Testing of Reactive Systems

Lecture 5: Some more Implementation Relations

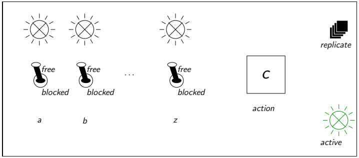
Henrik Bohnenkamp

Lehrstuhl Informatik 2 (MOVES)
RWTH Aachen

Summer Semester 2009

What happened so far?

- Bisimulation
- Comparison of preorders/equivalences (finer/coarser)
- Linear-Time/Branching-Time spectrum
- Effect of nondeterminism on the LTBT spectrum
- Observing and Manipulating Processes: The Generative Machine



What is coming?

We will

- formalise notion of (primitive) observation
- introduce so-called observers
- observers will be special processes: test expressions
- observers will manipulate and observe, i.e.: test
- characterise a few simple preorders by observers
- establish order in LTBT spectrum

What is coming?

We will

- formalise notion of (primitive) observation
- introduce so-called observers
- observers will be special processes: test expressions
- observers will manipulate and observe, i.e.: test
- characterise a few simple preorders by observers
- establish order in LTBT spectrum

What is coming?

We will

- formalise notion of (primitive) observation
- introduce so-called observers
- observers will be special processes: test expressions
- observers will manipulate and observe, i.e.: test
- characterise a few simple preorders by observers
- establish order in LTBT spectrum

What is coming?

We will

- formalise notion of (primitive) observation
- introduce so-called observers
- observers will be special processes: test expressions
- observers will manipulate and observe, i.e.: test
- characterise a few simple preorders by observers
- establish order in LTBT spectrum

What is coming?

We will

- formalise notion of (primitive) observation
- introduce so-called observers
- observers will be special processes: test expressions
- observers will manipulate and observe, i.e.: test
- characterise a few simple preorders by observers
- establish order in LTBT spectrum

What is coming?

We will

- formalise notion of (primitive) observation
- introduce so-called observers
- observers will be special processes: test expressions
- observers will manipulate and observe, i.e.: test
- characterise a few simple preorders by observers
- establish order in LTBT spectrum

1. Some more Implementation Relations

2. Failure Trace Preorder and Observers

We assume (again): no τ-steps

Trace preorder

$\mathcal{O}_{tr}$

Let $\mathcal{O}_{tr}$ be the set of test expressions defined by the following grammar:

$$o \quad \longrightarrow \quad a.o \quad | \quad \mathrm{t.STOP},$$

where $a \in Act$, and $\mathrm{t} \notin Act$ is a special action which denotes the end of the observation.

Example

For $Act = \{a, b, c\}$, the following are typical test expressions:

$$\mathrm{t.STOP},\ a.c.a.b.\mathrm{t.STOP},\ \text{etc.}$$

Convention

- We write $\mathrm{t.STOP}$ just as t
- We write $Act_{tr} = Act \cup \{\mathrm{t}\}$.

Trace preorder

$\mathcal{O}_{tr}$

Let $\mathcal{O}_{tr}$ be the set of test expressions defined by the following grammar:

$$o \longrightarrow a.o \mid \mathrm{t.STOP},$$

where $a \in Act$, and $\mathrm{t} \notin Act$ is a special action which denotes the end of the observation.

Example

For $Act = \{a, b, c\}$, the following are typical test expressions:

$$\mathrm{t.STOP},\ a.c.a.b.\mathrm{t.STOP},\ \text{etc.}$$

Convention

- We write $\mathrm{t.STOP}$ just as t
- We write $Act_{tr} = Act \cup \{\mathrm{t}\}$.

Trace preorder

$\mathcal{O}_{tr}$

Let $\mathcal{O}_{tr}$ be the set of test expressions defined by the following grammar:

$$o \longrightarrow a.o \mid \mathrm{t.STOP},$$

where $a \in Act$, and $\mathrm{t} \notin Act$ is a special action which denotes the end of the observation.

Example

For $Act = \{a, b, c\}$, the following are typical test expressions:

$$\mathrm{t.STOP},\ a.c.a.b.\mathrm{t.STOP},\ \text{etc.}$$

Convention

- We write $\mathrm{t.STOP}$ just as t
- We write $Act_{tr} = Act \cup \{\mathrm{t}\}$.

Test Execution

Definition 3.1.2: Test execution

Let o be a test-expression and p, p' processes. Test execution is described by the test-operator $\mathbin{\top\!\!\!\top}$, whose behaviour is defined by the following rules:

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{tr}) \qquad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \mathbin{\top\!\!\!\top} p \xrightarrow{a} o' \mathbin{\top\!\!\!\top} p'}(a \in Act) \qquad \frac{o \xrightarrow{t} o'}{o \mathbin{\top\!\!\!\top} p \xrightarrow{t} \mathrm{PASS}}$$

- Observations of o and p:

$$\mathrm{obs}_{tr}(o, p) := \{\sigma \mid o \mathbin{\top\!\!\!\top} p \xrightarrow{\sigma} \mathrm{PASS}\}$$

- All observations for p:

$$\mathrm{obs}_{tr}(p) = \bigcup \mathrm{obs}_{tr}(o, p).$$

Test Execution

Definition 3.1.2: Test execution

Let o be a test-expression and p, p' processes. Test execution is described by the test-operator $\mathbin{\top\!\!\top}$, whose behaviour is defined by the following rules:

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{tr}) \qquad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \mathbin{\top\!\!\top} p \xrightarrow{a} o' \mathbin{\top\!\!\top} p'}(a \in Act) \qquad \frac{o \xrightarrow{t} o'}{o \mathbin{\top\!\!\top} p \xrightarrow{t} \text{PASS}}$$

- Observations of o and p:

$$\text{obs}_{tr}(o, p) := \{\sigma \mid o \mathbin{\top\!\!\top} p \xrightarrow{\sigma} \text{PASS}\}$$

- All observations for p:

$$\text{obs}_{tr}(p) = \bigcup \text{obs}_{tr}(o, p).$$

Test Execution

Definition 3.1.2: Test execution

Let o be a test-expression and p, p' processes. Test execution is described by the test-operator $\mathbin{\top\!\!\!\top}$, whose behaviour is defined by the following rules:

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{tr}) \quad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \mathbin{\top\!\!\!\top} p \xrightarrow{a} o' \mathbin{\top\!\!\!\top} p'}(a \in Act) \quad \frac{o \xrightarrow{t} o'}{o \mathbin{\top\!\!\!\top} p \xrightarrow{t} \text{PASS}}$$

- Observations of o and p:

$$\text{obs}_{tr}(o, p) := \{\sigma \mid o \mathbin{\top\!\!\!\top} p \xrightarrow{\sigma} \text{PASS}\}$$

- All observations for p:

$$\text{obs}_{tr}(p) = \bigcup \text{obs}_{tr}(o, p).$$

Test Execution

Definition 3.1.2: Test execution

Let o be a test-expression and p, p' processes. Test execution is described by the test-operator $\mathbin{\top\!\!\!\top}$, whose behaviour is defined by the following rules:

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{tr}) \quad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \mathbin{\top\!\!\!\top} p \xrightarrow{a} o' \mathbin{\top\!\!\!\top} p'}(a \in Act) \quad \frac{o \xrightarrow{t} o'}{o \mathbin{\top\!\!\!\top} p \xrightarrow{t} \mathrm{PASS}}$$

- Observations of o and p:

$$obs_{tr}(o, p) := \{\sigma \mid o \mathbin{\top\!\!\!\top} p \xrightarrow{\sigma} \mathrm{PASS}\}$$

- All observations for p:

$$obs_{tr}(p) = \bigcup obs_{tr}(o, p).$$

Test Execution

Definition 3.1.2: Test execution

Let o be a test-expression and p, p' processes. Test execution is described by the test-operator $\talloblong$, whose behaviour is defined by the following rules:

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{tr}) \quad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \talloblong p \xrightarrow{a} o' \talloblong p'}(a \in Act) \quad \frac{o \xrightarrow{t} o'}{o \talloblong p \xrightarrow{t} \mathrm{PASS}}$$

- Observations of o and p:

$$\mathrm{obs}_{tr}(o, p) := \{\sigma \mid o \talloblong p \xrightarrow{\sigma} \mathrm{PASS}\}$$

- All observations for p:

$$\mathrm{obs}_{tr}(p) = \bigcup \mathrm{obs}_{tr}(o, p).$$

Test Execution

Definition 3.1.2: Test execution

Let o be a test-expression and p, p' processes. Test execution is described by the test-operator $\talloblong$, whose behaviour is defined by the following rules:

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{tr}) \quad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \talloblong p \xrightarrow{a} o' \talloblong p'}(a \in Act) \quad \frac{o \xrightarrow{t} o'}{o \talloblong p \xrightarrow{t} \text{PASS}}$$

- Observations of o and p:

$$\text{obs}_{tr}(o, p) := \{\sigma \mid o \talloblong p \xrightarrow{\sigma} \text{PASS}\}$$

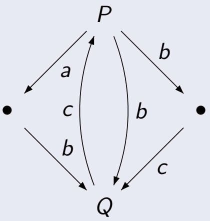
- All observations for p:

$$\text{obs}_{tr}(p) = \bigcup \text{obs}_{tr}(o, p).$$

Test Execution

Example 3.1.3

Let $P \triangleq b.c.Q + a.b.Q + b.Q$ and $Q \triangleq c.P$ and $o = b.c.c.\mathrm{t}$.



$$b.c.c.\mathrm{t} \mathbin{\top\!\!\top} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{\top\!\!\top} c.Q \xrightarrow{c} c.\mathrm{t} \mathbin{\top\!\!\top} Q \xrightarrow{c} \mathrm{t} \mathbin{\top\!\!\top} P \xrightarrow{\mathrm{t}} \text{PASS},$$

but also

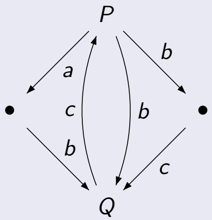
$$b.c.c.\mathrm{t} \mathbin{\top\!\!\top} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{\top\!\!\top} Q \xrightarrow{c} c.\mathrm{t} \mathbin{\top\!\!\top} P \not\rightarrow .$$

Observations: $\mathrm{obs}_{tr}(o, P) = \{bcct\}$.

Test Execution

Example 3.1.3

Let $P \mathrel{\hat{=}} b.c.Q + a.b.Q + b.Q$ and $Q \mathrel{\hat{=}} c.P$ and $o = b.c.c.\mathrm{t}$.



$$b.c.c.\mathrm{t} \mathbin{⫪} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{⫪} c.Q \xrightarrow{c} c.\mathrm{t} \mathbin{⫪} Q \xrightarrow{c} \mathrm{t} \mathbin{⫪} P \xrightarrow{\mathrm{t}} \mathrm{PASS},$$

but also

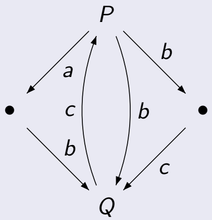
$$b.c.c.\mathrm{t} \mathbin{⫪} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{⫪} Q \xrightarrow{c} c.\mathrm{t} \mathbin{⫪} P \not\rightarrow .$$

Observations: $\mathrm{obs}_{tr}(o, P) = \{bcct\}$.

Test Execution

Example 3.1.3

Let $P \triangleq b.c.Q + a.b.Q + b.Q$ and $Q \triangleq c.P$ and $o = b.c.c.\mathrm{t}$.



$$b.c.c.\mathrm{t} \mathbin{\rceil\!\lceil} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{\rceil\!\lceil} c.Q \xrightarrow{c} c.\mathrm{t} \mathbin{\rceil\!\lceil} Q \xrightarrow{c} \mathrm{t} \mathbin{\rceil\!\lceil} P \xrightarrow{\mathrm{t}} \mathrm{PASS},$$

but also

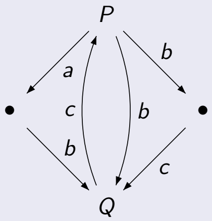
$$b.c.c.\mathrm{t} \mathbin{\rceil\!\lceil} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{\rceil\!\lceil} Q \xrightarrow{c} c.\mathrm{t} \mathbin{\rceil\!\lceil} P \not\rightarrow .$$

Observations: $\mathrm{obs}_{tr}(o, P) = \{bcct\}$.

Test Execution

Example 3.1.3

Let $P \triangleq b.c.Q + a.b.Q + b.Q$ and $Q \triangleq c.P$ and $o = b.c.c.t$.



$$b.c.c.t \Top P \xrightarrow{b} c.c.t \Top c.Q \xrightarrow{c} c.t \Top Q \xrightarrow{c} t \Top P \xrightarrow{t} \text{PASS},$$

but also

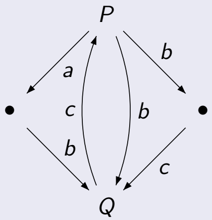
$$b.c.c.t \Top P \xrightarrow{b} c.c.t \Top Q \xrightarrow{c} c.t \Top P \not\rightarrow .$$

Observations: $\mathrm{obs}_{tr}(o, P) = \{bcct\}$.

Test Execution

Example 3.1.3

Let $P \mathrel{\hat{=}} b.c.Q + a.b.Q + b.Q$ and $Q \mathrel{\hat{=}} c.P$ and $o = b.c.c.\mathrm{t}$.



$$b.c.c.\mathrm{t} \mathbin{⫪} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{⫪} c.Q \xrightarrow{c} c.\mathrm{t} \mathbin{⫪} Q \xrightarrow{c} \mathrm{t} \mathbin{⫪} P \xrightarrow{\mathrm{t}} \mathrm{PASS},$$

but also

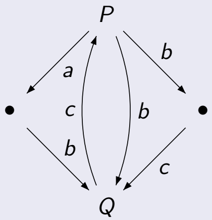
$$b.c.c.\mathrm{t} \mathbin{⫪} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{⫪} Q \xrightarrow{c} c.\mathrm{t} \mathbin{⫪} P \not\rightarrow .$$

Observations: $\mathrm{obs}_{tr}(o, P) = \{bcct\}$.

Test Execution

Example 3.1.3

Let $P \triangleq b.c.Q + a.b.Q + b.Q$ and $Q \triangleq c.P$ and $o = b.c.c.\mathrm{t}$.



$$b.c.c.\mathrm{t} \mathbin{⫪} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{⫪} c.Q \xrightarrow{c} c.\mathrm{t} \mathbin{⫪} Q \xrightarrow{c} \mathrm{t} \mathbin{⫪} P \xrightarrow{\mathrm{t}} \mathrm{PASS},$$

but also

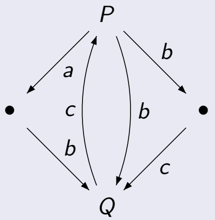
$$b.c.c.\mathrm{t} \mathbin{⫪} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{⫪} Q \xrightarrow{c} c.\mathrm{t} \mathbin{⫪} P \not\rightarrow .$$

Observations: $\mathrm{obs}_{tr}(o, P) = \{bcct\}$.

Test Execution

Example 3.1.3

Let $P \triangleq b.c.Q + a.b.Q + b.Q$ and $Q \triangleq c.P$ and $o = b.c.c.\mathrm{t}$.



$$b.c.c.\mathrm{t} \mathbin{\top\!\!\top} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{\top\!\!\top} c.Q \xrightarrow{c} c.\mathrm{t} \mathbin{\top\!\!\top} Q \xrightarrow{c} \mathrm{t} \mathbin{\top\!\!\top} P \xrightarrow{\mathrm{t}} \textsc{pass},$$

but also

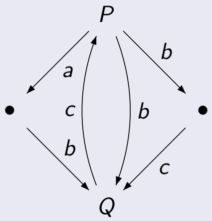
$$b.c.c.\mathrm{t} \mathbin{\top\!\!\top} P \xrightarrow{b} c.c.\mathrm{t} \mathbin{\top\!\!\top} Q \xrightarrow{c} c.\mathrm{t} \mathbin{\top\!\!\top} P \not\rightarrow .$$

Observations: $\mathrm{obs}_{tr}(o, P) = \{bcct\}$.

Test Execution

Example 3.1.3

Let $P \triangleq b.c.Q + a.b.Q + b.Q$ and $Q \triangleq c.P$ and $o = b.c.c.\mathsf{t}$.



$$b.c.c.\mathsf{t} \rceil\!\lceil P \xrightarrow{b} c.c.\mathsf{t} \rceil\!\lceil c.Q \xrightarrow{c} c.\mathsf{t} \rceil\!\lceil Q \xrightarrow{c} \mathsf{t} \rceil\!\lceil P \xrightarrow{\mathsf{t}} \text{PASS},$$

but also

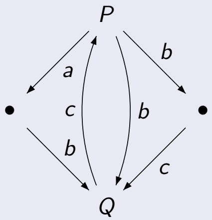
$$b.c.c.\mathsf{t} \rceil\!\lceil P \xrightarrow{b} c.c.\mathsf{t} \rceil\!\lceil Q \xrightarrow{c} c.\mathsf{t} \rceil\!\lceil P \not\rightarrow .$$

Observations: $\text{obs}_{tr}(o, P) = \{bcct\}$.

Test Execution

Example 3.1.3

Let $P \mathrel{\hat{=}} b.c.Q + a.b.Q + b.Q$ and $Q \mathrel{\hat{=}} c.P$ and $o = b.c.c.\mathrm{t}$.



$$b.c.c.\mathrm{t} \rceil\!\lceil P \xrightarrow{b} c.c.\mathrm{t} \rceil\!\lceil c.Q \xrightarrow{c} c.\mathrm{t} \rceil\!\lceil Q \xrightarrow{c} \mathrm{t} \rceil\!\lceil P \xrightarrow{\mathrm{t}} \mathrm{PASS},$$

but also

$$b.c.c.\mathrm{t} \rceil\!\lceil P \xrightarrow{b} c.c.\mathrm{t} \rceil\!\lceil Q \xrightarrow{c} c.\mathrm{t} \rceil\!\lceil P \not\rightarrow .$$

Observations: $\mathrm{obs}_{tr}(o, P) = \{bcct\}$.

Trace preorder

Lemma 3.1.4

For $p, q \in \mathbb{P}$:

$$p \leq_{tr} q \quad \text{iff} \quad \mathrm{obs}_{tr}(p) \subseteq \mathrm{obs}_{tr}(q)$$

Proof

Exercise.

Note

$$p \leq_{tr} q \quad \text{iff} \quad \forall o \in \mathcal{O}_{tr} : \mathrm{obs}_{tr}(o, p) \subseteq \mathrm{obs}_{tr}(o, q).$$

Trace preorder

Lemma 3.1.4

For $p, q \in \mathbb{P}$:

$$p \leq_{tr} q \quad \text{iff} \quad \text{obs}_{tr}(p) \subseteq \text{obs}_{tr}(q)$$

Proof

Exercise.

Note

$$p \leq_{tr} q \quad \text{iff} \quad \forall o \in \mathcal{O}_{tr} : \text{obs}_{tr}(o, p) \subseteq \text{obs}_{tr}(o, q).$$

Trace preorder

Lemma 3.1.4

For $p, q \in \mathbb{P}$:

$$p \leq_{tr} q \quad \text{iff} \quad \text{obs}_{tr}(p) \subseteq \text{obs}_{tr}(q)$$

Proof

Exercise.

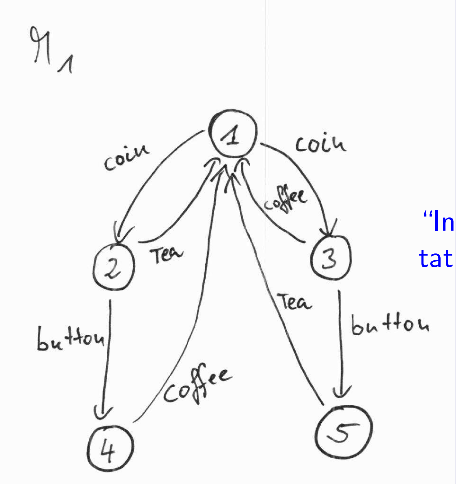
Note

$$p \leq_{tr} q \quad \text{iff} \quad \forall o \in \mathcal{O}_{tr} : \text{obs}_{tr}(o, p) \subseteq \text{obs}_{tr}(o, q).$$

Failures Preorder

Example 3.2.1: Crummy Tea & Coffee Inc.

Coffee & Tea Dispenser Inc. (CTD) Specification:

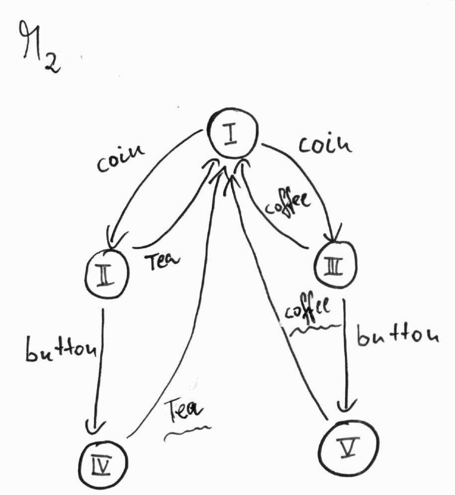


"Inviting offers for implementation, correct wrt. $\leq_{tr}$"

Failures Preorder

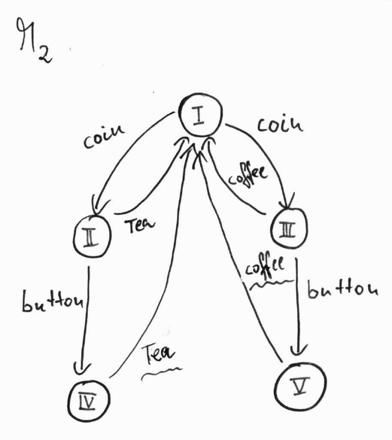
Example 3.2.1: Crummy Tea & Coffee Inc.

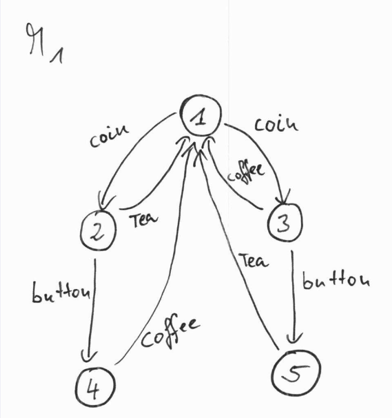
Crummy Tea & Coffee Inc. offers CTD the following:



Failures Preorder

On first sight: M2 seems ok

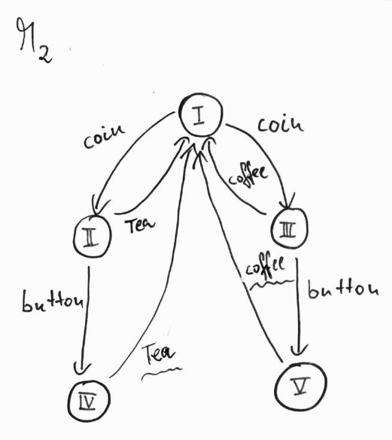


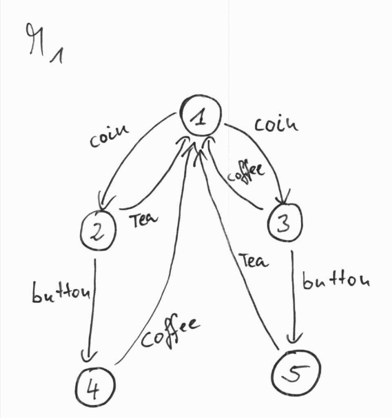


$I \leq_{tr} 1$

Failures Preorder

On first sight: M2 seems ok

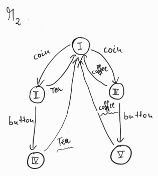


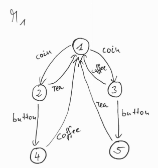


But trace inclusion is not enough! BUTTON in M_2 has no effect!

Failures Preorder

On first sight: M2 seems ok

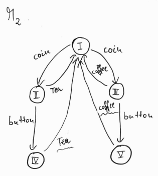


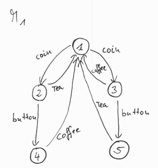


- We need a finer implementation relation.
- Approach: which actions can not be performed in the states?
- refusal sets
- refusal sets of state 1 and I are $\{\text{BUTTON}, \text{COFFEE}, \text{TEA}\}$

Failures Preorder

On first sight: M2 seems ok

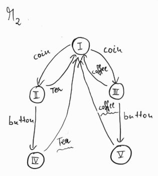


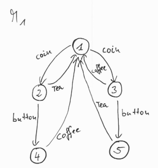


- We need a finer implementation relation.
- Approach: which actions can not be performed in the states?
- refusal sets
- refusal sets of state 1 and I are $\{\text{BUTTON}, \text{COFFEE}, \text{TEA}\}$

Failures Preorder

On first sight: M2 seems ok

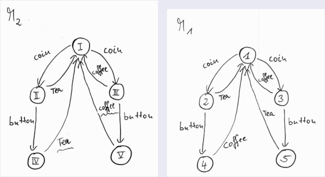




- We need a finer implementation relation.
- Approach: which actions can not be performed in the states?
- refusal sets
- refusal sets of state 1 and I are $\{\text{BUTTON}, \text{COFFEE}, \text{TEA}\}$

Failures Preorder

On first sight: M2 seems ok



- We need a finer implementation relation.
- Approach: which actions can not be performed in the states?
- refusal sets
- refusal sets of state 1 and I are $\{\text{BUTTON}, \text{COFFEE}, \text{TEA}\}$

Failures preorder

Refusals, Failure pairs and Failures preorder

- Refusals $R(Act) = \{\widetilde{X} \mid X \subseteq Act\}$
- $Act_{ft} = Act \cup R(Act)$.
- For $\widetilde{X} \in R(Act)$, X is called refusal set.
- Failure pairs are traces over Act_{ft} of the form $\sigma \cdot \widetilde{X}$, where $\sigma \in Act^*$
- The failure pairs $F(p)$ of process $p \in \mathbb{P}$ are defined as:
  - $\widetilde{X} \in F(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$.
  - $\sigma \cdot \widetilde{X} \in F(p)$ for $\sigma \in Act^*$, if $p \xrightarrow{\sigma} p'$ and $\widetilde{X} \in F(p')$.
- The failures preorder $\leq_f$ is defined as:

$$p \leq_f q \text{ if and only if } F(p) \subseteq F(q).$$

Failures preorder

Refusals, Failure pairs and Failures preorder

- Refusals $R(Act) = \{\widetilde{X} \mid X \subseteq Act\}$
- $Act_{ft} = Act \cup R(Act)$.
- For $\widetilde{X} \in R(Act)$, X is called refusal set.
- Failure pairs are traces over Act_{ft} of the form $\sigma \cdot \widetilde{X}$, where $\sigma \in Act^*$
- The failure pairs $F(p)$ of process $p \in \mathbb{P}$ are defined as:
  - $\widetilde{X} \in F(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$.
  - $\sigma \cdot \widetilde{X} \in F(p)$ for $\sigma \in Act^*$, if $p \xrightarrow{\sigma} p'$ and $\widetilde{X} \in F(p')$.
- The failures preorder $\leq_f$ is defined as:

$$p \leq_f q \text{ if and only if } F(p) \subseteq F(q).$$

Failures preorder

Refusals, Failure pairs and Failures preorder

- Refusals $R(Act) = \{\widetilde{X} \mid X \subseteq Act\}$
- $Act_{ft} = Act \cup R(Act)$.
- For $\widetilde{X} \in R(Act)$, X is called refusal set.
- Failure pairs are traces over Act_{ft} of the form $\sigma \cdot \widetilde{X}$, where $\sigma \in Act^*$
- The failure pairs $F(p)$ of process $p \in \mathbb{P}$ are defined as:
  - $\widetilde{X} \in F(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$.
  - $\sigma \cdot \widetilde{X} \in F(p)$ for $\sigma \in Act^*$, if $p \xrightarrow{\sigma} p'$ and $\widetilde{X} \in F(p')$.
- The failures preorder $\leq_f$ is defined as:

$$p \leq_f q \text{ if and only if } F(p) \subseteq F(q).$$

Failures preorder

Refusals, Failure pairs and Failures preorder

- Refusals $R(Act) = \{\widetilde{X} \mid X \subseteq Act\}$
- $Act_{ft} = Act \cup R(Act)$.
- For $\widetilde{X} \in R(Act)$, X is called refusal set.
- Failure pairs are traces over Act_{ft} of the form $\sigma \cdot \widetilde{X}$, where $\sigma \in Act^*$
- The failure pairs $F(p)$ of process $p \in \mathbb{P}$ are defined as:
  - $\widetilde{X} \in F(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$.
  - $\sigma \cdot \widetilde{X} \in F(p)$ for $\sigma \in Act^*$, if $p \xrightarrow{\sigma} p'$ and $\widetilde{X} \in F(p')$.
- The failures preorder $\leq_f$ is defined as:

$$p \leq_f q \text{ if and only if } F(p) \subseteq F(q).$$

Failures preorder

Refusals, Failure pairs and Failures preorder

- Refusals $R(Act) = \{\widetilde{X} \mid X \subseteq Act\}$
- $Act_{ft} = Act \cup R(Act)$.
- For $\widetilde{X} \in R(Act)$, X is called refusal set.
- Failure pairs are traces over Act_{ft} of the form $\sigma \cdot \widetilde{X}$, where $\sigma \in Act^*$
- The failure pairs $F(p)$ of process $p \in \mathbb{P}$ are defined as:
  - $\widetilde{X} \in F(p)$, if $p \stackrel{a}{\not\rightarrow}$ for all $a \in X$.
  - $\sigma \cdot \widetilde{X} \in F(p)$ for $\sigma \in Act^*$, if $p \stackrel{\sigma}{\rightarrow} p'$ and $\widetilde{X} \in F(p')$.
- The failures preorder $\leq_f$ is defined as:

$$p \leq_f q \text{ if and only if } F(p) \subseteq F(q).$$

Failures preorder

Refusals, Failure pairs and Failures preorder

- Refusals $R(Act) = \{\widetilde{X} \mid X \subseteq Act\}$
- $Act_{ft} = Act \cup R(Act)$.
- For $\widetilde{X} \in R(Act)$, X is called refusal set.
- Failure pairs are traces over Act_{ft} of the form $\sigma \cdot \widetilde{X}$, where $\sigma \in Act^*$
- The failure pairs $F(p)$ of process $p \in \mathbb{P}$ are defined as:
  - $\widetilde{X} \in F(p)$, if $p \stackrel{a}{\not\rightarrow}$ for all $a \in X$.
  - $\sigma \cdot \widetilde{X} \in F(p)$ for $\sigma \in Act^*$, if $p \stackrel{\sigma}{\rightarrow} p'$ and $\widetilde{X} \in F(p')$.
- The failures preorder $\leq_f$ is defined as:

$$p \leq_f q \text{ if and only if } F(p) \subseteq F(q).$$

Failures preorder

Notes

- Refusal $\widetilde{X}$: special kind of action
- corresponds one-to-one to $X \subseteq Act$.
- $\sigma \cdot \widetilde{X} \in F(p)$: there is a trace leading to state p' such that p' **refuses** X.
- p **refuses** Act: equivalent to p is deadlocked

Failures preorder

Notes

- Refusal $\widetilde{X}$: special kind of action
- corresponds one-to-one to $X \subseteq Act$.
- $\sigma \cdot \widetilde{X} \in F(p)$: there is a trace leading to state p' such that p' **refuses** X.
- p **refuses** Act: equivalent to p is deadlocked

Failures preorder

Notes

- Refusal $\widetilde{X}$: special kind of action
- corresponds one-to-one to $X \subseteq Act$.
- $\sigma \cdot \widetilde{X} \in F(p)$: there is a trace leading to state p' such that p' **refuses** X.
- p **refuses** Act: equivalent to p is deadlocked

Failures preorder

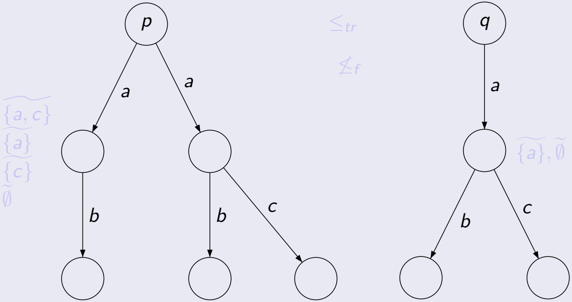
Notes

- Refusal $\widetilde{X}$: special kind of action
- corresponds one-to-one to $X \subseteq Act$.
- $\sigma \cdot \widetilde{X} \in F(p)$: there is a trace leading to state p' such that p' **refuses** X.
- p **refuses** Act: equivalent to p is deadlocked

Failures preorder

Example

Let $Act = \{a, b, c\}$



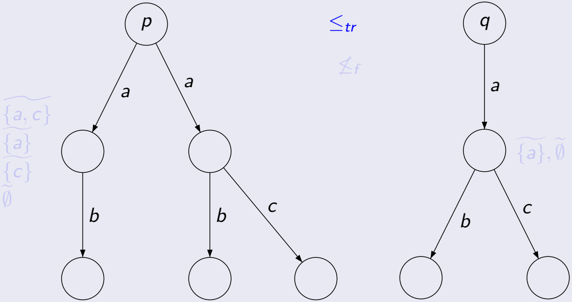
$p \leq_{tr} q$ is obvious

$p \not\leq_f q$ since $\sigma = a \cdot \widetilde{\{a, c\}} \in F(p)$, but not $a \cdot \widetilde{\{a, c\}} \in F(q)$.

Failures preorder

Example

Let $Act = \{a, b, c\}$



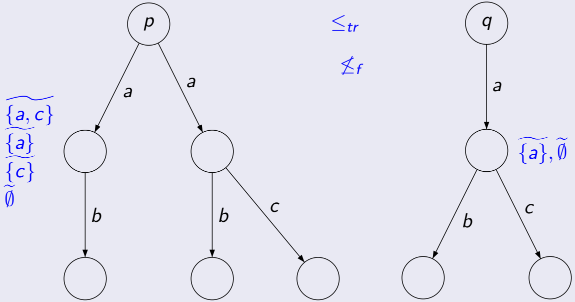
$p \leq_{tr} q$ is obvious

$p \not\leq_f q$ since $\sigma = a \cdot \widetilde{\{a,c\}} \in F(p)$, but not $a \cdot \widetilde{\{a,c\}} \in F(q)$.

Failures preorder

Example

Let $Act = \{a, b, c\}$



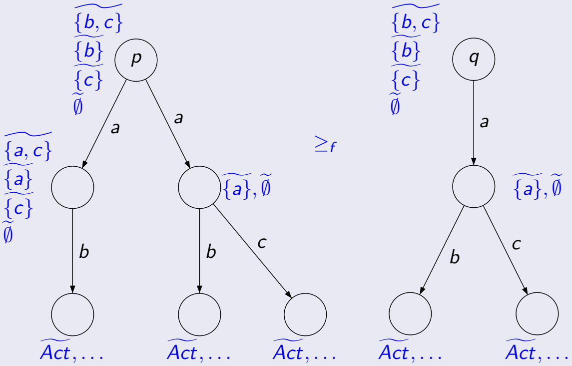
$p \leq_{tr} q$ is obvious

$p \not\leq_f q$ since $\sigma = a \cdot \widetilde{\{a, c\}} \in F(p)$, but not $a \cdot \widetilde{\{a, c\}} \in F(q)$.

Failures preorder

Example (cont.)

$q \leq_f p$: we annotate the states with the set of refusal sets:



- Failure pairs of p e.g.: $\widetilde{\emptyset}$, $\widetilde{\{b,c\}}$, $a\widetilde{\{a,c\}}$, $a\widetilde{\{a\}}$, $ac\widetilde{Act}, \ldots$.
- q is a part of p, $q \leq_f p$ obvious.

3.2.4 Example 3.2.1 continued

Coffee & Tea Dispenser Inc.:

- "Only implementations correct according to $\leq_f$ are acceptable"
- "obviously $M_2 \not\leq_f M_1$", therefore not acceptable

Right? Wrong!

3.2.4 Example 3.2.1 continued

Coffee & Tea Dispenser Inc.:

- "Only implementations correct according to $\leq_f$ are acceptable"
- "obviously $M_2 \not\leq_f M_1$", therefore not acceptable

Right? Wrong!

3.2.4 Example 3.2.1 continued

Coffee & Tea Dispenser Inc.:

- "Only implementations correct according to $\leq_f$ are acceptable"
- "obviously $M_2 \not\leq_f M_1$", therefore not acceptable

Right? Wrong!

3.2.4 Example 3.2.1 continued

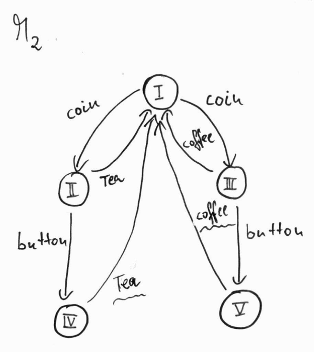
Coffee & Tea Dispenser Inc.:

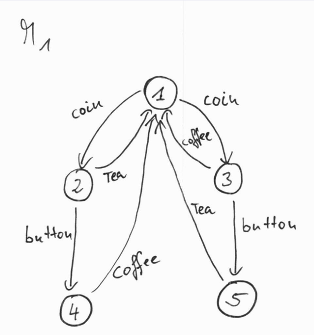
- "Only implementations correct according to $\leq_f$ are acceptable"
- "obviously $M_2 \not\leq_f M_1$", therefore not acceptable

Right? Wrong!

3.2.4 Example 3.2.1 continued

$M_2 \leq_f M_1$



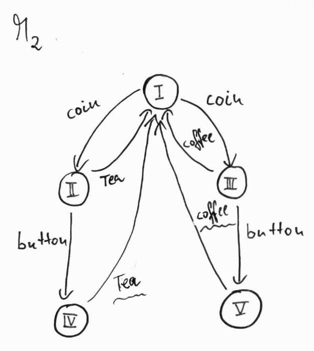


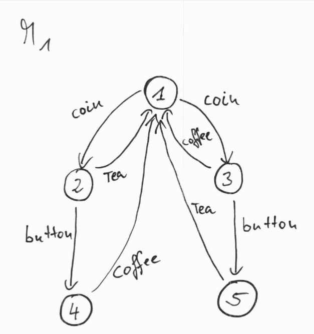
Failure pairs of M_2

COIN · {COIN, COFFEE}

3.2.4 Example 3.2.1 continued

$M_2 \leq_f M_1$



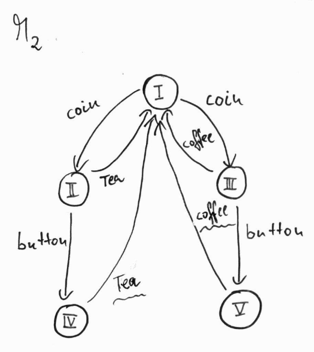


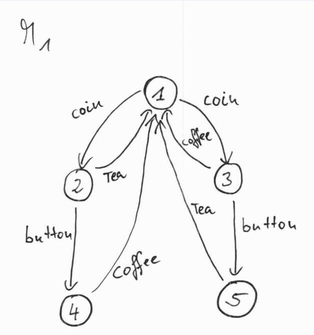
Failure pairs of M_2

$\text{COIN} \cdot \text{BUTTON} \cdot \widetilde{\{\text{COIN}, \text{COFFEE}\}}$

3.2.4 Example 3.2.1 continued

$M_2 \leq_f M_1$



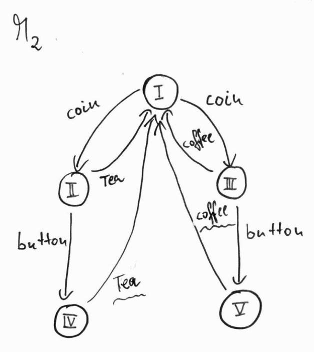


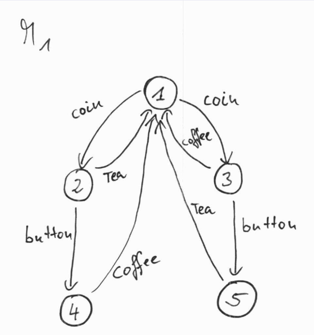
Failure pairs of M_2

$$\mathrm{COIN} \cdot \{\widetilde{\mathrm{COIN}, \mathrm{TEA}}\}$$

3.2.4 Example 3.2.1 continued

$M_2 \leq_f M_1$





Failure pairs of M_2

$\text{COIN} \cdot \text{BUTTON} \cdot \{\widetilde{\text{COIN}, \text{TEA}}\}$

3.2.4 Example 3.2.1 continued

- not only refusal sets of reached states important
- also what states with which refusals passed
- seems to be important to record also the refusals of intermediate states.

$\Longrightarrow$ failure traces

3.2.4 Example 3.2.1 continued

- not only refusal sets of reached states important
- also what states with which refusals passed
- seems to be important to record also the refusals of intermediate states.

$\Longrightarrow$ failure traces

3.2.4 Example 3.2.1 continued

- not only refusal sets of reached states important
- also what states with which refusals passed
- seems to be important to record also the refusals of intermediate states.

$\Longrightarrow$ failure traces

3.2.4 Example 3.2.1 continued

- not only refusal sets of reached states important
- also what states with which refusals passed
- seems to be important to record also the refusals of intermediate states.

$\Longrightarrow$ failure traces

Failure traces and failure trace preorder

Definition

1. Failure traces are the elements of Act^*_{ft}.
2. Set of failure traces of $p \in \mathbb{P}$:
   - $\varepsilon \in ftraces(p)$;
   - $a \in ftraces(p)$, if $p \xrightarrow{a}$;
   - $\widetilde{X} \in ftraces(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$;
   - $a \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $a \in Act$, if
     $$p \xrightarrow{a} p' \text{ and } \sigma \in ftraces(p');$$
   - $\widetilde{X} \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $\widetilde{X} \in R(Act)$, if
     $$\widetilde{X} \in ftraces(p) \text{ and } \sigma \in ftraces(p).$$
3. The failure trace preorder $\leq_{ft} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as
   $$p \leq_{ft} q \text{ iff } ftraces(p) \subseteq ftraces(q).$$

Failure traces and failure trace preorder

Definition

1. Failure traces are the elements of Act^*_{ft}.
2. Set of failure traces of $p \in \mathbb{P}$:
   - $\varepsilon \in ftraces(p)$;
   - $a \in ftraces(p)$, if $p \xrightarrow{a}$;
   - $\widetilde{X} \in ftraces(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$;
   - $a \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $a \in Act$, if
     $$p \xrightarrow{a} p' \text{ and } \sigma \in ftraces(p');$$
   - $\widetilde{X} \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $\widetilde{X} \in R(Act)$, if
     $$\widetilde{X} \in ftraces(p) \text{ and } \sigma \in ftraces(p).$$
3. The failure trace preorder $\leq_{ft} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as
   $$p \leq_{ft} q \text{ iff } ftraces(p) \subseteq ftraces(q).$$

Failure traces and failure trace preorder

Definition

1. **Failure traces** are the elements of Act^*_{ft}.
2. **Set of failure traces** of $p \in \mathbb{P}$:
   - $\varepsilon \in ftraces(p)$;
   - $a \in ftraces(p)$, if $p \xrightarrow{a}$;
   - $\widetilde{X} \in ftraces(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$;
   - $a \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $a \in Act$, if
     $$p \xrightarrow{a} p' \text{ and } \sigma \in ftraces(p');$$
   - $\widetilde{X} \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $\widetilde{X} \in R(Act)$, if
     $$\widetilde{X} \in ftraces(p) \text{ and } \sigma \in ftraces(p).$$
3. The **failure trace preorder** $\leq_{ft} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as
   $$p \leq_{ft} q \text{ iff } ftraces(p) \subseteq ftraces(q).$$

Failure traces and failure trace preorder

Definition

1. **Failure traces** are the elements of Act^*_{ft}.
2. **Set of failure traces** of $p \in \mathbb{P}$:
   - $\varepsilon \in ftraces(p)$;
   - $a \in ftraces(p)$, if $p \xrightarrow{a}$;
   - $\widetilde{X} \in ftraces(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$;
   - $a \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $a \in Act$, if
     $$p \xrightarrow{a} p' \text{ and } \sigma \in ftraces(p');$$
   - $\widetilde{X} \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $\widetilde{X} \in R(Act)$, if
     $$\widetilde{X} \in ftraces(p) \text{ and } \sigma \in ftraces(p).$$
3. The **failure trace preorder** $\leq_{ft} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as
   $$p \leq_{ft} q \text{ iff } ftraces(p) \subseteq ftraces(q).$$

Failure traces and failure trace preorder

Definition

1. Failure traces are the elements of Act^*_{ft}.
2. Set of failure traces of $p \in \mathbb{P}$:
   - $\varepsilon \in ftraces(p)$;
   - $a \in ftraces(p)$, if $p \xrightarrow{a}$;
   - $\widetilde{X} \in ftraces(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$;
   - $a \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $a \in Act$, if $$p \xrightarrow{a} p' \text{ and } \sigma \in ftraces(p');$$
   - $\widetilde{X} \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $\widetilde{X} \in R(Act)$, if $$\widetilde{X} \in ftraces(p) \text{ and } \sigma \in ftraces(p).$$
3. The failure trace preorder $\leq_{ft} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as $$p \leq_{ft} q \text{ iff } ftraces(p) \subseteq ftraces(q).$$

Failure traces and failure trace preorder

Definition

1. Failure traces are the elements of Act_{ft}^*.
2. Set of failure traces of $p \in \mathbb{P}$:
   - $\varepsilon \in ftraces(p)$;
   - $a \in ftraces(p)$, if $p \xrightarrow{a}$;
   - $\widetilde{X} \in ftraces(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$;
   - $a \cdot \sigma \in ftraces(p)$ for $\sigma \in Act_{ft}^*$, $a \in Act$, if

     $$p \xrightarrow{a} p' \text{ and } \sigma \in ftraces(p');$$

   - $\widetilde{X} \cdot \sigma \in ftraces(p)$ for $\sigma \in Act_{ft}^*$, $\widetilde{X} \in R(Act)$, if

     $$\widetilde{X} \in ftraces(p) \text{ and } \sigma \in ftraces(p).$$

3. The failure trace preorder $\leq_{ft} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

   $$p \leq_{ft} q \text{ iff } ftraces(p) \subseteq ftraces(q).$$

Failure traces and failure trace preorder

Definition

1. Failure traces are the elements of Act^*_{ft}.
2. Set of failure traces of $p \in \mathbb{P}$:
   - $\varepsilon \in ftraces(p)$;
   - $a \in ftraces(p)$, if $p \xrightarrow{a}$;
   - $\widetilde{X} \in ftraces(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$;
   - $a \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $a \in Act$, if
     $p \xrightarrow{a} p'$ and $\sigma \in ftraces(p')$;
   - $\widetilde{X} \cdot \sigma \in ftraces(p)$ for $\sigma \in Act^*_{ft}$, $\widetilde{X} \in R(Act)$, if
     $\widetilde{X} \in ftraces(p)$ and $\sigma \in ftraces(p)$.
3. The failure trace preorder $\leq_{ft} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as
   $$p \leq_{ft} q \text{ iff } ftraces(p) \subseteq ftraces(q).$$

Failure traces and failure trace preorder

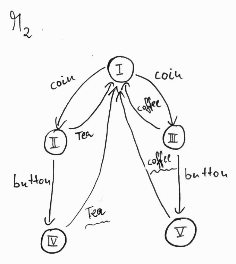
Definition

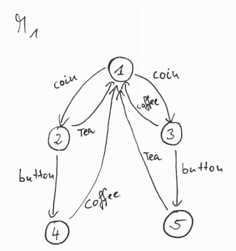
1. Failure traces are the elements of Act_{ft}^{*}.
2. Set of failure traces of $p \in \mathbb{P}$:
   - $\varepsilon \in ftraces(p)$;
   - $a \in ftraces(p)$, if $p \xrightarrow{a}$;
   - $\widetilde{X} \in ftraces(p)$, if $p \not\xrightarrow{a}$ for all $a \in X$;
   - $a \cdot \sigma \in ftraces(p)$ for $\sigma \in Act_{ft}^{*}$, $a \in Act$, if
     $p \xrightarrow{a} p'$ and $\sigma \in ftraces(p')$;
   - $\widetilde{X} \cdot \sigma \in ftraces(p)$ for $\sigma \in Act_{ft}^{*}$, $\widetilde{X} \in R(Act)$, if
     $\widetilde{X} \in ftraces(p)$ and $\sigma \in ftraces(p)$.
3. The failure trace preorder $\leq_{ft} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

$$p \leq_{ft} q \text{ iff } ftraces(p) \subseteq ftraces(q).$$

Failure traces and failure trace preorder

3.2.6 Example, 3.2.1 continued (2)

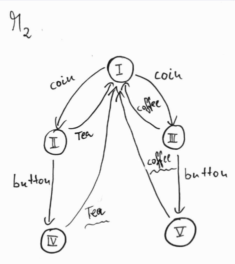


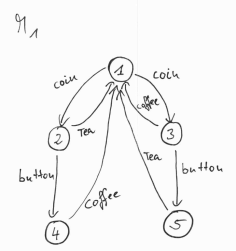


M_2 is not an implementation of M_1 according to $\leq_{ft}$!

Failure traces and failure trace preorder

3.2.6 Example, 3.2.1 continued (2)

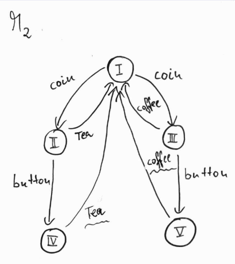


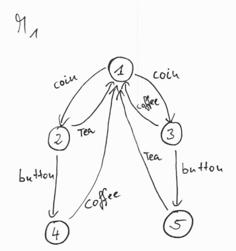


- $\sigma = \text{COIN}\{\widetilde{\text{COFFEE}, \text{COIN}}\}\text{BUTTON}\{\widetilde{\text{TEA}, \text{COIN}}\} \in \mathit{ftraces}(1)$
- σ not an element of $\mathit{ftraces}(\text{I})$:
  - $1 \xrightarrow{\text{COIN}\{\widetilde{\text{COFFEE},\text{COIN}}\}\text{BUTTON}} 4$
  - $I \xrightarrow{\text{COIN}\{\widetilde{\text{COFFEE},\text{COIN}}\}\text{BUTTON}} IV$
  - $\{\widetilde{\text{TEA}, \text{COIN}}\} \in \mathit{ftraces}(4)$
  - $\{\widetilde{\text{TEA}, \text{COIN}}\} \notin \mathit{ftraces}(IV)$

Failure traces and failure trace preorder

3.2.6 Example, 3.2.1 continued (2)

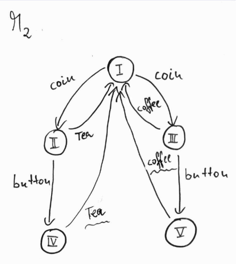


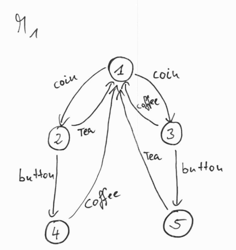


- $\sigma = \text{COIN}\widetilde{\{\text{COFFEE}, \text{COIN}\}}\text{BUTTON}\widetilde{\{\text{TEA}, \text{COIN}\}} \in \mathit{ftraces}(1)$
- σ not an element of $\mathit{ftraces}(\text{I})$:
  - $1 \xrightarrow{\text{COIN}\widetilde{\{\text{COFFEE},\text{COIN}\}}\text{BUTTON}} 4$
  - $I \xrightarrow{\text{COIN}\widetilde{\{\text{COFFEE},\text{COIN}\}}\text{BUTTON}} IV$
  - $\widetilde{\{\text{TEA}, \text{COIN}\}} \in \mathit{ftraces}(4)$
  - $\widetilde{\{\text{TEA}, \text{COIN}\}} \notin \mathit{ftraces}(IV)$

Failure traces and failure trace preorder

3.2.6 Example, 3.2.1 continued (2)

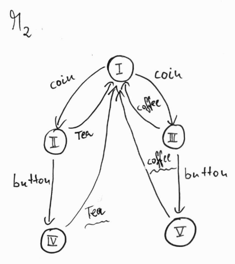


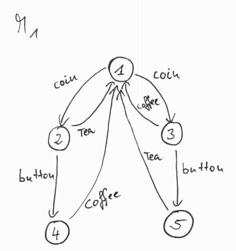


- $\sigma = \text{COIN}\{\widetilde{\text{COFFEE}, \text{COIN}}\}\text{BUTTON}\{\widetilde{\text{TEA}, \text{COIN}}\} \in \mathit{ftraces}(1)$
- σ not an element of $\mathit{ftraces}(\text{I})$:
  - $1 \xrightarrow{\text{COIN}\{\widetilde{\text{COFFEE},\text{COIN}}\}\text{BUTTON}} 4$
  - $I \xrightarrow{\text{COIN}\{\widetilde{\text{COFFEE},\text{COIN}}\}\text{BUTTON}} IV$
  - $\{\widetilde{\text{TEA}, \text{COIN}}\} \in \mathit{ftraces}(4)$
  - $\{\widetilde{\text{TEA}, \text{COIN}}\} \notin \mathit{ftraces}(IV)$

Failure traces and failure trace preorder

3.2.6 Example, 3.2.1 continued (2)

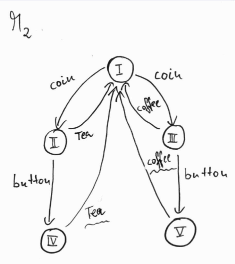


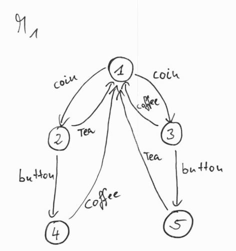


- $\sigma = \text{COIN}\widetilde{\{\text{COFFEE}, \text{COIN}\}}\text{BUTTON}\widetilde{\{\text{TEA}, \text{COIN}\}} \in \textit{ftraces}(1)$
- σ not an element of $\textit{ftraces}(\text{I})$:
  - $1 \xrightarrow{\text{COIN}\widetilde{\{\text{COFFEE},\text{COIN}\}}\text{BUTTON}} 4$
  - $I \xrightarrow{\text{COIN}\widetilde{\{\text{COFFEE},\text{COIN}\}}\text{BUTTON}} IV$
  - $\widetilde{\{\text{TEA}, \text{COIN}\}} \in \textit{ftraces}(4)$
  - $\widetilde{\{\text{TEA}, \text{COIN}\}} \notin \textit{ftraces}(IV)$

Failure traces and failure trace preorder

3.2.6 Example, 3.2.1 continued (2)

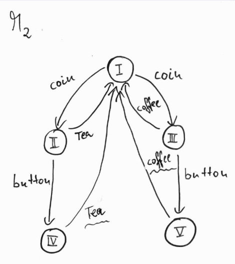


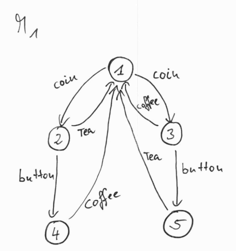


- $\sigma = \text{COIN}\{\widetilde{\text{COFFEE}, \text{COIN}}\}\text{BUTTON}\{\widetilde{\text{TEA}, \text{COIN}}\} \in \mathit{ftraces}(1)$
- σ not an element of $\mathit{ftraces}(\mathrm{I})$:
  - $1 \xrightarrow{\text{COIN}\{\widetilde{\text{COFFEE},\text{COIN}}\}\text{BUTTON}} 4$
  - $I \xrightarrow{\text{COIN}\{\widetilde{\text{COFFEE},\text{COIN}}\}\text{BUTTON}} IV$
  - $\{\widetilde{\text{TEA}, \text{COIN}}\} \in \mathit{ftraces}(4)$
  - $\{\widetilde{\text{TEA}, \text{COIN}}\} \notin \mathit{ftraces}(IV)$

Failure traces and failure trace preorder

3.2.6 Example, 3.2.1 continued (2)





- $\sigma = \text{COIN}\widetilde{\{\text{COFFEE}, \text{COIN}\}}\text{BUTTON}\widetilde{\{\text{TEA}, \text{COIN}\}} \in \mathit{ftraces}(1)$
- σ not an element of $\mathit{ftraces}(\text{I})$:
  - $1 \xrightarrow{\text{COIN}\widetilde{\{\text{COFFEE},\text{COIN}\}}\text{BUTTON}} 4$
  - $I \xrightarrow{\text{COIN}\widetilde{\{\text{COFFEE},\text{COIN}\}}\text{BUTTON}} IV$
  - $\widetilde{\{\text{TEA}, \text{COIN}\}} \in \mathit{ftraces}(4)$
  - $\widetilde{\{\text{TEA}, \text{COIN}\}} \notin \mathit{ftraces}(IV)$

Comparing the Implementation relations

3.2.7 Proposition

$\leq_{ft} \quad \preceq \quad \leq_{f} \quad \preceq \quad \leq_{tr}$

Proof: $\Longrightarrow$ Blackboard

Completed trace preorder

3.2.8 Definition $\leq_{ct}$

For $p, q \in \mathbb{P}$, the completed trace preorder $\leq_{ct} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

$p \leq_{ct} q$ iff

$$traces(p) \subseteq traces(q)$$

and

$$F(p) \cap Act^* \cdot \widetilde{Act} \subseteq F(q) \cap Act^* \cdot \widetilde{Act}$$

Note

- If $\widetilde{Act} \in F(p)$, then p is deadlocked.
- If $\sigma \cdot \widetilde{Act} \in F(p)$, then σ is called a completed trace of p.
- Apparently, $\leq_f \preceq \leq_{ct} \preceq \leq_{tr}$.
- $\leq_{ct}$ is finer than $\leq_{tr}$.

Completed trace preorder

3.2.8 Definition $\leq_{ct}$

For $p, q \in \mathbb{P}$, the completed trace preorder $\leq_{ct} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

$p \leq_{ct} q$ iff

$$traces(p) \subseteq traces(q)$$

and

$$F(p) \cap Act^* \cdot \widetilde{Act} \subseteq F(q) \cap Act^* \cdot \widetilde{Act}$$

Note

- If $\widetilde{Act} \in F(p)$, then p is deadlocked.
- If $\sigma \cdot \widetilde{Act} \in F(p)$, then σ is called a completed trace of p.
- Apparently, $\leq_f \preceq \leq_{ct} \preceq \leq_{tr}$.
- $\leq_{ct}$ is finer than $\leq_{tr}$.

Completed trace preorder

3.2.8 Definition $\leq_{ct}$

For $p, q \in \mathbb{P}$, the completed trace preorder $\leq_{ct} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

$p \leq_{ct} q$ iff

$$traces(p) \subseteq traces(q)$$

and

$$F(p) \cap Act^* \cdot \widetilde{Act} \subseteq F(q) \cap Act^* \cdot \widetilde{Act}$$

Note

- If $\widetilde{Act} \in F(p)$, then p is deadlocked.
- If $\sigma \cdot \widetilde{Act} \in F(p)$, then σ is called a completed trace of p.
- Apparently, $\leq_f \preceq \leq_{ct} \preceq \leq_{tr}$.
- $\leq_{ct}$ is finer than $\leq_{tr}$.

Completed trace preorder

3.2.8 Definition $\leq_{ct}$

For $p, q \in \mathbb{P}$, the completed trace preorder $\leq_{ct} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

$p \leq_{ct} q$ iff

$$traces(p) \subseteq traces(q)$$

and

$$F(p) \cap Act^* \cdot \widetilde{Act} \subseteq F(q) \cap Act^* \cdot \widetilde{Act}$$

Note

- If $\widetilde{Act} \in F(p)$, then p is deadlocked.
- If $\sigma \cdot \widetilde{Act} \in F(p)$, then σ is called a completed trace of p.
- Apparently, $\leq_f \preceq \leq_{ct} \preceq \leq_{tr}$.
- $\leq_{ct}$ is finer than $\leq_{tr}$.

Completed trace preorder

3.2.8 Definition $\leq_{ct}$

For $p, q \in \mathbb{P}$, the completed trace preorder $\leq_{ct} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

$p \leq_{ct} q$ iff

$$traces(p) \subseteq traces(q)$$

and

$$F(p) \cap Act^* \cdot \widetilde{Act} \subseteq F(q) \cap Act^* \cdot \widetilde{Act}$$

Note

- If $\widetilde{Act} \in F(p)$, then p is deadlocked.
- If $\sigma \cdot \widetilde{Act} \in F(p)$, then σ is called a completed trace of p.
- Apparently, $\leq_f \preceq \leq_{ct} \preceq \leq_{tr}$.
- $\leq_{ct}$ is finer than $\leq_{tr}$.

Completed trace preorder

3.2.8 Definition $\leq_{ct}$

For $p, q \in \mathbb{P}$, the completed trace preorder $\leq_{ct} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

$p \leq_{ct} q$ iff

$$traces(p) \subseteq traces(q)$$

and

$$F(p) \cap Act^* \cdot \widetilde{Act} \subseteq F(q) \cap Act^* \cdot \widetilde{Act}$$

Note

- If $\widetilde{Act} \in F(p)$, then p is deadlocked.
- If $\sigma \cdot \widetilde{Act} \in F(p)$, then σ is called a completed trace of p.
- Apparently, $\leq_f \preceq \leq_{ct} \preceq \leq_{tr}$.
- $\leq_{ct}$ is finer than $\leq_{tr}$.

Completed trace preorder

3.2.8 Definition $\leq_{ct}$

For $p, q \in \mathbb{P}$, the completed trace preorder $\leq_{ct} \subseteq \mathbb{P} \times \mathbb{P}$ is defined as

$p \leq_{ct} q$ iff

$$traces(p) \subseteq traces(q)$$

and

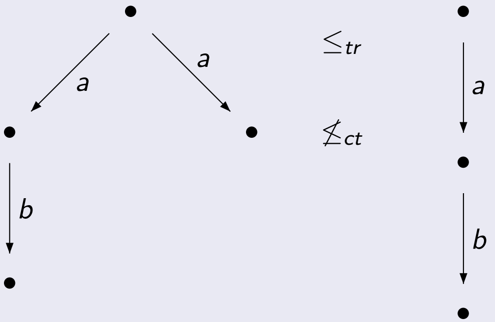
$$F(p) \cap Act^* \cdot \widetilde{Act} \subseteq F(q) \cap Act^* \cdot \widetilde{Act}$$

Note

- If $\widetilde{Act} \in F(p)$, then p is deadlocked.
- If $\sigma \cdot \widetilde{Act} \in F(p)$, then σ is called a completed trace of p.
- Apparently, $\leq_f \preceq \leq_{ct} \preceq \leq_{tr}$.
- $\leq_{ct}$ is finer than $\leq_{tr}$.

$\leq_{ct}$ is finer than $\leq_{tr}$

Example



1. Some more Implementation Relations
2. Failure Trace Preorder and Observers

Test Expressions for $\leq_{ft}$

Refusals become part of test expressions

Test Expressions $\mathcal{O}_{ft}$

$$o \longrightarrow a.o \mid \text{t.STOP} \mid \widetilde{X}.o$$

with $a \in Act$, and $\widetilde{X} \in R(Act)$.

Test Expressions for $\leq_{ft}$

Refusals become part of test expressions

Test Expressions $\mathcal{O}_{ft}$

$$o \longrightarrow a.o \mid \mathrm{t.STOP} \mid \widetilde{X}.o$$

with $a \in Act$, and $\widetilde{X} \in R(Act)$.

Test Expressions for $\leq_{ft}$

Refusals become part of test expressions

Test Expressions $\mathcal{O}_{ft}$

$$o \longrightarrow a.o \mid \mathrm{t.STOP} \mid \widetilde{X}.o$$

with $a \in Act$, and $\widetilde{X} \in R(Act)$.

Test Execution

Test Execution for $\leq_{ft}$

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{ft}) \qquad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \mathbin{\top\!\!\top} p \xrightarrow{a} o' \mathbin{\top\!\!\top} p'}(a \in Act) \qquad \frac{o \xrightarrow{t} o'}{o \mathbin{\top\!\!\top} p \xrightarrow{t} \text{PASS}}$$

$$\frac{o \xrightarrow{\widetilde{X}} o' \quad p \not\xrightarrow{a} \forall a \in X}{o \mathbin{\top\!\!\top} p \xrightarrow{\widetilde{X}} o' \mathbin{\top\!\!\top} p}$$

$$\text{obs}_{ft}(p) = \{\, o \in \mathcal{O}_{ft} \mid \exists \sigma \in Act^*_{ft} : o \mathbin{\top\!\!\top} p \xrightarrow{\sigma \cdot t} \text{PASS} \,\}$$

Test Execution

Test Execution for $\leq_{ft}$

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{ft}) \qquad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \Tint p \xrightarrow{a} o' \Tint p'}(a \in Act) \qquad \frac{o \xrightarrow{t} o'}{o \Tint p \xrightarrow{t} \text{PASS}}$$

$$\frac{o \xrightarrow{\widetilde{X}} o' \quad p \not\xrightarrow{a} \forall a \in X}{o \Tint p \xrightarrow{\widetilde{X}} o' \Tint p}$$

$$obs_{ft}(p) = \{\, o \in \mathcal{O}_{ft} \mid \exists \sigma \in Act_{ft}^{*} : o \Tint p \xRightarrow{\sigma \cdot t} \text{PASS} \,\}$$

Test Execution

Test Execution for $\leq_{ft}$

$$\frac{}{a.o \xrightarrow{a} o}(a \in Act_{ft}) \qquad \frac{o \xrightarrow{a} o' \quad p \xrightarrow{a} p'}{o \mathbin{⫪} p \xrightarrow{a} o' \mathbin{⫪} p'}(a \in Act) \qquad \frac{o \xrightarrow{t} o'}{o \mathbin{⫪} p \xrightarrow{t} \mathrm{PASS}}$$

$$\frac{o \xrightarrow{\widetilde{X}} o' \quad p \not\xrightarrow{a} \forall a \in X}{o \mathbin{⫪} p \xrightarrow{\widetilde{X}} o' \mathbin{⫪} p}$$

$$\mathrm{obs}_{ft}(p) = \{\ o \in \mathcal{O}_{ft} \mid \exists \sigma \in Act^*_{ft} : o \mathbin{⫪} p \xrightarrow{\sigma \cdot t} \mathrm{PASS}\ \}$$

Observing Failure Trace Preorder

3.2.10 Lemma

$p \leq_{ft} q$ iff $\mathsf{obs}_{ft}(p) \subseteq \mathsf{obs}_{ft}(q)$

Proof: blackboard