Operational Semantics of While programs

Hanne Riis Nielson

Informatics and Mathematical Modelling
Technical University of Denmark

What is operational semantics?

A method for specifying semantics of languages and systems based on syntactic transformations of programs and simple operations on discrete domains.

The semantics can be expressed at different levels of abstractions:

- structural operational semantics: some internal structure of the (not so) many steps
- natural semantics: lot of internal structure of the single step

Transition system: $(\Gamma, T, \rhd)$

- Γ: a set of configurations

  two forms: $\langle S, s\rangle$: a computation not yet completed

  s: a completed computation

- T: a set of terminal configurations

  one form: s: a completed computation

- $\rhd$: transition relation: $\rhd \subseteq \Gamma \times \Gamma$

  Natural semantics: $\langle S, s\rangle \rightarrow s'$

  Structural operational semantics: $\begin{cases} \langle S, s\rangle \Rightarrow \langle S', s'\rangle \\ \langle S, s\rangle \Rightarrow s' \end{cases}$

The two semantic models

- Natural semantics:
  - $\langle S, s\rangle \to s'$ means that the statement S is executed from the initial state s; it terminates and the final state is s'.
- Structural operational semantics:
  - $\langle S, s\rangle \Rightarrow \langle S', s'\rangle$ means that one step of execution of the statement S from the initial state s will result in a configuration where the remainder of the statement called S' has to be executed from the state s'.
  - $\langle S, s\rangle \Rightarrow s'$ means that one step of execution of the statement S from the initial state s will terminate and the final state is s'.

Natural semantics

$$\langle x := a\,,\, s\rangle \to s[x \mapsto \mathcal{A}[\![a]\!]s]$$

$$\langle \texttt{skip}\,,\, s\rangle \to s$$

$$\frac{\langle S_1\,,\, s\rangle \to s' \quad \langle S_2\,,\, s'\rangle \to s''}{\langle S_1; S_2\,,\, s\rangle \to s''}$$

$$\frac{\langle S_1\,,\, s\rangle \to s'}{\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2\,,\, s\rangle \to s'} \qquad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$\frac{\langle S_2\,,\, s\rangle \to s'}{\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2\,,\, s\rangle \to s'} \qquad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

$$\frac{\langle S\,,\, s\rangle \to s' \quad \langle \texttt{while}\ b\ \texttt{do}\ S\,,\, s'\rangle \to s''}{\langle \texttt{while}\ b\ \texttt{do}\ S\,,\, s\rangle \to s''} \qquad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$\langle \texttt{while}\ b\ \texttt{do}\ S\,,\, s\rangle \to s \qquad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

Structural operational semantics

$$\langle x := a\, ,\, s\rangle \Rightarrow s[x \mapsto \mathcal{A}[\![a]\!]s]$$

$$\langle \texttt{skip}\, ,\, s\rangle \Rightarrow s$$

$$\frac{\langle S_1\, ,\, s\rangle \Rightarrow \langle S_1'\, ,\, s'\rangle}{\langle S_1; S_2\, ,\, s\rangle \Rightarrow \langle S_1'; S_2\, ,\, s'\rangle}$$

$$\frac{\langle S_1\, ,\, s\rangle \Rightarrow s'}{\langle S_1; S_2\, ,\, s\rangle \Rightarrow \langle S_2\, ,\, s'\rangle}$$

$$\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2\, ,\, s\rangle \Rightarrow \langle S_1\, ,\, s\rangle \quad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2\, ,\, s\rangle \Rightarrow \langle S_2\, ,\, s\rangle \quad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

$$\langle \texttt{while}\ b\ \texttt{do}\ S\, ,\, s\rangle \Rightarrow \langle \texttt{if}\ b\ \texttt{then}\ (S;\ \texttt{while}\ b\ \texttt{do}\ S)\ \texttt{else skip}\, ,\, s\rangle$$

Terminating programs

- Natural semantics:

  The execution of the statement S from state s terminates if there exists a state s' such that $\langle S\,,\,s\rangle \to s'$

- Structural operational semantics:

  The execution of the statement S from state s terminates if there exists a finite derivation sequence $\langle S\,,\,s\rangle \Rightarrow \gamma_1 \Rightarrow \cdots \Rightarrow \gamma_n$ that cannot be extended

  OBS: it is not required that γ_n has the form s'; if it has the form $\langle S'\,,\,s'\rangle$ then we say that the computation is stuck

Looping programs

- Natural semantics:

  The execution of the statement S from state s loops if there does not exist a state s' such that $\langle S\,,\, s\rangle \to s'$

- Structural operational semantics:

  The execution of the statement S from state s loops if there exists an infinite derivation sequence $\langle S\,,\, s\rangle \Rightarrow \gamma_1 \Rightarrow \cdots \Rightarrow \gamma_n \Rightarrow \cdots$

  OBS: all the configurations γ_n will have the form $\langle S_n\,,\, s_n\rangle$

Semantic equivalence

- Natural semantics:

  Two statements S_1 and S_2 are semantically equivalent if for all states s and s'

  - $\langle S_1 \,,\, s\rangle \to s'$ if and only if $\langle S_2 \,,\, s\rangle \to s'$

- Structural operational semantics:

  Two statements S_1 and S_2 are semantically equivalent if for all states s

  - for all finite derivation sequences:

    $\langle S_1 \,,\, s\rangle \Rightarrow^* \gamma$ if and only if $\langle S_2 \,,\, s\rangle \Rightarrow^* \gamma$

  - there is an infinite derivation sequence starting in $\langle S_1 \,,\, s\rangle$ if and only if there is one starting in $\langle S_2 \,,\, s\rangle$.

Deterministic semantics

- Natural semantics:

  The semantics is deterministic if for all statements S and for all states s, s' and s'':

  $$\langle S\,,\,s\rangle \to s' \quad \text{and} \quad \langle S\,,\,s\rangle \to s''$$

  implies $s' = s''$

- Structural operational semantics:

  The semantics is deterministic if for all statements S and for all states s and configurations γ' and γ'':

  $$\langle S\,,\,s\rangle \Rightarrow \gamma' \quad \text{and} \quad \langle S\,,\,s\rangle \Rightarrow \gamma''$$

  implies $\gamma' = \gamma''$

Proof principle: structural induction

To prove a property of all the elements of the syntactic category do the following:

- Prove that the property holds for all the basis elements of the syntactic category.
- Prove that the property holds for all the composite elements of the syntactic category: Assume that the property holds for all the immediate constituents of the element — this is called the induction hypothesis — and prove that it also holds for the element itself.

Proof principle:
induction on shape of derivation trees

To prove a property of all the derivation trees of a natural semantics do the following:

- Prove that the property holds for all the simple derivation trees by showing that it holds for the axioms of the transition system.
- Prove that the property holds for all composite derivation trees: For each rule assume that the property holds for its premises — this is called the induction hypothesis — and prove that it also holds for the conclusion of the rule provided that the conditions of the rule are satisfied.

Proof principle:
induction on length of derivation sequences

To prove a property of all the derivation sequences of a structural operational semantics do the following:

- Prove that the property holds for all derivation sequences of length 0.
- Prove that the property holds for all other derivation sequences: Assume that the property holds for all derivation sequences of length at most k — this is called the induction hypothesis — and show that it holds for derivation sequences of length $k + 1$.