

Concurrency Theory

Weak bisimulation

Joost-Pieter Katoen and Thomas Noll

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

<http://moves.rwth-aachen.de/teaching/ws-1516/ct>

January 14, 2016



Overview

- 1 Aim of this lecture
- 2 Introduction
- 3 Weak bisimulation
- 4 Properties of weak bisimilarity
- 5 Observation congruence
- 6 Deciding weak bisimilarity and game interpretation
- 7 Summary

Overview

- 1 Aim of this lecture
- 2 Introduction
- 3 Weak bisimulation
- 4 Properties of weak bisimilarity
- 5 Observation congruence
- 6 Deciding weak bisimilarity and game interpretation
- 7 Summary

Summary so far

- ▶ Strong bisimulation is based on mutual mimicking of processes
- ▶ Strong bisimilarity (\sim) is a congruence, is deadlock sensitive
- ▶ Implies trace equivalence, and can be computed in polynomial time

But \sim does not distinguish between internal (τ -) actions and observable actions.

Aims of this lecture

1. A notion of bisimulation that treats τ -actions as **un**observable
2. How to treat **divergences**, i.e., loops of τ -actions?
3. A slight adaptation that yields a CCS congruence

Overview

- 1 Aim of this lecture
- 2 Introduction
- 3 Weak bisimulation
- 4 Properties of weak bisimilarity
- 5 Observation congruence
- 6 Deciding weak bisimilarity and game interpretation
- 7 Summary

Properties of strong bisimilarity

Properties of strong bisimilarity

1. \sim is an equivalence relation.
2. $P \sim Q \implies Tr(P) = Tr(Q)$.
3. \sim is a CCS congruence.
4. \sim is deadlock sensitive.
5. checking \sim is decidable for finite-state processes and can be done in polynomial time.¹
6. \sim has a nice game characterization.

Question: is there a need to consider another behavioural equivalence?

Yes.

¹In fact, computing \sim is P-complete. It is thus one of the “hardest problems” admitting a polynomial-time algorithm.

Strong bisimulation

Strong bisimulation

[Park, 1981, Milner, 1989]

A binary relation $\mathcal{R} \subseteq Proc \times Proc$ 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 $Q' \in Proc$ s.t. $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{R}$
2. if $Q \xrightarrow{\alpha} Q'$ then there exists $P' \in Proc$ 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 **strong bisimilarity**.

Inadequacy of strong bisimilarity

Sequential two-place buffer

$$\begin{aligned} sB_0 &= in.sB_1 \\ sB_1 &= in.sB_2 + \overline{out}.sB_0 \\ sB_2 &= \overline{out}.sB_1. \end{aligned}$$

Parallel two-place buffer

$$\begin{aligned} pB &= (oB[f] \parallel oB[g]) \setminus \{ com \} \\ &\text{with } f(in) = in \text{ and } f(out) = com \\ &\text{and } g(in) = com \text{ and } g(out) = out \\ oB &= in.\overline{out}.oB \end{aligned}$$

Inadequacy of strong bisimilarity

Sequential buffer $\not\sim$ parallel buffer



Problem: the sequential buffer cannot simulate the (invisible) τ -action.

Thus, the requirement in \sim to **exactly match all actions** is often too strong.

This suggests to weaken this and **not insist on exact matching of τ -actions**.

Rationale: τ -actions are special as they are **unobservable**.

Overview

- 1 Aim of this lecture
- 2 Introduction
- 3 **Weak bisimulation**
- 4 Properties of weak bisimilarity
- 5 Observation congruence
- 6 Deciding weak bisimilarity and game interpretation
- 7 Summary

The rationales for abstracting from τ

- ▶ The τ -actions are **internal** and thus **unobservable**.
- ▶ This is natural in parallel communication yielding τ : synchronization in CCS is binary and as observation means communication with the process, the **result of any communication is unobservable**
- ▶ Strong bisimilarity treats τ -actions as any other action.
- ▶ Can we yield the nice properties of \sim while “**abstracting**” from τ -actions?

Weak transition relation

Weak transition relation

$$\xRightarrow{\alpha} = \begin{cases} (-\tau \rightarrow)^* \circ \xrightarrow{\alpha} \circ (-\tau \rightarrow)^* & \text{if } \alpha \neq \tau \\ (-\tau \rightarrow)^* & \text{if } \alpha = \tau. \end{cases}$$

where $(-\tau \rightarrow)^*$ is the reflexive and transitive closure of the relation $-\tau \rightarrow$.

Informal meaning

1. If $\alpha \neq \tau$, then $s \xRightarrow{\alpha} t$ means that from s we can get to t by doing zero or more τ actions, followed by the action α , followed by zero or more τ actions.
2. If $\alpha = \tau$, then $s \xRightarrow{\alpha} t$ means that from s we can reach t by doing zero or more τ actions.

Weak bisimulation

Weak bisimulation

[Milner, 1989]

A binary relation $\mathcal{R} \subseteq Proc \times Proc$ is a **weak bisimulation** whenever for every $(P, Q) \in \mathcal{R}$, and $\alpha \in Act$ (including $\alpha = \tau$):

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

Weak bisimilarity

The processes P and Q are **weakly bisimilar**, denoted $P \approx Q$, iff there is a weak bisimulation \mathcal{R} with $(P, Q) \in \mathcal{R}$. Thus,

$$\approx = \bigcup \{ \mathcal{R} \mid \mathcal{R} \text{ is a weak bisimulation} \}.$$

Relation \approx is called an **observational equivalence** or **weak bisimilarity**.

Explanation

Weak bisimulation

[Milner, 1989]

A binary relation $\mathcal{R} \subseteq Proc \times Proc$ is a **weak bisimulation** whenever for every $(P, Q) \in \mathcal{R}$, and $\alpha \in Act$ (including $\alpha = \tau$):

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

Remark

Each clause in the definition of weak bisimulation subsumes **two cases**:

- ▶ $P \xrightarrow{\alpha} P'$ where $\alpha \neq \tau$
implies ex. $Q' \in Proc$ with $Q (\xrightarrow{\tau})^* \xrightarrow{\alpha} (\xrightarrow{\tau})^* Q'$ and $(P', Q') \in \mathcal{R}$
- ▶ $P \xrightarrow{\tau} P'$
implies ex. $Q' \in Proc$ such that $Q (\xrightarrow{\tau})^* Q'$ and $(P', Q') \in \mathcal{R}$
(where $Q' = Q$ is admissible)

Weak bisimulation

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

and

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

Examples

Weak bisimulation

[Milner, 1989]

A binary relation $\mathcal{R} \subseteq Proc \times Proc$ is a **weak bisimulation** whenever for every $(P, Q) \in \mathcal{R}$, and $\alpha \in Act$ (including $\alpha = \tau$):

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

A first example

Let $P = \tau.a.nil$ and $Q = a.nil$. Then $P \not\approx Q$. Claim: $P \approx Q$. Rewrite P as: $P = \tau.P_1$ with $P_1 = a.nil$. Let $\mathcal{R} = \{ (P, Q), (P_1, Q), (nil, nil) \}$. Check that \mathcal{R} is a weak bisimulation. As $(P, Q) \in \mathcal{R}$, it follows $P \approx Q$.

Buffers

Check that the parallel and sequential buffer are weakly bisimilar.

Overview

- 1 Aim of this lecture
- 2 Introduction
- 3 Weak bisimulation
- 4 Properties of weak bisimilarity
- 5 Observation congruence
- 6 Deciding weak bisimilarity and game interpretation
- 7 Summary

Divergence

A polling process

[Koomen, 1982]

Let:

$$\begin{aligned} A? &= a.nil + \tau.B? \\ B? &= b.nil + \tau.A? \end{aligned}$$

Claim: $A? \approx B? \approx a.nil + b.nil$. (Check this!)

But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ is a τ -loop, whereas $a.nil + b.nil$ does not have a loop, not even a τ -loop.

Thus, \approx assumes that if a process can escape from a τ -loop, it eventually will do so.²

Note that also $Div \approx nil$ where $Div = \tau.Div$. Thus, a **deadlock process is weakly bisimilar to a process that can only diverge**. This is justified by the fact that “observations” can only be made by interacting with the process.

²This is called **fair abstraction from divergence**. Divergence is a τ -loop.

Divergence

For every $P \in Proc$,

$$P \approx \tau.P$$

Proof.

We show that

$$\mathcal{R} = \{(P, \tau.P)\} \cup id_{Proc}$$

is a weak bisimulation:

1. every transition of P , $P \xrightarrow{\alpha} P'$ can be simulated by $\tau.P \xrightarrow{\tau} P \xrightarrow{\alpha} P'$ i.e., equivalently $\tau.P \xrightarrow{\alpha} P'$ with $(P', P') \in \mathcal{R}$ (since $id_{Proc} \subseteq \mathcal{R}$)
2. the only transition of $\tau.P$ is $\tau.P \xrightarrow{\tau} P$; it is simulated by $P \xrightarrow{\tau} P$ with $(P, P) \in \mathcal{R}$ (since $id_{Proc} \subseteq \mathcal{R}$).

□

Properties of weak bisimilarity

Properties of \approx

1. $P \sim Q$ implies $P \approx Q$.
2. \approx is an equivalence relation (reflexive, symmetric, transitive).
3. \approx is the largest weak bisimulation.
4. \approx is (non- τ) deadlock sensitive.³
5. \approx abstracts from τ -loops.

Proof.

1. Straightforward. 2.–4. Similar to the proofs for \sim . Left as an exercise.
5. Previous slide. □

³Where w -deadlocks are considered on observable traces w .

Weak bisimilarity versus trace equivalence

Observational trace language

The **observational trace language** of $P \in \text{Prc}$ is defined by:

$$\text{ObsTr}(P) = \{ \hat{w} \in \text{Act}^* \mid \exists P' \in \text{Prc}. P \xrightarrow{w} P' \}$$

where \hat{w} is obtained from w by omitting all τ -actions.

Trace equivalence

$P, Q \in \text{Prc}$ are **observational trace equivalent** if $\text{ObsTr}(P) = \text{ObsTr}(Q)$.

Theorem

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

Congruence

CCS congruence

Let $P, Q \in \text{Prc}$ be CCS processes. Assume $P \approx Q$. Then:

$$\begin{aligned} \alpha.P &\approx \alpha.Q \text{ for every action } \alpha \\ P \parallel R &\approx Q \parallel R \text{ for every process } R \\ P \setminus L &\approx Q \setminus L \text{ for every set } L \subseteq A \\ P[f] &\approx Q[f] \text{ for every relabelling } f. \end{aligned}$$

What about choice?

$$\tau.a.\text{nil} \approx a.\text{nil} \quad \text{but} \quad \tau.a.\text{nil} + b.\text{nil} \not\approx a.\text{nil} + b.\text{nil}$$

Thus, weak bisimilarity is **not** a congruence for CCS.
This motivates a slight adaptation of \approx .

Milner's τ -laws

Milner's τ -laws

$$\begin{aligned} \alpha.\tau.P &\approx \alpha.P \\ P + \tau.P &\approx \tau.P \\ \alpha.(P + \tau.Q) &\approx \alpha.(P + \tau.Q) + \alpha.Q. \end{aligned}$$

Proof.

Left as an exercise. Build appropriate weak bisimulation relations. \square

Overview

- 1 Aim of this lecture
- 2 Introduction
- 3 Weak bisimulation
- 4 Properties of weak bisimilarity
- 5 Observation congruence
- 6 Deciding weak bisimilarity and game interpretation
- 7 Summary

Observation congruence

Observation congruence

[Milner, 1989]

$P, Q \in \text{Prc}$ are **observationally congruent**, denoted $P \approx^c Q$, if for every $\alpha \in \text{Act}$ (including $\alpha = \tau$):

1. if $P \xrightarrow{\alpha} P'$ then there is a sequence of transitions $Q \xrightarrow{\tau} Q_1 \xrightarrow{\alpha} Q_2 \xrightarrow{\tau} Q'$ such that $P' \approx Q_2$
2. if $Q \xrightarrow{\alpha} Q'$ then there is a sequence of transitions $P \xrightarrow{\tau} P_1 \xrightarrow{\alpha} P_2 \xrightarrow{\tau} P'$ such that $P' \approx P_2$.

Remark

\approx^c differs from \approx only in that \approx^c requires τ -moves by P or Q to be mimicked by at least one τ -move in the other process. This only applies to the first step; the successors just have to satisfy $P' \approx Q'$ (and not necessarily $P' \approx^c Q'$).

Properties of observation congruence

Theorem

For every $P, Q \in \text{Prc}$, it holds:

1. $P \sim Q$ implies $P \approx^c Q$, and $P \approx^c Q$ implies $P \approx Q$
2. \approx^c is a CCS congruence
3. $P \approx^c Q$ if and only if $P + R \approx Q + R$ for every $R \in \text{Prc}$
4. \approx^c is an equivalence relation
5. $P \approx Q$ if and only if ($P \approx^c Q$ or $P \approx^c \tau.Q$ or $\tau.P \approx^c Q$)

Proof.

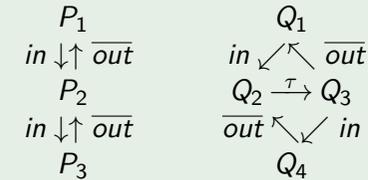
Omitted. □

Note: as \approx implies trace equivalence and is (non- τ) deadlock-sensitive, \approx^c implies trace equivalence and is (non- τ) deadlock-sensitive.

Examples

Example

1. Sequential and parallel two-place buffer:



$P_1 \approx^c Q_1$ since $P_1 \approx Q_1$ and neither P_1 nor Q_1 has initial τ -steps.

2. $\tau.b.\text{nil} \not\approx^c b.\text{nil}$ (since $\tau.b.\text{nil} \xrightarrow{\tau}$ but $b.\text{nil} \not\xrightarrow{\tau}$) thus the counterexample to congruence of \approx for $+$ does not apply.
3. $b.\tau.\text{nil} \approx^c b.\text{nil}$ (since $\tau.\text{nil} \approx \text{nil}$).

Overview

1. Aim of this lecture
2. Introduction
3. Weak bisimulation
4. Properties of weak bisimilarity
5. Observation congruence
6. Deciding weak bisimilarity and game interpretation
7. Summary

Weak bisimilarity as a game

Game rules

In each round the current configuration (s, t) is changed as follows:

1. the attacker chooses one of the processes in the current configuration, say t , and makes an $\xrightarrow{\alpha}$ -move for some $\alpha \in Act$ to t' , say, and
2. the defender must respond by making an $\xRightarrow{\alpha}$ -move in the other process s of the current configuration under the same action α , yielding $s \xRightarrow{\alpha} s'$.

The new pair of processes (s', t') becomes the current configuration. The game continues with another round.

Game results

1. If one player cannot move, the other player wins.
2. If the game can be played *ad infinitum*, the defender wins.

Deciding weak bisimilarity

Checking whether $P \approx Q$ (or $P \approx^c Q$) over finite-state processes can be reduced to checking strong bisimilarity \sim , using a technique called **saturation**.

Intuitively, saturation amounts to:

1. First pre-computing the weak transition relation \Rightarrow , and then
2. Constructing a new pair of finite-state processes whose original transitions are replaced by weak transitions.

The question whether $P \approx Q$ now boils down to checking \sim on the saturated processes. (Details are outside the scope of this lecture.)

As computing \Rightarrow and \sim can be done in polynomial time, $P \approx Q$ can be checked in polynomial time.

Game characterization of weak bisimilarity

Theorem

[Stirling, 1995], [Thomas, 1993]

1. $s \approx t$ iff the defender has a **universal** winning strategy from configuration (s, t) .
2. $s \not\approx t$ iff the attacker has a **universal** winning strategy from configuration (s, t) .

(By means of a universal winning strategy, a player can always win, regardless of how the other player selects her moves.)

Proof.

Similar as for strong bisimilarity. Left as an exercise. □

Overview

- 1 Aim of this lecture
- 2 Introduction
- 3 Weak bisimulation
- 4 Properties of weak bisimilarity
- 5 Observation congruence
- 6 Deciding weak bisimilarity and game interpretation
- 7 Summary

Summary

1. Weak bisimilarity is based on mutual mimicking processes
2. But: τ -actions do not need to be mimicked, as they are internal
3. Weak bisimilarity is not a congruence for choice (+)
4. Observation congruence remedies this by forcing initial τ -actions to be mimicked
5. Divergence is weakly bisimilar to a deadlock process
6. Checking (non-)weak bisimilarity can be done using a two-player game
7. Weak bisimilarity can be determined in polynomial-time