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)

RWTH AACHEN UNIVERSITY

noll@cs.rwth-aachen.de

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

Winter Semester 2011/12

Outline

Semantics of Statements

Definition (Denotational semantics of statements)

The (denotational) semantic functional for statements,

$$\mathfrak{C}[\![.]\!] : Cmd \to (\Sigma \dashrightarrow \Sigma),$$

is given by:

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

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

Characterization of $\mathrm{fix}(\Phi)$ II

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: continuity

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 $d_1, d_2 \in D$,

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

Interpretation: monotonic functions "preserve information"

Lemma

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

Theorem (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*

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

Proof.

on the board □

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

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

Corollary

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

□

Outline

Another Example

Example 8.1

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ – see Ex. 6.7)

Another Example

Example 8.1

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ – see Ex. 6.7)
- **Function:** $2^{\mathbb{N}} \to 2^{\mathbb{N}} : N \mapsto N \cup A$ 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: $F(\bigsqcup S) = F\left(\bigcup_{N \in S} N\right) = \bigcup_{N \in S} N \cup A = \bigcup_{N \in S}(N \cup A) = \bigcup_{N \in S} F(N) = \bigsqcup F(S)$

Another Example

Example 8.1

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ – see Ex. 6.7)
- **Function:** $2^{\mathbb{N}} \to 2^{\mathbb{N}} : N \mapsto N \cup A$ 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: $F(\bigsqcup S) = F\left(\bigcup_{N \in S} N\right) = \bigcup_{N \in S} N \cup A = \bigcup_{N \in S}(N \cup A) = \bigcup_{N \in S} F(N) = \bigsqcup F(S)$
- **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 $n \geq 1$

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

Another Example

Example 8.1

- **Domain:** $(2^{\mathbb{N}}, \subseteq)$ (CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ – see Ex. 6.7)
- **Function:** $2^{\mathbb{N}} \to 2^{\mathbb{N}} : N \mapsto N \cup A$ 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: $F(\bigsqcup S) = F\left(\bigcup_{N \in S} N\right) = \bigcup_{N \in S} N \cup A = \bigcup_{N \in S} (N \cup A) = \bigcup_{N \in S} F(N) = \bigsqcup F(S)$
- **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 $n \geq 1$

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

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

Outline

Summary: Denotational Semantics

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

Summary: Denotational Semantics

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

Summary: Denotational Semantics

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

Summary: Denotational Semantics

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

Outline

Equivalence of Semantics I

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

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

Equivalence of Semantics I

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

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

Theorem 8.2 (Coincidence Theorem)

For every $c \in Cmd$,

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

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

Equivalence of 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* $z \in \mathbb{Z}$:

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

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

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

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. see Exercise 4.2 (structural induction on b)

□

Equivalence of Semantics III

Proof (Theorem 8.2).

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) □

Definition (3.2; Execution relation for statements)

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

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

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

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

Definition (5.3; Denotational semantics of statements)

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