1 Lecture

1.1 Syntax

$$\begin{aligned} \mathit{Expr} ::= \; & \mathit{Num} \mid \\ & \mathit{Bool} \mid \\ & \triangle \mathit{Expr} \mid \\ & \mathit{Expr} \odot \mathit{Expr} \mid \\ & \mathit{Expr} \leq \mathit{Expr} \mid \\ & \mathit{Expr} \; \mathtt{nand} \; \mathit{Expr} \mid \\ & \mathtt{if} \; \mathit{Expr} \; \mathtt{then} \; \mathit{Expr} \; \mathtt{else} \; \mathit{Expr}, \end{aligned} \tag{1}$$

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

1.2 Typing

Convention: $e, e', e'', \ldots \in Expr$, $b, b' \in Bool$ and $n, n' \in Num$.

$$\frac{}{n : \mathit{Number}} \tag{2}$$

$$\frac{}{b : \mathit{Boolean}} \tag{3}$$

$$\frac{e : \mathit{Number}}{\triangle e : \mathit{Number}} \tag{4}$$

$$\frac{e : \mathit{Number} \quad e' : \mathit{Number}}{e \odot e' : \mathit{Number}} \tag{5}$$

$$\frac{e : \mathit{Number} \quad e' : \mathit{Number}}{e \leq e' : \mathit{Boolean}} \tag{6}$$

$$\frac{e : \mathit{Boolean} \quad e' : \mathit{Boolean}}{e \; \mathtt{nand} \; e' : \mathit{Boolean}} \tag{7}$$

$$\frac{e : \mathit{Boolean} \quad e' : \mathit{Number} \quad e'' : \mathit{Number}}{\mathtt{if} \; e \; \mathtt{then} \; e' \; \mathtt{else} \; e'' : \mathit{Number}} \tag{8}$$

$$\frac{e : \mathit{Boolean} \quad e' : \mathit{Boolean} \quad e'' : \mathit{Boolean}}{\mathtt{if} \; e \; \mathtt{then} \; e' \; \mathtt{else} \; e'' : \mathit{Boolean}} \tag{9}$$

2 Seminar

1. Write down the big-step operational semantics of the language described above and check its compliance of its rules with respective rules of the typing system.

2. Extend the language to include strings. Try to make type rules as compact as possible.

3. Extend the language with implicit coercions from numbers to booleans.

4. Add division operator to the language. Define its type system so that it avoids division-by-zero error. Hint: you will probably have to define a type representing nonzero number.