Semantics and Verification of Software

Lecture 9: Equivalence of Operational and Denotational 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/svsw10/

Summer Semester 2010

Exam

- In oral form
- By appointment (e-mail):
  - second half of July or
  - beginning of September to mid-October
- Registration:
  - Diplom: (V)ZPA
  - Master: CampusOffice by May 28

Outline

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

Goals:

- Prove existence of $\mathsf{fix}(\Phi)$ for $\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$
- 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 Completeness

Definition (Chain completeness)

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

Example

1. $(2^{\mathbb{N}}, \subseteq)$ is a CCPO with $\sqcup S = \bigcup_{M \in S} M$ 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).

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"

Example

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

Continuity

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

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 $S \subseteq D$,

$$F(\sqcup S) = \sqcup F(S).$$

Lemma

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

$$\mathsf{fix}(F) := \sqcup \{F^n(\sqcup\emptyset) \mid n \in \mathbb{N}\}$$

is the least fixpoint of F *where*

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

Proof.

on the board □

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

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* $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$. *Then*

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

Proof.

Using

- Lemma 7.9
  - $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ CCPO with least element $f_\emptyset$
  - LUB = union of graphs
- Lemma 8.6 (Φ continuous)
- Theorem 8.7 (Fixpoint Theorem)

□

Outline

Equivalence of Semantics I

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

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

Equivalence of Semantics I

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

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

Theorem 9.1 (Coincidence Theorem)

For every $c \in \mathit{Cmd}$,

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

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 9.1 employs the following auxiliary propositions:

Lemma 9.2

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 9.1 employs the following auxiliary propositions:

Lemma 9.2

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 9.1 employs the following auxiliary propositions:

Lemma 9.2

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 4.1 (structural induction on b)

□

Proof (Theorem 9.1).

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) □

Overview: Operational/Denotational Semantics

Definition (3.1; Execution relation for statements)

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

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

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

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

Definition (5.4; Denotational semantics of statements)

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