TFAE – FAE with Type System

1 INTRODUCTION

TFAE is a toy language for the COSE212 course at Korea University. TFAE stands for an extension of the FAE language with a **type system**, and it supports the following features:

- **number (integer) values**
- **basic arithmetic operators**: addition (+) and multiplication (*)
- **first-class functions**
- **immutable variables** (val)
- **static type checking**

   This document is the specification of TFAE. First, Section 2 describes the concrete syntax, and Section 3 describes the abstract syntax. Then, Section 4 describes the type system. Finally, Section 5 describes the big-step operational (natural) semantics of TFAE.

2 CONCRETE SYNTAX

The concrete syntax of TFAE is written in a variant of the extended Backus–Naur form (EBNF). The notation `<nt>` denotes a nonterminal, and `"t"` denotes a terminal. We use ? to denote an optional element and + (or *) to denote one or more (or zero or more) repetitions of the preceding element. We use butnot to denote a set difference to exclude some strings from a producible set of strings. We omit some obvious terminals using the ellipsis (. . .) notation.

```
// basic elements
<digit>    ::= "0" | "1" | "2" | ... | "9"
<number>   ::= "-"? <digit>+
<alphabet> ::= "A" | "B" | "C" | ... | "Z" | "a" | "b" | "c" | ... | "z"
<idstart>  ::= <alphabet> | "_"
<idcont>   ::= <alphabet> | "_" | <digit>
<keyword>  ::= "val"
<id>       ::= <idstart> <idcont>* butnot <keyword>

// expressions
<expr>     ::= <number> | <expr> "+" <expr> | <expr> "*" <expr>
             | "(" <expr> ")" | "{" <expr> "}"
             | "val" <id> "=" <expr> ";" <expr> | <id>
             | "(" <id> ":" <type> ")" "=>" <expr> | <expr> "(" <expr> ")"

// types
<type>     ::= "Num" | <type> "=>" <type>
```

For types, the arrow (=>) operator is right-associative. For expressions, the precedence and associativity of operators are defined as follows:

Operator	Associativity	Precedence
*	left	1
+	left	2

3 ABSTRACT SYNTAX

The abstract syntax of TFAE is defined as follows:

Expressions $\mathbb{E} \ni e ::= n$ (Num)   Numbers $n \in \mathbb{Z}$ (BigInt)

$\qquad\qquad\quad | \; e + e$ (Add)

$\qquad\qquad\quad | \; e \times e$ (Mul)   Identifiers $x \in \mathbb{X}$ (String)

$\qquad\qquad\quad | \; \mathsf{val}\ x{=}e;\ e$ (Val)

$\qquad\qquad\quad | \; x$ (Id)   Types $\mathbb{T} \ni \tau ::= \mathsf{Number}$ (NumT)

$\qquad\qquad\quad | \; \lambda x{:}\tau.e$ (Fun) $| \; \tau \to \tau$ (ArrowT)

$\qquad\qquad\quad | \; e(e)$ (App)

4 TYPE SYSTEM

This section explains type system of TFAE, and we use the following notations:

$$\text{Type Environments} \quad \Gamma \in \mathbb{X} \xrightarrow{\text{fin}} \mathbb{T} \quad (\mathsf{TypeEnv})$$

In the type system, type checking is defined with the following typing rules:

$$\boxed{\Gamma \vdash e : \tau}$$

$$\tau{-}\mathsf{Num} \; \frac{}{\Gamma \vdash n : \mathsf{Number}}$$

$$\tau{-}\mathsf{Add} \; \frac{\Gamma \vdash e_1 : \mathsf{Number} \qquad \Gamma \vdash e_2 : \mathsf{Number}}{\Gamma \vdash e_1 + e_2 : \mathsf{Number}} \qquad \tau{-}\mathsf{Mul} \; \frac{\Gamma \vdash e_1 : \mathsf{Number} \qquad \Gamma \vdash e_2 : \mathsf{Number}}{\Gamma \vdash e_1 \times e_2 : \mathsf{Number}}$$

$$\tau{-}\mathsf{Val} \; \frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma[x \mapsto \tau_1] \vdash e_2 : \tau_2}{\Gamma \vdash \mathsf{val}\ x{=}e_1;\ e_2 : \tau_2} \qquad \tau{-}\mathsf{Id} \; \frac{x \in \mathrm{Domain}(\Gamma)}{\Gamma \vdash x : \Gamma(x)}$$

$$\tau{-}\mathsf{Fun} \; \frac{\Gamma[x \mapsto \tau] \vdash e : \tau'}{\Gamma \vdash \lambda x{:}\tau.e : \tau \to \tau'} \qquad \tau{-}\mathsf{App} \; \frac{\Gamma \vdash e_0 : \tau_1 \to \tau_2 \qquad \Gamma \vdash e_1 : \tau_1}{\Gamma \vdash e_0(e_1) : \tau_2}$$

5 SEMANTICS

We use the following notations in the semantics:

Values $\mathbb{V} \ni v ::= n$ (NumV)   Environments $\sigma \in \mathbb{X} \xrightarrow{\text{fin}} \mathbb{V}$ (Env)

$\qquad\qquad | \; \langle \lambda x.e, \sigma \rangle$ (CloV)

The big-step operational (natural) semantics of TFAE is defined as follows:

$$\boxed{\sigma \vdash e \Rightarrow v}$$

$$\mathsf{Num} \; \frac{}{\sigma \vdash n \Rightarrow n} \qquad \mathsf{Add} \; \frac{\sigma \vdash e_1 \Rightarrow n_1 \qquad \sigma \vdash e_2 \Rightarrow n_2}{\sigma \vdash e_1 + e_2 \Rightarrow n_1 + n_2} \qquad \mathsf{Mul} \; \frac{\sigma \vdash e_1 \Rightarrow n_1 \qquad \sigma \vdash e_2 \Rightarrow n_2}{\sigma \vdash e_1 \times e_2 \Rightarrow n_1 \times n_2}$$

$$\mathsf{Val} \; \frac{\sigma \vdash e_1 \Rightarrow v_1 \qquad \sigma[x \mapsto v_1] \vdash e_2 \Rightarrow v_2}{\sigma \vdash \mathsf{val}\ x{=}e_1;\ e_2 \Rightarrow v_2} \qquad \mathsf{Fun} \; \frac{}{\sigma \vdash \lambda x{:}\tau.e \Rightarrow \langle \lambda x.e, \sigma \rangle}$$

$$\mathsf{Id} \; \frac{x \in \mathrm{Domain}(\sigma)}{\sigma \vdash x \Rightarrow \sigma(x)} \qquad \mathsf{App} \; \frac{\sigma \vdash e_0 \Rightarrow \langle \lambda x.e_2, \sigma' \rangle \qquad \sigma \vdash e_1 \Rightarrow v_1 \qquad \sigma'[x \mapsto v_1] \vdash e_2 \Rightarrow v_2}{\sigma \vdash e_0(e_1) \Rightarrow v_2}$$