Concurrency Theory

Winter 2025/26

Lecture 7: Bisimulation as a Fixed Point

Thomas Noll, Peter Thiemann
Programming Languages Group
University of Freiburg

https://proglang.github.io/teaching/25ws/ct.html

Thomas Noll, Peter Thiemann

Winter 2025/26

Outline of Lecture 7

Strong Bisimilarity as a Fixed Point

Question: How to (dis-)prove that $P \sim Q$ for $P, Q \in Prc$?

Strong Bisimilarity as a Fixed Point

Question: How to (dis-)prove that $P \sim Q$ for $P, Q \in Prc$?

So far: Game characterisation

Theorem (Game characterisation of bisimulation; cf. Theorem 5.2)

(1) $s \sim t$ iff *the defender has a universal winning strategy* from configuration (s, t).

(2) $s \not\sim t$ 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 their moves.)

Strong Bisimilarity as a Fixed Point

Question: How to (dis-)prove that $P \sim Q$ for $P, Q \in Prc$?

So far: Game characterisation

Theorem (Game characterisation of bisimulation; cf. Theorem 5.2)

(1) $s \sim t$ iff *the defender has a universal winning strategy* from configuration (s, t).

(2) $s \not\sim t$ 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 their moves.)

Goal: Show that $\sim$ can be characterised as the greatest fixed point of a monotonic function on a complete lattice[1].

[1] Later we will use similar methods to give meaning to recursive logical formulae

Outline of Lecture 7

Partial Orders

Definition 7.1 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \Rightarrow d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \Rightarrow d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Partial Orders

Definition 7.1 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \; \subseteq \; D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \Rightarrow d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \Rightarrow d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 7.2

(1) $(\mathbb{N}, \leq)$ is a total order.

Partial Orders

Definition 7.1 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \Rightarrow d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \Rightarrow d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 7.2

(1) $(\mathbb{N}, \leq)$ is a total order.

(2) $(\mathbb{N}, <)$ is not a partial order (since not reflexive).

Partial Orders

Definition 7.1 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \Rightarrow d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \Rightarrow d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 7.2

(1) $(\mathbb{N}, \leq)$ is a total order.

(2) $(\mathbb{N}, <)$ is not a partial order (since not reflexive).

(3) $(2^{\mathbb{N}}, \subseteq)$ is a (non-total) partial order.

Partial Orders

Definition 7.1 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \Rightarrow d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \Rightarrow d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 7.2

(1) $(\mathbb{N}, \leq)$ is a total order.

(2) $(\mathbb{N}, <)$ is not a partial order (since not reflexive).

(3) $(2^{\mathbb{N}}, \subseteq)$ is a (non-total) partial order.

(4) $(\Sigma^*, \sqsubseteq)$ is a (non-total) partial order, where Σ is some alphabet and $\sqsubseteq$ denotes prefix ordering

Upper and Lower Bounds

Definition 7.3 ((Least) upper bounds and (greatest) lower bounds)

Let $(D, \sqsubseteq)$ be a partial order and $T \subseteq D$.

(1) An element $d \in D$ is a upper bound of T if $t \sqsubseteq d$ for every $t \in T$ (notation: $T \sqsubseteq d$). It is a least upper bound (LUB) (or supremum) of T if additionally $d \sqsubseteq d'$ for every upper bound d' of T (notation: $d = \bigsqcup T$).

Upper and Lower Bounds

Definition 7.3 ((Least) upper bounds and (greatest) lower bounds)

Let $(D, \sqsubseteq)$ be a partial order and $T \subseteq D$.

(1) An element $d \in D$ is a upper bound of T if $t \sqsubseteq d$ for every $t \in T$ (notation: $T \sqsubseteq d$). It is a least upper bound (LUB) (or supremum) of T if additionally $d \sqsubseteq d'$ for every upper bound d' of T (notation: $d = \bigsqcup T$).

(2) An element $d \in D$ is a lower bound of T if $d \sqsubseteq t$ for every $t \in T$ (notation: $d \sqsubseteq T$). It is a greatest lower bound (GLB) (or infimum) of T if $d' \sqsubseteq d$ for every lower bound d' of T (notation: $d = \bigsqcap T$).

Upper and Lower Bounds

Definition 7.3 ((Least) upper bounds and (greatest) lower bounds)

Let $(D, \sqsubseteq)$ be a partial order and $T \subseteq D$.

(1) An element $d \in D$ is a upper bound of T if $t \sqsubseteq d$ for every $t \in T$ (notation: $T \sqsubseteq d$). It is a least upper bound (LUB) (or supremum) of T if additionally $d \sqsubseteq d'$ for every upper bound d' of T (notation: $d = \bigsqcup T$).

(2) An element $d \in D$ is a lower bound of T if $d \sqsubseteq t$ for every $t \in T$ (notation: $d \sqsubseteq T$). It is a greatest lower bound (GLB) (or infimum) of T if $d' \sqsubseteq d$ for every lower bound d' of T (notation: $d = \bigsqcap T$).

Example 7.4

(1) $T \subseteq \mathbb{N}$ has a LUB/GLB in $(\mathbb{N}, \leq)$ iff it is finite/non-empty.

Upper and Lower Bounds

Definition 7.3 ((Least) upper bounds and (greatest) lower bounds)

Let $(D, \sqsubseteq)$ be a partial order and $T \subseteq D$.

(1) An element $d \in D$ is a upper bound of T if $t \sqsubseteq d$ for every $t \in T$ (notation: $T \sqsubseteq d$). It is a least upper bound (LUB) (or supremum) of T if additionally $d \sqsubseteq d'$ for every upper bound d' of T (notation: $d = \bigsqcup T$).

(2) An element $d \in D$ is a lower bound of T if $d \sqsubseteq t$ for every $t \in T$ (notation: $d \sqsubseteq T$). It is a greatest lower bound (GLB) (or infimum) of T if $d' \sqsubseteq d$ for every lower bound d' of T (notation: $d = \bigsqcap T$).

Example 7.4

(1) $T \subseteq \mathbb{N}$ has a LUB/GLB in $(\mathbb{N}, \leq)$ iff it is finite/non-empty.

(2) In $(2^{\mathbb{N}}, \subseteq)$, every subset $T \subseteq 2^{\mathbb{N}}$ has an LUB and a GLB:

$$\bigsqcup T = \bigcup T \quad \text{and} \quad \bigsqcap T = \bigcap T.$$

Complete Lattices I

Definition 7.5 (Complete lattice)

A complete lattice is a partial order $(D, \sqsubseteq)$ such that all subsets of D have LUBs and GLBs.

Remark

In a complete lattice

$$\bot := \bigsqcup \emptyset \; (= \bigsqcap D) \qquad \text{and} \qquad \top := \bigsqcap \emptyset \; (= \bigsqcup D)$$

respectively denote the least and greatest element of D.

Complete Lattices I

Definition 7.5 (Complete lattice)

A complete lattice is a partial order $(D, \sqsubseteq)$ such that all subsets of D have LUBs and GLBs.

Remark

In a complete lattice

$$\bot := \bigsqcup \emptyset \; (= \bigsqcap D) \qquad \text{and} \qquad \top := \bigsqcap \emptyset \; (= \bigsqcup D)$$

respectively denote the least and greatest element of D.

Example 7.6

(1) $(\mathbb{N}, \leq)$ is not a complete lattice as, e.g., $\mathbb{N}$ does not have a LUB.

Complete Lattices I

Definition 7.5 (Complete lattice)

A complete lattice is a partial order $(D, \sqsubseteq)$ such that all subsets of D have LUBs and GLBs.

Remark

In a complete lattice

$$\bot := \bigsqcup \emptyset \; (= \bigsqcap D) \qquad \text{and} \qquad \top := \bigsqcap \emptyset \; (= \bigsqcup D)$$

respectively denote the least and greatest element of D.

Example 7.6

(1) $(\mathbb{N}, \leq)$ is not a complete lattice as, e.g., $\mathbb{N}$ does not have a LUB.

(2) $(\mathbb{N} \cup \{\infty\}, \leq)$ with $n \leq \infty$ for all $n \in \mathbb{N}$ is a complete lattice.

Complete Lattices I

Definition 7.5 (Complete lattice)

A complete lattice is a partial order $(D, \sqsubseteq)$ such that all subsets of D have LUBs and GLBs.

Remark

In a complete lattice

$$\bot := \bigsqcup \emptyset \; (= \bigsqcap D) \qquad \text{and} \qquad \top := \bigsqcap \emptyset \; (= \bigsqcup D)$$

respectively denote the least and greatest element of D.

Example 7.6

(1) $(\mathbb{N}, \leq)$ is not a complete lattice as, e.g., $\mathbb{N}$ does not have a LUB.

(2) $(\mathbb{N} \cup \{\infty\}, \leq)$ with $n \leq \infty$ for all $n \in \mathbb{N}$ is a complete lattice.

(3) $(2^{\mathbb{N}}, \subseteq)$ is a complete lattice (cf. Example 7.4).

Complete Lattices II

Lemma 7.7

Let S *be some (finite or infinite) set. Then* $(2^S, \subseteq)$ *is a complete lattice with*

- $\bigsqcup \mathcal{T} = \bigcup \mathcal{T} = \bigcup_{T \in \mathcal{T}} T$ *for all* $\mathcal{T} \subseteq 2^S$
- $\bigsqcap \mathcal{T} = \bigcap \mathcal{T} = \bigcap_{T \in \mathcal{T}} T$ *for all* $\mathcal{T} \subseteq 2^S$

Complete Lattices II

Lemma 7.7

Let S be some (finite or infinite) set. Then $(2^S, \subseteq)$ is a complete lattice with

- $\bigsqcup \mathcal{T} = \bigcup \mathcal{T} = \bigcup_{T \in \mathcal{T}} T$ *for all* $\mathcal{T} \subseteq 2^S$
- $\bigsqcap \mathcal{T} = \bigcap \mathcal{T} = \bigcap_{T \in \mathcal{T}} T$ *for all* $\mathcal{T} \subseteq 2^S$
- $\bot = \bigsqcup \emptyset = \bigsqcap 2^S = \emptyset$
- $\top = \bigsqcap \emptyset = \bigsqcup 2^S = S$

Proof.

omitted □

Outline of Lecture 7

Definition 7.8 (Fixed point)

Let D be some domain and $f : D \to D$. An element $d \in D$ is

- a fixed point of f if $f(d) = d$;
- a pre-fixed point of f if $f(d) \sqsubseteq d$;
- a post-fixed point of f if $d \sqsubseteq f(d)$.

Fixed Points

Definition 7.8 (Fixed point)

Let D be some domain and $f : D \to D$. An element $d \in D$ is

- a fixed point of f if $f(d) = d$;
- a pre-fixed point of f if $f(d) \sqsubseteq d$;
- a post-fixed point of f if $d \sqsubseteq f(d)$.

Example 7.9

(1) The (only) fixed points of $f_1 : \mathbb{N} \to \mathbb{N} : n \mapsto n^2$ are 0 and 1

Fixed Points

Definition 7.8 (Fixed point)

Let D be some domain and $f : D \to D$. An element $d \in D$ is

- a fixed point of f if $f(d) = d$;
- a pre-fixed point of f if $f(d) \sqsubseteq d$;
- a post-fixed point of f if $d \sqsubseteq f(d)$.

Example 7.9

(1) The (only) fixed points of $f_1 : \mathbb{N} \to \mathbb{N} : n \mapsto n^2$ are 0 and 1

(2) A subset $T \subseteq \mathbb{N}$ is a fixed point of $f_2 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : T \mapsto T \cup \{1, 2\}$ iff $\{1, 2\} \subseteq T$

Monotonicity of Functions

Definition 7.10 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders. A function $f : D \to D'$ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \Rightarrow f(d_1) \sqsubseteq' f(d_2).$$

Monotonicity of Functions

Definition 7.10 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders. A function $f : D \to D'$ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \Rightarrow f(d_1) \sqsubseteq' f(d_2).$$

Example 7.11

(1) $f_1 : \mathbb{N} \to \mathbb{N} : n \mapsto n^2$ is monotonic w.r.t. $(\mathbb{N}, \leq)$

Monotonicity of Functions

Definition 7.10 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders. A function $f : D \to D'$ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \Rightarrow f(d_1) \sqsubseteq' f(d_2).$$

Example 7.11

(1) $f_1 : \mathbb{N} \to \mathbb{N} : n \mapsto n^2$ is monotonic w.r.t. $(\mathbb{N}, \leq)$

(2) $f_2 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : T \mapsto T \cup \{1, 2\}$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$

Monotonicity of Functions

Definition 7.10 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders. A function $f : D \to D'$ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \Rightarrow f(d_1) \sqsubseteq' f(d_2).$$

Example 7.11

(1) $f_1 : \mathbb{N} \to \mathbb{N} : n \mapsto n^2$ is monotonic w.r.t. $(\mathbb{N}, \leq)$

(2) $f_2 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : T \mapsto T \cup \{1, 2\}$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$

(3) Let $\mathcal{T} := \{T \subseteq \mathbb{N} \mid T \text{ finite}\}$.
Then $f_3 : \mathcal{T} \to \mathbb{N} : T \mapsto \sum_{n \in T} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.

Monotonicity of Functions

Definition 7.10 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders. A function $f : D \to D'$ is called monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \Rightarrow f(d_1) \sqsubseteq' f(d_2).$$

Example 7.11

(1) $f_1 : \mathbb{N} \to \mathbb{N} : n \mapsto n^2$ is monotonic w.r.t. $(\mathbb{N}, \leq)$

(2) $f_2 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : T \mapsto T \cup \{1, 2\}$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$

(3) Let $\mathcal{T} := \{T \subseteq \mathbb{N} \mid T \text{ finite}\}$.
Then $f_3 : \mathcal{T} \to \mathbb{N} : T \mapsto \sum_{n \in T} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.

(4) $f_4 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : T \mapsto \mathbb{N} \setminus T$ is not monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$
(since, e.g., $\emptyset \subseteq \mathbb{N}$ but $f_4(\emptyset) = \mathbb{N} \not\subseteq f_4(\mathbb{N}) = \emptyset$).

The Fixed-Point Theorem I

Alfred Tarski (1901–1983)

Theorem 7.12 (Tarski's fixed-point theorem)

Let $(D, \sqsubseteq)$ be a complete lattice and $f : D \to D$ monotonic. Then f has a least fixed point $\mathrm{fix}(f)$ and a greatest fixed point $\mathrm{FIX}(f)$, which are given by

$$\mathrm{fix}(f) := \bigsqcap\{d \in D \mid f(d) \sqsubseteq d\} \qquad \textit{(GLB of all pre-fixed points of } f\textit{)}$$

$$\mathrm{FIX}(f) := \bigsqcup\{d \in D \mid d \sqsubseteq f(d)\} \qquad \textit{(LUB of all post-fixed points of } f\textit{)}$$

The Fixed-Point Theorem II

Example 7.13 (cf. Example 7.9)

- Let $(D, \sqsubseteq) := (2^{\mathbb{N}}, \subseteq)$ and $f : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : T \mapsto T \cup \{1, 2\}$.
- As seen in Example 7.9(2): $f(T) = T$ iff $\{1, 2\} \subseteq T$.

The Fixed-Point Theorem II

Example 7.13 (cf. Example 7.9)

- Let $(D, \sqsubseteq) := (2^{\mathbb{N}}, \subseteq)$ and $f : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : T \mapsto T \cup \{1, 2\}$.
- As seen in Example 7.9(2): $f(T) = T$ iff $\{1, 2\} \subseteq T$.
- Theorem 7.12 for fix:

$$
\begin{aligned}
\mathrm{fix}(f) &= \bigsqcap\{d \in D \mid f(d) \sqsubseteq d\} && \text{(Theorem 7.12)}\\
&= \bigcap\{T \subseteq \mathbb{N} \mid f(T) \subseteq T\} && \text{(Lemma 7.7)}\\
&= \bigcap\{T \subseteq \mathbb{N} \mid T \cup \{1, 2\} \subseteq T\} && \text{(Def. } f\text{)}\\
&= \bigcap\{T \subseteq \mathbb{N} \mid \{1, 2\} \subseteq T\}\\
&= \{1, 2\}
\end{aligned}
$$

The Fixed-Point Theorem II

Example 7.13 (cf. Example 7.9)

- Let $(D, \sqsubseteq) := (2^{\mathbb{N}}, \subseteq)$ and $f : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : T \mapsto T \cup \{1,2\}$.
- As seen in Example 7.9(2): $f(T) = T$ iff $\{1,2\} \subseteq T$.
- Theorem 7.12 for fix:

$$
\begin{aligned}
\mathrm{fix}(f) &= \bigsqcap\{d \in D \mid f(d) \sqsubseteq d\} && \text{(Theorem 7.12)}\\
&= \bigcap\{T \subseteq \mathbb{N} \mid f(T) \subseteq T\} && \text{(Lemma 7.7)}\\
&= \bigcap\{T \subseteq \mathbb{N} \mid T \cup \{1,2\} \subseteq T\} && \text{(Def. } f\text{)}\\
&= \bigcap\{T \subseteq \mathbb{N} \mid \{1,2\} \subseteq T\}\\
&= \{1,2\}
\end{aligned}
$$

- Theorem 7.12 for FIX:

$$
\begin{aligned}
\mathrm{FIX}(f) &= \bigsqcup\{d \in D \mid d \sqsubseteq f(d)\} && \text{(Theorem 7.12)}\\
&= \bigcup\{T \subseteq \mathbb{N} \mid T \subseteq f(T)\} && \text{(Lemma 7.7)}\\
&= \bigcup\{T \subseteq \mathbb{N} \mid T \subseteq T \cup \{1,2\}\} && \text{(Def. } f\text{)}\\
&= \bigcup 2^{\mathbb{N}}\\
&= \mathbb{N}
\end{aligned}
$$

The Fixed-Point Theorem III

Proof (Theorem 7.12).

First we show that $\mathrm{fix}(f) = \bigsqcap\{d \in D \mid f(d) \sqsubseteq d\}$ has the required properties:

(1) $\mathrm{fix}(f)$ is a fixed point, i.e., $f(\mathrm{fix}(f)) = \mathrm{fix}(f)$.

(2) $\mathrm{fix}(f)$ is “least”, i.e., $\forall d \in D : f(d) = d \Rightarrow \mathrm{fix}(f) \sqsubseteq d$.

The Fixed-Point Theorem III

Proof (Theorem 7.12).

First we show that $\text{fix}(f) = \bigsqcap\{d \in D \mid f(d) \sqsubseteq d\}$ has the required properties:

(1) $\text{fix}(f)$ is a fixed point, i.e., $f(\text{fix}(f)) = \text{fix}(f)$.

(2) $\text{fix}(f)$ is "least", i.e., $\forall d \in D : f(d) = d \Rightarrow \text{fix}(f) \sqsubseteq d$.

Let $A := \{d \in D \mid f(d) \sqsubseteq d\}$ (and thus $\text{fix}(f) = \bigsqcap A$).

(1) We prove both directions separately:

$$
\begin{array}{lll}
f(\text{fix}(f)) \sqsubseteq \text{fix}(f): & \text{fix}(f) = \bigsqcap A & \text{(def. fix}(f)\text{)} \\
& \Rightarrow \forall a \in A : \text{fix}(f) \sqsubseteq a & \text{(def. } \bigsqcap\text{)} \\
& \Rightarrow \forall a \in A : f(\text{fix}(f)) \sqsubseteq f(a) \sqsubseteq a & (f \text{ monotonic, def. } A) \\
& \Rightarrow f(\text{fix}(f)) \sqsubseteq \bigsqcap A & \\
& \Rightarrow f(\text{fix}(f)) \sqsubseteq \text{fix}(f) & (\text{fix}(f) = \bigsqcap A)
\end{array}
$$

The Fixed-Point Theorem III

Proof (Theorem 7.12).

First we show that $\mathrm{fix}(f) = \bigsqcap\{d \in D \mid f(d) \sqsubseteq d\}$ has the required properties:

(1) $\mathrm{fix}(f)$ is a fixed point, i.e., $f(\mathrm{fix}(f)) = \mathrm{fix}(f)$.

(2) $\mathrm{fix}(f)$ is "least", i.e., $\forall d \in D : f(d) = d \Rightarrow \mathrm{fix}(f) \sqsubseteq d$.

Let $A := \{d \in D \mid f(d) \sqsubseteq d\}$ (and thus $\mathrm{fix}(f) = \bigsqcap A$).

(1) We prove both directions separately:

$f(\mathrm{fix}(f)) \sqsubseteq \mathrm{fix}(f)$:

$$\begin{aligned} & \mathrm{fix}(f) = \bigsqcap A && \text{(def. } \mathrm{fix}(f)\text{)} \\ \Rightarrow\ & \forall a \in A : \mathrm{fix}(f) \sqsubseteq a && \text{(def. } \bigsqcap\text{)} \\ \Rightarrow\ & \forall a \in A : f(\mathrm{fix}(f)) \sqsubseteq f(a) \sqsubseteq a && \text{(} f \text{ monotonic, def. } A\text{)} \\ \Rightarrow\ & f(\mathrm{fix}(f)) \sqsubseteq \bigsqcap A && \\ \Rightarrow\ & f(\mathrm{fix}(f)) \sqsubseteq \mathrm{fix}(f) && \text{(} \mathrm{fix}(f) = \bigsqcap A\text{)} \end{aligned}$$

$f(\mathrm{fix}(f)) \sqsupseteq \mathrm{fix}(f)$:

$$\begin{aligned} & f(\mathrm{fix}(f)) \sqsubseteq \mathrm{fix}(f) && \text{(as shown)} \\ \Rightarrow\ & f(f(\mathrm{fix}(f))) \sqsubseteq f(\mathrm{fix}(f)) && \text{(} f \text{ monotonic)} \\ \Rightarrow\ & f(\mathrm{fix}(f)) \in A && \text{(def. } A\text{)} \\ \Rightarrow\ & \mathrm{fix}(f) \sqsubseteq f(\mathrm{fix}(f)) && \text{(} \mathrm{fix}(f) = \bigsqcap A\text{)} \end{aligned}$$

The Fixed-Point Theorem III

Proof (Theorem 7.12).

First we show that $\mathrm{fix}(f) = \bigsqcap\{d \in D \mid f(d) \sqsubseteq d\}$ has the required properties:

(1) $\mathrm{fix}(f)$ is a fixed point, i.e., $f(\mathrm{fix}(f)) = \mathrm{fix}(f)$.

(2) $\mathrm{fix}(f)$ is "least", i.e., $\forall d \in D : f(d) = d \Rightarrow \mathrm{fix}(f) \sqsubseteq d$.

Let $A := \{d \in D \mid f(d) \sqsubseteq d\}$ (and thus $\mathrm{fix}(f) = \bigsqcap A$).

(2) Let $d \in D$ such that $f(d) = d$.

$\Rightarrow f(d) \sqsubseteq d$

$\Rightarrow d \in A$ (def. A)

$\Rightarrow \mathrm{fix}(f) \sqsubseteq d$ ($\mathrm{fix}(f) = \bigsqcap A$)

The Fixed-Point Theorem III

Proof (Theorem 7.12).

First we show that $\mathrm{fix}(f) = \bigsqcap\{d \in D \mid f(d) \sqsubseteq d\}$ has the required properties:

(1) $\mathrm{fix}(f)$ is a fixed point, i.e., $f(\mathrm{fix}(f)) = \mathrm{fix}(f)$.

(2) $\mathrm{fix}(f)$ is “least”, i.e., $\forall d \in D : f(d) = d \Rightarrow \mathrm{fix}(f) \sqsubseteq d$.

$\mathrm{FIX}(f) = \bigsqcup\{d \in D \mid d \sqsubseteq f(d)\}$ is greatest fixed point of f: analogously □

Outline of Lecture 7

The Fixed-Point Theorem for Finite Lattices I

Theorem 7.14 (Fixed-point theorem for finite lattices)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

The Fixed-Point Theorem for Finite Lattices I

Theorem 7.14 (Fixed-point theorem for finite lattices)

Let $(D, \sqsubseteq)$ *be a finite complete lattice and* $f : D \to D$ *monotonic. Then*

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

for some $m, M \in \mathbb{N}$ *where* $f^0(d) := d$ *and* $f^{k+1}(d) := f(f^k(d))$.

Example 7.15

- Let $f : 2^{\{0,1,2\}} \to 2^{\{0,1,2\}} : T \mapsto T \cup \{1\} \setminus \{2\}$ (monotonic on $(2^{\{0,1,2\}}, \subseteq)$)

The Fixed-Point Theorem for Finite Lattices I

Theorem 7.14 (Fixed-point theorem for finite lattices)

Let $(D, \sqsubseteq)$ *be a* *finite* *complete lattice and* $f : D \to D$ *monotonic. Then*

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

for some $m, M \in \mathbb{N}$ *where* $f^0(d) := d$ *and* $f^{k+1}(d) := f(f^k(d))$.

Example 7.15

- Let $f : 2^{\{0,1,2\}} \to 2^{\{0,1,2\}} : T \mapsto T \cup \{1\} \setminus \{2\}$ (monotonic on $(2^{\{0,1,2\}}, \subseteq)$)
- $f^0(\bot) = \emptyset$, $f^1(\bot) = \{1\}$, $f^2(\bot) = \{1\} = f^1(\bot)$
  $\Rightarrow \mathrm{fix}(f) = \{1\}$ after $m = 1$ iteration

The Fixed-Point Theorem for Finite Lattices I

Theorem 7.14 (Fixed-point theorem for finite lattices)

Let $(D, \sqsubseteq)$ *be a finite complete lattice and* $f : D \to D$ *monotonic. Then*

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

for some $m, M \in \mathbb{N}$ *where* $f^0(d) := d$ *and* $f^{k+1}(d) := f(f^k(d))$.

Example 7.15

- Let $f : 2^{\{0,1,2\}} \to 2^{\{0,1,2\}} : T \mapsto T \cup \{1\} \setminus \{2\}$ (monotonic on $(2^{\{0,1,2\}}, \subseteq)$)
- $f^0(\bot) = \emptyset$, $f^1(\bot) = \{1\}$, $f^2(\bot) = \{1\} = f^1(\bot)$
  $\Rightarrow \mathrm{fix}(f) = \{1\}$ after $m = 1$ iteration
- $f^0(\top) = \{0, 1, 2\}$, $f^1(\top) = \{0, 1\}$, $f^2(\top) = \{0, 1\} = f^1(\top)$
  $\Rightarrow \mathrm{FIX}(f) = \{0, 1\}$ after $M = 1$ iteration

Proof (Theorem 7.14).

We first have to show that there ex. $m \in \mathbb{N}$ such that $\mathrm{fix}(f) = f^m(\perp)$:

The Fixed-Point Theorem for Finite Lattices II

Proof (Theorem 7.14).

We first have to show that there ex. $m \in \mathbb{N}$ such that $\mathrm{fix}(f) = f^m(\perp)$:

- Since $\perp = \bigsqcup \emptyset$ is the least element of D, $\perp \sqsubseteq f(\perp)$.

The Fixed-Point Theorem for Finite Lattices II

Proof (Theorem 7.14).

We first have to show that there ex. $m \in \mathbb{N}$ such that $\mathrm{fix}(f) = f^m(\perp)$:

- Since $\perp = \bigsqcup \emptyset$ is the least element of D, $\perp \sqsubseteq f(\perp)$.
- By monotonicity of f, we can iteratively apply f to this inequation, which yields the "chain"

$$\perp \sqsubseteq f(\perp) \sqsubseteq \ldots \sqsubseteq f^i(\perp) \sqsubseteq f^{i+1}(\perp) \sqsubseteq \ldots$$

The Fixed-Point Theorem for Finite Lattices II

Proof (Theorem 7.14).

We first have to show that there ex. $m \in \mathbb{N}$ such that $\mathrm{fix}(f) = f^m(\bot)$:

- Since $\bot = \bigsqcup \emptyset$ is the least element of D, $\bot \sqsubseteq f(\bot)$.
- By monotonicity of f, we can iteratively apply f to this inequation, which yields the "chain"

$$\bot \sqsubseteq f(\bot) \sqsubseteq \ldots \sqsubseteq f^i(\bot) \sqsubseteq f^{i+1}(\bot) \sqsubseteq \ldots$$

- By finiteness of D, there ex. $m \in \mathbb{N}$ such that $f^m(\bot) = f^{m+1}(\bot) = f^{m+2}(\bot) = \ldots$

The Fixed-Point Theorem for Finite Lattices II

Proof (Theorem 7.14).

We first have to show that there ex. $m \in \mathbb{N}$ such that $\mathrm{fix}(f) = f^m(\bot)$:

- Since $\bot = \bigsqcup \emptyset$ is the least element of D, $\bot \sqsubseteq f(\bot)$.
- By monotonicity of f, we can iteratively apply f to this inequation, which yields the "chain"
$$\bot \sqsubseteq f(\bot) \sqsubseteq \ldots \sqsubseteq f^i(\bot) \sqsubseteq f^{i+1}(\bot) \sqsubseteq \ldots$$
- By finiteness of D, there ex. $m \in \mathbb{N}$ such that $f^m(\bot) = f^{m+1}(\bot) = f^{m+2}(\bot) = \ldots$
- Thus $f(f^m(\bot)) = f^m(\bot)$, and hence $f^m(\bot)$ is a fixed point of f.

The Fixed-Point Theorem for Finite Lattices II

Proof (Theorem 7.14).

We first have to show that there ex. $m \in \mathbb{N}$ such that $\mathrm{fix}(f) = f^m(\perp)$:

- Since $\perp = \bigsqcup \emptyset$ is the least element of D, $\perp \sqsubseteq f(\perp)$.
- By monotonicity of f, we can iteratively apply f to this inequation, which yields the "chain"

$$\perp \sqsubseteq f(\perp) \sqsubseteq \ldots \sqsubseteq f^i(\perp) \sqsubseteq f^{i+1}(\perp) \sqsubseteq \ldots$$

- By finiteness of D, there ex. $m \in \mathbb{N}$ such that $f^m(\perp) = f^{m+1}(\perp) = f^{m+2}(\perp) = \ldots$
- Thus $f(f^m(\perp)) = f^m(\perp)$, and hence $f^m(\perp)$ is a fixed point of f.
- To show the minimality of $f^m(\perp)$, let $f(d) = d$ be another fixed point. Since $\perp \sqsubseteq d$, monotonicity of f yields $f(\perp) \sqsubseteq f(d) = d$, which can be iterated to show that $f^m(\perp) \sqsubseteq d$.

The Fixed-Point Theorem for Finite Lattices II

Proof (Theorem 7.14).

We first have to show that there ex. $m \in \mathbb{N}$ such that $\mathrm{fix}(f) = f^m(\perp)$:

- Since $\perp = \bigsqcup \emptyset$ is the least element of D, $\perp \sqsubseteq f(\perp)$.
- By monotonicity of f, we can iteratively apply f to this inequation, which yields the "chain"
$$\perp \sqsubseteq f(\perp) \sqsubseteq \ldots \sqsubseteq f^i(\perp) \sqsubseteq f^{i+1}(\perp) \sqsubseteq \ldots$$
- By finiteness of D, there ex. $m \in \mathbb{N}$ such that $f^m(\perp) = f^{m+1}(\perp) = f^{m+2}(\perp) = \ldots$
- Thus $f(f^m(\perp)) = f^m(\perp)$, and hence $f^m(\perp)$ is a fixed point of f.
- To show the minimality of $f^m(\perp)$, let $f(d) = d$ be another fixed point. Since $\perp \sqsubseteq d$, monotonicity of f yields $f(\perp) \sqsubseteq f(d) = d$, which can be iterated to show that $f^m(\perp) \sqsubseteq d$.

Altogether, $f^m(\perp) = \mathrm{fix}(f)$.

The Fixed-Point Theorem for Finite Lattices II

Proof (Theorem 7.14).

We first have to show that there ex. $m \in \mathbb{N}$ such that $\mathrm{fix}(f) = f^m(\bot)$:

- Since $\bot = \bigsqcup \emptyset$ is the least element of D, $\bot \sqsubseteq f(\bot)$.
- By monotonicity of f, we can iteratively apply f to this inequation, which yields the "chain"

$$\bot \sqsubseteq f(\bot) \sqsubseteq \ldots \sqsubseteq f^i(\bot) \sqsubseteq f^{i+1}(\bot) \sqsubseteq \ldots$$

- By finiteness of D, there ex. $m \in \mathbb{N}$ such that $f^m(\bot) = f^{m+1}(\bot) = f^{m+2}(\bot) = \ldots$
- Thus $f(f^m(\bot)) = f^m(\bot)$, and hence $f^m(\bot)$ is a fixed point of f.
- To show the minimality of $f^m(\bot)$, let $f(d) = d$ be another fixed point. Since $\bot \sqsubseteq d$, monotonicity of f yields $f(\bot) \sqsubseteq f(d) = d$, which can be iterated to show that $f^m(\bot) \sqsubseteq d$.

Altogether, $f^m(\bot) = \mathrm{fix}(f)$.

$f^M(\top) = \mathrm{FIX}(f)$ can be proven analogously. $\square$

Outline of Lecture 7

Strong Bisimilarity Revisited

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

Strong Bisimilarity Revisited

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

What about checking $\sim$ between two processes?

Strong Bisimilarity Revisited

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

What about checking $\sim$ between two processes?

Definition (Strong bisimilarity; cf. Definition 4.2)

Processes $P, Q \in Prc$ are strongly bisimilar, denoted $P \sim Q$, iff there is a strong bisimulation ρ with $P \, \rho \, Q$. Thus,

$$\sim \; = \bigcup \{\rho \subseteq Prc \times Prc \mid \rho \text{ is a strong bisimulation}\}.$$

Relation $\sim$ is called strong bisimilarity.

Strong Bisimilarity Revisited

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

What about checking $\sim$ between two processes?

Definition (Strong bisimilarity; cf. Definition 4.2)

Processes $P, Q \in Prc$ are strongly bisimilar, denoted $P \sim Q$, iff there is a strong bisimulation ρ with $P \; \rho \; Q$. Thus,

$$\sim \; = \bigcup \{\rho \subseteq Prc \times Prc \mid \rho \text{ is a strong bisimulation}\}.$$

Relation $\sim$ is called strong bisimilarity.

By Lemma 7.7, $(2^{Prc \times Prc}, \subseteq)$ is a complete lattice with $\bigcup$ and $\bigcap$ as least upper and greatest lower bound, respectively.

Strong Bisimilarity Revisited

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

What about checking $\sim$ between two processes?

Definition (Strong bisimilarity; cf. Definition 4.2)

Processes $P, Q \in Prc$ are strongly bisimilar, denoted $P \sim Q$, iff there is a strong bisimulation ρ with $P \; \rho \; Q$. Thus,

$$\sim = \bigcup \{\rho \subseteq Prc \times Prc \mid \rho \text{ is a strong bisimulation}\}.$$

Relation $\sim$ is called strong bisimilarity.

By Lemma 7.7, $(2^{Prc \times Prc}, \subseteq)$ is a complete lattice with $\bigcup$ and $\bigcap$ as least upper and greatest lower bound, respectively.

Show: $\sim$ can be characterised as the greatest fixed point of a monotonic function on this lattice.

Fixed-Point Characterisation of Strong Bisimilarity I

Definition 7.16 (Function on relations)

Let $\rho \subseteq Prc \times Prc$. Let $\mathcal{F} : 2^{Prc \times Prc} \to 2^{Prc \times Prc}$ be defined as follows:
for every $P, Q \in Prc$, $(P, Q) \in \mathcal{F}(\rho)$ iff

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \, \rho \, Q'$ and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \, \rho \, Q'$.

Fixed-Point Characterisation of Strong Bisimilarity I

Definition 7.16 (Function on relations)

Let $\rho \subseteq Prc \times Prc$. Let $\mathcal{F} : 2^{Prc \times Prc} \to 2^{Prc \times Prc}$ be defined as follows:
for every $P, Q \in Prc$, $(P, Q) \in \mathcal{F}(\rho)$ iff

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \rho Q'$ and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \rho Q'$.

Intuition: $\mathcal{F}(\rho)$ contains all pairs of processes from which, in one round of the bisimulation game, the defender can ensure that the players reach a configuration contained in ρ. Clearly, $\mathcal{F}$ is monotonic.

Fixed-Point Characterisation of Strong Bisimilarity I

Definition 7.16 (Function on relations)

Let $\rho \subseteq Prc \times Prc$. Let $\mathcal{F} : 2^{Prc \times Prc} \to 2^{Prc \times Prc}$ be defined as follows:
for every $P, Q \in Prc$, $(P, Q) \in \mathcal{F}(\rho)$ iff

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \, \rho \, Q'$ and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \, \rho \, Q'$.

Intuition: $\mathcal{F}(\rho)$ contains all pairs of processes from which, in one round of the bisimulation game, the defender can ensure that the players reach a configuration contained in ρ. Clearly, $\mathcal{F}$ is monotonic.

Corollary 7.17

ρ is a strong bisimulation iff $\rho \subseteq \mathcal{F}(\rho)$, and thus:

$$\sim = \bigcup \{\rho \in Prc \times Prc \mid \rho \subseteq \mathcal{F}(\rho)\}.$$

Fixed-Point Characterisation of Strong Bisimilarity II

Corollary

ρ is a strong bisimulation iff $\rho \subseteq \mathcal{F}(\rho)$, and thus:

$$\sim \; = \bigcup \{\rho \in \mathit{Prc} \times \mathit{Prc} \mid \rho \subseteq \mathcal{F}(\rho)\}.$$

Thus: $\sim$ is the LUB of all post-fixed points of $\mathcal{F}$.

Fixed-Point Characterisation of Strong Bisimilarity II

Corollary

ρ is a strong bisimulation iff $\rho \subseteq \mathcal{F}(\rho)$, and thus:

$$\sim = \bigcup \{\rho \in Prc \times Prc \mid \rho \subseteq \mathcal{F}(\rho)\}.$$

Thus: $\sim$ is the LUB of all post-fixed points of $\mathcal{F}$.

Theorem (Tarski's fixed-point theorem; cf. Theorem 7.12)

Let $(D, \sqsubseteq)$ be a complete lattice and $f : D \to D$ monotonic. Then f has a least fixed point $\mathrm{fix}(f)$ and a greatest fixed point $\mathrm{FIX}(f)$, which are given by

$$\mathrm{fix}(f) := \bigsqcap \{d \in D \mid f(d) \sqsubseteq d\} \qquad \text{(GLB of all pre-fixed points of } f\text{)}$$

$$\mathrm{FIX}(f) := \bigsqcup \{d \in D \mid d \sqsubseteq f(d)\} \qquad \text{(LUB of all post-fixed points of } f\text{)}$$

Application to Finite LTS ("Partition Refinement")

Theorem (Fixed-point theorem for finite lattices; cf. Theorem 7.14)

Let $(D, \sqsubseteq)$ *be a finite complete lattice and* $f : D \to D$ *monotonic. Then*

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

for some $m, M \in \mathbb{N}$ *where* $f^0(d) := d$ *and* $f^{k+1}(d) := f(f^k(d))$.

Application to Finite LTS ("Partition Refinement")

Theorem (Fixed-point theorem for finite lattices; cf. Theorem 7.14)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\text{fix}(f) = f^m(\bot) \quad \textit{and} \quad \text{FIX}(f) = f^M(\top)$$

for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

Corollary 7.18

For finite-state process P with state space S, $\sim$ can be computed by:

$$\begin{array}{rcl} \sim & = & \bigcap_{i=0}^{\infty} \sim_i \quad \textit{where} \\ \sim_0 & := & S \times S \\ \sim_{i+1} & := & \mathcal{F}(\sim_i) \end{array}$$

Application to Finite LTS ("Partition Refinement")

Theorem (Fixed-point theorem for finite lattices; cf. Theorem 7.14)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

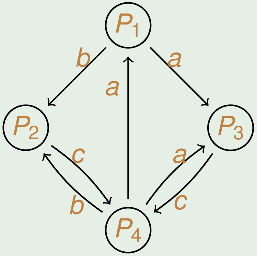
for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

Corollary 7.18

For finite-state process P with state space S, $\sim$ can be computed by:

$$\begin{aligned} \sim \quad &= \quad \textstyle\bigcap_{i=0}^{\infty} \sim_i \qquad \textit{where} \\ \sim_0 \quad &:= \quad S \times S \\ \sim_{i+1} \quad &:= \quad \mathcal{F}(\sim_i) \end{aligned}$$

Example 7.19



Equivalence classes:

$$\sim_0 \quad = \{\{P_1, P_2, P_3, P_4\}\}$$

Application to Finite LTS ("Partition Refinement")

Theorem (Fixed-point theorem for finite lattices; cf. Theorem 7.14)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

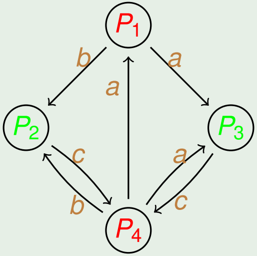
for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

Corollary 7.18

For finite-state process P with state space S, $\sim$ can be computed by:

$$\begin{array}{rcl} \sim & = & \bigcap_{i=0}^{\infty} \sim_i \qquad \textit{where} \\ \sim_0 & := & S \times S \\ \sim_{i+1} & := & \mathcal{F}(\sim_i) \end{array}$$

Example 7.19



Equivalence classes:

$$\sim_0 \; = \{\{P_1, P_2, P_3, P_4\}\}$$
$$\sim_1 \; = \{\{P_1, P_4\}, \{P_2, P_3\}\}$$

Application to Finite LTS ("Partition Refinement")

Theorem (Fixed-point theorem for finite lattices; cf. Theorem 7.14)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

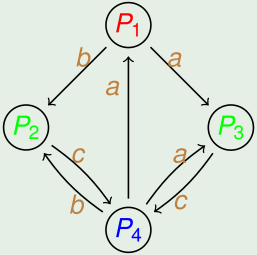
for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

Corollary 7.18

For finite-state process P with state space S, $\sim$ can be computed by:

$$\begin{aligned} \sim \; &= \; \textstyle\bigcap_{i=0}^{\infty} \sim_i \quad \textit{where} \\ \sim_0 \; &:= \; S \times S \\ \sim_{i+1} \; &:= \; \mathcal{F}(\sim_i) \end{aligned}$$

Example 7.19



Equivalence classes:

$$\begin{aligned} \sim_0 &= \{\{P_1, P_2, P_3, P_4\}\} \\ \sim_1 &= \{\{P_1, P_4\}, \{P_2, P_3\}\} \\ \sim_2 &= \{\{P_1\}, \{P_2, P_3\}, \{P_4\}\} \end{aligned}$$

Application to Finite LTS ("Partition Refinement")

Theorem (Fixed-point theorem for finite lattices; cf. Theorem 7.14)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

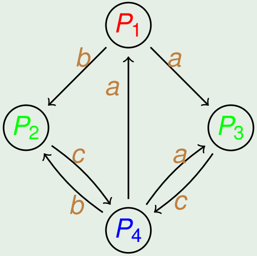
for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

Corollary 7.18

For finite-state process P with state space S, $\sim$ can be computed by:

$$\begin{array}{rcl} \sim & = & \bigcap_{i=0}^{\infty} \sim_i \quad \textit{where} \\ \sim_0 & := & S \times S \\ \sim_{i+1} & := & \mathcal{F}(\sim_i) \end{array}$$

Example 7.19



Equivalence classes:

$$\begin{array}{rl} \sim_0 & = \{\{P_1, P_2, P_3, P_4\}\} \\ \sim_1 & = \{\{P_1, P_4\}, \{P_2, P_3\}\} \\ \sim_2 & = \{\{P_1\}, \{P_2, P_3\}, \{P_4\}\} \\ \sim_3 & = \sim_2 \end{array}$$

Complexity of Checking Strong Bisimilarity

- The previous corollary yields a polynomial-time algorithm.
- More efficient algorithms do exist, but are not topic of this lecture.

Complexity of Checking Strong Bisimilarity

- The previous corollary yields a polynomial-time algorithm.
- More efficient algorithms do exist, but are not topic of this lecture.

Theorem 7.20 (Complexity) (Balcázar et al. 1992)

Deciding strong bisimilarity between finite LTSs is P-complete.[a]

[a] Recall that checking trace equivalence is PSPACE-complete.

Outline of Lecture 7

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.

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) pre-computing the weak transition relation $\stackrel{\alpha}{\Longrightarrow}$ (for $\alpha \neq \tau$) based on given relation $\longrightarrow$
  (2) constructing a new finite-state process by replacing original transitions with weak transitions

  (similarly to elimination of ε-transitions in ε-NFA; see following example)

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. pre-computing the weak transition relation $\stackrel{\alpha}{\Longrightarrow}$ (for $\alpha \neq \tau$) based on given relation $\longrightarrow$
  2. constructing a new finite-state process by replacing original transitions with weak transitions

  (similarly to elimination of ε-transitions in ε-NFA; see following example)
- The question whether $P \approx Q$ now boils down to checking $\sim$ on the saturated processes.

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) pre-computing the weak transition relation $\stackrel{\alpha}{\Longrightarrow}$ (for $\alpha \neq \tau$) based on given relation $\longrightarrow$
  (2) constructing a new finite-state process by replacing original transitions with weak transitions

  (similarly to elimination of ε-transitions in ε-NFA; see following example)
- The question whether $P \approx Q$ now boils down to checking $\sim$ on the saturated processes.
- As both computing $\Longrightarrow$ and $\sim$ can be done in polynomial time, $P \approx Q$ can also be checked in polynomial time.

Example 7.21 (Parallel two-place buffer; cf. Example 5.11)

For
$$
\begin{tikzcd}
 & Q_1 \arrow[dl, "in"'] & \\
Q_2 \arrow[rr, "\tau"] & & Q_3 \arrow[ul, "\overline{out}"'] \arrow[dl, "in"] \\
 & Q_4 \arrow[ul, "\overline{out}"] &
\end{tikzcd}
$$

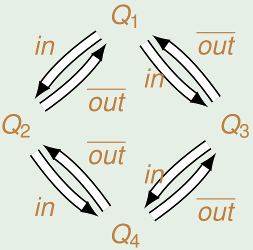
Deciding Weak Bisimilarity (Example)

Example 7.21 (Parallel two-place buffer; cf. Example 5.11)

For

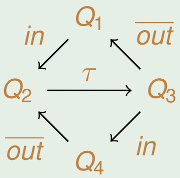
$$\begin{tikzcd}
& Q_1 \arrow[dl, "in"'] & \\
Q_2 \arrow[rr, "\tau"] & & Q_3 \arrow[ul, "\overline{out}"'] \arrow[dl, "in"] \\
& Q_4 \arrow[ul, "\overline{out}"] &
\end{tikzcd}$$

saturation yields

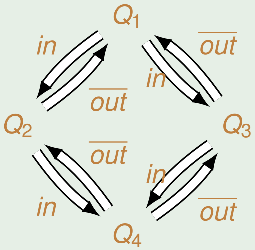


Deciding Weak Bisimilarity (Example)

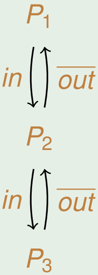
Example 7.21 (Parallel two-place buffer; cf. Example 5.11)

For 

saturation yields



$\approx$



Outline of Lecture 7

Summary: Bisimulation

Definitions:

- $\sim$: Strong bisimilarity (Definition 4.2)
- $\approx$: Weak bisimilarity (observation equivalence; Definition 5.10)
- $\approx^c$: Observation congruence (Definition 6.12)

Summary: Bisimulation

Definitions:

- $\sim$: Strong bisimilarity (Definition 4.2)
- $\approx$: Weak bisimilarity (observation equivalence; Definition 5.10)
- $\approx^c$: Observation congruence (Definition 6.12)

Properties:

- $\sim$, $\approx$ and $\approx^c$ are equivalence relations.
- $\sim$ is finer than $\approx^c$, and $\approx^c$ is finer than $\approx$.
- $\sim$ and $\approx^c$ are CCS congruences.
- $\sim$, $\approx$ and $\approx^c$ are (observationally) deadlock-sensitive.
- $\sim$ and $\approx$ can be characterised by a two-player game.
- $\sim$ and $\approx$ can be characterised as greatest fixed points of a monotonic function on a complete lattice.
- Both characterisations yield decision algorithms for finite-state processes.