

Semantics and Verification of Software

Lecture 6: Denotational Semantics of WHILE II (Chain-Complete Partial Orders)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)



noll@cs.rwth-aachen.de

<http://www-i2.informatik.rwth-aachen.de/i2/svsw11/>

Winter Semester 2011/12

- 1 Repetition: Denotational Semantics of WHILE
- 2 Making It Precise
- 3 Chain-Complete Partial Orders

Definition (Denotational semantics of statements)

The (denotational) semantic functional for statements,

$$\mathcal{C}[\![\cdot]\!] : \text{Cmd} \rightarrow (\Sigma \dashrightarrow \Sigma),$$

is given by:

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

where $\Phi : (\Sigma \dashrightarrow \Sigma) \rightarrow (\Sigma \dashrightarrow \Sigma) : f \mapsto \text{cond}(\mathcal{B}[\![b]\!], f \circ \mathcal{C}[\![c]\!], \text{id}_\Sigma)$

Why Fixpoints?

- Goal: preserve **validity of equivalence**

$$\mathcal{C}[\text{while } b \text{ do } c] \stackrel{(*)}{=} \mathcal{C}[\text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}]$$

(cf. Lemma 4.3)

- Using the known parts of Def. 5.3, we obtain:

$$\begin{aligned} \mathcal{C}[\text{while } b \text{ do } c] & \stackrel{(*)}{=} \mathcal{C}[\text{if } b \text{ then } (c; \text{while } b \text{ do } c) \text{ else skip}] \\ & \stackrel{\text{Def. 5.3}}{=} \text{cond}(\mathcal{B}[b], \mathcal{C}[c; \text{while } b \text{ do } c], \mathcal{C}[\text{skip}]) \\ & \stackrel{\text{Def. 5.3}}{=} \text{cond}(\mathcal{B}[b], \mathcal{C}[\text{while } b \text{ do } c] \circ \mathcal{C}[c], \text{id}_{\Sigma}) \end{aligned}$$

- Abbreviating $f := \mathcal{C}[\text{while } b \text{ do } c]$ this yields:

$$f = \text{cond}(\mathcal{B}[b], f \circ \mathcal{C}[c], \text{id}_{\Sigma})$$

- Hence f must be a **solution** of this recursive equation
- In other words: f must be a **fixpoint** of the mapping

$$\Phi : (\Sigma \dashrightarrow \Sigma) \rightarrow (\Sigma \dashrightarrow \Sigma) : f \mapsto \text{cond}(\mathcal{B}[b], f \circ \mathcal{C}[c], \text{id}_{\Sigma})$$

(since the equation can be stated as $f = \Phi(f)$)

Characterization of $\text{fix}(\Phi)$

For $\Phi(f_0) = f_0$ and initial state $\sigma_0 \in \Sigma$, case distinction yields:

- ① Loop `while b do c` terminates after n iterations ($n \in \mathbb{N}$)
 $\implies f_0(\sigma_0) = \sigma_n$
 - ② Body c diverges in the n th iteration
 $\implies f_0(\sigma_0) = \text{undefined}$
 - ③ Loop `while b do c` diverges
 $\implies$ no condition on f_0 (only $f_0(\sigma_0) = f_0(\sigma_i)$ for every $i \in \mathbb{N}$)
- Not surprising since, e.g., for the loop `while true do skip` every $f : \Sigma \dashrightarrow \Sigma$ is a fixpoint:
$$\Phi(f) = \text{cond}(\mathcal{B}[\text{true}], f \circ \mathcal{C}[\text{skip}], \text{id}_\Sigma) = f$$
 - On the other hand, our operational understanding requires, for every $\sigma_0 \in \Sigma$,
$$\mathcal{C}[\text{while true do skip}]\sigma_0 = \text{undefined}$$

Conclusion

$\text{fix}(\Phi)$ is the **least defined fixpoint** of Φ .

- 1 Repetition: Denotational Semantics of WHILE
- 2 Making It Precise
- 3 Chain-Complete Partial Orders

To use fixpoint theory, the notion of “least defined” has to be made precise.

- Given $f, g : \Sigma \dashrightarrow \Sigma$, let

$$f \sqsubseteq g \iff \text{for every } \sigma, \sigma' \in \Sigma : f(\sigma) = \sigma' \implies g(\sigma) = \sigma'$$

(g is “at least as defined” as f)

- Equivalent to requiring

$$\text{graph}(f) \subseteq \text{graph}(g)$$

where

$$\text{graph}(h) := \{(\sigma, \sigma') \mid \sigma \in \Sigma, \sigma' = h(\sigma) \text{ defined}\} \subseteq \Sigma \times \Sigma$$

for every $h : \Sigma \dashrightarrow \Sigma$

Example 6.1

Let $x \in \text{Var}$ be fixed, and let $f_0, f_1, f_2, f_3 : \Sigma \dashrightarrow \Sigma$ be given by

$$\begin{aligned} f_0(\sigma) &:= \text{undefined} \\ f_1(\sigma) &:= \begin{cases} \sigma & \text{if } \sigma(x) \text{ even} \\ \text{undefined} & \text{otherwise} \end{cases} \\ f_2(\sigma) &:= \begin{cases} \sigma & \text{if } \sigma(x) \text{ odd} \\ \text{undefined} & \text{otherwise} \end{cases} \\ f_3(\sigma) &:= \sigma \end{aligned}$$

This implies $f_0 \sqsubseteq f_1 \sqsubseteq f_3$, $f_0 \sqsubseteq f_2 \sqsubseteq f_3$, $f_1 \not\sqsubseteq f_2$, and $f_2 \not\sqsubseteq f_1$

Characterization of $\text{fix}(\Phi)$ I

Now $\text{fix}(\Phi)$ can be characterized by:

- $\text{fix}(\Phi)$ is a **fixpoint** of Φ , i.e.,

$$\Phi(\text{fix}(\Phi)) = \text{fix}(\Phi)$$

- $\text{fix}(\Phi)$ is **minimal** with respect to $\sqsubseteq$, i.e., for every $f_0 : \Sigma \dashrightarrow \Sigma$ such that $\Phi(f_0) = f_0$,

$$\text{fix}(\Phi) \sqsubseteq f_0$$

Example 6.2

For `while true do skip` we obtain for every $f : \Sigma \dashrightarrow \Sigma$:

$$\Phi(f) = \text{cond}(\mathcal{B}[\text{true}], f \circ \mathcal{C}[\text{skip}], \text{id}_\Sigma) = f$$

$\implies \text{fix}(\Phi) = f_\emptyset$ where $f_\emptyset(\sigma) := \text{undefined}$ for every $\sigma \in \Sigma$
(that is, $\text{graph}(f_\emptyset) = \emptyset$)

Goals:

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

Sufficient conditions:

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

on function Φ : **continuity**

- 1 Repetition: Denotational Semantics of WHILE
- 2 Making It Precise
- 3 Chain-Complete Partial Orders

Definition 6.1 (Partial order)

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

reflexivity: $d_1 \sqsubseteq d_1$

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

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

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

Example 6.2

- 1 $(\mathbb{N}, \leq)$ is a total partial order
- 2 $(2^{\mathbb{N}}, \subseteq)$ is a (non-total) partial order
- 3 $(\mathbb{N}, <)$ is not a partial order (since not reflexive)

Lemma 6.3

$(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ is a partial order.

Proof.

on the board □

Definition 6.4 (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 $s_1, s_2 \in S$,
 $s_1 \sqsubseteq s_2$ 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: $d = \bigsqcup S$).

Example 6.5

- 1 Every subset $S \subseteq \mathbb{N}$ is a chain in $(\mathbb{N}, \leq)$.
It has a LUB (its greatest element) iff it is finite.
- 2 $\{\emptyset, \{0\}, \{0, 1\}, \dots\}$ is a chain in $(2^{\mathbb{N}}, \subseteq)$ with LUB $\mathbb{N}$.
- 3 Let $x \in \text{Var}$, and let $f_i : \Sigma \dashrightarrow \Sigma$ for every $i \in \mathbb{N}$ be given by

$$f_i(\sigma) := \begin{cases} \sigma[x \mapsto \sigma(x) + 1] & \text{if } \sigma(x) \leq i \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then $\{f_0, f_1, f_2, \dots\}$ is a chain in $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$, since for every $i \in \mathbb{N}$ and $\sigma, \sigma' \in \Sigma$:

$$\begin{aligned} & f_i(\sigma) = \sigma' \\ \implies & \sigma(x) \leq i, \sigma' = \sigma[x \mapsto \sigma(x) + 1] \\ \implies & \sigma(x) \leq i + 1, \sigma' = \sigma[x \mapsto \sigma(x) + 1] \\ \implies & f_{i+1}(\sigma) = \sigma' \\ \implies & f_i \sqsubseteq f_{i+1} \end{aligned}$$

Definition 6.6 (Chain completeness)

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

Example 6.7

- 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).

Corollary 6.8

Every CCPO has a least element $\sqcup \emptyset$.

Proof.

Let $(D, \sqsubseteq)$ be a CCPO.

- By definition, $\emptyset$ is a chain in D .
- By definition, every $d \in D$ is an upper bound of $\emptyset$.
- Thus $\sqcup \emptyset$ exists and is the least element of D .



Lemma 6.9

- $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ is a CCPO with least element $f_\emptyset$ where $\text{graph}(f_\emptyset) = \emptyset$.
- In particular, for every chain $S \subseteq \Sigma \dashrightarrow \Sigma$,
$$\text{graph}(\bigsqcup S) = \bigcup_{f \in S} \text{graph}(f).$$

Proof.

on the board



Example 6.10 (cf. Example 6.5(3))

Let $x \in \text{Var}$, and let $f_i : \Sigma \dashrightarrow \Sigma$ for every $i \in \mathbb{N}$ be given by

$$f_i(\sigma) := \begin{cases} \sigma[x \mapsto \sigma(x) + 1] & \text{if } \sigma(x) \leq i \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then $S := \{f_0, f_1, f_2, \dots\}$ is a chain (Example 6.5(3)) with $\bigsqcup S = f$ where

$$f : \Sigma \rightarrow \Sigma : \sigma \mapsto \sigma[x \mapsto \sigma(x) + 1]$$