

ASSIGNMENT 2 — OPERATIONAL SEMANTICS

COMP 3010 — ORGANIZATION OF PROGRAMMING LANGUAGES

1. BIG-STEP OPERATIONAL SEMANTICS

For exercises 1 to 3, consider this definition of conditional arithmetic in terms of a big-step operational semantics.

Syntax of arithmetic (A) and boolean (B) expressions, and natural number (n) and boolean (b) values:

$$\begin{aligned}
 n &::= 0 \mid 1 \mid 2 \mid 3 \mid \dots \\
 A &::= \underline{n} \mid \mathbf{plus}(A_1, A_2) \mid \mathbf{minus}(A_1, A_2) \mid \mathbf{times}(A_1, A_2) \mid \mathbf{div}(A_1, A_2) \mid \mathbf{if}(B, A_1, A_2) \\
 b &::= \mathbf{true} \mid \mathbf{false} \\
 B &::= \underline{b} \mid \mathbf{and}(B_1, B_2) \mid \mathbf{or}(B_1, B_2) \mid \mathbf{zero?}(A)
 \end{aligned}$$

Big-step operational semantics of arithmetic expressions ($A \Downarrow n$):

$$\begin{array}{c}
 \overline{\underline{n} \Downarrow n} \\
 \frac{A_1 \Downarrow n_1 \quad A_2 \Downarrow n_2 \quad n_1 + n_2 = n}{\mathbf{plus}(A_1, A_2) \Downarrow n} \qquad \frac{A_1 \Downarrow n_1 \quad A_2 \Downarrow n_2 \quad n_1 - n_2 = n \quad n_1 \geq n_2}{\mathbf{minus}(A_1, A_2) \Downarrow n} \\
 \frac{A_1 \Downarrow n_1 \quad A_2 \Downarrow n_2 \quad n_1 \times n_2 = n}{\mathbf{times}(A_1, A_2) \Downarrow n} \qquad \frac{A_1 \Downarrow n_1 \quad A_2 \Downarrow n_2 \quad n_1 \div n_2 = n \quad n_2 \neq 0}{\mathbf{div}(A_1, A_2) \Downarrow n} \\
 \frac{B \Downarrow \mathbf{true} \quad A_1 \Downarrow n_1}{\mathbf{if}(B, A_1, A_2) \Downarrow n_1} \qquad \frac{B \Downarrow \mathbf{false} \quad A_2 \Downarrow n_2}{\mathbf{if}(B, A_1, A_2) \Downarrow n_2}
 \end{array}$$

Big-step operational semantics of boolean expressions ($B \Downarrow b$):

$$\begin{array}{c}
 \overline{\mathbf{true} \Downarrow \mathbf{true}} \qquad \overline{\mathbf{false} \Downarrow \mathbf{false}} \\
 \frac{B_1 \Downarrow \mathbf{true} \quad B_2 \Downarrow b}{\mathbf{and}(B_1, B_2) \Downarrow b} \qquad \frac{B_1 \Downarrow \mathbf{false}}{\mathbf{and}(B_1, B_2) \Downarrow \mathbf{false}} \\
 \frac{B_1 \Downarrow \mathbf{false} \quad B_2 \Downarrow b}{\mathbf{or}(B_1, B_2) \Downarrow b} \qquad \frac{B_1 \Downarrow \mathbf{true}}{\mathbf{or}(B_1, B_2) \Downarrow \mathbf{true}} \\
 \frac{A \Downarrow 0}{\mathbf{zero?}(A) \Downarrow \mathbf{true}} \qquad \frac{A \Downarrow n \quad n \neq 0}{\mathbf{zero?}(A) \Downarrow \mathbf{false}}
 \end{array}$$

For the natural number division $n_1 \div n_2$ returns only the whole number dividend and drops the remainder, so that $7 \div 2$ is 3 for example.

Exercise 1 (Multiple Choice). Which of the following evaluations of

$$\text{times}(\text{if}(\text{zero?}(\text{minus}(2, \text{plus}(1, 1))), 3, 1), 2)$$

can be derived by the operational semantics?

- (a) $\text{times}(\text{if}(\text{zero?}(\text{minus}(2, \text{plus}(1, 1))), 3, 1), 2) \Downarrow 3$
- (b) $\text{times}(\text{if}(\text{zero?}(\text{minus}(2, \text{plus}(1, 1))), 3, 1), 2) \Downarrow 1$
- (c) $\text{times}(\text{if}(\text{zero?}(\text{minus}(2, \text{plus}(1, 1))), 3, 1), 2) \Downarrow 2$
- (d) $\text{times}(\text{if}(\text{zero?}(\text{minus}(2, \text{plus}(1, 1))), 3, 1), 2) \Downarrow 6$
- (e) $\text{times}(\text{if}(\text{zero?}(\text{minus}(2, \text{plus}(1, 1))), 3, 1), 2) \Downarrow \text{true}$
- (f) $\text{times}(\text{if}(\text{zero?}(\text{minus}(2, \text{plus}(1, 1))), 3, 1), 2) \Downarrow \text{false}$

Exercise 2 (This or That). An arithmetic expression A *returns* if there is some number n such that $A \Downarrow n$, and *diverges* if there is no such n . For example, $\text{div}(1, 0)$ and $\text{minus}(0, 1)$ both diverge, and $\text{div}(0, 1)$ and $\text{minus}(1, 0)$ both return (since $\text{div}(0, 1) \Downarrow 0$ and $\text{minus}(1, 0) \Downarrow 1$). Similarly, a boolean expression B *returns* if there is some boolean value $b = \text{true}$ or $b = \text{false}$ such that $B \Downarrow b$, and *diverges* otherwise.

For each of the following arithmetic and boolean expressions, say if that expression returns or diverges.

- (a) $\text{minus}(\text{plus}(3, 1), 2)$
- (b) $\text{plus}(3, \text{minus}(1, 2))$
- (c) $\text{if}(\text{zero?}(\text{minus}(2, 2)), 0, \text{div}(3, \text{minus}(2, 2)))$
- (d) $\text{and}(\text{zero?}(\text{div}(0, 0)), \text{false})$
- (e) $\text{and}(\text{false}, \text{zero?}(\text{div}(0, 0)))$

Exercise 3 (Show Your Work). Determine the number the following expression evaluates to by drawing a derivation tree of the big-step operational semantics:

$$\text{if}(\text{zero?}(\text{minus}(\text{plus}(1, 1), 2)), \text{div}(4, 2), \text{div}(4, \text{minus}(\text{plus}(1, 1), 2)))$$

2. SMALL-STEP OPERATIONAL SEMANTICS

For exercises 4 to 6, consider this definition of a small-step operational semantics for the same conditional arithmetic language used previously in section 1.

Small-step reduction rules:

$$\begin{array}{llll}
 \text{plus}(n_1, n_2) \mapsto \underline{n} & (n = n_1 + n_2) & \text{minus}(n_1, n_2) \mapsto \underline{n} & (n = n_1 - n_2, n_1 \geq n_2) \\
 \text{times}(n_1, n_2) \mapsto \underline{n} & (n = n_1 \times n_2) & \text{div}(n_1, n_2) \mapsto \underline{n} & (n = n_1 \div n_2, n_2 \neq 0) \\
 \text{if}(\text{true}, A_1, A_2) \mapsto A_1 & & \text{if}(\text{false}, A_1, A_2) \mapsto A_2 & \\
 \text{and}(\text{true}, B) \mapsto B & & \text{and}(\text{false}, B) \mapsto \text{false} & \\
 \text{or}(\text{false}, B) \mapsto B & & \text{or}(\text{true}, B) \mapsto \text{true} & \\
 \text{zero?}(0) \mapsto \text{true} & & \text{zero?}(\underline{n}) \mapsto \text{false} & (n \neq 0)
 \end{array}$$

Evaluation contexts (E):

$$\begin{aligned}
 E ::= & \square \mid \text{plus}(E, A) \mid \text{plus}(n, E) \mid \text{minus}(E, A) \mid \text{minus}(n, E) \\
 & \mid \text{times}(E, A) \mid \text{times}(n, E) \mid \text{div}(E, A) \mid \text{div}(n, E) \mid \text{if}(E, A_1, A_2) \\
 & \mid \text{and}(E, B) \mid \text{or}(E, B) \mid \text{zero?}(E)
 \end{aligned}$$

Reducing sub-expressions is *only* allowed within evaluation contexts:

$$\frac{A \mapsto A'}{E[A] \mapsto E[A']} \qquad \frac{A \mapsto B'}{E[B] \mapsto E[B']}$$

Exercise 4 (Multiple Choice). What do you get from plugging the expression `plus(2, 3)` into the evaluation context `if(zero?(□), times(2, 4), minus(4, 1))`

- (a) `if(zero?(5), times(2, 4), minus(4, 1))`
- (b) `if(zero?(plus(2, 3)), times(2, 4), minus(4, 1))`
- (c) `if(plus(2, 3), times(2, 4), minus(4, 1))`
- (d) `if(times(2, 4), minus(4, 1), plus(2, 3))`

Exercise 5 (This or That). For each of the following pairs of a context and a sub-expression, identify which ones are valid or invalid decompositions of the expression

`if(zero?(times(plus(3, 4), minus(1, 1))), minus(5, 3), div(6, 2))`

according to the grammar of E in small-step operational semantics. (There may be multiple valid decompositions.)

- (a) `if(□, minus(5, 3), div(6, 2)) and zero?(times(plus(3, 4), minus(1, 1)))`
- (b) `if(zero?(times(plus(3, 4), minus(1, 1))), □, div(6, 2)) and minus(5, 3)`
- (c) `if(zero?(times(plus(3, 4), minus(1, 1))), minus(5, 3), □) and div(6, 2)`
- (d) `if(zero?(times(□, minus(1, 1))), minus(5, 3), div(6, 2)) and plus(3, 4)`
- (e) `if(zero?(times(plus(3, 4), □)), minus(5, 3), div(6, 2)) and minus(1, 1)`

Exercise 6 (Show Your Work). Write down the sequence of reduction step using the rules of the small-step operational semantics for simplifying

`if(zero?(times(plus(3, 4), minus(1, 1))), minus(5, 3), div(6, 2))`

to its final result.