



Concurrency Theory

Winter Semester 2019/20

Lecture 6: Mutually Recursive Equational Systems

Joost-Pieter Katoen and Thomas Noll

Software Modeling and Verification Group

RWTH Aachen University

<https://moves.rwth-aachen.de/teaching/ws-19-20/ct/>

Recap: Fixed-Point Theory

Outline of Lecture 6

Recap: Fixed-Point Theory

Fixed Points and System Properties

Mutually Recursive Equational Systems

Mixing Least and Greatest Fixed Points

Recap: Fixed-Point Theory

The Fixed-Point Theorem



Alfred Tarski (1901–1983)

Theorem (Tarski's fixed-point theorem)

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

$$\text{fix}(f) = \prod \{d \in D \mid f(d) \sqsubseteq d\} \quad (\text{GLB of all pre-fixed points of } f)$$

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

Proof.

on the board



Recap: Fixed-Point Theory

The Fixed-Point Theorem for Finite Lattices

Theorem (Fixed-point theorem for finite lattices)

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

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

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

Proof.

on the board □

Recap: Fixed-Point Theory

Application to HML with Recursion

Lemma

Let $(S, Act, \longrightarrow)$ be an LTS and $F \in HMF_X$. Then

1. $\llbracket F \rrbracket : 2^S \rightarrow 2^S$ is monotonic w.r.t. $(2^S, \subseteq)$
2. $\text{fix}(\llbracket F \rrbracket) = \bigcap \{T \subseteq S \mid \llbracket F \rrbracket(T) \subseteq T\}$
3. $\text{FIX}(\llbracket F \rrbracket) = \bigcup \{T \subseteq S \mid T \subseteq \llbracket F \rrbracket(T)\}$

If, in addition, S is finite, then

4. $\text{fix}(\llbracket F \rrbracket) = \llbracket F \rrbracket^m(\emptyset)$ for some $m \in \mathbb{N}$
5. $\text{FIX}(\llbracket F \rrbracket) = \llbracket F \rrbracket^M(S)$ for some $M \in \mathbb{N}$

Proof.

1. by induction on the structure of F (details omitted)
2. by Lemma 4.15 and Theorem 5.5
3. by Lemma 4.15 and Theorem 5.5
4. by Lemma 4.15 and Theorem 5.7
5. by Lemma 4.15 and Theorem 5.7



Fixed Points and System Properties

Outline of Lecture 6

Recap: Fixed-Point Theory

Fixed Points and System Properties

Mutually Recursive Equational Systems

Mixing Least and Greatest Fixed Points

Greatest Fixed Points and Invariants

- **Invariants** (cf. Example 4.5):
 - $Inv(F) \stackrel{max}{=} F \wedge [Act]Inv(F)$ for $F \in HMF$
 - $s \models Inv(F)$ if all states reachable from s satisfy F

Greatest Fixed Points and Invariants

- **Invariants** (cf. Example 4.5):
 - $Inv(F) \stackrel{max}{=} F \wedge [Act]Inv(F)$ for $F \in HMF$
 - $s \models Inv(F)$ if all states reachable from s satisfy F
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)

Greatest Fixed Points and Invariants

- **Invariants** (cf. Example 4.5):
 - $Inv(F) \stackrel{\text{max}}{=} F \wedge [Act]Inv(F)$ for $F \in HMF$
 - $s \models Inv(F)$ if all states reachable from s satisfy F
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)
- Let $inv : 2^S \rightarrow 2^S : T \mapsto \llbracket F \rrbracket \cap [\cdot Act \cdot](T)$ be the corresponding semantic function
- By Lemma 5.9, $FIX(inv) = \bigcup \{T \subseteq S \mid T \subseteq inv(T)\}$

Greatest Fixed Points and Invariants

- **Invariants** (cf. Example 4.5):
 - $Inv(F) \stackrel{max}{=} F \wedge [Act]Inv(F)$ for $F \in HMF$
 - $s \models Inv(F)$ if all states reachable from s satisfy F
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)
- Let $inv : 2^S \rightarrow 2^S : T \mapsto \llbracket F \rrbracket \cap [\cdot Act \cdot](T)$ be the corresponding semantic function
- By Lemma 5.9, $FIX(inv) = \bigcup \{T \subseteq S \mid T \subseteq inv(T)\}$
- **Direct formulation** of invariance property:

$$Inv = \{s \in S \mid \forall w \in Act^*, s' \in S : s \xrightarrow{w} s' \implies s' \in \llbracket F \rrbracket\}$$

Fixed Points and System Properties

Greatest Fixed Points and Invariants

- **Invariants** (cf. Example 4.5):
 - $Inv(F) \stackrel{\text{max}}{=} F \wedge [Act]Inv(F)$ for $F \in HMF$
 - $s \models Inv(F)$ if all states reachable from s satisfy F
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)
- Let $inv : 2^S \rightarrow 2^S : T \mapsto \llbracket F \rrbracket \cap [\cdot Act \cdot](T)$ be the corresponding semantic function
- By Lemma 5.9, $FIX(inv) = \bigcup \{T \subseteq S \mid T \subseteq inv(T)\}$
- **Direct formulation** of invariance property:

$$Inv = \{s \in S \mid \forall w \in Act^*, s' \in S : s \xrightarrow{w} s' \implies s' \in \llbracket F \rrbracket\}$$

Theorem 6.1

For every LTS $(S, Act, \longrightarrow)$, $Inv = FIX(inv)$ holds.

Fixed Points and System Properties

Greatest Fixed Points and Invariants

- **Invariants** (cf. Example 4.5):
 - $Inv(F) \stackrel{\text{max}}{=} F \wedge [Act]Inv(F)$ for $F \in HMF$
 - $s \models Inv(F)$ if all states reachable from s satisfy F
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)
- Let $inv : 2^S \rightarrow 2^S : T \mapsto \llbracket F \rrbracket \cap [\cdot Act \cdot](T)$ be the corresponding semantic function
- By Lemma 5.9, $FIX(inv) = \bigcup \{T \subseteq S \mid T \subseteq inv(T)\}$
- **Direct formulation** of invariance property:

$$Inv = \{s \in S \mid \forall w \in Act^*, s' \in S : s \xrightarrow{w} s' \implies s' \in \llbracket F \rrbracket\}$$

Theorem 6.1

For every LTS $(S, Act, \longrightarrow)$, $Inv = FIX(inv)$ holds.

Proof.

on the board



Least Fixed Points and Possibilities

- **Possibilities** (cf. Example 4.5):
 - $Pos(F) \stackrel{min}{=} F \vee \langle Act \rangle Pos(F)$
 - $s \models Pos(F)$ if a state satisfying F is reachable from s

Least Fixed Points and Possibilities

- **Possibilities** (cf. Example 4.5):
 - $Pos(F) \stackrel{min}{=} F \vee \langle Act \rangle Pos(F)$
 - $s \models Pos(F)$ if a state satisfying F is reachable from s
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)

Least Fixed Points and Possibilities

- **Possibilities** (cf. Example 4.5):
 - $Pos(F) \stackrel{min}{=} F \vee \langle Act \rangle Pos(F)$
 - $s \models Pos(F)$ if a state satisfying F is reachable from s
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)
- Let $pos : 2^S \rightarrow 2^S : T \mapsto \llbracket F \rrbracket \cup \langle \cdot Act \cdot \rangle(T)$ be the corresponding semantic function
- By Lemma 5.9, $fix(pos) = \bigcap \{ T \subseteq S \mid pos(T) \subseteq T \}$

Least Fixed Points and Possibilities

- **Possibilities** (cf. Example 4.5):
 - $Pos(F) \stackrel{min}{=} F \vee \langle Act \rangle Pos(F)$
 - $s \models Pos(F)$ if a state satisfying F is reachable from s
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)
- Let $pos : 2^S \rightarrow 2^S : T \mapsto \llbracket F \rrbracket \cup \langle \cdot Act \cdot \rangle(T)$ be the corresponding semantic function
- By Lemma 5.9, $fix(pos) = \bigcap \{ T \subseteq S \mid pos(T) \subseteq T \}$
- **Direct formulation** of possibility property:

$$Pos = \{ s \in S \mid \exists w \in Act^*, s' \in \llbracket F \rrbracket : s \xrightarrow{w} s' \}$$

Fixed Points and System Properties

Least Fixed Points and Possibilities

- **Possibilities** (cf. Example 4.5):
 - $Pos(F) \stackrel{min}{=} F \vee \langle Act \rangle Pos(F)$
 - $s \models Pos(F)$ if a state satisfying F is reachable from s
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)
- Let $pos : 2^S \rightarrow 2^S : T \mapsto \llbracket F \rrbracket \cup \langle \cdot Act \cdot \rangle(T)$ be the corresponding semantic function
- By Lemma 5.9, $fix(pos) = \bigcap \{T \subseteq S \mid pos(T) \subseteq T\}$
- **Direct formulation** of possibility property:

$$Pos = \{s \in S \mid \exists w \in Act^*, s' \in \llbracket F \rrbracket : s \xrightarrow{w} s'\}$$

Theorem 6.2

For every LTS $(S, Act, \longrightarrow)$, $Pos = fix(pos)$ holds.

Fixed Points and System Properties

Least Fixed Points and Possibilities

- **Possibilities** (cf. Example 4.5):
 - $Pos(F) \stackrel{min}{=} F \vee \langle Act \rangle Pos(F)$
 - $s \models Pos(F)$ if a state satisfying F is reachable from s
- Now: **formalise** argument and prove its **correctness** (for arbitrary LTSs)
- Let $pos : 2^S \rightarrow 2^S : T \mapsto \llbracket F \rrbracket \cup \langle \cdot Act \cdot \rangle(T)$ be the corresponding semantic function
- By Lemma 5.9, $fix(pos) = \bigcap \{ T \subseteq S \mid pos(T) \subseteq T \}$
- **Direct formulation** of possibility property:

$$Pos = \{ s \in S \mid \exists w \in Act^*, s' \in \llbracket F \rrbracket : s \xrightarrow{w} s' \}$$

Theorem 6.2

For every LTS $(S, Act, \longrightarrow)$, $Pos = fix(pos)$ holds.

Proof.

similar to Theorem 6.1



Mutually Recursive Equational Systems

Outline of Lecture 6

Recap: Fixed-Point Theory

Fixed Points and System Properties

Mutually Recursive Equational Systems

Mixing Least and Greatest Fixed Points

Mutually Recursive Equational Systems

Introducing Several Variables

Sometimes necessary: using more than one variable

Example 6.3

“It is always the case that a process can perform an a -labelled transition leading to a state where b -transitions can be executed forever.”

Mutually Recursive Equational Systems

Introducing Several Variables

Sometimes necessary: using more than one variable

Example 6.3

“It is always the case that a process can perform an a -labelled transition leading to a state where b -transitions can be executed forever.”

can be specified by

$$Inv(\langle a \rangle Forever(b))$$

where

$$\begin{aligned} Inv(F) &\stackrel{max}{=} F \wedge [Act]Inv(F) && \text{(cf. Theorem 6.1)} \\ Forever(b) &\stackrel{max}{=} \langle b \rangle Forever(b) \end{aligned}$$

Mutually Recursive Equational Systems

Syntax of Mutually Recursive Equational Systems

Definition 6.4 (Syntax of mutually recursive equational systems)

Let $\mathcal{X} = \{X_1, \dots, X_n\}$ be a set of **variables**. The set $HMF_{\mathcal{X}}$ of **Hennesy-Milner formulae over \mathcal{X}** is defined by the following syntax:

$F ::= X_i$	(variable)
tt	(true)
ff	(false)
$F_1 \wedge F_2$	(conjunction)
$F_1 \vee F_2$	(disjunction)
$\langle \alpha \rangle F$	(diamond)
$[\alpha] F$	(box)

where $1 \leq i \leq n$ and $\alpha \in Act$. A **mutually recursive equational system** has the form

$$(X_i = F_{X_i} \mid 1 \leq i \leq n)$$

where $F_{X_i} \in HMF_{\mathcal{X}}$ for every $1 \leq i \leq n$.

Mutually Recursive Equational Systems

Semantics of Recursive Equational Systems I

As before: semantics of formula depends on states satisfying the variables

Definition 6.5 (Semantics of mutually recursive equational systems)

Let $(S, Act, \longrightarrow)$ be an LTS and $E = (X_i = F_{X_i} \mid 1 \leq i \leq n)$ a mutually recursive equational system. The **semantics** of E , $\llbracket E \rrbracket : (2^S)^n \rightarrow (2^S)^n$, is defined by

$$\llbracket E \rrbracket (T_1, \dots, T_n) := (\llbracket F_{X_1} \rrbracket (T_1, \dots, T_n), \dots, \llbracket F_{X_n} \rrbracket (T_1, \dots, T_n))$$

where

$$\begin{aligned}\llbracket X_i \rrbracket (T_1, \dots, T_n) &:= T_i \\ \llbracket \text{tt} \rrbracket (T_1, \dots, T_n) &:= S \\ \llbracket \text{ff} \rrbracket (T_1, \dots, T_n) &:= \emptyset \\ \llbracket F_1 \wedge F_2 \rrbracket (T_1, \dots, T_n) &:= \llbracket F_1 \rrbracket (T_1, \dots, T_n) \cap \llbracket F_2 \rrbracket (T_1, \dots, T_n) \\ \llbracket F_1 \vee F_2 \rrbracket (T_1, \dots, T_n) &:= \llbracket F_1 \rrbracket (T_1, \dots, T_n) \cup \llbracket F_2 \rrbracket (T_1, \dots, T_n) \\ \llbracket \langle \alpha \rangle F \rrbracket (T_1, \dots, T_n) &:= \langle \cdot \alpha \cdot \rangle (\llbracket F \rrbracket (T_1, \dots, T_n)) \\ \llbracket [\alpha] F \rrbracket (T_1, \dots, T_n) &:= [\cdot \alpha \cdot] (\llbracket F \rrbracket (T_1, \dots, T_n))\end{aligned}$$

Mutually Recursive Equational Systems

Semantics of Recursive Equational Systems II

Lemma 6.6

Let $(S, Act, \longrightarrow)$ be a *finite* LTS and $E = (X_i = F_{X_i} \mid 1 \leq i \leq n)$ a mutually recursive equational system. Let (D, \sqsubseteq) be given by $D := (2^S)^n$ and

$$(T_1, \dots, T_n) \sqsubseteq (T'_1, \dots, T'_n)$$

iff $T_i \subseteq T'_i$ for every $1 \leq i \leq n$.

Mutually Recursive Equational Systems

Semantics of Recursive Equational Systems II

Lemma 6.6

Let $(S, Act, \longrightarrow)$ be a *finite* LTS and $E = (X_i = F_{X_i} \mid 1 \leq i \leq n)$ a mutually recursive equational system. Let (D, \sqsubseteq) be given by $D := (2^S)^n$ and

$$(T_1, \dots, T_n) \sqsubseteq (T'_1, \dots, T'_n)$$

iff $T_i \subseteq T'_i$ for every $1 \leq i \leq n$.

1. (D, \sqsubseteq) is a complete lattice with

$$\begin{aligned} \bigsqcup \{(T_1^i, \dots, T_n^i) \mid i \in I\} &= (\bigcup \{T_1^i \mid i \in I\}, \dots, \bigcup \{T_n^i \mid i \in I\}) \\ \bigsqcap \{(T_1^i, \dots, T_n^i) \mid i \in I\} &= (\bigcap \{T_1^i \mid i \in I\}, \dots, \bigcap \{T_n^i \mid i \in I\}) \end{aligned}$$

Mutually Recursive Equational Systems

Semantics of Recursive Equational Systems II

Lemma 6.6

Let $(S, Act, \longrightarrow)$ be a *finite* LTS and $E = (X_i = F_{X_i} \mid 1 \leq i \leq n)$ a mutually recursive equational system. Let (D, \sqsubseteq) be given by $D := (2^S)^n$ and

$$(T_1, \dots, T_n) \sqsubseteq (T'_1, \dots, T'_n)$$

iff $T_i \subseteq T'_i$ for every $1 \leq i \leq n$.

1. (D, \sqsubseteq) is a complete lattice with

$$\begin{aligned} \bigsqcup \{(T_1^i, \dots, T_n^i) \mid i \in I\} &= (\bigcup \{T_1^i \mid i \in I\}, \dots, \bigcup \{T_n^i \mid i \in I\}) \\ \bigsqcap \{(T_1^i, \dots, T_n^i) \mid i \in I\} &= (\bigcap \{T_1^i \mid i \in I\}, \dots, \bigcap \{T_n^i \mid i \in I\}) \end{aligned}$$

2. $\llbracket E \rrbracket$ is monotonic w.r.t. (D, \sqsubseteq)

Mutually Recursive Equational Systems

Semantics of Recursive Equational Systems II

Lemma 6.6

Let $(S, Act, \longrightarrow)$ be a *finite* LTS and $E = (X_i = F_{X_i} \mid 1 \leq i \leq n)$ a mutually recursive equational system. Let (D, \sqsubseteq) be given by $D := (2^S)^n$ and

$$(T_1, \dots, T_n) \sqsubseteq (T'_1, \dots, T'_n)$$

iff $T_i \subseteq T'_i$ for every $1 \leq i \leq n$.

1. (D, \sqsubseteq) is a complete lattice with

$$\begin{aligned} \bigsqcup \{(T_1^i, \dots, T_n^i) \mid i \in I\} &= (\bigcup \{T_1^i \mid i \in I\}, \dots, \bigcup \{T_n^i \mid i \in I\}) \\ \bigsqcap \{(T_1^i, \dots, T_n^i) \mid i \in I\} &= (\bigcap \{T_1^i \mid i \in I\}, \dots, \bigcap \{T_n^i \mid i \in I\}) \end{aligned}$$

2. $\llbracket E \rrbracket$ is monotonic w.r.t. (D, \sqsubseteq)

3. $\text{fix}(\llbracket E \rrbracket) = \llbracket E \rrbracket^m(\emptyset, \dots, \emptyset)$ for some $m \in \mathbb{N}$

4. $\text{FIX}(\llbracket E \rrbracket) = \llbracket E \rrbracket^M(S, \dots, S)$ for some $M \in \mathbb{N}$

Mutually Recursive Equational Systems

Semantics of Recursive Equational Systems II

Lemma 6.6

Let $(S, Act, \longrightarrow)$ be a *finite* LTS and $E = (X_i = F_{X_i} \mid 1 \leq i \leq n)$ a mutually recursive equational system. Let (D, \sqsubseteq) be given by $D := (2^S)^n$ and

$$(T_1, \dots, T_n) \sqsubseteq (T'_1, \dots, T'_n)$$

iff $T_i \subseteq T'_i$ for every $1 \leq i \leq n$.

1. (D, \sqsubseteq) is a complete lattice with

$$\begin{aligned} \bigsqcup \{(T_1^i, \dots, T_n^i) \mid i \in I\} &= (\bigcup \{T_1^i \mid i \in I\}, \dots, \bigcup \{T_n^i \mid i \in I\}) \\ \bigsqcap \{(T_1^i, \dots, T_n^i) \mid i \in I\} &= (\bigcap \{T_1^i \mid i \in I\}, \dots, \bigcap \{T_n^i \mid i \in I\}) \end{aligned}$$

2. $\llbracket E \rrbracket$ is monotonic w.r.t. (D, \sqsubseteq)

3. $\text{fix}(\llbracket E \rrbracket) = \llbracket E \rrbracket^m(\emptyset, \dots, \emptyset)$ for some $m \in \mathbb{N}$

4. $\text{FIX}(\llbracket E \rrbracket) = \llbracket E \rrbracket^M(S, \dots, S)$ for some $M \in \mathbb{N}$

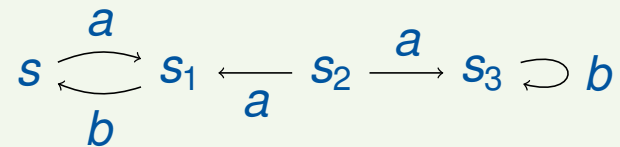
Proof.

omitted □

Mutually Recursive Equational Systems

A Mutually Recursive Specification

Example 6.7



- Let $S := \{s, s_1, s_2, s_3\}$ and E given by

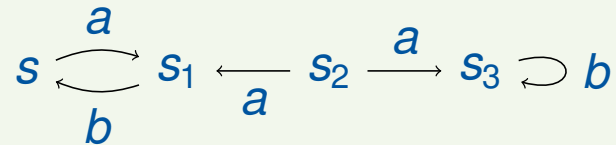
$$X = \langle a \rangle Y \wedge [a]Y \wedge [b]ff$$

$$Y = \langle b \rangle X \wedge [b]X \wedge [a]ff$$

Mutually Recursive Equational Systems

A Mutually Recursive Specification

Example 6.7



- Let $S := \{s, s_1, s_2, s_3\}$ and E given by

$$X = \langle a \rangle Y \wedge [a]Y \wedge [b]ff$$

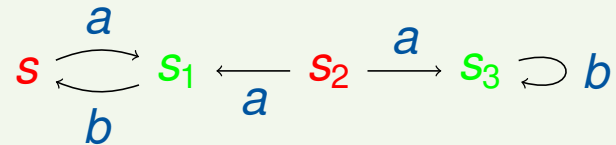
$$Y = \langle b \rangle X \wedge [b]X \wedge [a]ff$$

- Interpretation:
 - X : “has no b -successor and ≥ 1 a -successors that all satisfy Y ”
 - Y : “has no a -successor and ≥ 1 b -successors that all satisfy X ”

Mutually Recursive Equational Systems

A Mutually Recursive Specification

Example 6.7



- Let $S := \{s, s_1, s_2, s_3\}$ and E given by

$$X = \langle a \rangle Y \wedge [a]Y \wedge [b]ff$$

$$Y = \langle b \rangle X \wedge [b]X \wedge [a]ff$$

- Interpretation:

– X : “has no b -successor and ≥ 1 a -successors that all satisfy Y ”

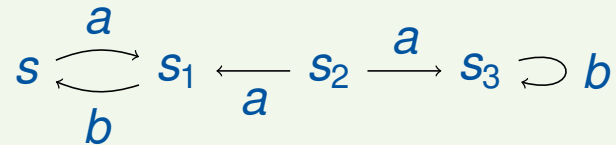
– Y : “has no a -successor and ≥ 1 b -successors that all satisfy X ”

\Rightarrow expected: $X = \{s\}$, $Y = \{s_1\}$

Mutually Recursive Equational Systems

A Mutually Recursive Specification

Example 6.7



- Let $S := \{s, s_1, s_2, s_3\}$ and E given by

$$X = \langle a \rangle Y \wedge [a]Y \wedge [b]ff$$

$$Y = \langle b \rangle X \wedge [b]X \wedge [a]ff$$

- Interpretation:
 - X : “has no b -successor and ≥ 1 a -successors that all satisfy Y ”
 - Y : “has no a -successor and ≥ 1 b -successors that all satisfy X ” \Rightarrow expected: $X = \{s\}$, $Y = \{s_1\}$
- Computation of $\text{FIX}(\llbracket E \rrbracket)$: on the board

Mixing Least and Greatest Fixed Points

Outline of Lecture 6

Recap: Fixed-Point Theory

Fixed Points and System Properties

Mutually Recursive Equational Systems

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points I

- **So far:** least/greatest fixed point of **overall** system
- **But:** too **restrictive**

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points I

- **So far:** least/greatest fixed point of **overall** system
- **But:** too **restrictive**

Example 6.8

“It is possible for the system to reach a state which has a livelock (i.e., an infinite sequence of internal steps).”

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points I

- **So far:** least/greatest fixed point of **overall** system
- **But:** too **restrictive**

Example 6.8

“It is possible for the system to reach a state which has a livelock (i.e., an infinite sequence of internal steps).”

can be specified by

$$Pos(Livelock)$$

where

$$Pos(F) \stackrel{min}{=} F \vee \langle Act \rangle Pos(F) \quad (\text{cf. Theorem 6.2})$$
$$Livelock \stackrel{max}{=} \langle \tau \rangle Livelock$$

(thus, $Livelock \equiv Forever(\tau)$ [cf. Example 6.3])

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points II

Caveat: arbitrary mixing can entail **non-monotonic behaviour**

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points II

Caveat: arbitrary mixing can entail **non-monotonic behaviour**

Example 6.9

$$E : X \stackrel{\min}{=} Y$$
$$Y \stackrel{\max}{=} X$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points II

Caveat: arbitrary mixing can entail **non-monotonic behaviour**

Example 6.9

$$E : X \stackrel{\min}{=} Y$$
$$Y \stackrel{\max}{=} X$$

Fixed-point iteration:

$$(\perp, \top) = (\emptyset, S)$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points II

Caveat: arbitrary mixing can entail **non-monotonic behaviour**

Example 6.9

$$E : X \stackrel{\min}{=} Y$$
$$Y \stackrel{\max}{=} X$$

Fixed-point iteration:

$$(\perp, \top) = (\emptyset, S) \xrightarrow{[[E]]} (S, \emptyset)$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points II

Caveat: arbitrary mixing can entail **non-monotonic behaviour**

Example 6.9

$$E : X \stackrel{\min}{=} Y \\ Y \stackrel{\max}{=} X$$

Fixed-point iteration:

$$(\perp, \top) = (\emptyset, S) \xrightarrow{[E]} (S, \emptyset) \xrightarrow{[E]} (\emptyset, S) \xrightarrow{[E]} \dots$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points II

Caveat: arbitrary mixing can entail **non-monotonic behaviour**

Example 6.9

$$E : X \stackrel{\min}{=} Y \\ Y \stackrel{\max}{=} X$$

Fixed-point iteration:

$$(\perp, \top) = (\emptyset, S) \xrightarrow{[E]} (S, \emptyset) \xrightarrow{[E]} (\emptyset, S) \xrightarrow{[E]} \dots$$

Solution: **nesting** of specifications by partitioning equations into a sequence of blocks such that all equations in one block

- are of **same type** (either *min* or *max*) and
- use only variables defined in **the same or subsequent blocks**

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points II

Caveat: arbitrary mixing can entail **non-monotonic behaviour**

Example 6.9

$$\begin{aligned} E : X &\stackrel{\min}{=} Y \\ Y &\stackrel{\max}{=} X \end{aligned}$$

Fixed-point iteration:

$$(\perp, \top) = (\emptyset, S) \xrightarrow{[E]} (S, \emptyset) \xrightarrow{[E]} (\emptyset, S) \xrightarrow{[E]} \dots$$

Solution: **nesting** of specifications by partitioning equations into a sequence of blocks such that all equations in one block

- are of **same type** (either *min* or *max*) and
- use only variables defined in **the same or subsequent blocks**

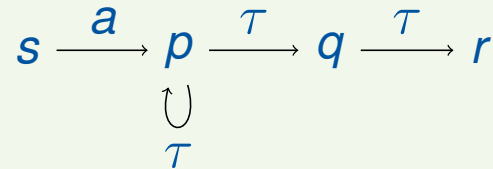
⇒ **bottom-up, block-wise evaluation** by fixed-point iteration

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$

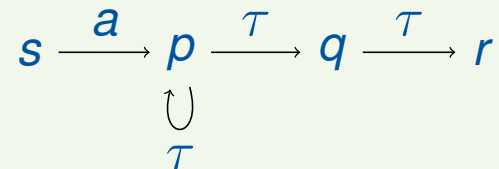


Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$



1. Greatest fixed-point iteration for $\text{Livelock} : T \mapsto \langle \cdot \tau \cdot \rangle (T)$:

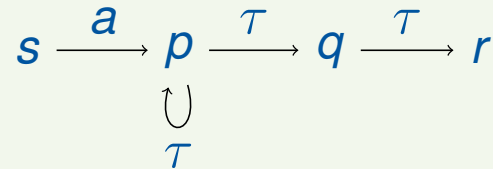
$$\top = S = \{s, p, q, r\}$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$



1. Greatest fixed-point iteration for $\text{Livelock} : T \mapsto \langle \cdot \tau \cdot \rangle (T)$:

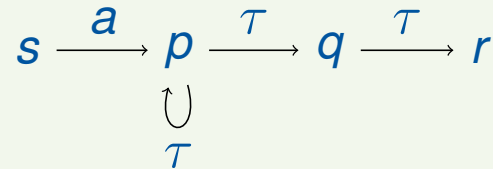
$$\top = S = \{s, p, q, r\} \mapsto \{p, q\}$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$



1. Greatest fixed-point iteration for $\text{Livelock} : T \mapsto \langle \cdot \tau \cdot \rangle (T)$:

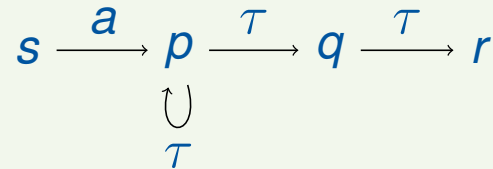
$$\top = S = \{s, p, q, r\} \mapsto \{p, q\} \mapsto \{p\}$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$



1. Greatest fixed-point iteration for $\text{Livelock} : T \mapsto \langle \cdot \tau \cdot \rangle (T)$:

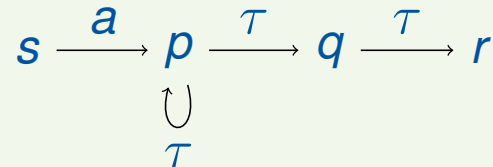
$$\top = S = \{s, p, q, r\} \mapsto \{p, q\} \mapsto \{p\} \mapsto \{p\}$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$



1. Greatest fixed-point iteration for $\text{Livelock} : T \mapsto \langle \cdot \tau \cdot \rangle (T)$:

$$\top = S = \{s, p, q, r\} \mapsto \{p, q\} \mapsto \{p\} \mapsto \{p\}$$

2. Least fixed-point iteration for $\text{PosLL} : T \mapsto \{p\} \cup \langle \cdot \text{Act} \cdot \rangle (T)$:

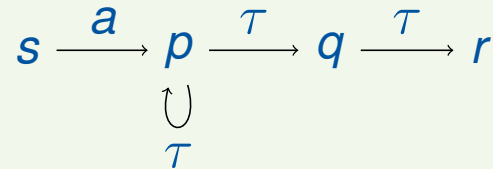
$$\perp = \emptyset$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$



1. Greatest fixed-point iteration for $\text{Livelock} : T \mapsto \langle \cdot \tau \cdot \rangle (T)$:

$$\top = S = \{s, p, q, r\} \mapsto \{p, q\} \mapsto \{p\} \mapsto \{p\}$$

2. Least fixed-point iteration for $\text{PosLL} : T \mapsto \{p\} \cup \langle \cdot \text{Act} \cdot \rangle (T)$:

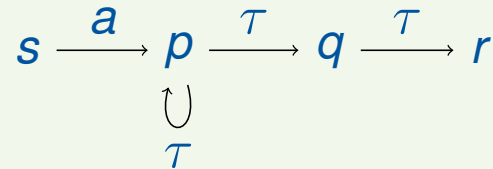
$$\perp = \emptyset \mapsto \{p\}$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$



1. Greatest fixed-point iteration for $\text{Livelock} : T \mapsto \langle \cdot \tau \cdot \rangle (T)$:

$$\top = S = \{s, p, q, r\} \mapsto \{p, q\} \mapsto \{p\} \mapsto \{p\}$$

2. Least fixed-point iteration for $\text{PosLL} : T \mapsto \{p\} \cup \langle \cdot \text{Act} \cdot \rangle (T)$:

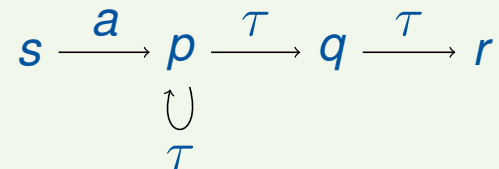
$$\perp = \emptyset \mapsto \{p\} \mapsto \{s, p\}$$

Mixing Least and Greatest Fixed Points

Mixing Least and Greatest Fixed Points III

Example 6.10 (cf. Example 6.8)

$$\begin{aligned} \text{PosLL} &\stackrel{\min}{=} \text{Livelock} \vee \langle \text{Act} \rangle \text{PosLL} \\ \text{Livelock} &\stackrel{\max}{=} \langle \tau \rangle \text{Livelock} \end{aligned}$$



1. Greatest fixed-point iteration for $\text{Livelock} : T \mapsto \langle \cdot \tau \cdot \rangle (T)$:

$$\top = S = \{s, p, q, r\} \mapsto \{p, q\} \mapsto \{p\} \mapsto \{p\}$$

2. Least fixed-point iteration for $\text{PosLL} : T \mapsto \{p\} \cup \langle \cdot \text{Act} \cdot \rangle (T)$:

$$\perp = \emptyset \mapsto \{p\} \mapsto \{s, p\} \mapsto \{s, p\}$$

Mixing Least and Greatest Fixed Points

The Modal μ -Calculus

- Logic that supports free mixing of least and greatest fixed points:
 - D. Kozen: *Results on the Propositional μ -Calculus*, Theoretical Computer Science 27, 1983, 333–354

Mixing Least and Greatest Fixed Points

The Modal μ -Calculus

- Logic that supports free mixing of least and greatest fixed points:
 - D. Kozen: *Results on the Propositional μ -Calculus*, Theoretical Computer Science 27, 1983, 333–354
- HML variants are fragments thereof
- Expressivity increases with alternation of least and greatest fixed points:
 - J.C. Bradfield: *The Modal Mu-Calculus Alternation Hierarchy is Strict*, Theoretical Computer Science 195(2), 1998, 133–153

Mixing Least and Greatest Fixed Points

The Modal μ -Calculus

- Logic that supports free mixing of least and greatest fixed points:
 - D. Kozen: *Results on the Propositional μ -Calculus*, Theoretical Computer Science 27, 1983, 333–354
- HML variants are fragments thereof
- Expressivity increases with alternation of least and greatest fixed points:
 - J.C. Bradfield: *The Modal Mu-Calculus Alternation Hierarchy is Strict*, Theoretical Computer Science 195(2), 1998, 133–153
- **Decidable** model-checking problem for **finite** LTSs
(in $NP \cap co-NP$; linear for HML with one variable)
- Generally **undecidable** for **infinite** LTSs and HML with one variable (CCS, Petri nets, ...)

Mixing Least and Greatest Fixed Points

The Modal μ -Calculus

- Logic that supports free mixing of least and greatest fixed points:
 - D. Kozen: *Results on the Propositional μ -Calculus*, Theoretical Computer Science 27, 1983, 333–354
- HML variants are fragments thereof
- Expressivity increases with alternation of least and greatest fixed points:
 - J.C. Bradfield: *The Modal Mu-Calculus Alternation Hierarchy is Strict*, Theoretical Computer Science 195(2), 1998, 133–153
- **Decidable** model-checking problem for **finite** LTSs
(in $NP \cap co-NP$; linear for HML with one variable)
- Generally **undecidable** for **infinite** LTSs and HML with one variable (CCS, Petri nets, ...)
- Overview paper:
 - O. Burkart, D. Caucal, F. Moller, B. Steffen: *Verification on Infinite Structures*, Chapter 9 of *Handbook of Process Algebra*, Elsevier, 2001, 545–623