Semantics (NAT, SOS), elements of proofs

Laure Gonnord

October 1, 2017

Abstract syntax

Recall the abstract syntax of the course for expressions:

$$
\begin{array}{rcll}
e & ::= & c & \textit{constant} \\
 & | & x & \textit{variable} \\
 & | & e + e & \textit{addition} \\
 & | & e \times e & \textit{multiplication} \\
 & | & \ldots &
\end{array}
$$

and the mini-while language:

$$
\begin{array}{rcll}
S(Smt) & ::= & x := expr & \text{assign} \\
 & | & skip & \text{do nothing} \\
 & | & S_1; S_2 & \text{sequence} \\
 & | & \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2 & \text{test} \\
 & | & \texttt{while } b \texttt{ do } S \texttt{ done} & \text{loop}
\end{array}
$$

1 Semantic rules

Notations are in the course, as well as the inductive definitions for $\mathcal{A}[e]\sigma$ and $\mathcal{B}[b]\sigma$. A *configuration* is a tuple (statement, state).

1.1 Natural "big steps" semantics (NAT)

Name	Rule
skip	$(x := a, \sigma) \to \sigma[x \mapsto \mathcal{A}[a]\sigma]$
assign	$\dfrac{(S_1, \sigma) \to \sigma' \ (S_2, \sigma') \to \sigma''}{\big((S_1; S_2), \sigma\big) \to \sigma''}$
iftrue	$\mathcal{B}[b]\sigma = tt : \dfrac{(S_1, \sigma) \to \sigma'}{(\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2, \sigma) \to \sigma'}$
iffalse	$\mathcal{B}[b]\sigma = ff : \dfrac{(S_2, \sigma) \to \sigma'}{(\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2, \sigma) \to \sigma'}$
whiletrue	$\mathcal{B}[b]\sigma = tt : \dfrac{(S, \sigma) \to \sigma', (\texttt{while } b \texttt{ do } S, \sigma') \to \sigma''}{(\texttt{while } b \texttt{ do } S, \sigma) \to \sigma''}$
whilefalse	$\mathcal{B}[b]\sigma = ff : (\texttt{while } b \texttt{ do } S, \sigma) \to \sigma$

Proposition 1 (Determinism). *For all S, for all $\sigma, \sigma', \sigma''$, if $(S, \sigma) \to \sigma'$ and $(S, \sigma) \to \sigma''$ then $\sigma' = \sigma''$.*

Proof. TODO ! □

1.2 Structural Operational "small steps" Semantics (SOS)

Name	Rule
skip	$(\texttt{skip}, \sigma) \Rightarrow \sigma$
assign	$(x := a, \sigma) \Rightarrow \sigma[x \mapsto \mathcal{A}[a]\sigma]$
seq1	$\dfrac{(S_1, \sigma) \Rightarrow \sigma'}{\big((S_1; S_2), \sigma\big) \Rightarrow (S_2, \sigma')}$
seq2	$\dfrac{(S_1, \sigma) \Rightarrow (S_1', \sigma')}{\big((S_1; S_2), \sigma\big) \Rightarrow (S_1'; S_2, \sigma')}$
iffalse	$\mathcal{B}[b]\sigma = \mathit{ff} : (\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, \sigma) \Rightarrow (S_2, \sigma)$
iftrue	$\mathcal{B}[b]\sigma = \mathit{tt} : (\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, \sigma) \Rightarrow (S_1, \sigma)$
while	$(\texttt{while}\ b\ \texttt{do}\ S, \sigma) \Rightarrow (\texttt{if}\ b\ \texttt{then}\ (S; \texttt{while}\ b\ \texttt{do}\ S)\ \texttt{else skip}), \sigma)$

2 NAT/SOS equivalence

Theorem 1 (Semantic equivalence). *$(S, \sigma) \to \sigma' \iff (S, \sigma) \Rightarrow \sigma'$.*

Proposition 2. *If $(S, \sigma) \to \sigma'$ then $(S, \sigma) \Rightarrow^* \sigma'$.*

Proof. TODO □

Lemma 1 (Extension of seq1 SOS rule). *If $(S_1, \sigma) \Rightarrow^k \sigma'$ then $((S_1; S_2), \sigma) \Rightarrow^k (S_2, \sigma')$*

Proposition 3. *For all S, σ, k, σ', if $(S, \sigma) \Rightarrow^k \sigma'$ then $(S, \sigma) \to \sigma'$.*

Proof. TODO □

Lemma 2. *If $(S_1; S_2, \sigma) \Rightarrow^k \sigma''$ then there exists σ', k_1 such that $(S_1, \sigma) \Rightarrow^{k_1} \sigma'$ and $(S_2, \sigma) \Rightarrow^{k-k_1} \sigma''$*