3. Small-step operational semantics of the simple imperative language.

① Let's try to give the operational semantics to the simple imperative language that we studied. Here is a reminder of its abstract grammar:

$$\langle comm \rangle ::= \mathbf{skip} \mid \langle var \rangle := \langle intexp \rangle \mid \langle comm \rangle ; \langle comm \rangle \mid \mathbf{if}\ \langle boolexp \rangle\ \mathbf{then}\ \langle comm \rangle\ \mathbf{else}\ \langle comm \rangle \mid \mathbf{while}\ \langle boolexp \rangle\ \mathbf{do}\ \langle comm \rangle$$

② What should we do? First, we have to define the set of nonterminal configurations and that of terminal configurations.

Γ_N Γ_T

Here are our definitions:

$$\Gamma_N \stackrel{def}{=} \langle comm \rangle \times \Sigma \qquad \Gamma_T \stackrel{def}{=} \Sigma$$

(the set of states, i.e. $[\langle var \rangle \to \mathbb{Z}]$)

The $\langle comm \rangle$ part is missing because there is no remaining computation.

the current state of a machine

Command that records the remaining computation.

The set of configurations is the union of the above two sets.

③ Second, we should define a binary relation

$$\to\ \subseteq \Gamma_N \times \Gamma,$$

called transition relation, that describes single-step computation. We write $\gamma \to \gamma'$ to mean $(\gamma, \gamma') \in \to$. We define the transition relation $\to$ using the inference-rule notation.

$$\frac{}{\langle \mathbf{skip}, \sigma \rangle \to \sigma}$$

$$\frac{}{\langle v := e, \sigma \rangle \to [\sigma \mid v : [\![e]\!]\sigma]}$$

$$\frac{\langle c_1, \sigma \rangle \to \sigma'}{\langle c_1 ; c_2, \sigma \rangle \to \langle c_2, \sigma' \rangle}$$

$$\frac{\langle c_1, \sigma \rangle \to \langle c_1', \sigma' \rangle}{\langle c_1 ; c_2, \sigma \rangle \to \langle c_1' ; c_2, \sigma' \rangle}$$

$$\frac{}{\langle \mathbf{if}\ b\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2, \sigma \rangle \to \langle c_1, \sigma \rangle} \ ([\![b]\!]\sigma = tt)$$

$$\frac{}{\langle \mathbf{if}\ b\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2, \sigma \rangle \to \langle c_2, \sigma \rangle} \ ([\![b]\!]\sigma = ff)$$

$$\frac{}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma \rangle \to \sigma} \ ([\![b]\!]\sigma = ff)$$

$$\frac{}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma \rangle \to \langle c ; \mathbf{while}\ b\ \mathbf{do}\ c, \sigma \rangle} \ ([\![b]\!]\sigma = tt)$$

Note that the right-hand side of $\to$ may include a command that is not a sub-command of the one on the left-hand side. Look at and . This indicates that the semantics is not compositional. All these rules correspond to our intuitive understanding of one computation step. They can form the basis of the implementation of a simple interpreter, which just needs to run the $\to$ step repeatedly.