# Denotational Semantics

Hongfei Fu

John Hopcroft Center for Computer Science
Shanghai Jiao Tong University

Mar. 25th

# Today's Topic

## Denotational Semantics

- complete partial orders
- continuous functions
- a least-fixed-point theorem
- rigorous definition for denotational semantics

# Denotational Semantics: An Informal View

Textbook, Page 55 – Page 61

# Motivation

## Denotational Semantics

- a functional viewpoint for programs
- programs as input-output transformers

# Motivation

## Equivalence over Commands

- $c_0, c_1$: two commands

$$c_0 \sim c_1 \quad \text{iff} \quad \forall \sigma, \sigma'. \left( \langle c_0, \sigma \rangle \to \sigma' \Leftrightarrow \langle c_1, \sigma \rangle \to \sigma' \right)$$

$$\text{iff} \quad \{(\sigma, \sigma') \mid \langle c_0, \sigma \rangle \to \sigma'\} = \{(\sigma, \sigma') \mid \langle c_1, \sigma \rangle \to \sigma'\}$$

# Motivation

- $c$ is represented by $\{(\sigma, \sigma') \mid \langle c, \sigma \rangle \to \sigma'\}$.
- commands as partial functions from inputs to outputs

# Motivation

## The Mathematical Layout

- arithmetic expressions $a$: $\mathcal{A}[\![a]\!] : \Sigma \to \mathbb{Z}$
- boolean expressions $b$: $\mathcal{B}[\![b]\!] : \Sigma \to \{\textbf{true}, \textbf{false}\}$
- commands $c$: $\mathcal{C}[\![c]\!] : \Sigma \rightharpoonup \Sigma$

# Arithmetic Expressions

## Definition through Well-Founded Recursion

- $\mathcal{A}[\![n]\!](\sigma) := n$ for any state $\sigma$;
- $\mathcal{A}[\![X]\!](\sigma) := \sigma(X)$ for any state $\sigma$;

# Arithmetic Expressions

## Definition through Well-Founded Recursion

- $\mathcal{A}[\![n]\!](\sigma) := n$ for any state $\sigma$;
- $\mathcal{A}[\![X]\!](\sigma) := \sigma(X)$ for any state $\sigma$;
- $\mathcal{A}[\![a_0 + a_1]\!](\sigma) := \mathcal{A}[\![a_0]\!](\sigma) + \mathcal{A}[\![a_1]\!](\sigma)$ for any state $\sigma$;
- $\mathcal{A}[\![a_0 - a_1]\!](\sigma) := \mathcal{A}[\![a_0]\!](\sigma) - \mathcal{A}[\![a_1]\!](\sigma)$ for any state $\sigma$;
- $\mathcal{A}[\![a_0 \times a_1]\!](\sigma) := \mathcal{A}[\![a_0]\!](\sigma) \times \mathcal{A}[\![a_1]\!](\sigma)$ for any state $\sigma$;

# Boolean Expressions

## Definition through Well-Founded Recursion

- $\mathcal{B}[\![\textbf{true}]\!](\sigma) := \textbf{true}$;
- $\mathcal{B}[\![\textbf{false}]\!](\sigma) := \textbf{false}$;

# Boolean Expressions

## Definition through Well-Founded Recursion

- $$\mathcal{B}[\![a_0 = a_1]\!](\sigma) := \begin{cases} \textbf{true} & \text{if } \mathcal{A}[\![a_0]\!](\sigma) = \mathcal{A}[\![a_1]\!](\sigma) \\ \textbf{false} & \text{if } \mathcal{A}[\![a_0]\!](\sigma) \neq \mathcal{A}[\![a_1]\!](\sigma) \end{cases}$$
- $$\mathcal{B}[\![a_0 \leq a_1]\!](\sigma) := \begin{cases} \textbf{true} & \text{if } \mathcal{A}[\![a_0]\!](\sigma) \leq \mathcal{A}[\![a_1]\!](\sigma) \\ \textbf{false} & \text{if } \mathcal{A}[\![a_0]\!](\sigma) > \mathcal{A}[\![a_1]\!](\sigma) \end{cases}$$

# Boolean Expressions

## Definition through Well-Founded Recursion

- $\mathcal{B}[\![\neg b]\!](\sigma) := \neg \mathcal{B}[\![b]\!](\sigma)$
- $\mathcal{B}[\![b_0 \wedge b_1]\!](\sigma) := \mathcal{B}[\![b_0]\!](\sigma)$ & $\mathcal{B}[\![b_1]\!](\sigma)$
- $\mathcal{B}[\![b_0 \vee b_1]\!](\sigma) := \mathcal{B}[\![b_0]\!](\sigma)$ or $\mathcal{B}[\![b_1]\!](\sigma)$

## Exercise

Prove by structural induction that for all arithmetic expressions $a$ and boolean expressions $b$, $\mathcal{A}[\![a]\!]$ and $\mathcal{B}[\![b]\!]$ are indeed functions.

# Commands

## Skip and Assignment

- $\mathcal{C}[\![\textbf{skip}]\!] := \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$;
- $\mathcal{C}[\![X := a]\!] := \{(\sigma, \sigma\,[\mathcal{A}[\![a]\!](\sigma)/X]) \mid \sigma \in \Sigma\}$;

# Commands

## Sequential Composition

- $\mathcal{C}[\![c_0; c_1]\!] := \mathcal{C}[\![c_1]\!] \circ \mathcal{C}[\![c_0]\!]$;

# Commands

## Conditional Branch

- $\mathcal{C}[\![\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1]\!]$ is the union of the following two sets:
  - $\{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![c_0]\!]\}$
  - $\{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![c_1]\!]\}$

# Commands

## While Loop

- $w =$ **while** $b$ **do** $c$;

# Commands

## While Loop

- $w =$ **while** $b$ **do** $c$;
- How??

# Commands

## A First Attempt

- $w =$ **while** $b$ **do** $c$;
- $w \sim$ **if** $b$ **then** $c$; $w$ **else skip**;

# Commands

## A First Attempt

- $w =$ **while** $b$ **do** $c$;
- $w \sim$ **if** $b$ **then** $c$; $w$ **else skip**;
-

$$\mathcal{C}[\![w]\!] = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup$$
$$\{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\}$$

# Commands

## The Problem

- $$\mathcal{C}[\![w]\!] = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\}$$
- $\mathcal{C}[\![w]\!]$ is not recursively defined.

# Commands

## The Fixed-Point Phenomenon

- $w =$ **while** $b$ **do** $c$;
- $$\mathcal{C}[\![w]\!] = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\}$$
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

# Commands

## The Fixed-Point Phenomenon

- $w = \textbf{while}\ b\ \textbf{do}\ c;$
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

# Commands

## The Fixed-Point Phenomenon

- $w =$ **while** $b$ **do** $c$;
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

- Does any solution $R$ work ?

# Commands

## The Fixed-Point Phenomenon

- $w = \textbf{while } b \textbf{ do } c;$
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

- Does any solution $R$ work ?
  - The set $R = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$ is a solution when $c = \textbf{skip}$.

# Commands

## The Fixed-Point Phenomenon

- $w = \textbf{while } b \textbf{ do } c$;
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

- Does any solution $R$ work ?
  - The set $R = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$ is a solution when $c = \textbf{skip}$.
  - However, we desire $\mathcal{C}[\![w]\!] = \emptyset$ when $c = \textbf{skip}$ and $b = \textbf{true}$.

# Commands

## The Fixed-Point Phenomenon

- $w = \textbf{while } b \textbf{ do } c;$
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

- Does any solution $R$ work ?
  - The set $R = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$ is a solution when $c = \textbf{skip}$.
  - However, we desire $\mathcal{C}[\![w]\!] = \emptyset$ when $c = \textbf{skip}$ and $b = \textbf{true}$.
- What do we desire about $\mathcal{C}[\![w]\!]$ ?

# Commands

## The Fixed-Point Phenomenon

- $w = \textbf{while } b \textbf{ do } c;$
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

- Does any solution $R$ work ?
  - The set $R = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$ is a solution when $c = \textbf{skip}$.
  - However, we desire $\mathcal{C}[\![w]\!] = \emptyset$ when $c = \textbf{skip}$ and $b = \textbf{true}$.
- What do we desire about $\mathcal{C}[\![w]\!]$ ?
  - The set $R = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$ contains too much information.

# Commands

## The Fixed-Point Phenomenon

- $w = \textbf{while } b \textbf{ do } c;$
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

- Does any solution $R$ work ?
  - The set $R = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$ is a solution when $c = \textbf{skip}$.
  - However, we desire $\mathcal{C}[\![w]\!] = \emptyset$ when $c = \textbf{skip}$ and $b = \textbf{true}$.
- What do we desire about $\mathcal{C}[\![w]\!]$ ?
  - The set $R = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$ contains too much information.
  - $\mathcal{C}[\![w]\!]$ should be the solution with the least information.

# Complete Partial Orders

Textbook, Page 68 – Page 70

# Complete Partial Orders

## Motivation

- a partial order to compare elements
- a complete property in infinitely ascending sequences
- a fundamental characterization with least fixed points

# Complete Partial Orders

## Recall: Partial Orders

A partial order is an ordered pair $(P, \sqsubseteq)$ such that $P$ is a set and $\sqsubseteq$ is a binary relation $\sqsubseteq \subseteq P \times P$ satisfying the following conditions:

- (reflexibility) $\forall p \in P. p \sqsubseteq p$;
- (transitivity) $\forall p, q, r \in P. [(p \sqsubseteq q \ \& \ q \sqsubseteq r) \Rightarrow p \sqsubseteq r]$;
- (antisymmetry) $\forall p, q \in P. [(p \sqsubseteq q \ \& \ q \sqsubseteq p) \Rightarrow p = q]$.

# Complete Partial Orders

## Upper Bounds

- $(P, \sqsubseteq)$: a partial order
- $X$: a subset of $P$ (i.e., that satisfies $X \subseteq P$)

$p \in P$ is an upper bound of $X$ if $\forall q \in X. q \sqsubseteq p$.

# Complete Partial Orders

## Upper Bounds

- $(P, \sqsubseteq)$: a partial order
- $X$: a subset of $P$ (i.e., that satisfies $X \subseteq P$)

$p \in P$ is an upper bound of $X$ if $\forall q \in X. q \sqsubseteq p$.

## Least Upper Bounds

$p \in P$ is a least upper bound (in short, lub) of $X$ if

- $p$ is an upper bound of $X$, and
- for all upper bounds $q$ of $X$, $p \sqsubseteq q$

# Complete Partial Orders

## Upper Bounds

- $(P, \sqsubseteq)$: a partial order
- $X$: a subset of $P$ (i.e., that satisfies $X \subseteq P$)

$p \in P$ is an upper bound of $X$ if $\forall q \in X. q \sqsubseteq p$.

## Least Upper Bounds

$p \in P$ is a least upper bound (in short, lub) of $X$ if

- $p$ is an upper bound of $X$, and
- for all upper bounds $q$ of $X$, $p \sqsubseteq q$

## Exercise

For any $X \subseteq P$, $X$ has at most one least upper bound.

# Complete Partial Orders

## Least Upper Bounds

$p \in P$ is a least upper bound (in short, lub) of $X$ if

- $p$ is an upper bound of $X$, and
- for all upper bounds $q$ of $X$, $p \sqsubseteq q$

# Complete Partial Orders

## Least Upper Bounds

$p \in P$ is a least upper bound (in short, lub) of $X$ if

- $p$ is an upper bound of $X$, and
- for all upper bounds $q$ of $X$, $p \sqsubseteq q$

## Notation

- The least upper bound of $X$ (if exists) is denoted by $\bigsqcup X$.
- If $X = \{d_1, \ldots, d_n\}$, then $d_1 \sqcup \cdots \sqcup d_n := \bigsqcup X$.

# Complete Partial Orders

## $\omega$-Chains

- $(P, \sqsubseteq)$: a partial order

An $\omega$-chain in $P$ is an infinite sequence $d_0, d_1, \ldots, d_n, \ldots$ in $P$ such that $d_0 \sqsubseteq d_1 \sqsubseteq \cdots \sqsubseteq d_n \sqsubseteq \ldots$.

# Complete Partial Orders

## $\omega$-Chains

- $(P, \sqsubseteq)$: a partial order

An $\omega$-chain in $P$ is an infinite sequence $d_0, d_1, \ldots, d_n, \ldots$ in $P$ such that $d_0 \sqsubseteq d_1 \sqsubseteq \cdots \sqsubseteq d_n \sqsubseteq \ldots$.

## Complete Partial Orders (CPOs)

$(P, \sqsubseteq)$ is a complete partial order (cpo) if for any $\omega$-chain

$$d_0 \sqsubseteq d_1 \sqsubseteq \cdots \sqsubseteq d_n \sqsubseteq \ldots$$

in $P$, the least upper bound

$$\bigsqcup_{n \in \omega} d_n := \bigsqcup \{d_n \mid n \in \omega\} = \bigsqcup \{d_0, d_1, \ldots, d_n, \ldots\}$$

exists in $P$.

# Complete Partial Orders

## Least Elements

- $(P, \sqsubseteq)$: a partial order

$p \in P$ is a least element if $\forall q \in P. p \sqsubseteq q$.

# Complete Partial Orders

## Least Elements

- $(P, \sqsubseteq)$: a partial order

$p \in P$ is a least element if $\forall q \in P. p \sqsubseteq q$.

## Exercise

Show that the least element, if exists, is unique.

# Complete Partial Orders

## Least Elements

- $(P, \sqsubseteq)$: a partial order

$p \in P$ is a least element if $\forall q \in P. p \sqsubseteq q$.

## Exercise

Show that the least element, if exists, is unique.

## CPOs with Bottom

- $(P, \sqsubseteq)$: a cpo

$(P, \sqsubseteq)$ is a cpo with bottom if $P$ has a (unique) least element $\bot_P$.

# Complete Partial Orders

## Set Inclusion

- $A$: a set
- $D := 2^A$
- $\sqsubseteq := \{(X, Y) \in D \times D \mid X \subseteq Y\}$

# Complete Partial Orders

## Set Inclusion

- $A$: a set
- $D := 2^A$
- $\sqsubseteq := \{(X, Y) \in D \times D \mid X \subseteq Y\}$

## Exercise

Verify that $(D, \sqsubseteq)$ is a cpo with bottom.

# Complete Partial Orders

## Set Inclusion

- $A$: a set
- $D := 2^A$
- $\sqsubseteq := \{(X, Y) \in D \times D \mid X \subseteq Y\}$

## Exercise

Verify that $(D, \sqsubseteq)$ is a cpo with bottom.

- $\bigsqcup_{n \in \omega} A_n = \bigcup_n A_n$ given $A_0 \subseteq A_1 \subseteq \ldots$
- $\bot_D = \emptyset$

# Complete Partial Orders

## Partial Functions

- $B, C$: sets
- $D := \{F \mid F : B \rightharpoonup C\}$
- $\sqsubseteq := \{(F, G) \in D \times D \mid F \subseteq G\}$

# Complete Partial Orders

## Partial Functions

- $B, C$: sets
- $D := \{F \mid F : B \rightharpoonup C\}$
- $\sqsubseteq := \{(F, G) \in D \times D \mid F \subseteq G\}$

## Exercise

Verify that $(D, \sqsubseteq)$ is a cpo with bottom.

# Complete Partial Orders

## Partial Functions

- $B, C$: sets
- $D := \{F \mid F : B \rightharpoonup C\}$
- $\sqsubseteq := \{(F, G) \in D \times D \mid F \subseteq G\}$

## Exercise

Verify that $(D, \sqsubseteq)$ is a cpo with bottom.

- $\bigsqcup_{n \in \omega} F_n = \bigcup_n F_n$ given $F_0 \subseteq F_1 \subseteq \ldots$
- (important!) $\bigcup_n F_n$ is a function!
- $\bot_D = \emptyset$

# Complete Partial Orders

## Intervals

- $D := [0, \infty) \cup \{\infty\}$
- $\sqsubseteq := \{(x, y) \in D \times D \mid x \leq y\}$

# Complete Partial Orders

## Intervals

- $D := [0, \infty) \cup \{\infty\}$
- $\sqsubseteq := \{(x, y) \in D \times D \mid x \leq y\}$

## Exercise

Verify that $(D, \sqsubseteq)$ is a cpo with bottom.

# Complete Partial Orders

## Intervals

- $D := [0, \infty) \cup \{\infty\}$
- $\sqsubseteq := \{(x, y) \in D \times D \mid x \leq y\}$

## Exercise

Verify that $(D, \sqsubseteq)$ is a cpo with bottom.

- $\bigsqcup_{n \in \omega} x_n = \sup_n x_n$ if $x_0 \leq x_1 \leq \ldots$
- $\bot_D = 0$

# Complete Partial Orders

## Intervals

- $D := [0, 1)$
- $\sqsubseteq := \{(x, y) \in D \times D \mid x \leq y\}$

# Complete Partial Orders

## Intervals

- $D := [0, 1)$
- $\sqsubseteq := \{(x, y) \in D \times D \mid x \leq y\}$

## Exercise

Is $(D, \sqsubseteq)$ a cpo (with bottom) ?

# Complete Partial Orders

## Real Numbers

- $D := \mathbb{R}$
- $\sqsubseteq := \{(x, y) \in D \times D \mid x \leq y\}$

## Exercise

Is $(D, \sqsubseteq)$ a cpo (with bottom) ?

## Continuous Functions

Textbook, Page 71 – Page 72

# Monotonic Functions

## Definition

- $(D, \sqsubseteq_D)$ and $(E, \sqsubseteq_E)$: partial orders

A function $f : D \to E$ is monotonic if

$$\forall d, d' \in D. \left[ d \sqsubseteq_D d' \Rightarrow f(d) \sqsubseteq_E f(d') \right]$$

## Example

- partial order: $(\mathbb{R}, \leq)$
- $f(x) = 2 \cdot x$ is a monotonic function.

# Continuous Functions

## Definition

- $(D, \sqsubseteq_D)$ and $(E, \sqsubseteq_E)$: cpo's

A function $f : D \to E$ is continuous if the followings hold:

- $f$ is monotonic;
- for all $\omega$-chains $d_0 \sqsubseteq d_1 \sqsubseteq \cdots \sqsubseteq d_n \sqsubseteq \ldots$ in $D$, we have that

$$\bigsqcup_{n \in \omega} f(d_n) = f\left(\bigsqcup_{n \in \omega} d_n\right)$$

# Continuous Functions

## Definition

- $(D, \sqsubseteq_D)$ and $(E, \sqsubseteq_E)$: cpo's

A function $f : D \to E$ is continuous if the followings hold:

- $f$ is monotonic;
- for all $\omega$-chains $d_0 \sqsubseteq d_1 \sqsubseteq \cdots \sqsubseteq d_n \sqsubseteq \ldots$ in $D$, we have that

$$\bigsqcup_{n \in \omega} f(d_n) = f\left(\bigsqcup_{n \in \omega} d_n\right)$$

## Example

- the cpo: $([0, 1], \leq)$
- $f(x) = 2 \cdot x$ is a continuous function.

# Continuous Functions

## Definition

- $(D, \sqsubseteq_D)$ and $(E, \sqsubseteq_E)$: cpo's

A function $f : D \to E$ is continuous if the followings hold:

- $f$ is monotonic;
- for all $\omega$-chains $d_0 \sqsubseteq d_1 \sqsubseteq \cdots \sqsubseteq d_n \sqsubseteq \ldots$ in $D$, we have that

$$\bigsqcup_{n \in \omega} f(d_n) = f\left(\bigsqcup_{n \in \omega} d_n\right)$$

# Continuous Functions

## Definition

- $(D, \sqsubseteq_D)$ and $(E, \sqsubseteq_E)$: cpo's

A function $f : D \to E$ is continuous if the followings hold:

- $f$ is monotonic;
- for all $\omega$-chains $d_0 \sqsubseteq d_1 \sqsubseteq \cdots \sqsubseteq d_n \sqsubseteq \ldots$ in $D$, we have that

$$\bigsqcup_{n \in \omega} f(d_n) = f\left(\bigsqcup_{n \in \omega} d_n\right)$$

## Question

Can one construct a monotonic function which is not continuous?

# Fixed Points

## Definition

- $(D, \sqsubseteq_D)$: a partial order
- $f : D \to D$: a function

An element $d \in D$ is:

- a fixed point of $f$ if $f(d) = d$;
- a prefixed point of $f$ if $f(d) \sqsubseteq d$;

# Fixed Points

## The Fixed-Point Theorem

Suppose

- $(D, \sqsubseteq_D)$: a cpo with bottom $\bot_D$
- $f : D \to D$: a continuous function
- $\bot_D \sqsubseteq_D f(\bot_D) \sqsubseteq_D \cdots \sqsubseteq_D f^n(\bot_D) \sqsubseteq_D \ldots$
- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$

# Fixed Points

## The Fixed-Point Theorem

Suppose

- $(D, \sqsubseteq_D)$: a cpo with bottom $\bot_D$
- $f : D \to D$: a continuous function
- $\bot_D \sqsubseteq_D f(\bot_D) \sqsubseteq_D \cdots \sqsubseteq_D f^n(\bot_D) \sqsubseteq_D \ldots$
- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$

Then

- $fix(f)$ is a fixed point of $f$: $f(fix(f)) = fix(f)$
- $fix(f)$ is the least prefixed point of $f$: $f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$
- $fix(f)$ is the least fixed point of $f$: $f(d) = d \Rightarrow fix(f) \sqsubseteq d$

# Fixed Points

## Proof

- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$
- $fix(f)$ is a fixed point of $f$: $f(fix(f)) = fix(f)$

# Fixed Points

## Proof

- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$
- $fix(f)$ is a fixed point of $f$: $f(fix(f)) = fix(f)$

$$
\begin{aligned}
f(fix(f)) &= f(\bigsqcup_{n \in \omega} f^n(\bot_D)) \\
&= \bigsqcup_{n \in \omega} f^{n+1}(\bot_D) \\
&= \bigsqcup_{n \in \omega} f^{n+1}(\bot_D) \sqcup \bot_D \\
&= \bigsqcup_{n \in \omega} f^n(\bot_D) \\
&= fix(f)
\end{aligned}
$$

# Fixed Points

## Fixed-Point Theorem: Proof

- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$
- $fix(f)$ is the least prefixed point of $f$: $f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$

# Fixed Points

## Fixed-Point Theorem: Proof

- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$
- $fix(f)$ is the least prefixed point of $f$: $f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$
- $d$: a prefixed point (i.e., $f(d) \sqsubseteq d$)

# Fixed Points

## Fixed-Point Theorem: Proof

- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$
- $fix(f)$ is the least prefixed point of $f$: $f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$
- $d$: a prefixed point (i.e., $f(d) \sqsubseteq d$)
- $\bot_D \sqsubseteq d$

# Fixed Points

## Fixed-Point Theorem: Proof

- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$
- $fix(f)$ is the least prefixed point of $f$: $f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$
- $d$: a prefixed point (i.e., $f(d) \sqsubseteq d$)
- $\bot_D \sqsubseteq d$
- $f^n(\bot_D) \sqsubseteq d \Rightarrow f^{n+1}(\bot_D) \sqsubseteq f(d) \sqsubseteq d$

# Fixed Points

## Fixed-Point Theorem: Proof

- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$
- $fix(f)$ is the least prefixed point of $f$: $f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$
- $d$: a prefixed point (i.e., $f(d) \sqsubseteq d$)
- $\bot_D \sqsubseteq d$
- $f^n(\bot_D) \sqsubseteq d \Rightarrow f^{n+1}(\bot_D) \sqsubseteq f(d) \sqsubseteq d$
- $\forall n. (f^n(\bot_D) \sqsubseteq d)$

# Fixed Points

## Fixed-Point Theorem: Proof

- $fix(f) := \bigsqcup_{n \in \omega} f^n(\bot_D)$
- $fix(f)$ is the least prefixed point of $f$: $f(d) \sqsubseteq d \Rightarrow fix(f) \sqsubseteq d$
- $d$: a prefixed point (i.e., $f(d) \sqsubseteq d$)
- $\bot_D \sqsubseteq d$
- $f^n(\bot_D) \sqsubseteq d \Rightarrow f^{n+1}(\bot_D) \sqsubseteq f(d) \sqsubseteq d$
- $\forall n. (f^n(\bot_D) \sqsubseteq d)$
- $fix(f) = \bigsqcup_{n \in \omega} f^n(\bot_D) \sqsubseteq d$

# Denotational Semantics: Formal Definition

Textbook, Page 55 – Page 61

# Recall: Denotational Semantics

- commands as partial functions from inputs to outputs
- $c$ is represented by the partial function $\{(\sigma, \sigma') \mid \langle c, \sigma \rangle \to \sigma'\}$.

# Denotational Semantics

## Recall: The Mathematical Layout

- arithmetic expressions $a$: $\mathcal{A}[\![a]\!] : \Sigma \to \mathbb{N}$
- boolean expressions $b$: $\mathcal{B}[\![b]\!] : \Sigma \to \{\textbf{true}, \textbf{false}\}$
- commands $c$: $\mathcal{C}[\![c]\!] : \Sigma \rightharpoonup \Sigma$

# Arithmetic Expressions

## Recall: Definition through Well-Founded Recursion

- $\mathcal{A}[\![n]\!](\sigma) := n$ for any state $\sigma$;
- $\mathcal{A}[\![X]\!](\sigma) := \sigma(X)$ for any state $\sigma$;

# Arithmetic Expressions

## Recall: Definition through Well-Founded Recursion

- $\mathcal{A}[\![n]\!](\sigma) := n$ for any state $\sigma$;
- $\mathcal{A}[\![X]\!](\sigma) := \sigma(X)$ for any state $\sigma$;
- $\mathcal{A}[\![a_0 + a_1]\!](\sigma) := \mathcal{A}[\![a_0]\!](\sigma) + \mathcal{A}[\![a_1]\!](\sigma)$ for any state $\sigma$;
- $\mathcal{A}[\![a_0 - a_1]\!](\sigma) := \mathcal{A}[\![a_0]\!](\sigma) - \mathcal{A}[\![a_1]\!](\sigma)$ for any state $\sigma$;
- $\mathcal{A}[\![a_0 \times a_1]\!](\sigma) := \mathcal{A}[\![a_0]\!](\sigma) \times \mathcal{A}[\![a_1]\!](\sigma)$ for any state $\sigma$;

# Boolean Expressions

## Recall: Definition through Well-Founded Recursion

- $\mathcal{B}[\![\textbf{true}]\!](\sigma) := \textbf{true}$;
- $\mathcal{B}[\![\textbf{false}]\!](\sigma) := \textbf{false}$;

# Boolean Expressions

## Recall: Definition through Well-Founded Recursion

- $$\mathcal{B}[\![a_0 = a_1]\!](\sigma) := \begin{cases} \textbf{true} & \text{if } \mathcal{A}[\![a_0]\!](\sigma) = \mathcal{A}[\![a_1]\!](\sigma) \\ \textbf{false} & \text{if } \mathcal{A}[\![a_0]\!](\sigma) \neq \mathcal{A}[\![a_1]\!](\sigma) \end{cases}$$
- $$\mathcal{B}[\![a_0 \leq a_1]\!](\sigma) := \begin{cases} \textbf{true} & \text{if } \mathcal{A}[\![a_0]\!](\sigma) \leq \mathcal{A}[\![a_1]\!](\sigma) \\ \textbf{false} & \text{if } \mathcal{A}[\![a_0]\!](\sigma) > \mathcal{A}[\![a_1]\!](\sigma) \end{cases}$$

# Boolean Expressions

## Recall: Definition through Well-Founded Recursion

- $\mathcal{B}[\![\neg b]\!](\sigma) := \neg \mathcal{B}[\![b]\!](\sigma)$
- $\mathcal{B}[\![b_0 \wedge b_1]\!](\sigma) := \mathcal{B}[\![b_0]\!](\sigma)$ & $\mathcal{B}[\![b_1]\!](\sigma)$
- $\mathcal{B}[\![b_0 \vee b_1]\!](\sigma) := \mathcal{B}[\![b_0]\!](\sigma)$ or $\mathcal{B}[\![b_1]\!](\sigma)$

# Commands

## Recall: Assignment and Skip

- $\mathcal{C}[\![\textbf{skip}]\!] := \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$;
- $\mathcal{C}[\![X := a]\!] := \{(\sigma, \sigma\,[\mathcal{A}[\![a]\!](\sigma)/X]) \mid \sigma \in \Sigma\}$;

# Commands

## Recall: Sequential Composition

- $\mathcal{C}[\![c_0; c_1]\!] := \mathcal{C}[\![c_1]\!] \circ \mathcal{C}[\![c_0]\!]$;

# Commands

## If Branch

- $\mathcal{C}[\![\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1]\!]$ is the union of the following two sets:
  - $\{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![c_0]\!]\}$;
  - $\{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![c_1]\!]\}$

# Commands

## While Loop

- $w =$ **while** $b$ **do** $c$;
- $w \sim$ **if** $b$ **then** $c$; $w$ **else skip**;

# Commands

## While Loop

- $w =$ **while** $b$ **do** $c$;
- $w \sim$ **if** $b$ **then** $c$; $w$ **else skip**;
-

$$\mathcal{C}[\![w]\!] = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\}$$

# Commands

## The Fixed-Point Phenomenon

- $w =$ **while** $b$ **do** $c$;
- $$\mathcal{C}[\![w]\!] = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\}$$
- $\mathcal{C}[\![w]\!]$ should be a solution to the following set equation:

$$R = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in R \circ \mathcal{C}[\![c]\!]\}$$

# Commands

## The Fixed-Point Phenomenon

- $w =$ **while** $b$ **do** $c$;
- $$\mathcal{C}[\![w]\!] = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\}$$
- Define $\Gamma : (\Sigma \rightharpoonup \Sigma) \to (\Sigma \rightharpoonup \Sigma)$ by

$$\Gamma(F) := \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in F \circ \mathcal{C}[\![c]\!]\}$$

# Commands

## The Fixed-Point Phenomenon

- $w =$ **while** $b$ **do** $c$;
- $$\mathcal{C}[\![w]\!] = \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\}$$
- Define $\Gamma : (\Sigma \rightharpoonup \Sigma) \to (\Sigma \rightharpoonup \Sigma)$ by
$$\Gamma(F) := \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in F \circ \mathcal{C}[\![c]\!]\}$$
- $\Gamma(\mathcal{C}[\![w]\!]) = \mathcal{C}[\![w]\!]$.

# Commands

## The Fixed-Point Phenomenon

- $w = \textbf{while}\ b\ \textbf{do}\ c$;
- Define $\Gamma : (\Sigma \rightharpoonup \Sigma) \to (\Sigma \rightharpoonup \Sigma)$ by

$$\Gamma(F) := \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in F \circ \mathcal{C}[\![c]\!]\}$$

- $((\Sigma \rightharpoonup \Sigma), \subseteq)$: the complete partial order
- $\Gamma$: a continuous function for $((\Sigma \rightharpoonup \Sigma), \subseteq)$

# Commands

## The Fixed-Point Phenomenon

- $w = \textbf{while}\ b\ \textbf{do}\ c$;
- Define $\Gamma : (\Sigma \rightharpoonup \Sigma) \to (\Sigma \rightharpoonup \Sigma)$ by

$$\Gamma(F) := \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in F \circ \mathcal{C}[\![c]\!]\}$$

- $((\Sigma \rightharpoonup \Sigma), \subseteq)$: the complete partial order
- $\Gamma$: a continuous function for $((\Sigma \rightharpoonup \Sigma), \subseteq)$

## Exercise

- $((\Sigma \rightharpoonup \Sigma), \subseteq)$ is a complete partial order.
- $\Gamma$ is a continuous function for $((\Sigma \rightharpoonup \Sigma), \subseteq)$.

# Commands

## The Fixed-Point Phenomenon

- $w =$ **while** $b$ **do** $c$;
- Define $\Gamma : (\Sigma \rightharpoonup \Sigma) \to (\Sigma \rightharpoonup \Sigma)$ by

$$\Gamma(F) := \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in F \circ \mathcal{C}[\![c]\!]\}$$

- $((\Sigma \rightharpoonup \Sigma), \subseteq)$: the complete partial order
- $\Gamma$: a continuous function for $((\Sigma \rightharpoonup \Sigma), \subseteq)$

## Definition for $\mathcal{C}[\![w]\!]$

- $\mathcal{C}[\![w]\!] := fix(\Gamma) = \bigsqcup_{n \in \omega} \Gamma^n(\bot) = \bigcup_{n \in \omega} \Gamma^n(\emptyset)$;

# Commands

## The Intuition

- $w =$ **while** $b$ **do** $c$;
- $\Gamma : (\Sigma \rightharpoonup \Sigma) \to (\Sigma \rightharpoonup \Sigma)$ is given by

$$\Gamma(F) := \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in F \circ \mathcal{C}[\![c]\!]\}$$

- $\mathcal{C}[\![w]\!] := \mathit{fix}(\Gamma) = \bigsqcup_{n \in \omega} \Gamma^n(\bot) = \bigcup_{n \in \omega} \Gamma^n(\emptyset)$;

# Commands

## The Intuition

- $w =$ **while** $b$ **do** $c$;
- $\Gamma : (\Sigma \rightharpoonup \Sigma) \to (\Sigma \rightharpoonup \Sigma)$ is given by

$$\Gamma(F) := \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{false}\} \cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!](\sigma) = \textbf{true} \text{ and } (\sigma, \sigma') \in F \circ \mathcal{C}[\![c]\!]\}$$

- $\mathcal{C}[\![w]\!] := \mathit{fix}(\Gamma) = \bigsqcup_{n \in \omega} \Gamma^n(\bot) = \bigcup_{n \in \omega} \Gamma^n(\emptyset)$;

## Example

- $w =$ **while true do skip**
- $\mathcal{C}[\![w]\!] = \emptyset$

# Commands

**Theorem**

For all commands $c$, $\mathcal{C}[\![c]\!]$ is a partial function from $\Sigma$ to $\Sigma$.

# Commands

**Theorem**

For all commands $c$, $\mathcal{C}[\![c]\!]$ is a partial function from $\Sigma$ to $\Sigma$.

**Proof**

By structural induction.

# Summary

- complete partial orders
- continuous functions
- a fixed-point theorem
- denotational semantics

# Homework

## Problem 1

Let $D$ be a non-empty set and $(D \rightharpoonup D)$ be the set of all partial functions from $D$ to $D$. Prove that the partial order $((D \rightharpoonup D), \subseteq)$ (i.e., the set of partial functions ordered by set inclusion) is a complete partial order with bottom.

# Homework

## Problem 2

- Prove that $(\mathbb{N}, \geq)$ is a cpo.
- Prove that $(\mathcal{P}(\mathbb{N})\backslash\{\emptyset\}, \subseteq)$ is a cpo.
- Determine whether the function $F : \mathcal{P}(\mathbb{N})\backslash\{\emptyset\} \to \mathbb{N}$ given by

$$F(A) := \text{“the minimal number in } A\text{”}$$

is a continuous function from $(\mathcal{P}(\mathbb{N})\backslash\{\emptyset\}, \subseteq)$ to $(\mathbb{N}, \geq)$. Prove your answer.