1 Lecture

1.1 Syntax

$$
\begin{aligned}
Expr ::= {} & Num \mid \\
& Bool \mid \\
& \triangle Expr \mid \\
& Expr \odot Expr \mid \\
& Expr \leq Expr \mid \\
& Expr \texttt{ nand } Expr \mid \\
& \texttt{if } Expr \texttt{ then } Expr \texttt{ else } Expr \mid \\
& VarName \\
Type ::= {} & \texttt{Number} \mid \texttt{Boolean} \\
Cmd ::= {} & Type\ VarName \mid \\
& VarName \texttt{ = } Expr \\
Program ::= {} & Cmd\texttt{;}\ Program \mid \\
& Cmd,
\end{aligned}
\tag{1}
$$

where Num is a predefined set of integer numbers (a.k.a. Z), $Bool$ is a predefined set of boolean values and $VarName$ is a predefined set of variable names.

1.2 Operational Semantics

Convention: $e, e' \in Expr$, $n, n' \in Num$, $b \in Bool$, $v \in VarName$, $t \in Type$, $c \in Cmd$, $p \in Program$ and $s, s', s'' \in (VarName \to (Num \cup Bool))$.

$$\frac{}{(s, n) \Rightarrow n} \tag{2}$$

$$\frac{}{(s, b) \Rightarrow b} \tag{3}$$

$$\frac{(s, e) \Rightarrow n}{(s, \triangle e) \Rightarrow -n} \tag{4}$$

$$\frac{(s, e) \Rightarrow n \quad (s, e') \Rightarrow n'}{(s, e \odot e') \Rightarrow n + n'} \tag{5}$$

$$\vdots$$

$$\frac{}{(s, v) \Rightarrow s(v)} \tag{6}$$

$$\frac{}{(s, t\ v) \Longrightarrow s} \tag{7}$$

$$\frac{(s, e) \Rightarrow n}{(s, v \texttt{ = } e) \Longrightarrow s[v \mapsto n]} \tag{8}$$

$$\frac{(s, e) \Rightarrow b}{(s, v \texttt{ = } e) \Longrightarrow s[v \mapsto b]} \tag{9}$$

$$\frac{(s, c) \Longrightarrow s' \quad (s', p) \Longrightarrow s''}{(s, c;\ p) \Longrightarrow s''} \tag{10}$$

1.3 Typing

Convention: $e, e', e'', \ldots \in Expr$, $b \in Bool$, $n \in Num$, $v \in VarName$, $p \in Program$ and $t \in \{Number, Boolean\}$.

$$\frac{}{\Gamma \vdash n : Number} \tag{11}$$

$$\frac{}{\Gamma \vdash b : Boolean} \tag{12}$$

$$\frac{\Gamma \vdash e : Number}{\Gamma \vdash \triangle e : Number} \tag{13}$$

$$\frac{\Gamma \vdash e : Number \quad \Gamma \vdash e' : Number}{\Gamma \vdash e \odot e' : Number} \tag{14}$$

$$\frac{\Gamma \vdash e : Number \quad \Gamma \vdash e' : Number}{e \leq e' : Boolean} \tag{15}$$

$$\frac{\Gamma \vdash e : Boolean \quad \Gamma \vdash e' : Boolean}{e \texttt{ nand } e' : Boolean} \tag{16}$$

$$\frac{\Gamma \vdash e : Boolean \quad \Gamma \vdash e' : t \quad \Gamma \vdash e'' : t}{\Gamma \vdash \texttt{if } e \texttt{ then } e' \texttt{ else } e'' : t} \tag{17}$$

$$\frac{}{\Gamma \vdash v : \Gamma(v)} \tag{18}$$

$$\frac{\Gamma \vdash v : t \quad \Gamma \vdash e : t}{\Gamma \vdash v \texttt{ = } e : \diamond} \tag{19}$$

$$\frac{v \notin dom(\Gamma)}{\Gamma \vdash t\ v : \diamond} \tag{20}$$

$$\frac{\Gamma \vdash v \texttt{ = } e : \diamond \quad \Gamma \vdash p : \diamond}{\Gamma \vdash v \texttt{ = } e\texttt{; } p : \diamond} \tag{21}$$

$$\frac{\Gamma \vdash t\ v : \diamond \quad \Gamma \cup \{(v, t)\} \vdash p : \diamond}{\Gamma \vdash t\ v\texttt{; } p : \diamond} \tag{22}$$

2 Seminar

1. Write down the missing semantic rules of our imperative language.
2. Add loops, extend both syntax and semantics.
3. Turn ternary expression into if statement.
4. Extend the language with blocks and variable scoping.