

Modeling Concurrent and Probabilistic Systems

Lecture 5: Properties of Strong Bisimulation

Joost-Pieter Katoen Thomas Noll

Software Modeling and Verification Group

RWTH Aachen University

`noll@cs.rwth-aachen.de`

`http://www-i2.informatik.rwth-aachen.de/i2/mcps07/`

Winter Semester 2007/08

- 1 Repetition: Definition of Strong Bisimulation
- 2 Properties of Strong Bisimulation
- 3 Decidability of Strong Bisimulation

Repetition: Definition of Strong Bisimulation

Definition (Strong bisimulation)

A relation $\rho \subseteq Prc \times Prc$ is called a **strong bisimulation** if $P\rho Q$ implies, for every $\alpha \in Act$,

- ① $P \xrightarrow{\alpha} P' \implies \text{ex. } Q' \in Prc \text{ such that } Q \xrightarrow{\alpha} Q' \text{ and } P'\rho Q'$
- ② $Q \xrightarrow{\alpha} Q' \implies \text{ex. } P' \in Prc \text{ such that } P \xrightarrow{\alpha} P' \text{ and } P'\rho Q'$

$P, Q \in Prc$ are called **strongly bisimilar** (notation: $P \sim Q$) if there exists a strong bisimulation ρ such that $P\rho Q$.

Theorem

$\sim$ is an equivalence relation.

- 1 Repetition: Definition of Strong Bisimulation
- 2 Properties of Strong Bisimulation
- 3 Decidability of Strong Bisimulation

It remains to show that strong bisimulation has the required properties of a process equivalence:

- ① Identification of processes with **identical LTSs**:
since the definition of strong bisimulation directly relies on the transition relation, processes with identical transition trees are clearly strongly bisimilar
- ② Implication of **trace equivalence**: following slides
- ③ **CCS congruence**: following slides
- ④ **Deadlock sensitivity**: following slides

It remains to show that strong bisimulation has the required properties of a process equivalence:

- ① Identification of processes with **identical LTSs**:
since the definition of strong bisimulation directly relies on the transition relation, processes with identical transition trees are clearly strongly bisimilar
- ② Implication of **trace equivalence**: following slides
- ③ **CCS congruence**: following slides
- ④ **Deadlock sensitivity**: following slides

Strong Bisimulation Implies Trace Equivalence

Definition (Trace language; repetition)

The **trace language** of $P \in \text{Prc}$ is given by

$$\text{Tr}(P) := \{w \in \text{Act}^* \mid \text{ex. } P' \in \text{Prc} \text{ such that } P \xrightarrow{w}^* P'\}.$$

Theorem 5.1

For every $P, Q \in \text{Prc}$, $P \sim Q$ implies $\text{Tr}(P) = \text{Tr}(Q)$.

Proof.

- Assume that $P \sim Q$ but $w \in \text{Tr}(P) \setminus \text{Tr}(Q)$.
- Let $v \in \text{Act}^*$ be the longest prefix of w such that $v \in \text{Tr}(Q)$ (i.e., $w = v\alpha u$ for some $\alpha \in \text{Act}$ and $u \in \text{Act}^*$).
- Let $P', P'' \in \text{Prc}$ such that $P \xrightarrow{v}^* P' \xrightarrow{\alpha} P''$.
- Since $P \sim Q$ there exists $Q' \in \text{Prc}$ such that $Q \xrightarrow{v}^* Q'$ and $P' \sim Q'$ (by induction on $|v|$).
- But we have that $P' \xrightarrow{\alpha} P''$ whereas $Q' \not\xrightarrow{\alpha} \Rightarrow$ contradiction.



Strong Bisimulation Implies Trace Equivalence

Definition (Trace language; repetition)

The **trace language** of $P \in \text{Prc}$ is given by

$$\text{Tr}(P) := \{w \in \text{Act}^* \mid \text{ex. } P' \in \text{Prc} \text{ such that } P \xrightarrow{w}^* P'\}.$$

Theorem 5.1

For every $P, Q \in \text{Prc}$, $P \sim Q$ implies $\text{Tr}(P) = \text{Tr}(Q)$.

Proof.

- Assume that $P \sim Q$ but $w \in \text{Tr}(P) \setminus \text{Tr}(Q)$.
- Let $v \in \text{Act}^*$ be the longest prefix of w such that $v \in \text{Tr}(Q)$ (i.e., $w = v\alpha u$ for some $\alpha \in \text{Act}$ and $u \in \text{Act}^*$).
- Let $P', P'' \in \text{Prc}$ such that $P \xrightarrow{v}^* P' \xrightarrow{\alpha} P''$.
- Since $P \sim Q$ there exists $Q' \in \text{Prc}$ such that $Q \xrightarrow{v}^* Q'$ and $P' \sim Q'$ (by induction on $|v|$).
- But we have that $P' \xrightarrow{\alpha} P''$ whereas $Q' \not\xrightarrow{\alpha} \Rightarrow$ contradiction



Strong Bisimulation Implies Trace Equivalence

Definition (Trace language; repetition)

The **trace language** of $P \in \text{Prc}$ is given by

$$\text{Tr}(P) := \{w \in \text{Act}^* \mid \text{ex. } P' \in \text{Prc} \text{ such that } P \xrightarrow{w}^* P'\}.$$

Theorem 5.1

For every $P, Q \in \text{Prc}$, $P \sim Q$ implies $\text{Tr}(P) = \text{Tr}(Q)$.

Proof.

- Assume that $P \sim Q$ but $w \in \text{Tr}(P) \setminus \text{Tr}(Q)$.
- Let $v \in \text{Act}^*$ be the longest prefix of w such that $v \in \text{Tr}(Q)$ (i.e., $w = v\alpha u$ for some $\alpha \in \text{Act}$ and $u \in \text{Act}^*$).
- Let $P', P'' \in \text{Prc}$ such that $P \xrightarrow{v}^* P' \xrightarrow{\alpha} P''$.
- Since $P \sim Q$ there exists $Q' \in \text{Prc}$ such that $Q \xrightarrow{v}^* Q'$ and $P' \sim Q'$ (by induction on $|v|$).
- But we have that $P' \xrightarrow{\alpha} P''$ whereas $Q' \not\xrightarrow{\alpha} \implies$ contradiction



Strong Bisimulation Implies Trace Equivalence

Definition (Trace language; repetition)

The **trace language** of $P \in \text{Prc}$ is given by

$$\text{Tr}(P) := \{w \in \text{Act}^* \mid \text{ex. } P' \in \text{Prc} \text{ such that } P \xrightarrow{w}^* P'\}.$$

Theorem 5.1

For every $P, Q \in \text{Prc}$, $P \sim Q$ implies $\text{Tr}(P) = \text{Tr}(Q)$.

Proof.

- Assume that $P \sim Q$ but $w \in \text{Tr}(P) \setminus \text{Tr}(Q)$.
- Let $v \in \text{Act}^*$ be the longest prefix of w such that $v \in \text{Tr}(Q)$ (i.e., $w = v\alpha u$ for some $\alpha \in \text{Act}$ and $u \in \text{Act}^*$).
- Let $P', P'' \in \text{Prc}$ such that $P \xrightarrow{v}^* P' \xrightarrow{\alpha} P''$.
- Since $P \sim Q$ there exists $Q' \in \text{Prc}$ such that $Q \xrightarrow{v}^* Q'$ and $P' \sim Q'$ (by induction on $|v|$).
- But we have that $P' \xrightarrow{\alpha} P''$ whereas $Q' \not\xrightarrow{\alpha} \implies$ contradiction



Strong Bisimulation Implies Trace Equivalence

Definition (Trace language; repetition)

The **trace language** of $P \in \text{Prc}$ is given by

$$\text{Tr}(P) := \{w \in \text{Act}^* \mid \text{ex. } P' \in \text{Prc} \text{ such that } P \xrightarrow{w}^* P'\}.$$

Theorem 5.1

For every $P, Q \in \text{Prc}$, $P \sim Q$ implies $\text{Tr}(P) = \text{Tr}(Q)$.

Proof.

- Assume that $P \sim Q$ but $w \in \text{Tr}(P) \setminus \text{Tr}(Q)$.
- Let $v \in \text{Act}^*$ be the longest prefix of w such that $v \in \text{Tr}(Q)$ (i.e., $w = v\alpha u$ for some $\alpha \in \text{Act}$ and $u \in \text{Act}^*$).
- Let $P', P'' \in \text{Prc}$ such that $P \xrightarrow{v}^* P' \xrightarrow{\alpha} P''$.
- Since $P \sim Q$ there exists $Q' \in \text{Prc}$ such that $Q \xrightarrow{v}^* Q'$ and $P' \sim Q'$ (by induction on $|v|$).
- But we have that $P' \xrightarrow{\alpha} P''$ whereas $Q' \not\xrightarrow{\alpha} \implies$ contradiction



Strong Bisimulation Implies Trace Equivalence

Definition (Trace language; repetition)

The **trace language** of $P \in \text{Prc}$ is given by

$$\text{Tr}(P) := \{w \in \text{Act}^* \mid \text{ex. } P' \in \text{Prc} \text{ such that } P \xrightarrow{w}^* P'\}.$$

Theorem 5.1

For every $P, Q \in \text{Prc}$, $P \sim Q$ implies $\text{Tr}(P) = \text{Tr}(Q)$.

Proof.

- Assume that $P \sim Q$ but $w \in \text{Tr}(P) \setminus \text{Tr}(Q)$.
- Let $v \in \text{Act}^*$ be the longest prefix of w such that $v \in \text{Tr}(Q)$ (i.e., $w = v\alpha u$ for some $\alpha \in \text{Act}$ and $u \in \text{Act}^*$).
- Let $P', P'' \in \text{Prc}$ such that $P \xrightarrow{v}^* P' \xrightarrow{\alpha} P''$.
- Since $P \sim Q$ there exists $Q' \in \text{Prc}$ such that $Q \xrightarrow{v}^* Q'$ and $P' \sim Q'$ (by induction on $|v|$).
- But we have that $P' \xrightarrow{\alpha} P''$ whereas $Q' \not\xrightarrow{\alpha} \implies$ contradiction



Strong Bisimulation Implies Trace Equivalence

Definition (Trace language; repetition)

The **trace language** of $P \in \text{Prc}$ is given by

$$\text{Tr}(P) := \{w \in \text{Act}^* \mid \text{ex. } P' \in \text{Prc} \text{ such that } P \xrightarrow{w}^* P'\}.$$

Theorem 5.1

For every $P, Q \in \text{Prc}$, $P \sim Q$ implies $\text{Tr}(P) = \text{Tr}(Q)$.

Proof.

- Assume that $P \sim Q$ but $w \in \text{Tr}(P) \setminus \text{Tr}(Q)$.
- Let $v \in \text{Act}^*$ be the longest prefix of w such that $v \in \text{Tr}(Q)$ (i.e., $w = v\alpha u$ for some $\alpha \in \text{Act}$ and $u \in \text{Act}^*$).
- Let $P', P'' \in \text{Prc}$ such that $P \xrightarrow{v}^* P' \xrightarrow{\alpha} P''$.
- Since $P \sim Q$ there exists $Q' \in \text{Prc}$ such that $Q \xrightarrow{v}^* Q'$ and $P' \sim Q'$ (by induction on $|v|$).
- But we have that $P' \xrightarrow{\alpha} P''$ whereas $Q' \not\xrightarrow{\alpha} \implies$ contradiction



The congruence proof employs the following lemma.

Lemma 5.2

For every $P, Q, R \in \text{Prc}$,

- ① $P + Q \sim Q + P$
- ② $P + (Q + R) \sim (P + Q) + R$
- ③ $P + \text{nil} \sim P$
- ④ $P \parallel Q \sim Q \parallel P$
- ⑤ $P \parallel (Q \parallel R) \sim (P \parallel Q) \parallel R$
- ⑥ $P \parallel \text{nil} \sim P$

Proof.

- ① on the board
- ② on the board
- ③ 3rd ex. sheet
- ④ on the board
- ⑤ 3rd ex. sheet
- ⑥ on the board



Congruence Property of Strong Bisimulation I

The congruence proof employs the following lemma.

Lemma 5.2

For every $P, Q, R \in \text{Prc}$,

- ① $P + Q \sim Q + P$
- ② $P + (Q + R) \sim (P + Q) + R$
- ③ $P + \text{nil} \sim P$
- ④ $P \parallel Q \sim Q \parallel P$
- ⑤ $P \parallel (Q \parallel R) \sim (P \parallel Q) \parallel R$
- ⑥ $P \parallel \text{nil} \sim P$

Proof.

- ① on the board
- ② on the board
- ③ 3rd ex. sheet
- ④ on the board
- ⑤ 3rd ex. sheet
- ⑥ on the board



Congruence Property of Strong Bisimulation II

Definition (CCS congruence; repetition)

An equivalence relation $\cong \subseteq \text{Prc} \times \text{Prc}$ is said to be a **CCS congruence** if it is preserved by the CCS constructs; that is, if $P \cong Q$ then

$$\alpha.P \cong \alpha.Q$$

$$P + R \cong Q + R$$

$$R + P \cong R + Q$$

$$P \parallel R \cong Q \parallel R$$

$$R \parallel P \cong R \parallel Q$$

$$\text{new } a P \cong \text{new } a Q$$

for every $\alpha \in \text{Act}$, $R \in \text{Prc}$, and $a \in N$.

Theorem 5.3

$\sim$ is a CCS congruence.

Proof.

on the board



Congruence Property of Strong Bisimulation II

Definition (CCS congruence; repetition)

An equivalence relation $\cong \subseteq \text{Prc} \times \text{Prc}$ is said to be a **CCS congruence** if it is preserved by the CCS constructs; that is, if $P \cong Q$ then

$$\alpha.P \cong \alpha.Q$$

$$P + R \cong Q + R$$

$$R + P \cong R + Q$$

$$P \parallel R \cong Q \parallel R$$

$$R \parallel P \cong R \parallel Q$$

$$\text{new } a P \cong \text{new } a Q$$

for every $\alpha \in \text{Act}$, $R \in \text{Prc}$, and $a \in N$.

Theorem 5.3

$\sim$ is a CCS congruence.

Proof.

on the board



Congruence Property of Strong Bisimulation II

Definition (CCS congruence; repetition)

An equivalence relation $\cong \subseteq \text{Prc} \times \text{Prc}$ is said to be a **CCS congruence** if it is preserved by the CCS constructs; that is, if $P \cong Q$ then

$$\alpha.P \cong \alpha.Q$$

$$P + R \cong Q + R$$

$$R + P \cong R + Q$$

$$P \parallel R \cong Q \parallel R$$

$$R \parallel P \cong R \parallel Q$$

$$\text{new } a P \cong \text{new } a Q$$

for every $\alpha \in \text{Act}$, $R \in \text{Prc}$, and $a \in N$.

Theorem 5.3

$\sim$ is a CCS congruence.

Proof.

on the board



Deadlock Sensitivity of Strong Bisimulation

Definition (Deadlock; repetition)

Let $P, Q \in \text{Prc}$ and $w \in \text{Act}^*$ such that $P \xrightarrow{w}^* Q$ and $Q \not\rightarrow$. Then Q is called a **w -deadlock** of P .

An equivalence relation $\cong \subseteq \text{Prc} \times \text{Prc}$ is called **deadlock sensitive** if for every $P \cong Q$ such that P has a w -deadlock, Q also has a w -deadlock.

Theorem 5.4

$\sim$ is deadlock sensitive.

Proof.

on the board □

Deadlock Sensitivity of Strong Bisimulation

Definition (Deadlock; repetition)

Let $P, Q \in \text{Prc}$ and $w \in \text{Act}^*$ such that $P \xrightarrow{w}^* Q$ and $Q \not\rightarrow$. Then Q is called a **w -deadlock** of P .

An equivalence relation $\cong \subseteq \text{Prc} \times \text{Prc}$ is called **deadlock sensitive** if for every $P \cong Q$ such that P has a w -deadlock, Q also has a w -deadlock.

Theorem 5.4

$\sim$ is deadlock sensitive.

Proof.

on the board □

Deadlock Sensitivity of Strong Bisimulation

Definition (Deadlock; repetition)

Let $P, Q \in \text{Prc}$ and $w \in \text{Act}^*$ such that $P \xrightarrow{w}^* Q$ and $Q \not\rightarrow$. Then Q is called a **w -deadlock** of P .

An equivalence relation $\cong \subseteq \text{Prc} \times \text{Prc}$ is called **deadlock sensitive** if for every $P \cong Q$ such that P has a w -deadlock, Q also has a w -deadlock.

Theorem 5.4

$\sim$ is deadlock sensitive.

Proof.

on the board

