TeSyL Operational Semantics

Mathias Rud, `https://github.com/Cenaras/TeSyL`

May 2022

1 Syntax

In the following, let σ be the environment variable, i.e. $\sigma : Id \rightarrow Exp$, where Id is an identifier, and Exp is an expression. We will use the syntax $\sigma' = \sigma[x \rightarrow v]$ to indicate that the environment is copied and extended with the binding from $x \rightarrow v$.
We will use e_i as the syntax for expression i. For now, the evaluation of an expression returns a $Value$ and also the updated environment, σ. We will denote the $UnitVal$ with ().

2 Operational Semantics

Notice: This is a work in progress. The operational semantics might have typo's (in particular with respect to the environments σ) and might not explain the entirety of the implemented language.

$$\text{Int:}\frac{i \in IntExp(v),\ v \in \mathbb{Z}}{\sigma \vdash i \Rightarrow v, \sigma} \qquad \text{Bool:}\frac{b \in BoolExp(v),\ v \in \{\top, \bot\}}{\sigma \vdash b \Rightarrow v, \sigma}$$

$$\text{BinOp(1):}\frac{\begin{array}{c}\sigma \vdash e_1 \Rightarrow v_1, \sigma \vdash e_2 \Rightarrow v_2, \\ op \in \{+, -, *\}\ v = v_1\ op\ v_2\end{array}}{\sigma \vdash e_1\ +\ e_2 \Rightarrow v, \sigma} \qquad \text{BinOp(2):}\frac{\begin{array}{c}\sigma \vdash e_1 \Rightarrow v_1,\ \sigma \vdash e_2 \Rightarrow v_2, \\ v_2 \neq 0,\ v = v_1/v_2\end{array}}{\sigma \vdash e_1/e_2 \Rightarrow v, \sigma}$$

$$\text{Let:}\frac{\sigma \vdash e \Rightarrow v,\ \sigma' = \sigma[x \rightarrow v]}{\sigma \vdash let\ x = e \Rightarrow (), \sigma'} \qquad \text{Assignment:}\frac{\sigma \vdash e \Rightarrow v,\ v \neq \emptyset, \sigma' = \sigma[x \rightarrow v]}{\sigma \vdash x = e \Rightarrow (), \sigma'}$$

$$\text{Var:}\frac{\sigma(x) \Rightarrow v}{\sigma \vdash x \Rightarrow v, \sigma} \qquad \text{Unit}\frac{\sigma \vdash e = ()}{\sigma \vdash e \Rightarrow (), \sigma}$$

$$\text{IfTrue:}\frac{\sigma \vdash e \Rightarrow g,\ g = false, \sigma \vdash e_2 \Rightarrow v}{\sigma \vdash if\ e\ then\ e_1\ else\ e_2 \Rightarrow v, \sigma} \qquad \text{IfFalse}\frac{\sigma \vdash e \Rightarrow g,\ g = true, \sigma \vdash e_1 \Rightarrow v}{\sigma \vdash if\ e\ then\ e_1\ else\ e_2 \Rightarrow v, \sigma}$$

$$\text{Sequence:}\frac{\begin{array}{c}\sigma_{i-1} \vdash e_i \Rightarrow v_i, \sigma_i, for\ i \in \{1, ..., n-1\} \\ \sigma_{n-1} \vdash e_n \Rightarrow v, \sigma'\end{array}}{\sigma_0 \vdash \{e_1; \ldots e_n\} \Rightarrow v, \sigma'}$$

$$\text{WhileFalse:}\frac{\sigma \vdash e_1 \Rightarrow false, \sigma'}{\sigma \vdash while\ e_1\ e_2 \Rightarrow (), \sigma'} \qquad \text{WhileTrue:}\frac{\begin{array}{c}\sigma \vdash e_1 \Rightarrow true, \sigma',\ \sigma' \vdash e_2 \Rightarrow v', \sigma'', \\ \sigma'' \vdash while\ e_1\ e_2 \Rightarrow v, \sigma'''\end{array}}{\sigma \vdash while\ e_1\ e_2 \Rightarrow v, \sigma'''}$$

To explain the while-rule a bit. Evaluating the guard to false immediately returns the unit value, with the environment potentially updated. If the guard evaluates to true, we evaluate the body in the possibly updated environment. This produces a new environment, in which we evaluate the entire while expression again.

$$\text{EqTrue:}\frac{\sigma \vdash e_1 \Rightarrow v_1,\ \sigma \vdash e_2 \Rightarrow v_2,\ v_1 = v_2}{\sigma \vdash e_1 = e_2 \Rightarrow true, \sigma} \qquad \text{EqFalse:}\frac{\sigma \vdash e_1 \Rightarrow v_1,\ \sigma \vdash e_2 \Rightarrow v_2,\ v_1 \neq v_2}{\sigma \vdash e_1 = e_2 \Rightarrow false, \sigma}$$

$$\text{EqTrue:}\frac{\begin{array}{c}\sigma \vdash e_1 \Rightarrow v_1,\ \sigma \vdash e_2 \Rightarrow v_2, \\ v_1 \in \mathbb{Z},\ v_2 \in \mathbb{Z}, v_1 \leq v_2\end{array}}{\sigma \vdash e_1 <= e_2 \Rightarrow true, \sigma} \qquad \text{EqFalse:}\frac{\begin{array}{c}\sigma \vdash e_1 \Rightarrow v_1,\ \sigma \vdash e_2 \Rightarrow v_2, \\ v_1 \in \mathbb{Z},\ v_2 \in \mathbb{Z}, v_1 \nleq v_2\end{array}}{\sigma \vdash e_1 <= e_2 \Rightarrow false, \sigma}$$

The rules for the other relational operations follow the same structure as the above.