Concurrency Theory

Strong bisimulation

Joost-Pieter Katoen and Thomas Noll

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://www-i2.informatik.rwth-aachen.de/i2/ct13

December 05, 2013

RWTH AACHEN UNIVERSITY

Overview

Overview

Summary so far

- Trace equivalence is a possible behavioural equivalence, is a congruence, but does not preserve deadlocks.
- Main problem:
  $$\alpha.(P + Q) \equiv \alpha.P + \alpha.Q,$$
  whereas their deadlock behaviour in a context differs.
- Solution: consider finer behavioural equivalences such that:
  $$\alpha.(P + Q) \not\equiv \alpha.P + \alpha.Q$$
- Our (serious) attempt today: Milner's strong bisimulation.

Overview

Robin Milner (1934-2010)

Rationale

Observation

In order for a behavioural equivalence to be deadlock sensitive, it has to take the branching structure of processes into account.

This is achieved by an equivalence that is defined according to the scheme:

Bisimulation scheme

$P, Q \in Prc$ are equivalent iff, for every action α, every α-successor of P is equivalent to some α-successor of Q, and vice versa.

Three versions will be considered in these lecture series:

1. Strong bisimulation: ignore the special function of τ-actions
2. Weak bisimulation: treat τ-actions as invisible
3. Simulation relations: unidirectional versions of bisimulation

Strong bisimulation

Strong bisimulation [Park, 1981, Milner, 1989]

A binary relation $\mathcal{R} \subseteq Prc \times Prc$ is a strong bisimulation whenever for every $(P, Q) \in \mathcal{R}$, and $\alpha \in Act$:

1. if $P \xrightarrow{\alpha} P'$ then there exists $Q' \in Prc$ s.t. $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{R}$,
2. if $Q \xrightarrow{\alpha} Q'$ then there exists $P' \in Prc$ s.t. $P \xrightarrow{\alpha} P'$ and $(P', Q') \in \mathcal{R}$.

Strong bisimilarity

The processes P and Q are strongly bisimilar, denoted $P \sim Q$, iff there is a strong bisimulation $\mathcal{R}$ with $(P, Q) \in \mathcal{R}$. Thus,

$$\sim \; = \; \bigcup \{ \mathcal{R} \mid \mathcal{R} \text{ is a strong bisimulation} \}.$$

Relation $\sim$ is called a strong bisimulation equivalence or strong bisimilarity.

Strong bisimulation

$$\begin{array}{ccc} P & \xrightarrow{\alpha} & P' \\ \mathcal{R} & & \\ Q & & \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} P & \xrightarrow{\alpha} & P' \\ \mathcal{R} & & \mathcal{R} \\ Q & \xrightarrow{\alpha} & Q' \end{array}$$

and

$$\begin{array}{ccc} P & & \\ \mathcal{R} & & \\ Q & \xrightarrow{\alpha} & Q' \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} P & \xrightarrow{\alpha} & P' \\ \mathcal{R} & & \mathcal{R} \\ Q & \xrightarrow{\alpha} & Q' \end{array}$$

Example

A first example

Claim: $P \sim Q$ where:

$$\begin{array}{rcl} P & = & a.P_1 + a.P_2 \\ P_1 & = & b.P_2 \\ P_2 & = & b.P_2 \\ Q & = & a.Q_1 \\ Q_1 & = & b.Q_1. \end{array}$$

Proof: $\mathcal{R} = \{ (P, Q), (P_1, Q_1), (P_2, Q_1) \}$ is a strong bisimulation.

Relating a finite to an infinite-state process

Claim: $P_0 \sim Q$ where:

$$\begin{array}{rcl} P_i & = & a.P_{i+1} \text{ for } i \in \mathbb{N} \\ Q & = & a.Q \end{array}$$

Proof: $\mathcal{R} = \{ (P_i, Q) \mid i \in \mathbb{N} \}$ is a strong bisimulation.

Properties of strong bisimilarity

Properties of $\sim$

1. $\sim$ is an equivalence relation (reflexive, symmetric, transitive).
2. $\sim$ is the largest strong bisimulation.
3. $s \sim t$ if and only if for every $\alpha \in Act$:
   - 3.1 if $s \xrightarrow{\alpha} s'$ then there is a transition $t \xrightarrow{\alpha} t'$ with $s' \sim t'$
   - 3.2 if $t \xrightarrow{\alpha} t'$ then there is a transition $s \xrightarrow{\alpha} s'$ with $s' \sim t'$.

Proof.

On the board. □

Overview

Bisimulation on paths

Whenever we have:

$$\begin{array}{ccccccccc} s_0 & \xrightarrow{\alpha_1} & s_1 & \xrightarrow{\alpha_2} & s_2 & \xrightarrow{\alpha_3} & s_3 & \xrightarrow{\alpha_4} & s_4 \ldots\ldots \\ \mathcal{R} & & & & & & & & \\ t_0 & & & & & & & & \end{array}$$

this can be completed to

$$\begin{array}{ccccccccc} s_0 & \xrightarrow{\alpha_1} & s_1 & \xrightarrow{\alpha_2} & s_2 & \xrightarrow{\alpha_3} & s_3 & \xrightarrow{\alpha_4} & s_4 \ldots\ldots \\ \mathcal{R} & & \mathcal{R} & & \mathcal{R} & & \mathcal{R} & & \mathcal{R} \\ t_0 & \xrightarrow{\alpha_1} & t_1 & \xrightarrow{\alpha_2} & t_2 & \xrightarrow{\alpha_3} & t_3 & \xrightarrow{\alpha_4} & t_4 \ldots\ldots \end{array}$$

proof: by induction on the length of a path

Strong bisimulation versus trace equivalence

Theorem

$P \sim Q$ implies that P and Q are trace equivalent. The reverse does not hold.

Proof.

The implication from left-to-right follows from the previous slide.

Consider the other direction.
Take $P = a.P_1$ with $P_1 = b.\text{nil} + c.\text{nil}$ and $Q = a.b.nil + a.c.\text{nil}$.
Then: $Tr(P) = \{\, \epsilon, a, ab, ac \,\} = Tr(Q)$.
Thus, P and Q are trace equivalent.
But: $P \not\sim Q$, as there is no state in Q that is bisimilar to P_1.
Why? There is no state in Q can can perform either b or cs. □

Deterministic transition systems

Determinism

$P \in Prc$ is deterministic whenever for every of its states s it holds:

$$(s \xrightarrow{\alpha} t \text{ and } s \xrightarrow{\alpha} u) \quad \text{implies} \quad t = u.$$

Determinism implies $\sim$ and trace equivalence coincide **[Park]**

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

Left as an exercise. In fact, for deterministic processes, trace equivalence, complete trace, failure trace, and ready trace equivalence all coincide. □

Overview

Congruence

CCS congruence

Strong bisimilarity $\sim$ is a CCS congruence. Let $P, Q \in Prc$ be CCS processes. Assume $P \sim Q$. Then:

$$\begin{array}{rcll} \alpha.P & \sim & \alpha.Q & \text{for every action } \alpha \\ P+R & \sim & Q+R & \text{for every process } R \\ P||R & \sim & Q||R & \text{for every process } R \\ P \backslash L & \sim & Q \backslash L & \text{for every set } L \subseteq A \\ P[f] & \sim & Q[f] & \text{for every relabelling } f. \end{array}$$

Proof.

Provide the proof for $||$ and $\backslash L$ on the board. The proofs for the other CCS operators is left as an exercise. □

Deadlock sensitivity of $\sim$

Deadlock

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

Deadlock sensitive

Relation $\equiv \subseteq Prc \times Prc$ is deadlock sensitive whenever:

$$P \equiv Q \text{ implies } (\forall w.\, P \text{ has a } w\text{-deadlock iff } Q \text{ has a } w\text{-deadlock})$$

Theorem

$\sim$ is deadlock sensitive.

Proof.

On the board. □

Two buffers

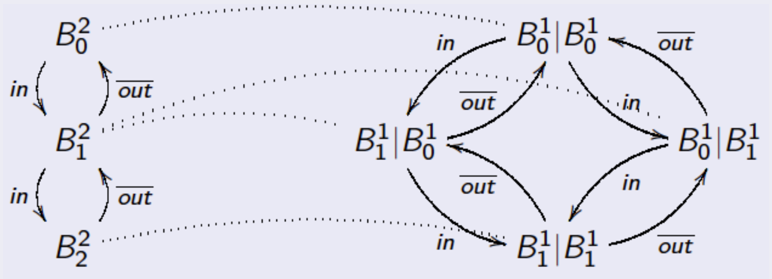
Buffer of capacity one

$$\begin{array}{rcl} B_0^1 & = & in.B_1^1 \\ B_1^1 & = & \overline{out}.B_0^1. \end{array}$$

Buffer of capacity two

$$\begin{array}{rcl} B_0^2 & = & in.B_1^2 \\ B_1^2 & = & in.B_2^2 + \overline{out}.B_0^2 \\ B_2^2 & = & \overline{out}.B_1^2. \end{array}$$

$B_0^2 \sim B_0^1 \,||\, B_0^1$



Buffers: a generalisation

An n-place buffer

Let B_i^n stand for a buffer of capacity n holding i items:

$$\begin{array}{rcll} B_0^n & = & in.B_1^n & \\ B_i^n & = & in.B_{i+1}^n + \overline{out}.B_{i-1}^n & \text{for } 0 < i < n \\ B_n^n & = & \overline{out}.B_{n-1}^n. & \end{array}$$

This buffer is strongly bisimilar to n parallel buffers of capacity one:

Proposition

For every $n \in \mathbb{N}_{>0}$, we have: $B_0^n \sim \underbrace{B_0^1 \,||\, \cdots \,||\, B_0^1}_{n \text{ times}}$.

Buffers

Proposition

For every $n \in \mathbb{N}_{>0}$, we have: $B_0^n \sim \underbrace{B_0^1 \,||\, \cdots \,||\, B_0^1}_{n \text{ times}}$.

Proof.

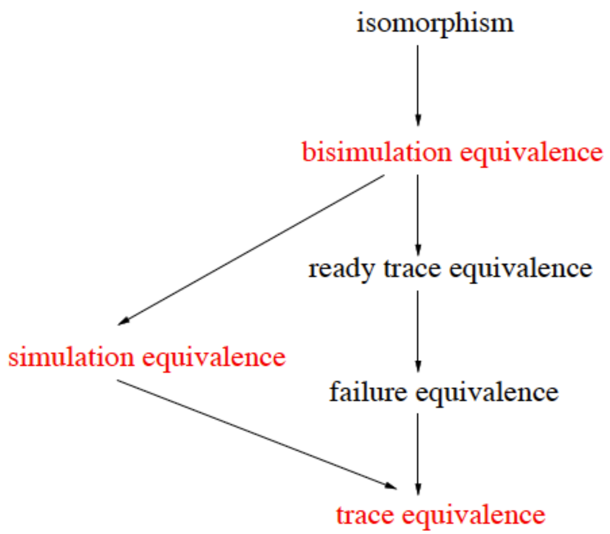
Consider the following binary relation where $i_1, i_2, \ldots, i_n \in \{0, 1\}$:

$$\mathcal{R} \;=\; \left\{ (B_i^n, B_{i_1}^1 \,||\, \cdots \,||\, B_{i_n}^1 \mid \sum_{j=1}^{n} i_j = i \right\}$$

Then: $\mathcal{R}$ is a strong bisimulation and $(B_0^n, \underbrace{B_0^1 \,||\, \cdots \,||\, B_0^1}_{n \text{ times}}) \in \mathcal{R}$. □

Overview

Overview of some behavioural equivalences



Summary

1. Strong bisimulation is based on mutual mimicking each other
2. Strong bisimilarity $\sim$:
   - 2.1 is the largest strong bisimulation
   - 2.2 is an equivalence
   - 2.3 is a congruence (for CCS)
   - 2.4 is strictly finer than trace equivalence
   - 2.5 is deadlock sensitive