Arith Proof

Paul Franosch

October 2025

1 Inference Rules

Expressions

$$\text{Exp} ::= \text{Const } n \mid e_1 + e_2 \mid -e \mid e_1 * e_2$$

Values

Constants are values; they do not step:

$$\text{Value } v ::= \text{Const } n$$

Small-step rules

$$\langle \text{Const } n_1 + \text{Const } n_2 \rangle \to \text{Const } (n_1 + n_2)$$

$$\frac{\langle e_1 \rangle \to e_1'}{\langle e_1 + e_2 \rangle \to \langle e_1' + e_2 \rangle}$$

$$\frac{\langle e_2 \rangle \to e_2'}{\langle \text{Const } n + e_2 \rangle \to \langle \text{Const } n + e_2' \rangle}$$

$$\langle -\text{Const } n \rangle \to \text{Const } (-n)$$

$$\frac{\langle e \rangle \to e'}{\langle -e \rangle \to -e'}$$

$$\langle \text{Const } n_1 * \text{Const } n_2 \rangle \to \text{Const } (n_1 * n_2)$$

$$\frac{\langle e_1 \rangle \to e_1'}{\langle e_1 * e_2 \rangle \to \langle e_1' * e_2 \rangle}$$

$$\frac{\langle e_2 \rangle \to e_2'}{\langle \text{Const } n * e_2 \rangle \to \langle \text{Const } n * e_2' \rangle}$$

Proof

For every expression e, there exists a finite sequence of small-step reductions leading to the value computed by the standard evaluation function $\text{eval}(e)$:

Proof. We prove by structural induction.

Induction Hypothesis

For any expression e, there exists a finite sequence of small-step reductions

$$\exists n \geq 0, \exists e_0, \ldots, e_n : e_0 = e \wedge e_n = \text{Const}(\text{eval}(e)) \wedge \forall i < n, e_i \to e_{i+1}.$$

Base Case

$e = \text{Const } n$
Constants are values, so no reduction is needed:

$$\text{eval}(\text{Const } n) = n.$$

Induction Step

We perform case distinction on e.

Case 1: Addition $e = e_1 + e_2$ By the induction hypothesis (IH), there exist sequences

$$e_1 \to^* v_1 = \text{Const}(\text{eval}(e_1)), \qquad e_2 \to^* v_2 = \text{Const}(\text{eval}(e_2)).$$

Then, using the small-step rules for addition:

$$\begin{aligned} e_1 + e_2 &\to^* v_1 + e_2 \\ &\to^* v_1 + v_2 \\ &\to \text{Const}(\text{eval}(e_1) + \text{eval}(e_2)). \end{aligned}$$

Case 2: Negation $e = -e_1$ By IH, $e_1 \to^* \text{Const}(\text{eval}(e_1))$. Then, applying the negation rule:

$$\begin{aligned} -e_1 &\to^* -\text{Const}(\text{eval}(e_1)) \\ &\to \text{Const}(-\text{eval}(e_1)). \end{aligned}$$

Case 3: Multiplication $e = e_1 * e_2$ By IH, $e_1 \to^* v_1 = \text{Const}(\text{eval}(e_1))$, $e_2 \to^* v_2 = \text{Const}(\text{eval}(e_2))$. Then, by the multiplication rules:

$$\begin{aligned} e_1 * e_2 &\to^* v_1 * e_2 \\ &\to^* v_1 * v_2 \\ &\to \text{Const}(\text{eval}(e_1) * \text{eval}(e_2)). \end{aligned}$$

By structural induction on e, every expression reduces in finitely many small steps to $\mathrm{Const}(\mathrm{eval}(e))$, as required.

$\square$