CS208: Mathematical Foundations of CS **Fall 2018**

Lecture 21: September 28

Lecturer: Samar *Scribes: Arpit(B17009), Aniket(B17076)*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

21.1 Arithmetic Expressions

This recursive data type is defined as follows:

- **Base Case**:

$$x \in AEXP$$

$$k \in AEXP, \quad k \in Z$$

- **Constructors**:

$$e, f \in AEXP$$

$$e + f \in AEXP$$

$$e - f \in AEXP$$

$$e \times f \in AEXP$$

$$-e \in AEXP$$

Exercise: Show that $3x^2 + x + 1$ is an Arithmetic Expression.

21.1.1 Evaluation of Arithmetic Expressions

Evaluation of an arithmetic expression e at a value n is represented as $eval(e, n)$. It is a recursive data type which maps from $AEXP \times Z$ to Z

- **Base Case**:

$$eval(x, n) = n$$

$$eval(k, n) = k, \quad k \in Z$$

- **Constructors**:

$$eval(e_1 + e_2, n) = eval(e_1, n) + eval(e_2, n)$$

$$eval(e_1 \times e_2, n) = eval(e_1, n) \times eval(e_2, n)$$

$$eval(-e_1, n) = -eval(e_1, n)$$

Exercise: Find the value of $eval(3 + x^2, 2)$.

21.1.2 Substitution of Arithmetic Expressions

It is represented as $subst(f, e)$. The definition is as follows:

- **Base Case**:
$$subst(f, x) = f$$
$$subst(f, k) = k$$

- **Constructors**:
$$subst(f, e_1 + e_2) = subst(f, e_1) + subst(f, e_2)$$
$$subst(f, e_1 \times e_2) = subst(f, e_1) \times subst(f, e_2)$$
$$subst(f, -e) = -subst(f, e)$$

  Exercise: Evaluate $subst(3x, x(x - 1))$.

Lemma 1 $\forall \quad e, f \in AEXP \quad and \quad x \in Z$
$$eval(subst(f, e), n) = eval(e, eval(f, n))$$

Proof: This proof is by structural induction.

Base Case: e = x

$$\begin{aligned} L.H.S. &= eval(subst(f, x), n) \\ &= eval(f, n) \end{aligned}$$

$$\begin{aligned} R.H.S. &= eval(e, eval(f, n)) \\ &= eval(x, eval(f, n) \\ &= eval(f, n) \end{aligned}$$

$\Rightarrow$ Base case is true.

Induction Step: Let's say for $e = e_i$, the given statement holds. Now,

$$\begin{aligned} & eval(subst(f, e_1 + e_2), n) \\ & \qquad = eval(subst(f, e_1) + subst(f, e_2), n) \\ & \qquad = eval(subst(f, e_1), n) + eval(subst(f, e_2), n) \\ & \qquad = eval(e_1, eval(f, n)) + eval(e_2, eval(f, n)) \\ & \qquad = eval(e_1 + e_2, eval(f, n)) \end{aligned}$$

$\Rightarrow$ The lemma is true for first constructor case.

$$\begin{aligned} & eval(subst(f, e_1 \times e_2), n) \\ & \qquad = eval(subst(f, e_1) \times subst(f, e_2), n) \\ & \qquad = eval(subst(f, e_1), n) \times eval(subst(f, e_2), n) \\ & \qquad = eval(e_1, eval(f, n)) \times eval(e_2, eval(f, n)) \\ & \qquad = eval(e_1 \times e_2, eval(f, n)) \end{aligned}$$

$\Rightarrow$ The lemma is true for second constructor case.

$$\begin{aligned}
&eval(subst(f, -e), n) \\
&\qquad = eval(-subst(f, e), n) \\
&\qquad = -eval(subst(f, e), n) \\
&\qquad = -eval(e, eval(f, n)) \\
&\qquad = eval(-e, eval(f, n))
\end{aligned}$$

$\Rightarrow$ The lemma is true for last constructor case.
$\Rightarrow$ The claim is true $\quad \forall \quad e, f \in AEXP \quad and \quad x \in Z.$ ■

Exercise: Evaluate $eval(subst(3x, x(x-1), 2)$