Es muss dazu nur jeweils an den Stellen, an denen $PrET(P) \cup MIT(P)$ steht auch noch eine Vereinigung mit $PrDT(P)$ vorgenommen werden. Für Fall I und II aus dem Beweis der oben genannten Inklusion von Satz 3.8 ist jeweils kein weiterer Unterfall für $v_2'$ notwendig da, wenn $v_2'$ nicht ausführbar ist, bereits ein Fehler-Zustand in der Parallelkomposition entsteht. Somit ist egal, ob auch noch Divergenz vorlag. Falls $v_2'$ ausführbar, ist nicht relevant, ob eine Divergenz-Möglichkeit bestanden hat, da diese nicht an der Ausführbarkeit ändert. Am Ende ist ein zusätzlicher Fall für $v_1 \in PrDT_1$ zu ergänzen: TODO: erzwungenen Zeilenumbruch kontrollieren

- Fall III ($v_1 \in PrDT_1$): Es existiert ein $u_1$ aus $O_1^*$, sodass $p_{01} \overset{v_1}{\Rightarrow}_1 p_1 \overset{u_1}{\Rightarrow}_1 p_1'$ mit $p_1' \in Div_1$ gilt. Da es hier keine disjunkten Inputmengen gibt kann das $a$, auf das $v_1$ im Fall $v_1 \neq \varepsilon$ endet, ebenfalls der letzte Buchstabe von $v_2$ sein. Im Fall von $v_2 \in MIT_2$ kann somit $a = b$ gelten und damit wäre $v_2 = v_2'$. Dieser Fall verläuft jedoch analog zu Fall Ic) aus dem Beweis der oben genannten Inklusion von Satz 3.8 und wird somit hier nicht weiter betrachtet. Deshalb gilt für alle im folgenden betrachteten Fälle $p_{02} \overset{v_2'}{\Rightarrow}_2 p_2$ mit $(p_{01}, p_{02}) \overset{v'}{\Rightarrow}_2$.
  - Fall IIIa) $\Big(u_2 \in (O_1 \cap I_2)^*, c \in (O_1 \cap I_2)$, sodass $u_2 c$ ein Präfix von $u_1|_{I_2}$ mit $p_2 \overset{u_2}{\Rightarrow}_2 p_2' \overset{c}{\not\rightarrow}_2\Big)$: Für ein Präfix von $u_1' c$ von $u_1$ mit $(u_1' c)|_{I_2} = u_2 c$ weiß man, dass $p_1 \overset{u_1'}{\Rightarrow}_1 p_1'' \overset{c}{\dashrightarrow}_1$. Somit gilt $u_1' \in u_1' \| u_2$ und $(p_1, p_2) \overset{u_1'}{\Rightarrow}_{12} (p_1'', p_2') \in E_{12}$, 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 Fehler. Es wird $v := \text{prune}(v' u_1') \in PrET_{12}$ gewählt, dies ist ein Präfix von $v'$, da $u_1 \in O_1^*$.

– Fall IIIb) $\left(p_2 \overset{u_2}{\Longrightarrow}_2 p_2' \text{ mit } u_2 = u_1|_{I_2}\right)$: Somit ist $u_1 \in u_1 \| u_2$ und $(p_1, p_2) \overset{u_1}{=\!\Rightarrow}_{12} (p_1', p_2') \in Div_{12}$, da $p_1 \in Div_1$. $P_{12}$ hat also die Divergenz von $P_1$ geerbt. Es wird nun $v := \text{prune}(v'u_1) \in PrDT_{12}$ gewählt, das wiederum ein Präfix von $v'$ ist.

2. „$\subseteq$“:
Diese Inklusionsrichtung kann analog zum Beweis derselben Inklusionsrichtung des zweiten Punktes von Satz 4.9 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 $w \in StQT_{12} \backslash EDT_{12}$ der erreichte Zustand $(p_1, p_2)$ kein Fehler 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 $w \in QDT_1 \| QDT_2$ 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 $w \in w_1 \| w_2$ unterschiedliche Möglichkeiten:

- Fall 1 ($w_1 \in EDT_1 \vee w_2 \in EDT_2$): OBdA gilt $w_1 \in EDT_1$. Es kann nun $w_2 \in StQT_2 \subseteq L_2$ gelten oder $w_2 \in EDT_2 \subseteq EDL_2$ und somit gilt auf jeden Fall $w_2 \in EDL_2$. Daraus kann mit dem ersten Punkt dieses Satzes gefolgert werden, dass $w \in EDT_{12}$ gilt und somit $w$ in der linken Seite der Gleichung enthalten ist.

- Fall 2 ($w_1 \in StQT_1 \backslash EDT_1 \wedge w_2 \in StQT_2 \backslash EDT_2$): Dieser Fall läuft analog zu Fall 2 derselben Inklusionsrichtung des Beweises von Satz 4.9. Hierfür muss die Menge $QET_{12}$ durch $QDT_{12}$ ersetzt werden.

3.:
Durch die Definition 5.5 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:

$$\begin{aligned}
(EDL_1 \| EDL_2) \cup EDT_{12} &\overset{5.5}{=} ((L_1 \cup EDT_1) \,\|\, (L_2 \cup EDT_2)) \cup EDT_{12} \\
&= (L_1 \| L_2) \cup \underbrace{(L_1 \| EDT_2)}_{\substack{\subseteq (EDL_1 \| EDT_2) \\ \overset{1.}{\subseteq} EDT_{12}}} \cup \underbrace{(EDT_1 \| L_2)}_{\substack{\subseteq (EDT_1 \| EDL_2) \\ \overset{1.}{\subseteq} EDT_{12}}} \\
&\quad \cup \underbrace{(EDT_1 \| EDT_2)}_{\substack{\subseteq (EDL_1 \| EDT_2) \\ \overset{1.}{\subseteq} EDT_{12}}} \cup EDT_{12} \\
&= (L_1 \| L_2) \cup EDT_{12} \\
&\overset{2.2}{=} L_{12} \cup EDT_{12} \\
&\overset{5.5}{=} EDL_{12}.
\end{aligned}$$

□

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

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

*Beweis.* Um zu zeigen, dass es sich bei $\sqsubseteq_{Div}$ um eine Präkongruenz handelt, muss nachgewiesen werden, dass 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 3.9 und 4.10, aus der Monotonie von cont, prune und $\cdot \| \cdot$ auf Sprachen wie folgt:

- $$\begin{aligned}
  EDT_{31} &\overset{5.8\ 1.}{=} \text{cont}\,(\text{prune}\,((EDT_3 \| EDL_1) \cup (EDL_3 \| EDT_1))) \\
  &\overset{\substack{EDT_1 \subseteq EDT_2 \\ \text{und} \\ EDL_1 \subseteq EDL_2}}{\subseteq} \text{cont}\,(\text{prune}\,((EDT_3 \| EDL_2) \cup (EDL_3 \| EDT_2))) \\
  &\overset{5.8\ 1.}{=} EDT_{32},
  \end{aligned}$$

- $$\begin{aligned}
  QDT_{31} &\overset{5.8\ 2.}{=} (QDT_3 \| QDT_1) \cup EDT_{31} \\
  &\overset{\substack{EDT_{31} \subseteq EDT_{32}, \\ \text{und} \\ QDT_1 \subseteq QDT_2}}{\subseteq} (QDT_3 \| QDT_2) \cup EDT_{32} \\
  &\overset{5.8\ 2.}{=} QDT_{32}.
  \end{aligned}$$

- $EDL_{31} \overset{5.8\ 3.}{=} (EDL_3 \| EDL_1) \cup EDT_{31}$
  $\overset{\substack{EDT_{31} \subseteq EDT_{32}, \\ \text{und} \\ EDL_1 \subseteq EDL_2}}{\subseteq} (EDL_3 \| EDL_2) \cup EDT_{32}$
  $\overset{5.8\ 3.}{=} EDL_{32}.$

□

Im nächsten Lemme soll eine Verfeinerung bezüglich guter Kommunikation betrachtet werden. Die Vorgaben für gute Kommunikation gibt hierbei die Definition der Tests und die daraus resultierende Verfeinerung in 5.2 vor. Es muss in diesem Lemma eine Veränderung zu den analogen Lemmata aus den vorangegangenen Kapitel vorgenommen werden. Die Einschränkung der Tests $T$ auf Partner, kann nicht mehr beibehalten werden, da die Strategie zur Vermeidung von Ruhe im Beweis aus dem letzten Kapitel hier zu Divergenz führen würde. Somit werden für die Ruhe-Vermeidung in diesem Kapitel Aktionen außerhalb der Menge Synch benötigt, die nicht die interne Aktionen $\tau$ sind. Jedoch müssen trotzdem nicht alle Tests $T$ betrachtet werden. Es kann eine Einschränkung gemacht werden, sodass $T$ fast ein Partner ist. Zur Vereinfachung von umständlichen Formulierungen im Folgenden wird hierfür nun ein neuer Begriff definiert. Der jedoch auch bereits so in z.B. [Sch16] für EIOs verwendet und definiert wurde.

**Definition 5.10 (*ω-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.*

Ein $\omega$-Partner $P_1$ von $P_2$ unterscheidet sich von einem Partner von $P_2$ nur um den Output $\omega$, der nicht in der Menge Synch$(P_1, P_2)$ enthalten ist.

**Lemma 5.11 (*Testing-Verfeinerung mit Divergenz*).** *Gegeben sind zwei MEIOs $P_1$ und $P_2$ mit der gleichen Signatur. Wenn für alle Tests $T$, die $\omega$-Partner von $P_1$ bzw. $P_2$ sind, $P_2\, sat_{\text{as}}^{Div}\, T \Rightarrow P_1\, sat_{\text{as}}^{Div}\, T$ gilt, dann folgt daraus die Gültigkeit von $P_1 \sqsubseteq_{Div} P_2$.*

*Beweis.* TODO: zu beweisen □

**Satz 5.12.** *Falls $P_1 \sqsubseteq_{Div} P_2$ gilt folgt daraus auch, dass $P_1$ $P_2$ Divergenz-verfeinert.*

*Beweis.* TODO: zu beweisen □

Wie in den letzten beiden Kapitel, wurde eine Folgerungskette gezeigt, die sich zu einem Ring schließt. Dies ist in Abbildung 5.1 dargestellt.

TODO: Satz für den Zusammenhang der Verfeinerungs-Relationen erweitern: $\sqsubseteq_{\text{as}} \Rightarrow \sqsubseteq_{Div} \overset{?}{\Rightarrow} \sqsubseteq_{Qui}$ und $\sqsubseteq_{\text{w-as}} \not\Rightarrow \sqsubseteq_{Div}$

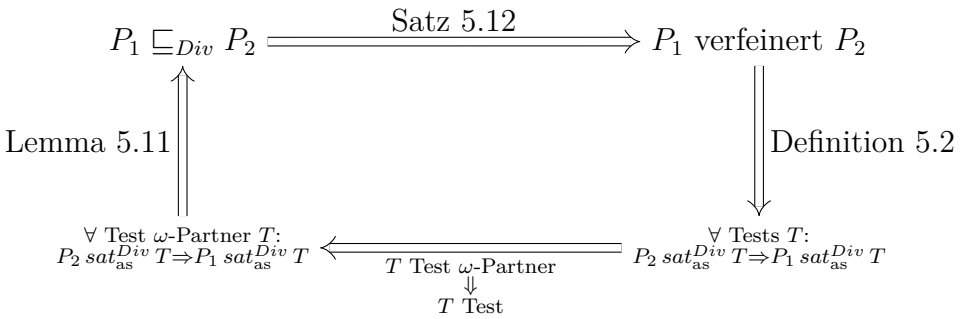


Abbildung 5.1: Folgerungskette der Testing-Verfeinerung und Divergenz-Relation

# Literaturverzeichnis


[BFLV16] Ferenc Bujtor, Sascha Fendrich, Gerald Lüttgen, und Walter Vogler, *Nondeterministic Modal Interfaces*, Theor. Comput. Sci. **642** (2016), 24–53.

[BV15a] Ferenc Bujtor und Walter Vogler, *Error-pruning in interface automata*, Theor. Comput. Sci. **597** (2015), 18–39.

[BV15b] ———, *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.