Semantics and Verification of Software

Lecture 4: Operational Semantics of WHILE III (Properties of Execution Relation)

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

Outline

Execution of Statements

Remember:

$c ::= \texttt{skip} \mid x \texttt{ := } a \mid c_1 \texttt{;} c_2 \mid \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2 \mid \texttt{while } b \texttt{ do } c \in \mathit{Cmd}$

Definition (Execution relation for statements)

For $c \in \mathit{Cmd}$ and $\sigma, \sigma' \in \Sigma$, the execution relation $\langle c, \sigma \rangle \to \sigma'$ is defined by the following rules:

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

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

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

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

Determinism of Execution Relation I

This operational semantics is well defined in the following sense:

Theorem

The execution relation for statements is deterministic, i.e., whenever $c \in Cmd$ *and* $\sigma, \sigma', \sigma'' \in \Sigma$ *such that* $\langle c, \sigma \rangle \to \sigma'$ *and* $\langle c, \sigma \rangle \to \sigma''$, *then* $\sigma' = \sigma''$.

- How to prove this theorem?
- Idea:
  - use induction on the syntactic structure of c
  - employ corresponding result for expressions (Lemma 3.5)

Determinism of Execution Relation II

- But: proof of Theorem 3.4 fails!
- Problematic case:

$$c = \texttt{while}\ b\ \texttt{do}\ c_0 \text{ where } \langle b, \sigma\rangle \to \mathsf{true}$$

- Here $\langle c, \sigma\rangle \to \sigma'$ and $\langle c, \sigma\rangle \to \sigma''$ require $\sigma_1, \sigma_2 \in \Sigma$ such that

$$\text{(wh-t)}\frac{\langle b, \sigma\rangle \to \mathsf{true}\ \ \langle c_0, \sigma\rangle \to \sigma_1\ \ \langle c, \sigma_1\rangle \to \sigma'}{\langle c, \sigma\rangle \to \sigma'}$$

and

$$\text{(wh-t)}\frac{\langle b, \sigma\rangle \to \mathsf{true}\ \ \langle c_0, \sigma\rangle \to \sigma_2\ \ \langle c, \sigma_2\rangle \to \sigma''}{\langle c, \sigma\rangle \to \sigma''}$$

- c_0 proper substatement of c
  $\Longrightarrow$ induction hypothesis yields $\sigma_1 = \sigma_2$
- c not proper substatement of c $\Longrightarrow$ conclusion $\sigma' = \sigma''$ invalid!

Determinism of Execution Relation III

Induction on derivation trees of execution relation

Induction base: $P\left(\dfrac{}{\langle \texttt{skip}, \sigma\rangle \to \sigma}\right)$ holds for every $\sigma \in \Sigma$, and $P(s)$ holds for every derivation tree s for an arithmetic or Boolean expression.

Induction hypothesis: $P(s_1)$, $P(s_2)$ und $P(s_3)$ holds.

Induction step: it also holds that

(asgn): $P\left(\dfrac{s_1}{\langle x \texttt{:=} a, \sigma\rangle \to \sigma[x \mapsto z]}\right)$

(seq): $P\left(\dfrac{s_1 \quad s_2}{\langle c_1 \texttt{;} c_2, \sigma\rangle \to \sigma''}\right)$

(if-t): $P\left(\dfrac{s_1 \quad s_2}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \to \sigma'}\right)$

(if-f): analogously

(wh-t): $P\left(\dfrac{s_1 \quad s_2 \quad s_3}{\langle \texttt{while } b \texttt{ do } c, \sigma\rangle \to \sigma''}\right)$

(wh-f): $P\left(\dfrac{s_1}{\langle \texttt{while } b \texttt{ do } c, \sigma\rangle \to \sigma}\right)$

Determinism of Execution Relation IV

Proof (Theorem 3.4).

To show:

$$\langle c, \sigma \rangle \to \sigma', \langle c, \sigma \rangle \to \sigma'' \implies \sigma' = \sigma''$$

(by structural induction on derivation trees; on the board) □

Outline

Functional of the Operational Semantics

The determinism of the execution relation (Theorem 3.4) justifies the following definition:

Definition 4.1 (Operational functional)

The functional of the operational semantics,

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

assigns to every statement $c \in Cmd$ a partial state transformation $\mathfrak{O}[\![c]\!] : \Sigma \dashrightarrow \Sigma$, which is defined as follows:

$$\mathfrak{O}[\![c]\!]\sigma := \begin{cases} \sigma' & \text{if } \langle c, \sigma\rangle \to \sigma' \text{ for some } \sigma' \in \Sigma \\ \text{undefined} & \text{otherwise} \end{cases}$$

Remark: $\mathfrak{O}[\![c]\!]\sigma$ can indeed be undefined
(consider e.g. $c =$ `while true do skip`; see Corollary 3.3)

Equivalence of Statements

Definition 4.2 (Operational equivalence)

Two statements $c_1, c_2 \in \mathit{Cmd}$ are called (operationally) equivalent (notation: $c_1 \sim c_2$) if

$$\mathfrak{O}[\![c_1]\!] = \mathfrak{O}[\![c_2]\!].$$

Thus:

- $c_1 \sim c_2$ iff $\mathfrak{O}[\![c_1]\!]\sigma = \mathfrak{O}[\![c_2]\!]\sigma$ for every $\sigma \in \Sigma$
- In particular, $\mathfrak{O}[\![c_1]\!]\sigma$ is undefined iff $\mathfrak{O}[\![c_2]\!]\sigma$ is undefined