Semantics of Programming Languages

Operational Semantics

Prof. Peter Müller

Software Component Technology

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

2. Operational Semantics

2.1 Big-Step Semantics

2.2 Small-Step Semantics

2.3 Equivalence

2.4 Applications of Operational Semantics

Semantic Functions

- The meaning of statements can be expressed as a **partial function** from State to State:

$$\mathcal{S}_{NS} : \text{Stm} \to (\text{State} \hookrightarrow \text{State})$$

$$\mathcal{S}_{NS}[\![s]\!]\sigma = \begin{cases} \sigma' & \text{if } \langle s, \sigma \rangle \to \sigma' \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\mathcal{S}_{SOS} : \text{Stm} \to (\text{State} \hookrightarrow \text{State})$$

$$\mathcal{S}_{SOS}[\![s]\!]\sigma = \begin{cases} \sigma' & \text{if } \langle s, \sigma \rangle \to_1^* \sigma' \\ \text{undefined} & \text{otherwise} \end{cases}$$

- The semantic functions are well-defined because the semantics are deterministic

Equivalence Theorem

Theorem: For every statement of IMP we have $\mathcal{S}_{NS}[\![s]\!] = \mathcal{S}_{SOS}[\![s]\!]$

- If the execution of s from some state terminates in one of the semantics then it also terminates in the other and the resulting states will be equal
- If the execution of s from some state loops in one of the semantics then it will also loop in the other

Equivalence Lemma 1

Lemma: For every statement s of IMP and states σ and σ' we have

$$\langle s, \sigma \rangle \rightarrow \sigma' \Rightarrow \langle s, \sigma \rangle \rightarrow_1^* \sigma'$$

- If the execution of s from σ terminates in the natural semantics then it will terminate in the same state in the structural operational semantics
- The proof runs by induction on the shape of the derivation tree for $\langle s, \sigma \rangle \rightarrow \sigma'$

Induction Base

- Case assign-axiom:
  The derivation tree is the axiom instance
  $$\langle x := e, \sigma \rangle \rightarrow \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma].$$
  From the SOS rule we get
  $$\langle x := e, \sigma \rangle \rightarrow_1 \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]$$
- Case skip-axiom: Analogously
- Case while-rule ($\mathcal{B}[\![b]\!]\sigma = ff$): Analogously

Induction Step: Seq. Composition

- Case sequence-rule:
  The root of the derivation tree is $\langle s_1 ; s_2, \sigma \rangle \rightarrow \sigma'$.
  - There are derivation trees for $\langle s_1, \sigma \rangle \rightarrow \sigma_0$ and $\langle s_2, \sigma_0 \rangle \rightarrow \sigma'$ for some state σ_0
  - By the induction hypothesis, we get $\langle s_1, \sigma \rangle \rightarrow_1^* \sigma_0$ and $\langle s_2, \sigma_0 \rangle \rightarrow_1^* \sigma'$
  - By Exercise 16, we get $\langle s_1 ; s_2, \sigma \rangle \rightarrow_1^* \langle s_2, \sigma_0 \rangle$
  - Finally, $\langle s_1 ; s_2, \sigma \rangle \rightarrow_1^* \langle s_2, \sigma_0 \rangle$ and $\langle s_2, \sigma_0 \rangle \rightarrow_1^* \sigma'$ imply $\langle s_1 ; s_2, \sigma \rangle \rightarrow_1^* \sigma'$

Induction Step: `if`

- Case if-rule ($\mathcal{B}[\![b]\!]\sigma = tt$):
  The root of the derivation tree is
  $\langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \sigma\rangle \rightarrow \sigma'$
  - There is a derivation tree for $\langle s_1, \sigma\rangle \rightarrow \sigma'$
  - By $\mathcal{B}[\![b]\!]\sigma = tt$ and the induction hypothesis, we get
    $\langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \sigma\rangle \rightarrow_1 \langle s_1, \sigma\rangle \rightarrow_1^* \sigma'$
- Case if-rule ($\mathcal{B}[\![b]\!]\sigma = ff$): Analogously

Induction Step: `while`

- Case while-rule ($\mathcal{B}[\![b]\!]\sigma = tt$):
  The root of the derivation tree is
  $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma\rangle \rightarrow \sigma'$
  - There are derivation trees for $\langle s, \sigma\rangle \rightarrow \sigma_0$ and
    $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma_0\rangle \rightarrow \sigma'$ for some state σ_0
  - By the induction hypothesis, we get $\langle s, \sigma\rangle \rightarrow_1^* \sigma_0$ and
    $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma_0\rangle \rightarrow_1^* \sigma'$
  - We derive:

$\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma\rangle \rightarrow_1$ [while-rule]

$\langle \texttt{if}\ b\ \texttt{then}\ s\texttt{;while}\ b\ \texttt{do}\ s\ \texttt{end}\ \texttt{end}, \sigma\rangle \rightarrow_1$ [$\mathcal{B}[\![b]\!]\sigma = tt$]

$\langle s\texttt{;while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma\rangle \rightarrow_1^*$ [Exercise 16]

$\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma_0\rangle \rightarrow_1^* \sigma'$

Equivalence Lemma 2

Lemma: For every statement s of IMP, states σ and σ', and natural number k we have that

$$\langle s, \sigma \rangle \rightarrow_1^k \sigma' \Rightarrow \langle s, \sigma \rangle \rightarrow \sigma'$$

- If the execution of s from σ terminates in the structural operational semantics then it will terminate in the same state in the natural semantics
- The proof runs by induction on the length of the derivation sequence for $\langle s, \sigma \rangle \rightarrow_1^k \sigma'$, that is, by induction on k
- Induction base: For $k = 0$, the result holds trivially

Induction Step

- Assume that lemma holds for $k \leq m$
- Prove that lemma holds for $m + 1$:
  $\langle s, \sigma \rangle \rightarrow_1^{m+1} \sigma' \Rightarrow \langle s, \sigma \rangle \rightarrow \sigma'$
- We consider the first step of the derivation sequence
  $\langle s, \sigma \rangle \rightarrow_1 \gamma \rightarrow_1^m \sigma' \Rightarrow \langle s, \sigma \rangle \rightarrow \sigma'$
- We inspect the derivation tree for the first step
  $\langle s, \sigma \rangle \rightarrow_1 \gamma$

Induction Step (cont'd)

- Case assign-axiom
  - We have $\langle x := e, \sigma \rangle \rightarrow_1 \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma]$
  - In this case $\gamma = \sigma[x \mapsto \mathcal{A}[\![e]\!]\sigma] = \sigma'$ is a state and $m = 0$
  - From the NS-axiom we get $\langle s, \sigma \rangle \rightarrow \sigma'$
- Case skip-axiom: Analogously
- Case Sequential composition
  - We have $\langle s_1 ; s_2, \sigma \rangle \rightarrow_1^{m+1} \sigma'$
  - There are a state σ'' and numbers k_1, k_2 such that $\langle s_1, \sigma \rangle \rightarrow_1^{k_1} \sigma''$ and $\langle s_2, \sigma'' \rangle \rightarrow_1^{k_2} \sigma'$ where $k_1 + k_2 = m + 1$
  - By the induction hypothesis, we get $\langle s_1, \sigma \rangle \rightarrow \sigma''$ and $\langle s_2, \sigma'' \rangle \rightarrow \sigma'$
  - By the NS-rule, we get $\langle s_1 ; s_2, \sigma \rangle \rightarrow \sigma'$

Induction Step (cont'd)

- Case if-axiom ($\mathcal{B}[\![b]\!]\sigma = tt$)
  - We have $\langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \sigma\rangle \rightarrow_1 \langle s_1, \sigma\rangle \rightarrow_1^m \sigma'$
  - By the induction hypothesis, we get $\langle s_1, \sigma\rangle \rightarrow \sigma'$
  - By the NS-rule, we get $\langle \texttt{if}\ b\ \texttt{then}\ s_1\ \texttt{else}\ s_2\ \texttt{end}, \sigma\rangle \rightarrow \sigma'$
- Case if-axiom ($\mathcal{B}[\![b]\!]\sigma = ff$): Analogously
- Case while-axiom
  - We have $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma\rangle \rightarrow_1$ $\langle \texttt{if}\ b\ \texttt{then}\ s\texttt{;while}\ b\ \texttt{do}\ s\ \texttt{end}\ \texttt{end}, \sigma\rangle \rightarrow_1^m \sigma'$
  - By the induction hypothesis, we get $\langle \texttt{if}\ b\ \texttt{then}\ s\texttt{;while}\ b\ \texttt{do}\ s\ \texttt{end}\ \texttt{end}, \sigma\rangle \rightarrow \sigma'$
  - By the lemma about unfolding loops, we get $\langle \texttt{while}\ b\ \texttt{do}\ s\ \texttt{end}, \sigma\rangle \rightarrow \sigma'$

Equivalence Theorem: Proof

$$\mathcal{S}_{NS}[\![s]\!]\sigma = \begin{cases} \sigma' & \text{if } \langle s, \sigma\rangle \to \sigma' \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\mathcal{S}_{SOS}[\![s]\!]\sigma = \begin{cases} \sigma' & \text{if } \langle s, \sigma\rangle \to_1^* \sigma' \\ \text{undefined} & \text{otherwise} \end{cases}$$

- We have proved: $\mathcal{S}_{NS}[\![s]\!]\sigma = \sigma' \Leftrightarrow \mathcal{S}_{SOS}[\![s]\!]\sigma = \sigma'$
- This is sufficient to prove $\mathcal{S}_{NS}[\![s]\!] = \mathcal{S}_{SOS}[\![s]\!]$ because one function is defined iff the other is defined

Equivalence: Summary

- The natural semantics and structural operational semantics are equivalent
  - Proof of Lemma 1 runs by induction on the shape of the derivation tree
  - Proof of Lemma 2 runs by induction on the length of the derivation sequence
- For extended languages, different formalization of the equivalence theorem could be necessary
  - Non-deterministic languages
  - Consider only finite derivation sequences that end in terminal configurations