

Semantics and Verification of Software

Lecture 8: Denotational Semantics of WHILE IV (Equivalence with Operational Semantics)

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 Another Example
- 3 Summary: Denotational Semantics
- 4 Equivalence of Operational and Denotational Semantics

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

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**

Monotonicity

Definition (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”

Lemma

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



Continuity

A function F is continuous if the order of applying F and taking LUBs can be reversed:

Definition (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

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

Proof.

omitted □

The Fixpoint Theorem

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

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

Corollary

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

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

Proof.

Using

- Lemma 7.4
 - $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ CCPO with least element $f_\emptyset$
 - LUB = union of graphs
- Lemma 7.6 (Φ continuous)
- Theorem 7.7 (Fixpoint Theorem)



- 1 Repetition: Denotational Semantics of WHILE
- 2 Another Example
- 3 Summary: Denotational Semantics
- 4 Equivalence of Operational and Denotational Semantics

Example 8.1

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ – see Ex. 6.7)
- **Function:** $2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}} : N \mapsto N \cup A$ for some fixed $A \subseteq \mathbb{N}$
 - F monotonic: $M \subseteq N \implies F(M) = M \cup A \subseteq N \cup A = F(N)$
 - F continuous: $F(\bigsqcup S) = F(\bigcup_{N \in S} N) = \bigcup_{N \in S} N \cup A = \bigcup_{N \in S} (N \cup A) = \bigcup_{N \in S} F(N) = \bigsqcup F(S)$
- **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 $n \geq 1$ $\implies \text{fix}(F) = A$
- Alternatively: $F(N) := N \cap A$ $\implies \text{fix}(F) = \emptyset$

- 1 Repetition: Denotational Semantics of WHILE
- 2 Another Example
- 3 Summary: Denotational Semantics
- 4 Equivalence of Operational and 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**

- 1 Repetition: Denotational Semantics of WHILE
- 2 Another Example
- 3 Summary: Denotational Semantics
- 4 Equivalence of Operational and Denotational Semantics

Remember: in Def. 4.1, $\mathfrak{D}[\cdot] : Cmd \rightarrow (\Sigma \dashrightarrow \Sigma)$ was given by

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

Theorem 8.2 (Coincidence Theorem)

For every $c \in Cmd$,

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

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

Equivalence of Semantics II

The proof of Theorem 8.2 employs the following auxiliary propositions:

Lemma 8.3

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

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

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

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

Proof.

- ① structural induction on a
- ② see Exercise 4.2 (structural induction on b)



Proof (Theorem 8.2).

We have to show that

$$\langle c, \sigma \rangle \rightarrow \sigma' \iff \mathcal{E}[\![c]\!](\sigma) = \sigma'$$

- $\Rightarrow$ by structural induction over the derivation tree of $\langle c, \sigma \rangle \rightarrow \sigma'$
- $\Leftarrow$ by structural induction over c (with a nested complete induction over fixpoint index n)

(on the board)



Overview: Operational/Denotational Semantics

Definition (3.2; Execution relation for statements)

$$\begin{array}{lcl} \text{(skip)} \frac{}{\langle \text{skip}, \sigma \rangle \rightarrow \sigma} & \text{(asgn)} \frac{\langle a, \sigma \rangle \rightarrow z}{\langle x := a, \sigma \rangle \rightarrow \sigma[x \mapsto z]} \\ \text{(seq)} \frac{\langle c_1, \sigma \rangle \rightarrow \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow \sigma''}{\langle c_1; c_2, \sigma \rangle \rightarrow \sigma''} & \text{(if-t)} \frac{\langle b, \sigma \rangle \rightarrow \text{true} \quad \langle c_1, \sigma \rangle \rightarrow \sigma'}{\langle \text{if } b \text{ then } c_1 \text{ else } c_2, \sigma \rangle \rightarrow \sigma'} \\ \text{(if-f)} \frac{\langle b, \sigma \rangle \rightarrow \text{false} \quad \langle c_2, \sigma \rangle \rightarrow \sigma'}{\langle \text{if } b \text{ then } c_1 \text{ else } c_2, \sigma \rangle \rightarrow \sigma'} & \text{(wh-f)} \frac{\langle b, \sigma \rangle \rightarrow \text{false}}{\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma} \\ \text{(wh-t)} \frac{\langle b, \sigma \rangle \rightarrow \text{true} \quad \langle c, \sigma \rangle \rightarrow \sigma' \quad \langle \text{while } b \text{ do } c, \sigma' \rangle \rightarrow \sigma''}{\langle \text{while } b \text{ do } c, \sigma \rangle \rightarrow \sigma''} \end{array}$$

Definition (5.3; Denotational semantics of statements)

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