# 1 Lecture

## 1.1 Syntax

$$\begin{aligned} Expr ::= {} & Num \mid \\ & \triangle Expr \mid \\ & Expr \odot Expr, \end{aligned} \tag{1}$$

where $Num$ is a predefined set of integer numbers (a.k.a. $Z$).

$$\begin{aligned} Expr = \{ & \dots, 0, 1, 2, \dots, \triangle 0, \triangle 1, \triangle 2, \dots, 0 \odot 0, 0 \odot 1, 0 \odot 2, \dots, (\triangle 0) \odot 0, \\ & (\triangle 0) \odot 1, (\triangle 0) \odot 2, 0 \odot (\triangle 0), 0 \odot (\triangle 1), 0 \odot (\triangle 2), \triangle(0 \odot 0), \triangle(0 \odot 1), \\ & \triangle(0 \odot 2), \dots \} \end{aligned} \tag{2}$$

## 1.2 Small-Step Operational Semantics

Convention: $e, e', e_1, e_2, \dots \in Expr$ and $n, n', n_1, n_2, \dots \in Num$.

$$\frac{}{\triangle n \Rightarrow -n} \tag{3}$$

$$\frac{}{n \odot n' \Rightarrow n + n'} \tag{4}$$

$$\frac{e \Rightarrow e'}{\triangle e \Rightarrow \triangle e'} \tag{5}$$

$$\frac{e_1 \Rightarrow e'}{e_1 \odot e_2 \Rightarrow e' \odot e_2} \tag{6}$$

$$\frac{e_2 \Rightarrow e'}{e_1 \odot e_2 \Rightarrow e_1 \odot e'} \tag{7}$$

## 1.3 Big-Step Operational Semantics

$$\frac{}{n \Longrightarrow n} \tag{8}$$

$$\frac{e \Longrightarrow n}{\triangle e \Longrightarrow -n} \tag{9}$$

$$\frac{e \Longrightarrow n \quad e' \Longrightarrow n'}{e \odot e' \Longrightarrow n + n'} \tag{10}$$

Please note that $\Rightarrow$ is relation on expressions (i.e. $\Rightarrow \in Expr \times Expr$), whereas $\Longrightarrow$ is relation between expressions and numbers (i.e. $\Longrightarrow \in Expr \times Num$).

## 2 Seminar

In the following semantics, consider both small-step and big-step semantics described above.

1. Write down the sequences of all possible rewritings of the following term: $(\triangle 1) \odot (\triangle 2)$.
2. Is there any term that could be rewritten indefinitely?
3. Change the rules of the small-step operational semantics so that right-hand sides of the $\odot$ operator always get evaluated first. (Hint: evaluate left-hand side only if the right-hand side consists of a number.)
4. Define the absolute value operator.
5. Define the C-like ternary conditional expression (in C, zero equals false and all other integers equal true).

## 3 Homework

$$\begin{aligned} Expr ::= {} & Num \mid \\ & \mathtt{X} \mid \\ & Expr + Expr \mid \\ & Expr \cdot Expr \mid \\ & Expr^{Num} \mid \\ & Expr' \end{aligned} \tag{11}$$

Convention: $n \in Num$ and $e, f, g \in Expr$.

$$\frac{}{n' \Rightarrow 0} \tag{12}$$

$$\frac{}{\mathtt{X}' \Rightarrow 1} \tag{13}$$

$$\frac{}{(e + f)' \Rightarrow e' + f'} \tag{14}$$

$$\frac{}{(e \cdot f)' \Rightarrow e \cdot f' + e' \cdot f} \tag{15}$$

$$\frac{}{(e^n)' \Rightarrow n \cdot e^{n-1}} \tag{16}$$

$$\frac{e \Rightarrow g}{e + f \Rightarrow g + f} \tag{17}$$

$$\frac{f \Rightarrow g}{e + f \Rightarrow e + g} \tag{18}$$

$$\frac{e \Rightarrow g}{e \cdot f \Rightarrow g \cdot f} \tag{19}$$

$$\frac{f \Rightarrow g}{e \cdot f \Rightarrow e \cdot g} \tag{20}$$

$$\frac{e \Rightarrow f}{e^n \Rightarrow f^n} \tag{21}$$