AE – Arithmetic Expressions

1 INTRODUCTION

AE is a toy language for the COSE212 course at Korea University. AE stands for a language of **a**rithmetic **e**xpressions that supports the following features:

- **integers**
- **basic arithmetic operators**: addition (`+`) and multiplication (`*`)

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

2 CONCRETE SYNTAX

The concrete syntax of AE 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, `+` to denote one or more repetitions of the preceding element, and `*` to denote zero or more repetitions of the preceding element.

```
<digit>  ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
<number> ::= "-"? <digit>+
<expr>   ::= <number> | <expr> "+" <expr> | <expr> "*" <expr> | "(" <expr> ")"
```

The precedence and associativity of operators are defined as follows:

Operator	Associativity	Precedence
`*`	left	1
`+`	left	2

3 ABSTRACT SYNTAX

The abstract syntax of AE is defined as follows:

$$\begin{array}{llll} \text{Expressions} & e & ::= n & (\texttt{Num}) \\ & & \mid e + e & (\texttt{Add}) \\ & & \mid e * e & (\texttt{Mul}) \end{array}$$

where

$$\text{Integers} \quad n \in \mathbb{Z} \quad (\texttt{BigInt})$$

4 SEMANTICS

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

$$\boxed{\vdash e \Rightarrow n}$$

$$\texttt{Num}\ \frac{}{\vdash n \Rightarrow n} \qquad \texttt{Add}\ \frac{\vdash e_1 \Rightarrow n_1 \qquad \vdash e_2 \Rightarrow n_2}{\vdash e_1 + e_2 \Rightarrow n_1 + n_2} \qquad \texttt{Mul}\ \frac{\vdash e_1 \Rightarrow n_1 \qquad \vdash e_2 \Rightarrow n_2}{\vdash e_1 * e_2 \Rightarrow n_1 \times n_2}$$