Semantics and Verification of Software

Summer Semester 2015

Lecture 8: Denotational Semantics of WHILE III (Fixpoint & Coincidence Theorem)

Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University

`http://moves.rwth-aachen.de/teaching/ss-15/sv-sw/`

Recap: CCPOs and Continuous Functions

Outline of Lecture 8

Recap: CCPOs and Continuous Functions

The Fixpoint Theorem

Application to fix(Φ)

Summary: Denotational Semantics

Equivalence of Operational and Denotational Semantics

Recap: CCPOs and Continuous Functions

Characterisation of $\mathrm{fix}(\Phi)$

Goals:

- Prove existence of $\mathrm{fix}(\Phi)$ for $\Phi(f) = \mathrm{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathrm{id}_\Sigma)$
- Show how it can be "computed" (more exactly: approximated)

Sufficient conditions:

on domain $\Sigma \dashrightarrow \Sigma$: chain-complete partial order

on function Φ: monotonicity and continuity

Recap: CCPOs and Continuous Functions

Chains and Least Upper Bounds I

Definition (Chain, (least) upper bound)

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

1. S is called a chain in D if, for every $s_1, s_2 \in S$,

   $$s_1 \sqsubseteq s_2 \text{ or } s_2 \sqsubseteq s_1$$

   (that is, S is a totally ordered subset of D).
2. An element $d \in D$ is called an upper bound of S if $s \sqsubseteq d$ for every $s \in S$ (notation: $S \sqsubseteq d$).
3. An upper bound d of S is called least upper bound (LUB) or supremum of S if $d \sqsubseteq d'$ for every upper bound d' of S (notation: $d = \bigsqcup S$).

Recap: CCPOs and Continuous Functions

Chain Completeness

Definition (Chain completeness)

A partial order is called chain complete (CCPO) if every of its chains has a least upper bound.

Example

1. $(2^{\mathbb{N}}, \subseteq)$ is a CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ for every chain $S \subseteq 2^{\mathbb{N}}$.
2. $(\mathbb{N}, \leq)$ is not chain complete (since, e.g., the chain $\mathbb{N}$ has no upper bound).

Recap: CCPOs and Continuous Functions

Monotonicity

Definition (Monotonicity)

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

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

Interpretation: monotonic functions “preserve information”

Example

1. Let $T := \{S \subseteq \mathbb{N} \mid S \text{ finite}\}$. Then $F_1 : T \to \mathbb{N} : S \mapsto \sum_{n \in S} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.
2. $F_2 : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : S \mapsto \mathbb{N} \setminus S$ is not monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ (since, e.g., $\emptyset \subseteq \mathbb{N}$ but $F_2(\emptyset) = \mathbb{N} \not\subseteq F_2(\mathbb{N}) = \emptyset$).

Recap: CCPOs and Continuous Functions

Continuity

A function F is continuous if applying F and taking LUBs is commutable:

Definition (Continuity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs and $F : D \to D'$ monotonic. Then F is called continuous (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every non-empty chain $S \subseteq D$,

$$F\left(\bigsqcup S\right) = \bigsqcup F(S).$$

Lemma

Let $b \in \mathit{BExp}$, $c \in \mathit{Cmd}$, *and* $\Phi(f) := \mathrm{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathrm{id}_\Sigma)$. *Then* Φ *is continuous w.r.t.* $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$.

Proof.

omitted ☐

The Fixpoint Theorem

Outline of Lecture 8

Recap: CCPOs and Continuous Functions

The Fixpoint Theorem

Application to fix(Φ)

Summary: Denotational Semantics

Equivalence of Operational and Denotational Semantics

The Fixpoint Theorem

The Fixpoint Theorem

Alfred Tarski (1901–1983)

Bronislaw Knaster (1893–1990)

Theorem 8.1 (Fixpoint Theorem by Tarski and Knaster)

Let $(D, \sqsubseteq)$ *be a CCPO and* $F : D \to D$ *continuous. Then*

$$\mathrm{fix}(F) := \bigsqcup \left\{ F^n \left(\bigsqcup \emptyset \right) \mid n \in \mathbb{N} \right\}$$

is the least fixpoint of F *where* $F^0(d) := d$ *and* $F^{n+1}(d) := F(F^n(d))$.

The Fixpoint Theorem

The Fixpoint Theorem

Alfred Tarski (1901–1983)

Bronislaw Knaster (1893–1990)

Theorem 8.1 (Fixpoint Theorem by Tarski and Knaster)

Let $(D, \sqsubseteq)$ *be a CCPO and* $F : D \to D$ *continuous. Then*

$$\mathrm{fix}(F) := \bigsqcup \left\{ F^n \left(\bigsqcup \emptyset \right) \mid n \in \mathbb{N} \right\}$$

is the least fixpoint of F *where* $F^0(d) := d$ *and* $F^{n+1}(d) := F(F^n(d))$.

Proof.

on the board ☐

The Fixpoint Theorem

An Example

Example 8.2

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{N \in S} N$ – see Example 7.7)

The Fixpoint Theorem

An Example

Example 8.2

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{N \in S} N$ – see Example 7.7)
- **Function:** $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : N \mapsto N \cup A$ for some fixed $A \subseteq \mathbb{N}$
  - F monotonic: $M \subseteq N \Rightarrow F(M) = M \cup A \subseteq N \cup A = F(N)$
  - F continuous: $F(\bigsqcup S) = F\left(\bigcup_{N \in S} N\right) = \left(\bigcup_{N \in S} N\right) \cup A = \bigcup_{N \in S} (N \cup A) = \bigcup_{N \in S} F(N) = \bigsqcup F(S)$

The Fixpoint Theorem

An Example

Example 8.2

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{N \in S} N$ – see Example 7.7)
- **Function:** $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : N \mapsto N \cup A$ for some fixed $A \subseteq \mathbb{N}$
  - F monotonic: $M \subseteq N \Rightarrow F(M) = M \cup A \subseteq N \cup A = F(N)$
  - F continuous: $F(\bigsqcup S) = F\left(\bigcup_{N \in S} N\right) = \left(\bigcup_{N \in S} N\right) \cup A = \bigcup_{N \in S} (N \cup A) = \bigcup_{N \in S} F(N) = \bigsqcup F(S)$
- **Fixpoint iteration:** $N_n := F^n(\bigsqcup \emptyset)$ where $\bigsqcup \emptyset = \emptyset$
  - $N_0 = \bigsqcup \emptyset = \emptyset$
  - $N_1 = F(N_0) = \emptyset \cup A = A$
  - $N_2 = F(N_1) = A \cup A = A = N_n$ for every $n \geq 1$

$\Rightarrow \text{fix}(F) = A$

The Fixpoint Theorem

An Example

Example 8.2

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{N \in S} N$ – see Example 7.7)
- **Function:** $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}} : N \mapsto N \cup A$ for some fixed $A \subseteq \mathbb{N}$
  - F monotonic: $M \subseteq N \Rightarrow F(M) = M \cup A \subseteq N \cup A = F(N)$
  - F continuous: $F(\bigsqcup S) = F\left(\bigcup_{N \in S} N\right) = \left(\bigcup_{N \in S} N\right) \cup A = \bigcup_{N \in S} (N \cup A) = \bigcup_{N \in S} F(N) = \bigsqcup F(S)$
- **Fixpoint iteration:** $N_n := F^n(\bigsqcup \emptyset)$ where $\bigsqcup \emptyset = \emptyset$
  - $N_0 = \bigsqcup \emptyset = \emptyset$
  - $N_1 = F(N_0) = \emptyset \cup A = A$
  - $N_2 = F(N_1) = A \cup A = A = N_n$ for every $n \geq 1$

  $\Rightarrow \text{fix}(F) = A$
- **Alternatively:** $F(N) := N \cap A$

  $\Rightarrow \text{fix}(F) = \emptyset$

Application to fix(Φ)

Outline of Lecture 8

Recap: CCPOs and Continuous Functions

The Fixpoint Theorem

Application to fix(Φ)

Summary: Denotational Semantics

Equivalence of Operational and Denotational Semantics

Application to $\mathrm{fix}(\Phi)$

Application to $\mathrm{fix}(\Phi)$

Altogether this completes the definition of $\mathfrak{C}[\![.]\!]$. In particular, for the `while` statement:

Corollary 8.3

Let $b \in BExp$, $c \in Cmd$, *and* $\Phi(f) := \mathrm{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathrm{id}_\Sigma)$. *Then*

$$\mathrm{graph}(\mathrm{fix}(\Phi)) = \bigcup_{n\in\mathbb{N}} \mathrm{graph}(\Phi^n(f_\emptyset))$$

Application to $\mathrm{fix}(\Phi)$

Application to $\mathrm{fix}(\Phi)$

Altogether this completes the definition of $\mathfrak{C}[\![.]\!]$. In particular, for the `while` statement:

Corollary 8.3

Let $b \in BExp$, $c \in Cmd$, *and* $\Phi(f) := \mathrm{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathrm{id}_\Sigma)$. *Then*

$$\mathrm{graph}(\mathrm{fix}(\Phi)) = \bigcup_{n\in\mathbb{N}} \mathrm{graph}(\Phi^n(f_\emptyset))$$

Proof.

Using

- Lemma 7.9
  - $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ CCPO with least element $f_\emptyset$
  - LUB = union of graphs
- Lemma 7.16 (Φ continuous)
- Theorem 8.1 (Fixpoint Theorem)

□

Denotational Semantics of Factorial Program I

Example 8.4 (Factorial program)

- Let $c \in \mathit{Cmd}$ be given by `y:=1; while ¬(x=1) do y:=y*x; x:=x-1 end`

Application to $\mathrm{fix}(\Phi)$

Denotational Semantics of Factorial Program I

Example 8.4 (Factorial program)

- Let $c \in \mathit{Cmd}$ be given by `y:=1; while ¬(x=1) do y:=y*x; x:=x-1 end`
- For every initial state $\sigma_0 \in \Sigma$, Definition 6.3 yields:

$$\mathfrak{C}[\![c]\!](\sigma_0) = \mathrm{fix}(\Phi)(\sigma_1)$$

where $\sigma_1 := \sigma_0[\mathtt{y} \mapsto 1]$ and, for every $f : \Sigma \dashrightarrow \Sigma$ and $\sigma \in \Sigma$,

$$\begin{aligned}\Phi(f)(\sigma) &= \mathrm{cond}(\mathfrak{B}[\![\neg\mathtt{(x=1)}]\!], f \circ \mathfrak{C}[\![\mathtt{y:=y*x;\ \ x:=x-1}]\!], \mathrm{id}_\Sigma)(\sigma)\\ &= \begin{cases}\sigma & \text{if } \sigma(\mathtt{x}) = 1\\ f(\sigma') & \text{otherwise}\end{cases}\end{aligned}$$

with $\sigma' := \sigma[\mathtt{y} \mapsto \sigma(\mathtt{y}) * \sigma(\mathtt{x}), \mathtt{x} \mapsto \sigma(\mathtt{x}) - 1]$.

Application to $\mathsf{fix}(\Phi)$

Denotational Semantics of Factorial Program I

Example 8.4 (Factorial program)

- Let $c \in Cmd$ be given by `y:=1; while ¬(x=1) do y:=y*x; x:=x-1 end`
- For every initial state $\sigma_0 \in \Sigma$, Definition 6.3 yields:

$$\mathfrak{C}[\![c]\!](\sigma_0) = \mathsf{fix}(\Phi)(\sigma_1)$$

where $\sigma_1 := \sigma_0[\mathtt{y} \mapsto 1]$ and, for every $f : \Sigma \dashrightarrow \Sigma$ and $\sigma \in \Sigma$,

$$\begin{aligned}\Phi(f)(\sigma) &= \mathsf{cond}(\mathfrak{B}[\![\neg(\mathtt{x=1})]\!], f \circ \mathfrak{C}[\![\mathtt{y:=y*x;\ \ x:=x-1}]\!], \mathsf{id}_\Sigma)(\sigma)\\ &= \begin{cases}\sigma & \text{if } \sigma(\mathtt{x}) = 1\\ f(\sigma') & \text{otherwise}\end{cases}\end{aligned}$$

with $\sigma' := \sigma[\mathtt{y} \mapsto \sigma(\mathtt{y}) * \sigma(\mathtt{x}), \mathtt{x} \mapsto \sigma(\mathtt{x}) - 1]$.
- Approximations of least fixpoint of Φ according to Theorem 8.1:

$$\mathsf{fix}(\Phi) = \bigsqcup\{\Phi^n(f_\emptyset) \mid n \in \mathbb{N}\}$$

(where $\mathsf{graph}(f_\emptyset) = \emptyset$)

Application to fix(Φ)

Denotational Semantics of Factorial Program II

Reminder: $\Phi(f)(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f(\sigma') & \text{otherwise} \end{cases} \qquad \sigma' = \sigma[\mathrm{y} \mapsto \sigma(\mathrm{y}) * \sigma(\mathrm{x}), \mathrm{x} \mapsto \sigma(\mathrm{x}) - 1]$

Example 8.4 (Factorial program; continued)

$$\begin{aligned} f_0(\sigma) &:= \Phi^0(f_\emptyset)(\sigma) \\ &= f_\emptyset(\sigma) \\ &= \text{undefined} \end{aligned}$$

Application to $\mathrm{fix}(\Phi)$

Denotational Semantics of Factorial Program II

Reminder: $\Phi(f)(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f(\sigma') & \text{otherwise} \end{cases}$ $\qquad \sigma' = \sigma[\mathrm{y} \mapsto \sigma(\mathrm{y}) * \sigma(\mathrm{x}), \mathrm{x} \mapsto \sigma(\mathrm{x}) - 1]$

Example 8.4 (Factorial program; continued)

$$\begin{aligned} f_0(\sigma) &:= \Phi^0(f_\emptyset)(\sigma) \\ &= f_\emptyset(\sigma) \\ &= \text{undefined} \end{aligned}$$

$$\begin{aligned} f_1(\sigma) &:= \Phi^1(f_\emptyset)(\sigma) \\ &= \Phi(f_0)(\sigma) \\ &= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f_0(\sigma') & \text{otherwise} \end{cases} \\ &= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ \text{undefined} & \text{otherwise} \end{cases} \end{aligned}$$

Denotational Semantics of Factorial Program II

Reminder: $\Phi(f)(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f(\sigma') & \text{otherwise} \end{cases} \qquad \sigma' = \sigma[\mathrm{y} \mapsto \sigma(\mathrm{y}) * \sigma(\mathrm{x}), \mathrm{x} \mapsto \sigma(\mathrm{x}) - 1]$

Example 8.4 (Factorial program; continued)

$$\begin{aligned} f_0(\sigma) &:= \Phi^0(f_\emptyset)(\sigma) \\ &= f_\emptyset(\sigma) \\ &= \text{undefined} \end{aligned}$$

$$\begin{aligned} f_1(\sigma) &:= \Phi^1(f_\emptyset)(\sigma) \\ &= \Phi(f_0)(\sigma) \\ &= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f_0(\sigma') & \text{otherwise} \end{cases} \\ &= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ \text{undefined} & \text{otherwise} \end{cases} \end{aligned}$$

$$\begin{aligned} f_2(\sigma) &:= \Phi^2(f_\emptyset)(\sigma) \\ &= \Phi(f_1)(\sigma) \\ &= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f_1(\sigma') & \text{otherwise} \end{cases} \\ &= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ \sigma' & \text{if } \sigma(\mathrm{x}) \neq 1, \sigma'(\mathrm{x}) = 1 \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \neq 1, \sigma'(\mathrm{x}) \neq 1 \end{cases} \\ &= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ \sigma' & \text{if } \sigma(\mathrm{x}) = 2 \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \neq 1, \sigma(\mathrm{x}) \neq 2 \end{cases} \\ &= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ \sigma[\mathrm{y} \mapsto 2 * \sigma(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } \sigma(\mathrm{x}) = 2 \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \neq 1, \sigma(\mathrm{x}) \neq 2 \end{cases} \end{aligned}$$

Application to $\mathrm{fix}(\Phi)$

Denotational Semantics of Factorial Program III

Reminder: $\Phi(f)(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f(\sigma') & \text{otherwise} \end{cases} \qquad \sigma' = \sigma[\mathrm{y} \mapsto \sigma(\mathrm{y}) * \sigma(\mathrm{x}), \mathrm{x} \mapsto \sigma(\mathrm{x}) - 1]$

Example 8.4 (Factorial program; continued)

$$
\begin{aligned}
f_3(\sigma) &:= \Phi^3(f_\emptyset)(\sigma) \\
&= \Phi(f_2)(\sigma) \\
&= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f_2(\sigma') & \text{otherwise} \end{cases} \\
&= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ \sigma' & \text{if } \sigma(\mathrm{x}) \neq 1, \sigma'(\mathrm{x}) = 1 \\ \sigma'[\mathrm{y} \mapsto 2 * \sigma'(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } \sigma(\mathrm{x}) \neq 1, \sigma'(\mathrm{x}) = 2 \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \neq 1, \sigma'(\mathrm{x}) \neq 1, \sigma'(\mathrm{x}) \neq 2 \end{cases} \\
&= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ \sigma' & \text{if } \sigma(\mathrm{x}) = 2 \\ \sigma'[\mathrm{y} \mapsto 2 * \sigma'(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } \sigma(\mathrm{x}) = 3 \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \notin \{1, 2, 3\} \end{cases} \\
&= \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ \sigma[\mathrm{y} \mapsto 2 * \sigma(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } \sigma(\mathrm{x}) = 2 \\ \sigma[\mathrm{y} \mapsto 3 * 2 * \sigma(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } \sigma(\mathrm{x}) = 3 \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \notin \{1, 2, 3\} \end{cases}
\end{aligned}
$$

Application to $\mathsf{fix}(\Phi)$

Denotational Semantics of Factorial Program IV

Reminder: $\Phi(f)(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f(\sigma') & \text{otherwise} \end{cases} \qquad \sigma' = \sigma[\mathrm{y} \mapsto \sigma(\mathrm{y}) * \sigma(\mathrm{x}), \mathrm{x} \mapsto \sigma(\mathrm{x}) - 1]$

Example 8.4 (Factorial program; continued)

- n-th approximation:

$$\begin{aligned} & f_n(\sigma) \\ & := \Phi^n(f_\emptyset)(\sigma) \\ & = \begin{cases} \sigma[\mathrm{y} \mapsto \sigma(\mathrm{x}) * (\sigma(\mathrm{x}) - 1) * \ldots * 2 * \sigma(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } 1 \leq \sigma(\mathrm{x}) \leq n \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \notin \{1, \ldots, n\} \end{cases} \\ & = \begin{cases} \sigma[\mathrm{y} \mapsto (\sigma(\mathrm{x}))! * \sigma(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } 1 \leq \sigma(\mathrm{x}) \leq n \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \notin \{1, \ldots, n\} \end{cases} \end{aligned}$$

Application to $\mathrm{fix}(\Phi)$

Denotational Semantics of Factorial Program IV

Reminder: $\Phi(f)(\sigma) = \begin{cases} \sigma & \text{if } \sigma(\mathrm{x}) = 1 \\ f(\sigma') & \text{otherwise} \end{cases} \qquad \sigma' = \sigma[\mathrm{y} \mapsto \sigma(\mathrm{y}) * \sigma(\mathrm{x}), \mathrm{x} \mapsto \sigma(\mathrm{x}) - 1]$

Example 8.4 (Factorial program; continued)

- n-th approximation:

$$\begin{aligned} & f_n(\sigma) \\ & := \Phi^n(f_\emptyset)(\sigma) \\ & = \begin{cases} \sigma[\mathrm{y} \mapsto \sigma(\mathrm{x}) * (\sigma(\mathrm{x}) - 1) * \ldots * 2 * \sigma(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } 1 \leq \sigma(\mathrm{x}) \leq n \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \notin \{1, \ldots, n\} \end{cases} \\ & = \begin{cases} \sigma[\mathrm{y} \mapsto (\sigma(\mathrm{x}))! * \sigma(\mathrm{y}), \mathrm{x} \mapsto 1] & \text{if } 1 \leq \sigma(\mathrm{x}) \leq n \\ \text{undefined} & \text{if } \sigma(\mathrm{x}) \notin \{1, \ldots, n\} \end{cases} \end{aligned}$$

- Fixpoint:

$$\mathfrak{C}[\![c]\!](\sigma_0) = \mathrm{fix}(\Phi)(\sigma_1) = \begin{cases} \sigma[\mathrm{y} \mapsto (\sigma(\mathrm{x}))!, \mathrm{x} \mapsto 1] & \text{if } \sigma(\mathrm{x}) \geq 1 \\ \text{undefined} & \text{otherwise} \end{cases}$$

Summary: Denotational Semantics

Outline of Lecture 8

Recap: CCPOs and Continuous Functions

The Fixpoint Theorem

Application to fix(Φ)

Summary: Denotational Semantics

Equivalence of Operational and Denotational Semantics

Summary: Denotational Semantics

Summary: Denotational Semantics

- Semantic model: partial state transformations ($\Sigma \dashrightarrow \Sigma$)

Summary: Denotational Semantics

Summary: Denotational Semantics

- Semantic model: partial state transformations ($\Sigma \dashrightarrow \Sigma$)
- Compositional definition of functional $\mathfrak{C}[\![.]\!] : \mathit{Cmd} \to (\Sigma \dashrightarrow \Sigma)$

Summary: Denotational Semantics

Summary: Denotational Semantics

- Semantic model: partial state transformations $(\Sigma \dashrightarrow \Sigma)$
- Compositional definition of functional $\mathfrak{C}[\![.]\!] : \mathit{Cmd} \to (\Sigma \dashrightarrow \Sigma)$
- Capturing the recursive nature of loops by a fixpoint definition (for a continuous function on a CCPO)

Summary: Denotational Semantics

Summary: Denotational Semantics

- Semantic model: partial state transformations $(\Sigma \dashrightarrow \Sigma)$
- Compositional definition of functional $\mathfrak{C}[\![.]\!] : \mathit{Cmd} \to (\Sigma \dashrightarrow \Sigma)$
- Capturing the recursive nature of loops by a fixpoint definition (for a continuous function on a CCPO)
- Approximation by fixpoint iteration

Equivalence of Operational and Denotational Semantics

Outline of Lecture 8

Recap: CCPOs and Continuous Functions

The Fixpoint Theorem

Application to fix(Φ)

Summary: Denotational Semantics

Equivalence of Operational and Denotational Semantics

Equivalence of Operational and Denotational Semantics

Equivalence of Semantics I

Remember: in Definition 4.1, $\mathfrak{O}[\![.]\!] : \mathit{Cmd} \to (\Sigma \dashrightarrow \Sigma)$ was given by

$$\mathfrak{O}[\![c]\!](\sigma) = \sigma' \iff \langle c, \sigma\rangle \to \sigma'$$

Equivalence of Operational and Denotational Semantics

Equivalence of Semantics I

Remember: in Definition 4.1, $\mathfrak{O}[\![.]\!] : \mathit{Cmd} \to (\Sigma \dashrightarrow \Sigma)$ was given by

$$\mathfrak{O}[\![c]\!](\sigma) = \sigma' \iff \langle c, \sigma \rangle \to \sigma'$$

Theorem 8.5 (Coincidence Theorem)

For every $c \in \mathit{Cmd}$,

$$\mathfrak{O}[\![c]\!] = \mathfrak{C}[\![c]\!],$$

i.e., $\langle c, \sigma \rangle \to \sigma'$ iff $\mathfrak{C}[\![c]\!](\sigma) = \sigma'$, and thus $\mathfrak{O}[\![.]\!] = \mathfrak{C}[\![.]\!]$.

Equivalence of Operational and Denotational Semantics

Equivalence of Semantics II

The proof of Theorem 8.5 employs the following auxiliary propositions:

Lemma 8.6

1. *For every* $a \in AExp$, $\sigma \in \Sigma$, *and* $z \in \mathbb{Z}$*:*

$$\langle a, \sigma \rangle \to z \iff \mathfrak{A}[\![a]\!](\sigma) = z.$$

Equivalence of Operational and Denotational Semantics

Equivalence of Semantics II

The proof of Theorem 8.5 employs the following auxiliary propositions:

Lemma 8.6

1. *For every* $a \in AExp$, $\sigma \in \Sigma$, *and* $z \in \mathbb{Z}$:

$$\langle a, \sigma \rangle \to z \iff \mathfrak{A}[\![a]\!](\sigma) = z.$$

2. *For every* $b \in BExp$, $\sigma \in \Sigma$, *and* $t \in \mathbb{B}$:

$$\langle b, \sigma \rangle \to t \iff \mathfrak{B}[\![b]\!](\sigma) = t.$$

Equivalence of Operational and Denotational Semantics

Equivalence of Semantics II

The proof of Theorem 8.5 employs the following auxiliary propositions:

Lemma 8.6

1. *For every $a \in AExp$, $\sigma \in \Sigma$, and $z \in \mathbb{Z}$:*

$$\langle a, \sigma \rangle \to z \iff \mathfrak{A}[\![a]\!](\sigma) = z.$$

2. *For every $b \in BExp$, $\sigma \in \Sigma$, and $t \in \mathbb{B}$:*

$$\langle b, \sigma \rangle \to t \iff \mathfrak{B}[\![b]\!](\sigma) = t.$$

Proof.

1. structural induction on a
2. structural induction on b

□

Equivalence of Semantics III

Proof (Theorem 8.5).

We have to show that

$$\langle c, \sigma \rangle \to \sigma' \iff \mathfrak{C}[\![c]\!](\sigma) = \sigma'$$

$\Rightarrow$ by structural induction over the derivation tree of $\langle c, \sigma \rangle \to \sigma'$

$\Leftarrow$ by structural induction over c (with a nested complete induction over fixpoint index n)

(on the board) $\square$

Equivalence of Operational and Denotational Semantics

Overview: Operational/Denotational Semantics

Definition (3.2; Execution relation for statements)

$$\text{(skip)}\;\frac{}{\langle \texttt{skip}, \sigma\rangle \to \sigma} \qquad \text{(asgn)}\;\frac{\langle a, \sigma\rangle \to z}{\langle x := a, \sigma\rangle \to \sigma[x \mapsto z]}$$

$$\text{(seq)}\;\frac{\langle c_1, \sigma\rangle \to \sigma' \quad \langle c_2, \sigma'\rangle \to \sigma''}{\langle c_1 ; c_2, \sigma\rangle \to \sigma''} \qquad \text{(if-t)}\;\frac{\langle b, \sigma\rangle \to \mathsf{true} \quad \langle c_1, \sigma\rangle \to \sigma'}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2 \texttt{ end}, \sigma\rangle \to \sigma'}$$

$$\text{(if-f)}\;\frac{\langle b, \sigma\rangle \to \mathsf{false} \quad \langle c_2, \sigma\rangle \to \sigma'}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2 \texttt{ end}, \sigma\rangle \to \sigma'} \qquad \text{(wh-f)}\;\frac{\langle b, \sigma\rangle \to \mathsf{false}}{\langle \texttt{while } b \texttt{ do } c \texttt{ end}, \sigma\rangle \to \sigma}$$

$$\text{(wh-t)}\;\frac{\langle b, \sigma\rangle \to \mathsf{true} \quad \langle c, \sigma\rangle \to \sigma' \quad \langle \texttt{while } b \texttt{ do } c \texttt{ end}, \sigma'\rangle \to \sigma''}{\langle \texttt{while } b \texttt{ do } c \texttt{ end}, \sigma\rangle \to \sigma''}$$

Definition (6.3; Denotational semantics of statements)

$$\begin{aligned}
\mathfrak{C}[\![\texttt{skip}]\!] &:= \mathrm{id}_\Sigma \\
\mathfrak{C}[\![x := a]\!]\sigma &:= \sigma[x \mapsto \mathfrak{A}[\![a]\!]\sigma] \\
\mathfrak{C}[\![c_1 ; c_2]\!] &:= \mathfrak{C}[\![c_2]\!] \circ \mathfrak{C}[\![c_1]\!] \\
\mathfrak{C}[\![\texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2 \texttt{ end}]\!] &:= \mathrm{cond}(\mathfrak{B}[\![b]\!], \mathfrak{C}[\![c_1]\!], \mathfrak{C}[\![c_2]\!]) \\
\mathfrak{C}[\![\texttt{while } b \texttt{ do } c \texttt{ end}]\!] &:= \mathrm{fix}(\Phi) \text{ where } \Phi(f) := \mathrm{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathrm{id}_\Sigma)
\end{aligned}$$