GrumpyIR Dynamic Semantics

April 4, 2021

The GrumpyIR dynamic (small-step operational) semantics is given as a six-place relation between a function environment δ, a variable environment ρ, heap stores μ and μ', and expressions e and e', written $\delta \mid \rho \vdash e \mid \mu \longrightarrow e' \mid \mu'$, pronounced "under δ and ρ, in memory state μ, e steps to e' resulting in new state μ'". Formally, the dynamic semantics is taken to be the smallest relation satisfying the following transition rules, where metavariables n and n_i range over numbers, b over booleans or binary operators depending on context, e and e_i over expressions, v and v_i over values, l over locations, p over function pointers, and x over variables:

Variables

$$\textsc{E-var}\quad \frac{\rho(x) = v}{\delta \mid \rho \vdash x \mid \mu \longrightarrow v \mid \mu}$$

A variable x steps to a value v whenever ρ maps x to v.

Unary and binary operators

$$\textsc{E-neg}\quad \frac{b \in \{\text{true}, \text{false}\}}{\delta \mid \rho \vdash (\text{neg } b) \mid \mu \longrightarrow \neg b \mid \mu}$$

$$\textsc{E-neg1}\quad \frac{\delta \mid \rho \vdash e \mid \mu \longrightarrow e' \mid \mu'}{\delta \mid \rho \vdash (\text{neg } e) \mid \mu \longrightarrow (\text{neg } e') \mid \mu'}$$

$$\textsc{E-binop}\quad \frac{n_1 \; b \; n_2 = n \qquad b \in \{+, *, -, /\}}{\delta \mid \rho \vdash (b \; n_1 \; n_2) \mid \mu \longrightarrow n \mid \mu}$$

$$\textsc{E-binop1}\quad \frac{\delta \mid \rho \vdash e_1 \mid \mu \longrightarrow e_1' \mid \mu'}{\delta \mid \rho \vdash (b \; e_1 \; e_2) \mid \mu \longrightarrow (b \; e_1' \; e_2) \mid \mu'}$$

$$\textsc{E-binop2}\quad \frac{\delta \mid \rho \vdash e_2 \mid \mu \longrightarrow e_2' \mid \mu'}{\delta \mid \rho \vdash (b \; n_1 \; e_2) \mid \mu \longrightarrow (b \; n_1 \; e_2') \mid \mu'}$$

Let-expressions and sequencing

$$\textsc{E-let}\quad \frac{}{\delta \mid \rho \vdash (\text{let } x\ v_1\ v_2) \mid \mu \longrightarrow v_2 \mid \mu}$$

$$\textsc{E-let1}\quad \frac{\delta \mid \rho \vdash e_1 \mid \mu \longrightarrow e_1' \mid \mu'}{\delta \mid \rho \vdash (\text{let } x\ e_1\ e_2) \mid \mu \longrightarrow (\text{let } x\ e_1'\ e_2) \mid \mu'}$$

$$\textsc{E-let2}\quad \frac{\delta \mid [x \mapsto v_1]\rho \vdash e_2 \mid \mu \longrightarrow e_2' \mid \mu'}{\delta \mid \rho \vdash (\text{let } x\ v_1\ e_2) \mid \mu \longrightarrow (\text{let } x\ v_1\ e_2') \mid \mu'}$$

$$\textsc{E-seq}\quad \frac{}{\delta \mid \rho \vdash (\text{seq v } e_2) \mid \mu \longrightarrow e_2 \mid \mu}$$

$$\textsc{E-seq1}\quad \frac{\delta \mid \rho \vdash e_1 \mid \mu \longrightarrow e_1' \mid \mu'}{\delta \mid \rho \vdash (\text{seq } e_1\ e_2) \mid \mu \longrightarrow (\text{seq } e_1'\ e_2) \mid \mu'}$$

Arrays

E-ALLOC
$$\frac{l \text{ is a freshly allocated store location} \qquad 0 \leq n \qquad \forall i, v_i = v}{\delta \mid \rho \vdash (\text{alloc } n\ v) \mid \mu \longrightarrow l \mid [l \mapsto (v_1, v_2, ..., v_n)]\mu}$$

E-ALLOC1
$$\frac{\delta \mid \rho \vdash e_{size} \mid \mu \longrightarrow e'_{size} \mid \mu'}{\delta \mid \rho \vdash (\text{alloc } e_{size}\ e_{init}) \mid \mu \longrightarrow (\text{alloc } e'_{size}\ e_{init}) \mid \mu'}$$

E-ALLOC2
$$\frac{\delta \mid \rho \vdash e_{init} \mid \mu \longrightarrow e'_{init} \mid \mu'}{\delta \mid \rho \vdash (\text{alloc } n\ e_{init}) \mid \mu \longrightarrow (\text{alloc } n\ e'_{init}) \mid \mu'}$$

E-SET
$$\frac{0 \leq i \qquad \forall j \neq i, v_j = \mu(l)_j}{\delta \mid \rho \vdash (\text{set } l\ i\ v) \mid \mu \longrightarrow \text{tt} \mid [l \mapsto (v_1, ..., v_{i-1}, v, v_{i+1}, ..., v_n)]\mu}$$

E-SET1
$$\frac{\delta \mid \rho \vdash e_{arr} \mid \mu \longrightarrow e'_{arr} \mid \mu'}{\delta \mid \rho \vdash (\text{set } e_{arr}\ e_{ix}\ e) \mid \mu \longrightarrow (\text{set } e'_{arr}\ e_{ix}\ e) \mid \mu'}$$

E-SET2
$$\frac{\delta \mid \rho \vdash e_{ix} \mid \mu \longrightarrow e'_{ix} \mid \mu'}{\delta \mid \rho \vdash (\text{set } l\ e_{ix}\ e) \mid \mu \longrightarrow (\text{set } l\ e'_{ix}\ e) \mid \mu'}$$

E-SET3
$$\frac{\delta \mid \rho \vdash e \mid \mu \longrightarrow e' \mid \mu'}{\delta \mid \rho \vdash (\text{set } l\ i\ e) \mid \mu \longrightarrow (\text{set } l\ i\ e') \mid \mu'}$$

E-GET
$$\frac{0 \leq i \qquad \mu(l)_i = v}{\delta \mid \rho \vdash (\text{get } l\ i) \mid \mu \longrightarrow v \mid \mu}$$

E-GET1
$$\frac{\delta \mid \rho \vdash e_{arr} \mid \mu \longrightarrow e'_{arr} \mid \mu'}{\delta \mid \rho \vdash (\text{get } e_{arr}\ e_{ix}) \mid \mu \longrightarrow (\text{get } e'_{arr}\ e_{ix}) \mid \mu'}$$

E-GET2
$$\frac{\delta \mid \rho \vdash e_{ix} \mid \mu \longrightarrow e'_{ix} \mid \mu'}{\delta \mid \rho \vdash (\text{get } l\ e_{ix}) \mid \mu \longrightarrow (\text{get } l\ e'_{ix}) \mid \mu'}$$

Conditionals and function calls

E-COND-TRUE

$$\frac{}{\delta \mid \rho \vdash (\text{cond true}\, e_1\; e_2) \mid \mu \longrightarrow e_1 \mid \mu}$$

E-COND-FALSE

$$\frac{}{\delta \mid \rho \vdash (\text{cond false}\, e_1\; e_2) \mid \mu \longrightarrow e_2 \mid \mu}$$

E-COND

$$\frac{\delta \mid \rho \vdash e_{cond} \mid \mu \longrightarrow e'_{cond} \mid \mu'}{\delta \mid \rho \vdash (\text{cond}\; e_{cond}\, e_1\; e_2) \mid \mu \longrightarrow (\text{cond}\; e'_{cond}\, e_1\; e_2) \mid \mu'}$$

E-CALL

$$\frac{\delta(p) = ((x_1, x_2, ..., x_n), e) \qquad \delta \mid [x_i \mapsto v_i]\rho_0 \vdash e \longrightarrow^* v \mid \mu'}{\delta \mid \rho \vdash (\text{call}\; p\; v_1\; v_2\; ...\; v_n) \mid \mu \longrightarrow v \mid \mu'}$$

E-CALL1

$$\frac{\delta \mid \rho \vdash e \mid \mu \longrightarrow e' \mid \mu'}{\delta \mid \rho \vdash (\text{call}\; e\; e_1\; e_2\; ...\; e_n) \mid \mu \longrightarrow (\text{call}\; e'\; e_1\; e_2\; ...\; e_n) \mid \mu'}$$

E-CALL2

$$\frac{\delta \mid \rho \vdash e_i \mid \mu \longrightarrow e'_i \mid \mu'}{\delta \mid \rho \vdash (\text{call}\; p\; v_1\; ...\; e_i\; ...\; e_n) \mid \mu \longrightarrow (\text{call}\; p\; e_1\; ...\; e'_i\; ...\; e_n) \mid \mu'}$$

where ρ_0 is the empty variable environment and $\longrightarrow^*$ is the reflexive transitive closure of $\longrightarrow$.