

Semantics and Verification of Software

Lecture 7: Continuous Functions and Fixpoint Theorem

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

`noll@cs.rwth-aachen.de`

`http://www-i2.informatik.rwth-aachen.de/i2/svsw08/`

Winter semester 2008/09

- 1 Repetition: Chain-Complete Partial Orders
- 2 Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics

Goals:

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

Sufficient conditions:

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

on function Φ : **continuity**

Chain-Complete Partial Orders

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: $d = \bigsqcup S$).

Definition (Chain completeness)

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

- 1 Repetition: Chain-Complete Partial Orders
- 2 Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics

Definition 7.1 (Monotonicity)

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

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

Interpretation: monotonic functions “preserve information”

Example 7.2

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

Lemma 7.3

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

Proof.

on the board



The following lemma states how chains behave under monotonic functions.

Lemma 7.4

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs, $F : D \rightarrow D'$ monotonic, and $S \subseteq D$ a chain in D . Then:

- ① $F(S) := \{F(d) \mid d \in S\}$ is a chain in D' .
- ② $\bigsqcup F(S) \sqsubseteq' F(\bigsqcup S)$.

Proof.

on the board



Continuity

A function F is continuous if applying F and taking LUBs can be exchanged

Definition 7.5 (Continuity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs and $F : D \rightarrow D'$ 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 7.6

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

Proof.

on the board



- 1 Repetition: Chain-Complete Partial Orders
- 2 Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics

The Fixpoint Theorem

Theorem 7.7 (Fixpoint Theorem by Tarski and Knaster)

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

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

is the least fixpoint of F where

$$F^0(d) := d \text{ and } F^{n+1}(d) := F(F^n(d)).$$

Proof.

on the board



Application to $\text{fix}(\Phi)$ II

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

Corollary 7.8

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

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

Proof.

Using

- Lemma 6.9 ($(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ CCPO with least element $f_\emptyset$; LUB = union of graphs)
- Lemma 7.6 (Φ continuous)
- Theorem 7.7 (Fixpoint Theorem)



- 1 Repetition: Chain-Complete Partial Orders
- 2 Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics

Example 7.9 (Factorial program)

- Let $c \in Cmd$ be given by

$y := 1; \text{ while } \neg(x=1) \text{ do } (y := y * x; x := x - 1)$

- For every initial state $\sigma_0 \in \Sigma$, Def. 4.6 yields:

$$\mathcal{E}[[c]](\sigma_0) = \text{fix}(\Phi)(\sigma_1)$$

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

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

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

- Approximations of least fixpoint of Φ according to Theorem 7.7:

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

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

Example 7.9 (Factorial program; continued)

$$\begin{aligned}
 f_0(\sigma) &:= \Phi^0(f_\emptyset)(\sigma) \\
 &= f_\emptyset(\sigma) \\
 &= \text{undefined} \\
 f_1(\sigma) &:= \Phi^1(f_\emptyset)(\sigma) \\
 &= \Phi(f_0)(\sigma) \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ f_0(\sigma') & \text{otherwise} \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma & \text{if } \sigma(x) = 1 \\ f_0(\sigma') & \text{otherwise} \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma & \text{if } \sigma(x) = 1 \\ \text{undefined} & \text{otherwise} \end{cases} \\
 f_2(\sigma) &:= \Phi^2(f_\emptyset)(\sigma) \\
 &= \Phi(f_1)(\sigma) \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ f_1(\sigma') & \text{otherwise} \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma' & \text{if } \sigma(x) \neq 1 \text{ and } \sigma'(x) = 1 \\ \text{undefined} & \text{if } \sigma(x) \neq 1 \text{ and } \sigma'(x) \neq 1 \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma' & \text{if } \sigma(x) = 2 \\ \text{undefined} & \text{if } \sigma(x) \neq 1 \text{ and } \sigma(x) \neq 2 \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma[y \mapsto 2 * \sigma(y), & \text{if } \sigma(x) = 2 \\ \quad x \mapsto 1] & \\ \text{undefined} & \text{if } \sigma(x) \neq 1 \\ & \text{and } \sigma(x) \neq 2 \end{cases}
 \end{aligned}$$

Example 7.9 (Factorial program; continued)

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

Example 7.9 (Factorial program; continued)

- n -th approximation:

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

- Fixpoint:

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

- 1 Repetition: Chain-Complete Partial Orders
- 2 Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics

Summary: Denotational Semantics

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