Testing of Reactive Systems

Lecture 4: Implementation Relations, Observing Behaviour

Henrik Bohnenkamp

Lehrstuhl Informatik 2 (MOVES)
RWTH Aachen

Summer Semester 2009

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p'Rq'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p'Rq'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p'Rq'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p'Rq'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p'Rq'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p'Rq'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p'Rq'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p'Rq'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p'Rq'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p'Rq'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p' R q'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p' R q'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p'Rq'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p'Rq'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p'Rq'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p'Rq'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

What happened so far?

- SOS of processes
- Preorders, Equivalences
- Trace preorder
- Bisimulation equivalence

Definition Bisimulation Equivalence

A bisimulation is a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying:
For all $a \in Act$:

1) if pRq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{a} q'$ and $p'Rq'$.

2) if pRq and $q \xrightarrow{a} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{a} p'$ and $p'Rq'$.

We say p is bisimulation equivalent to q ($\sim_B$) if there is a bisimulation R such that pRq.

1. Some Implementation Relations (cont)

2. A little surprise: Determinism and the LTBT spectrum

3. Introduction Part 3

Lifting bisimulation to $\xrightarrow{\sigma}$

Lemma 2.2.5

R is a bisimulation iff:

For all $\sigma \in Act^*$:

1) if pRq and $p \xrightarrow{\sigma} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{\sigma} q'$ and $p' R\, q'$.

2) if pRq and $q \xrightarrow{\sigma} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{\sigma} p'$ and $p' R\, q'$.

Proof:

$\Longrightarrow$ Blackboard

Lifting bisimulation to $\xrightarrow{\sigma}$

Lemma 2.2.5

R is a bisimulation iff:
For all $\sigma \in Act^*$:

1) if pRq and $p \xrightarrow{\sigma} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{\sigma} q'$ and $p' R\, q'$.

2) if pRq and $q \xrightarrow{\sigma} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{\sigma} p'$ and $p' R\, q'$.

Proof:

$\Longrightarrow$ Blackboard

Lifting bisimulation to $\xrightarrow{\sigma}$

Lemma 2.2.5

R is a bisimulation iff:
For all $\sigma \in Act^*$:

1) if pRq and $p \xrightarrow{\sigma} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{\sigma} q'$ and $p' R\, q'$.

2) if pRq and $q \xrightarrow{\sigma} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{\sigma} p'$ and $p' R\, q'$.

Proof:

$\Longrightarrow$ Blackboard

Lifting bisimulation to $\xrightarrow{\sigma}$

Lemma 2.2.5

R is a bisimulation iff:
For all $\sigma \in Act^*$:

1) if pRq and $p \xrightarrow{\sigma} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{\sigma} q'$ and $p' R\, q'$.

2) if pRq and $q \xrightarrow{\sigma} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{\sigma} p'$ and $p' R\, q'$.

Proof:

$\Longrightarrow$ Blackboard

Lifting bisimulation to $\xrightarrow{\sigma}$

Lemma 2.2.5

R is a bisimulation iff:
For all $\sigma \in Act^*$:

1) if pRq and $p \xrightarrow{\sigma} p'$, then $\exists q' \in \mathbb{P} : q \xrightarrow{\sigma} q'$ and $p' R\, q'$.

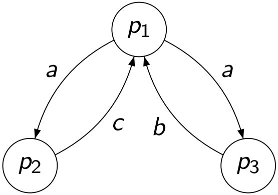
2) if pRq and $q \xrightarrow{\sigma} q'$, then $\exists p' \in \mathbb{P} : p \xrightarrow{\sigma} p'$ and $p' R\, q'$.

Proof:

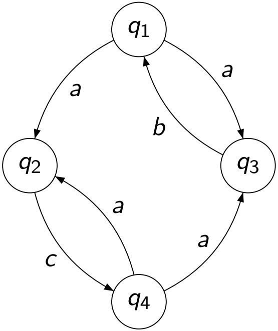
$\Longrightarrow$ Blackboard

Example 2.2.6

L_1 :



L_2 :



Comparing preorders

- A preorder is a set
- We can relate different preorders by set inclusion.

Definition 2.2.7 (Finer and Coarser)

Let $\leq$, $\leq'$ be preorders and $\leq \subseteq \leq'$.
We say

1. $\leq$ is finer than $\leq'$ (written $\leq \preceq \leq'$),
2. $\leq'$ is coarser than $\leq$.

Analogously for equivalences.

Comparing preorders

- A preorder is a set
- We can relate different preorders by set inclusion.

Definition 2.2.7 (Finer and Coarser)

Let $\leq$, $\leq'$ be preorders and $\leq \subseteq \leq'$.

We say

1. $\leq$ is finer than $\leq'$ (written $\leq \preceq \leq'$),
2. $\leq'$ is coarser than $\leq$.

Analogously for equivalences.

Comparing preorders

- A preorder is a set
- We can relate different preorders by set inclusion.

Definition 2.2.7 (Finer and Coarser)

Let $\leq$, $\leq'$ be preorders and $\leq \subseteq \leq'$.
We say

1. $\leq$ is finer than $\leq'$ (written $\leq \preceq \leq'$),
2. $\leq'$ is coarser than $\leq$.

Analogously for equivalences.

Comparing preorders

- A preorder is a set
- We can relate different preorders by set inclusion.

Definition 2.2.7 (Finer and Coarser)

Let $\leq$, $\leq'$ be preorders and $\leq \subseteq \leq'$.
We say

1. $\leq$ is finer than $\leq'$ (written $\leq \preceq \leq'$),
2. $\leq'$ is coarser than $\leq$.

Analogously for equivalences.

Comparing preorders

- A preorder is a set
- We can relate different preorders by set inclusion.

Definition 2.2.7 (Finer and Coarser)

Let $\leq$, $\leq'$ be preorders and $\leq \subseteq \leq'$.
We say

1. $\leq$ is finer than $\leq'$ (written $\leq \preceq \leq'$),
2. $\leq'$ is coarser than $\leq$.

Analogously for equivalences.

Finer and Coarser

Note:

Finer preorder:

1. is more "picky"
2. distinguishes more processes from each other
3. equivalences: the partitioning is finer
4. $\leq$ finer than $\leq'$ means:

$$p \leq q \quad \text{implies} \quad p \leq' q$$

5. We write $pre_1 \preceq pre_2$ iff pre_1 is finer than pre_2.

Question:

How are trace equivalence and bisimulation equivalence related?

Finer and Coarser

Note:

Finer preorder:

1. is more "picky"
2. distinguishes more processes from each other
3. equivalences: the partitioning is finer
4. $\leq$ finer than $\leq'$ means:

$$p \leq q \quad \text{implies} \quad p \leq' q$$

5. We write $pre_1 \preceq pre_2$ iff pre_1 is finer than pre_2.

Question:

How are trace equivalence and bisimulation equivalence related?

Finer and Coarser

Note:

Finer preorder:

1. is more "picky"
2. distinguishes more processes from each other
3. equivalences: the partitioning is finer
4. $\leq$ finer than $\leq'$ means:

$$p \leq q \quad \text{implies} \quad p \leq' q$$

5. We write $pre_1 \preceq pre_2$ iff pre_1 is finer than pre_2.

Question:

How are trace equivalence and bisimulation equivalence related?

Finer and Coarser

Note:

Finer preorder:

1. is more "picky"
2. distinguishes more processes from each other
3. equivalences: the partitioning is finer
4. $\leq$ finer than $\leq'$ means:

$$p \leq q \quad \text{implies} \quad p \leq' q$$

5. We write $pre_1 \preceq pre_2$ iff pre_1 is finer than pre_2.

Question:

How are trace equivalence and bisimulation equivalence related?

Finer and Coarser

Note:

Finer preorder:

1. is more "picky"
2. distinguishes more processes from each other
3. equivalences: the partitioning is finer
4. $\leq$ finer than $\leq'$ means:

$$p \leq q \quad \text{implies} \quad p \leq' q$$

5. We write $pre_1 \preceq pre_2$ iff pre_1 is finer than pre_2.

Question:

How are trace equivalence and bisimulation equivalence related?

Finer and Coarser

Note:

Finer preorder:

1. is more "picky"
2. distinguishes more processes from each other
3. equivalences: the partitioning is finer
4. $\leq$ finer than $\leq'$ means:

$$p \leq q \quad \text{implies} \quad p \leq' q$$

5. We write $pre_1 \preceq pre_2$ iff pre_1 is finer than pre_2.

Question:

How are trace equivalence and bisimulation equivalence related?

Comparing Trace and Bisimulation Equivalence

Proposition 2.2.8

$\sim_{tr}$ is coarser than $\sim_B$, but not finer.

Proof:

$\Longrightarrow$ Blackboard

Comparing Trace and Bisimulation Equivalence

Proposition 2.2.8

$\sim_{tr}$ is coarser than $\sim_B$, but not finer.

Proof:

$\Longrightarrow$ Blackboard

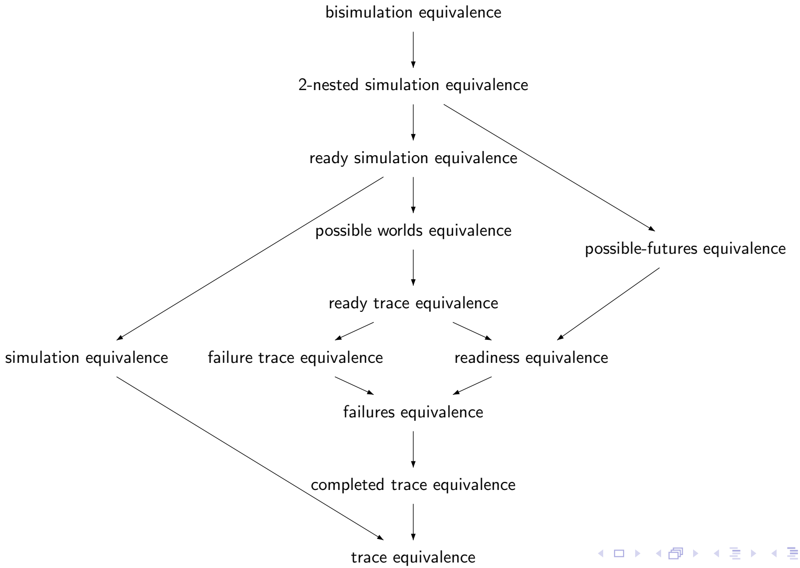
The picture so far

bisimulation equivalence

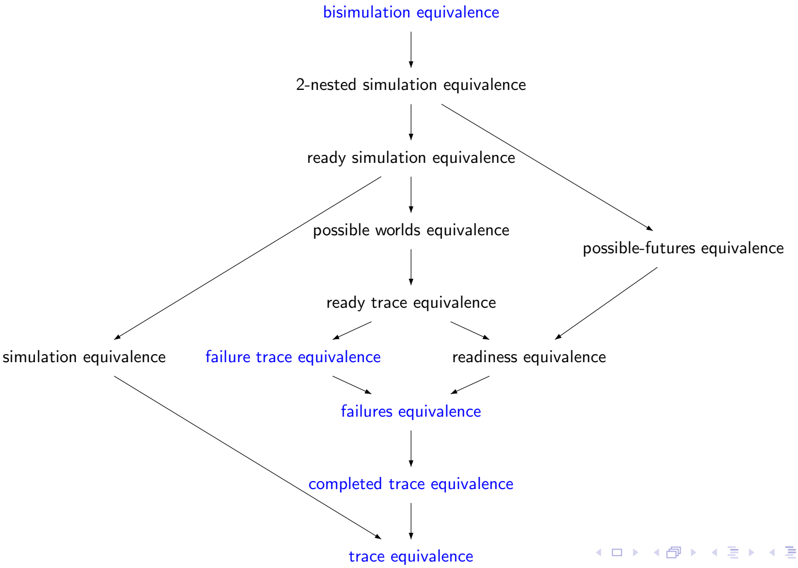
$\downarrow$ $\subsetneq$

trace equivalence

The whole picture: linear time – branching time spectrum



The whole picture: linear time – branching time spectrum



1. Some Implementation Relations (cont)
2. A little surprise: Determinism and the LTBT spectrum
3. Introduction Part 3

Determinism and the LTBT spectrum

Let $\mathbb{P}_{\text{det}}$ be the set of all deterministic processes.

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\text{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Restricted means, we look at:

- $\sim_B \ \cap\ \mathbb{P}_{\text{det}} \times \mathbb{P}_{\text{det}}$
- $\sim_{tr} \ \cap\ \mathbb{P}_{\text{det}} \times \mathbb{P}_{\text{det}}$

Determinism and the LTBT spectrum

Let $\mathbb{P}_{\text{det}}$ be the set of all deterministic processes.

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\text{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Restricted means, we look at:

- $\sim_B \ \cap \mathbb{P}_{\text{det}} \times \mathbb{P}_{\text{det}}$
- $\sim_{tr} \ \cap \mathbb{P}_{\text{det}} \times \mathbb{P}_{\text{det}}$

Determinism and the LTBT spectrum

Let $\mathbb{P}_{\mathrm{det}}$ be the set of all deterministic processes.

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\mathrm{det}}$, then

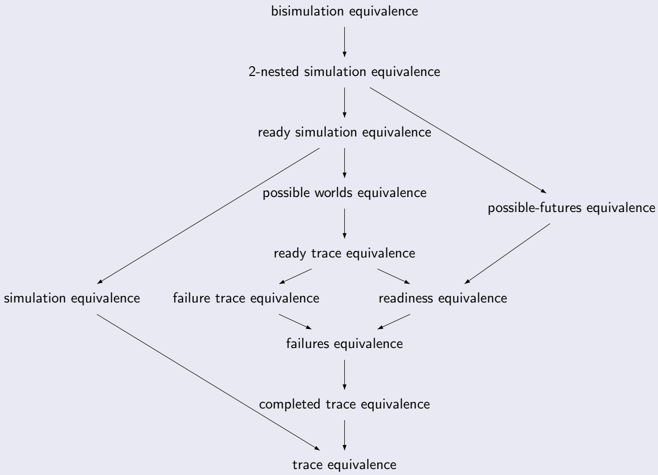
$$\sim_B \quad = \quad \sim_{tr}$$

Restricted means, we look at:

- $\sim_B \;\cap\; \mathbb{P}_{\mathrm{det}} \times \mathbb{P}_{\mathrm{det}}$
- $\sim_{tr} \;\cap\; \mathbb{P}_{\mathrm{det}} \times \mathbb{P}_{\mathrm{det}}$

Implications

Collapse of the linear time – branching time spectrum



Implications

Collapse of the linear time – branching time spectrum

$\sim$

bisimulation equivalence
2-nested simulation equivalence
possible-futures equivalence
ready simulation equivalence
possible worlds equivalence
simulation equivalence
ready trace equivalence
readiness equivalence
failure trace equivalence
failures equivalence
completed trace equivalence
trace equivalence

Implications

Differences between equivalences

The equivalences differ in the way how they treat nondeterminism in processes

Implications

Position in the diagram

. . . is an indication on how much nondeterminism is taken into account to distinguish processes

Implications

Trace equivalence

...ignores non-determinism completely

Bisimulation equivalence

...takes most of the information on nondeterminism into account

Implications

Trace equivalence

. . . ignores non-determinism completely

Bisimulation equivalence

. . . takes most of the information on nondeterminism into account

Implications

Why linear-time - branching-time?

- trace equivalence works on traces, linear sequences of actions $\Longrightarrow$ "linear time"
- bisimulation takes branching, esp. nondeterministic branching into account $\Longrightarrow$ "branching time"

Studying preorders and implementation relations on processes is thus studying nondeterminism

Implications

Why linear-time - branching-time?

- trace equivalence works on traces, linear sequences of actions $\Longrightarrow$ “linear time”
- bisimulation takes branching, esp. nondeterministic branching into account $\Longrightarrow$ “branching time”

Studying preorders and implementation relations on processes is thus studying nondeterminism

Implications

Why linear-time - branching-time?

- trace equivalence works on traces, linear sequences of actions $\Longrightarrow$ "linear time"
- bisimulation takes branching, esp. nondeterministic branching into account $\Longrightarrow$ "branching time"

Studying preorders and implementation relations on processes is thus studying nondeterminism

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\mathrm{det}}$, then

$$\sim_B \; = \; \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\mathrm{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p \; R \; q$.
   This implies $p \; \sim_B \; q$

$\Longrightarrow$ Blackboard

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\text{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\text{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p \; R \; q$.
   This implies $p \;\; \sim_B \;\; q$

$\Longrightarrow$ Blackboard

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\text{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\text{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p \; R \; q$.
   This implies $p \sim_B q$

$\Longrightarrow$ Blackboard

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\mathrm{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\mathrm{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p \; R \; q$.
   This implies $p \; \sim_B \; q$

$\Longrightarrow$ Blackboard

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\mathrm{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\mathrm{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p \; R \; q$.
   This implies $p \sim_B q$

$\Longrightarrow$ Blackboard

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\mathrm{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\mathrm{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p\ R\ q$.
   This implies $p \sim_B q$

$\Longrightarrow$ Blackboard

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\mathrm{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\mathrm{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p \; R \; q$.

   This implies $p \quad \sim_B \quad q$

Blackboard

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\text{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\text{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p \; R \; q$.
   This implies $p \;\sim_B\; q$

Blackboard

Proof of Proposition 2.3.1

Proposition 2.3.1

If we restrict $\sim_B$ and $\sim_{tr}$ on $\mathbb{P}_{\text{det}}$, then

$$\sim_B \quad = \quad \sim_{tr}$$

Proof idea

To show: 1) $\sim_B \subseteq \sim_{tr}$ and 2) $\sim_{tr} \subseteq \sim_B$

1) is already clear, since $\sim_B$ is finer than $\sim_{tr}$
2) We assume
   - $p, q \in \mathbb{P}_{\text{det}}$
   - $p \sim_{tr} q$

   Approach: construct a bisimulation R such that $p \; R \; q$.
   This implies $p \; \sim_B \; q$

$\Longrightarrow$ Blackboard

Summary Part 2

- Preorders, equivalences
- Trace inclusion
- Bisimulation equivalence
- Comparison of preorders, the LTBT spectrum
- The collapse of the LTBT spectrum for deterministic processes

Part 3: Distinguishing Processes by Manipulation and Observation

Remember

Even though it does not feel this way yet. . .

. . . this lecture is about testing!

Remember

Even though it does not feel this way yet. . .

. . . this lecture is about testing!

Next step

How can we relate *implementation relations* to testing?

Testing has much of experimenting:

- test-object is manipulated
- reactions are observed

How to compare the behaviour of processes (our test-objects) by manipulation and observation?

Next step

How can we relate *implementation relations* to testing?

Testing has much of experimenting:

- test-object is manipulated
- reactions are observed

How to compare the behaviour of processes (our test-objects) by manipulation and observation?

Next step

How can we relate *implementation relations* to testing?

Testing has much of experimenting:

- test-object is manipulated
- reactions are observed

How to compare the behaviour of processes (our test-objects) by manipulation and observation?

Next step

How can we relate *implementation relations* to testing?

Testing has much of experimenting:

- test-object is manipulated
- reactions are observed

How to compare the behaviour of processes (our test-objects) by manipulation and observation?

Next step

How can we relate *implementation relations* to testing?

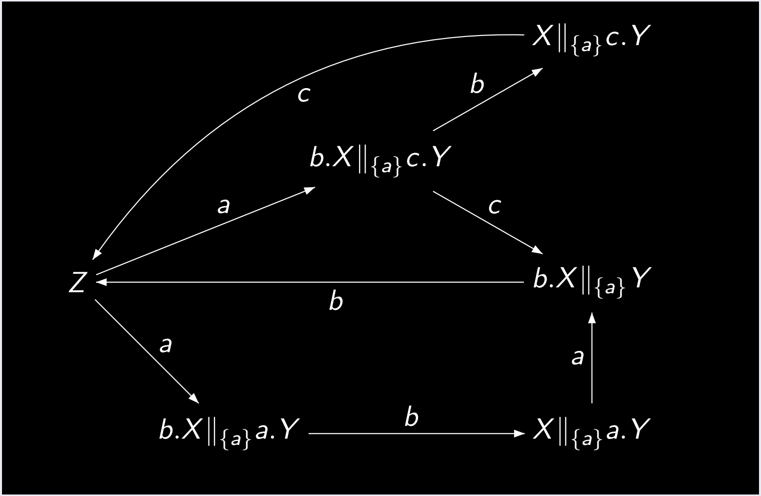
Testing has much of experimenting:

- test-object is manipulated
- reactions are observed

How to compare the behaviour of processes (our test-objects) by manipulation and observation?

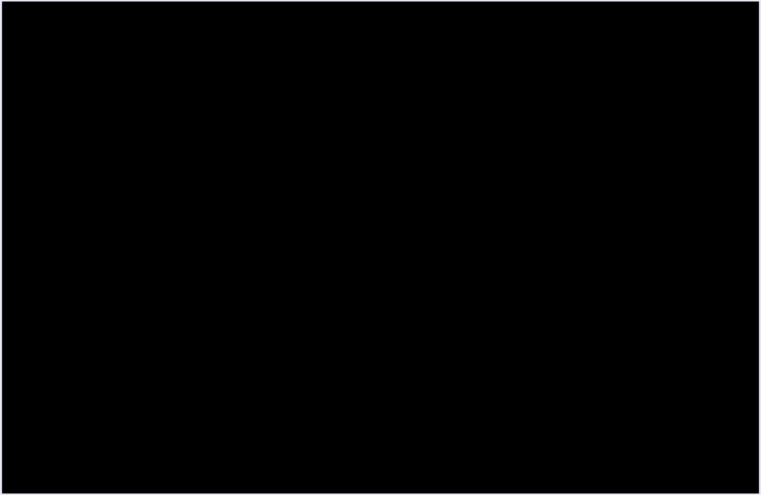
Framework

Process in black-box



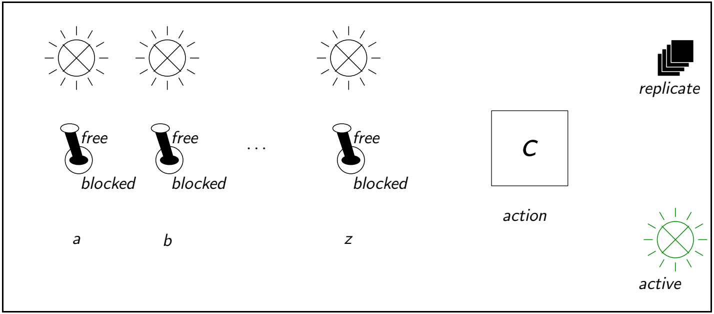
Framework

Inner structure not visible



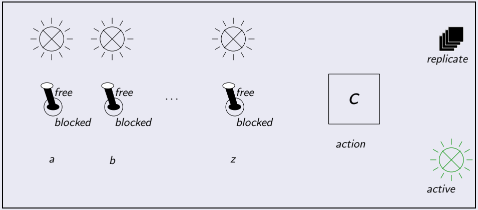
Framework

Manipulating and Observing Processes: the User Panel



The Generative machine

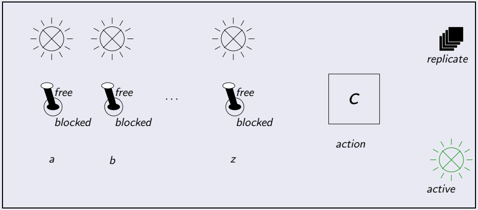
User panel



- Process in black box executes actions spontaneously
- Gadgets on user panel: possible observations
- Display: current action
- Switches: restricting corresponding action
- Menu Lights: actions that can be executed

The Generative machine

User panel

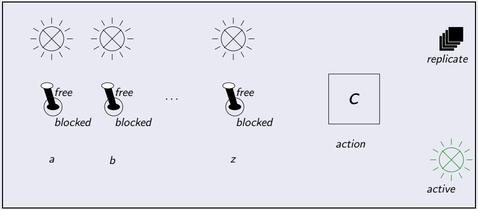


Convention

- Switches manipulate process
- Possible manipulation can be seen as observation
- Convention: manipulations are observations

The Generative machine

User panel

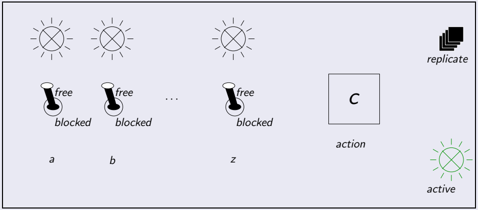


Convention

- Switches manipulate process
- Possible manipulation can be seen as observation
- Convention: manipulations are observations

The Generative machine

User panel

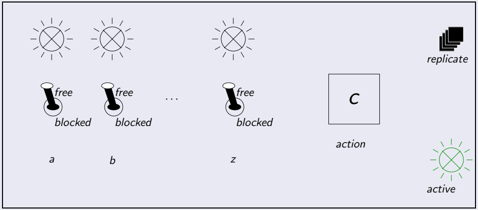


Convention

- Switches manipulate process
- Possible manipulation can be seen as observation
- Convention: manipulations are observations

What are observations?

User panel

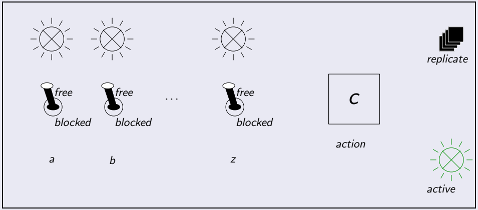


Display

- shows the action that is currently executed

What are observations?

User panel

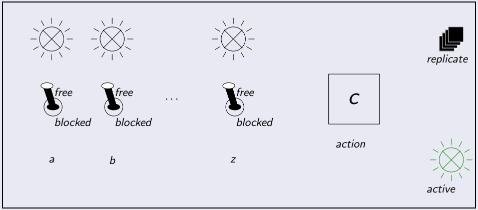


Switches

- set on free: action can be executed
- set on blocked: action can not be executed
- machine idles, if no free action can be executed, display is empty
- all switches on free, display empty: machine is deadlocked

What are observations?

User panel



Menus

- For each action one light
- Machine idle: light on for actions that can be executed
- Set of lighted actions: menu

Observations

Primitive observations $\mathcal{PO}$

1. Actions: $Act \subseteq \mathcal{PO}$
2. Refusals: $\widetilde{X} \in \mathcal{PO}$, for all $X \in 2^{Act}$

   actions $a \in X$ are set on free
3. Menus: $(X) \in \mathcal{PO}$, for all $X \in 2^{Act}$

   actions $a \in X$ can be executed

Observations

Sequences $\sigma \in \mathcal{PO}^*$ are our observations

Observations

Primitive observations $\mathcal{PO}$

1. Actions: $Act \subseteq \mathcal{PO}$
2. Refusals: $\widetilde{X} \in \mathcal{PO}$, for all $X \in 2^{Act}$

   actions $a \in X$ are set on free
3. Menus: $(X) \in \mathcal{PO}$, for all $X \in 2^{Act}$

   actions $a \in X$ can be executed

Observations

Sequences $\sigma \in \mathcal{PO}^*$ are our observations

Observations

Primitive observations $\mathcal{PO}$

1. Actions: $Act \subseteq \mathcal{PO}$
2. Refusals: $\widetilde{X} \in \mathcal{PO}$, for all $X \in 2^{Act}$

   actions $a \in X$ are set on free
3. Menus: $(X) \in \mathcal{PO}$, for all $X \in 2^{Act}$

   actions $a \in X$ can be executed

Observations

Sequences $\sigma \in \mathcal{PO}^*$ are our observations

Observations

Primitive observations $\mathcal{PO}$

1. Actions: $Act \subseteq \mathcal{PO}$
2. Refusals: $\widetilde{X} \in \mathcal{PO}$, for all $X \in 2^{Act}$

   actions $a \in X$ are set on free
3. Menus: $(X) \in \mathcal{PO}$, for all $X \in 2^{Act}$

   actions $a \in X$ can be executed

Observations

Sequences $\sigma \in \mathcal{PO}^*$ are our observations

Observations

Primitive observations $\mathcal{PO}$

1. Actions: $Act \subseteq \mathcal{PO}$
2. Refusals: $\widetilde{X} \in \mathcal{PO}$, for all $X \in 2^{Act}$
   actions $a \in X$ are set on free
3. Menus: $(X) \in \mathcal{PO}$, for all $X \in 2^{Act}$
   actions $a \in X$ can be executed

Observations

Sequences $\sigma \in \mathcal{PO}^*$ are our observations

Observations

Primitive observations $\mathcal{PO}$

1. Actions: $Act \subseteq \mathcal{PO}$
2. Refusals: $\widetilde{X} \in \mathcal{PO}$, for all $X \in 2^{Act}$
   actions $a \in X$ are set on free
3. Menus: $(X) \in \mathcal{PO}$, for all $X \in 2^{Act}$
   actions $a \in X$ can be executed

Observations

Sequences $\sigma \in \mathcal{PO}^*$ are our observations

Observations and implementation relations

Given:

- processes p, q inside black box
- $\mathrm{obs}(p), \mathrm{obs}(q) \subseteq \mathcal{PO}^*$: the sequences of observations of p and q
- the elements of $\mathrm{obs}(\cdot)$ depend on the primitive observation chosen

Implementation relations

- $p \leq^{may} q$ iff $\mathrm{obs}(p) \subseteq \mathrm{obs}(q)$
- $p \leq^{must} q$ iff $\mathrm{obs}(p) \supseteq \mathrm{obs}(q)$

We consider only *may* preorders

Observations and implementation relations

Given:

- processes p, q inside black box
- $\mathrm{obs}(p), \mathrm{obs}(q) \subseteq \mathcal{PO}^*$: the sequences of observations of p and q
- the elements of $\mathrm{obs}(\cdot)$ depend on the primitive observation chosen

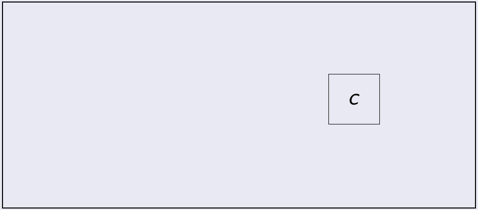
Implementation relations

- $p \leq^{may} q$ iff $\mathrm{obs}(p) \subseteq \mathrm{obs}(q)$
- $p \leq^{must} q$ iff $\mathrm{obs}(p) \supseteq \mathrm{obs}(q)$

We consider only *may* preorders

Trace preorder

The panel

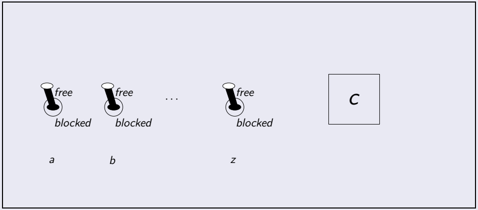


Observations and preorder

- $\text{obs}_{tr}(p)$: the sequence of actions executed by p
- $p \leq_{tr} q$ iff $\text{obs}_{tr}(p) \subseteq \text{obs}_{tr}(q)$

Failure preorder

Switches and display



Observations

- Switches on free or blocked, blocked actions not executable.
- $X \subseteq Act$: actions set on free.
- deadlock: machine refuses X
- observations: pairs $\langle \sigma, X \rangle$: trace σ leads up to refusal X

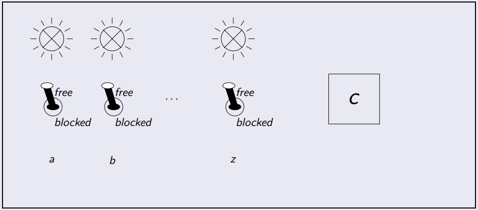
Failure preorder

Preorder

- $\langle \sigma, X \rangle$: failure pair
- $\text{obs}_f(p) = \{\langle \sigma, X \rangle \mid p \text{ executes } \sigma, \text{ then refuses } X\}$
- $p \leq_f q$ iff $\text{obs}_f(p) \subseteq \text{obs}_f(q)$
- failure preorder

Readiness preorder

Switches and display



Observations

- Situation as for failures preorder
- If machine idles: menu lights indicate which actions could have continued
- $X \subseteq Act$: $\langle \sigma, X \rangle$ ready pair of p iff

$$\exists p' : p \xrightarrow{\sigma} p' \text{ and } p' \xrightarrow{a} \forall a \in X.$$

Failure trace and Readiness trace preorder

Failures scenario and readiniess scenario

- Observation stops after machine idles
- machine is reset to starting state
- new observation starts

Alternative

- after machine idles continue observing
- unlock machine by setting switches to free
- Observations:
  - $\sigma \in (\left\{\widetilde{X} \mid X \in 2^{Act}\right\} \cup Act)^*$: failure traces
  - $\sigma \in (\{(X) \mid X \in 2^{Act}\} \cup Act)^*$: ready traces
- failure trace preorder, ready trace preorder

More Gadgets

Green light

Light is off: process idle or deadlocked

Light is on: process active with action

internal: if display is empty

observable: if display shows name

- Allows to distinguish deadlock and internal activity
- useless if process has no τ transitions

More Gadgets

Replicate Button

- if pressed, process (in its current state) is replicated:
  - one or finitely many copies
  - infinite nr. of copies (only interesting for systems with infinite branching)

  - observation of different futures
  - $\Longrightarrow$ trees rather than traces
  - necessary to characterise bisimulation

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