POPL17 Dynamics

Jonathan Lam

2022/01/03

Jon's note: This is copied from the hazelgrove/hazelnut repository, and the comments are de-commented.

Cyrus: We took a look at the dynamics. Overall it seems like the right idea. We noticed:

Done: 1) The stepping rules are non-deterministic (i.e. you can step the right or left of e1 + e2 in any order). Might be useful to make them deterministic.

Done (mostly): 2) The premises that have disjunctions in them could be broken out into two rules – this would take a little more space but follows the usual conventions more closely.

Done (mostly): 3) We need to add the "ceil" forms to the grammar of $\dot{e}$ (you just used e) and give them a static semantics.

TODO: 4) We need to figure out (the analogs of) canonical forms, preservation and progress – I guess we had decided on defining a declarative statics to do that. Ian has started to prove the correspondence (sans the ceil forms).

#4 seems like the most important next step.

$\boxed{\dot{e}\ \mathsf{value}}$ H-Expression $\dot{e}$ is a closed value

$$\frac{}{\underline{n}\ \mathsf{value}}\ \text{V-num} \qquad \frac{}{(\lambda x.\dot{e})\ \mathsf{value}}\ \text{V-lam}$$

Figure 1: Value forms

$\boxed{\dot{e}\ \mathsf{final}}$ H-Expression $\dot{e}$ is final

$$\frac{\dot{e}\ \mathsf{value}}{\dot{e}\ \mathsf{final}}\ \text{F-val} \qquad \frac{\dot{e}\ \mathsf{final}}{(\!|\dot{e}|\!)\ \mathsf{final}}\ \text{F-filled} \qquad \frac{}{(\!|\,|\!)\ \mathsf{final}}\ \text{F-unfilled} \qquad \frac{\dot{e}\ \mathsf{indet}}{\lceil\dot{e}\rceil\ \mathsf{final}}\ \text{F-indet}$$

Figure 2: Final forms

$\boxed{\dot{e}\ \mathsf{indet}}$ H-Expression $\dot{e}$ is indeterminate

$$\frac{\dot{e}_1\ \mathsf{final} \qquad \dot{e}_2\ \mathsf{final} \qquad \dot{e}_1 \neq \underline{n_1}}{(e_1 + e_2)\ \mathsf{indet}}\ \text{I-plus}_1$$

$$\frac{\dot{e}_1\ \mathsf{final} \qquad \dot{e}_2\ \mathsf{final} \qquad \dot{e}_2 \neq \underline{n_2}}{(\dot{e}_1 + \dot{e}_2)\ \mathsf{indet}}\ \text{I-plus}_2$$

$$\frac{\dot{e}_1\ \mathsf{final} \qquad \dot{e}_2\ \mathsf{final} \qquad \dot{e}_1 \neq (\lambda x.\dot{e}_1')}{\dot{e}_1(\dot{e}_2)\ \mathsf{indet}}\ \text{I-app}$$

Figure 3: Indeterminate forms

$\boxed{\dot{e}_1 \longrightarrow \dot{e}_2}$ H-Expression $\dot{e}_1$ steps to $\dot{e}_2$

$$\frac{\dot{e}_1 \longrightarrow \dot{e}_1'}{(\dot{e}_1 + \dot{e}_2) \longrightarrow (\dot{e}_1' + \dot{e}_2)}\ \text{S-plus}_1 \qquad \frac{\dot{e}_1\ \mathsf{final} \qquad \dot{e}_2 \longrightarrow \dot{e}_2'}{(\dot{e}_1 + \dot{e}_2) \longrightarrow (\dot{e}_1 + \dot{e}_2')}\ \text{S-plus}_2$$

$$\frac{n_1 + n_2 = n_3}{(\underline{n_1} + \underline{n_2}) \longrightarrow \underline{n_3}}\ \text{S-plus}_3 \qquad \frac{\begin{array}{c}\dot{e}_1\ \mathsf{final} \qquad \dot{e}_2\ \mathsf{final} \\ \big(\dot{e}_1 \neq \underline{n_1} \vee \dot{e}_2 \neq \underline{n_2}\big)\end{array}}{(\dot{e}_1 + \dot{e}_2) \longrightarrow \lceil(\dot{e}_1 + \dot{e}_2)\rceil}\ \text{S-plus}_4$$

$$\frac{\dot{e}_1 \longrightarrow \dot{e}_1'}{\dot{e}_1(\dot{e}_2) \longrightarrow \dot{e}_1'(\dot{e}_2)}\ \text{S-ap}_1 \qquad \frac{\dot{e}_2 \longrightarrow \dot{e}_2' \qquad \dot{e}_1\ \mathsf{final}}{\dot{e}_1(\dot{e}_2) \longrightarrow \dot{e}_1(\dot{e}_2')}\ \text{S-ap}_2$$

$$\frac{\dot{e}_2\ \mathsf{final}}{(\lambda x.\dot{e}_1)(\dot{e}_2) \longrightarrow [\dot{e}_2/x]\dot{e}_1}\ \text{S-ap}_3 \qquad \frac{\begin{array}{c}\dot{e}_1\ \mathsf{final} \qquad \dot{e}_2\ \mathsf{final} \\ \dot{e}_1 \neq (\lambda x.\dot{e}_1')\end{array}}{\dot{e}_1(\dot{e}_2) \longrightarrow \lceil \dot{e}_1(\dot{e}_2)\rceil}\ \text{S-ap}_4$$

Figure 4: Small-step operational semantics.

We extend the syntax of H-Expressions as follows:

$$\dot{e} ::= \cdots \mid \lceil \dot{e} \rceil$$

Here are some things that we want to prove:

Def (Ascription erasure).
$|\dot{e}|_{\mathsf{erase}}$ is the same as $\dot{e}$, but without its type ascriptions. All cases are congruences, except for the ascription case, where $|\dot{e} : \dot{\tau}|_{\mathsf{erase}} = \dot{e}$.

Def (Declarative typing).
The judgement $\dot{\Gamma} \vdash \dot{e} : \dot{\tau}$ is a type assignment system for erased terms. It is declarative, and unlike the bidirectional rules, is not algorithmic. We employ it to relate bidirectionally-typed terms to (erased) terms that enjoy type soundness with respect to the dynamics.

Conjecture (Bidirectional implies declarative).
(i) If $\dot{\Gamma} \vdash \dot{e} \Rightarrow \dot{\tau}$ then $\dot{\Gamma} \vdash |\dot{e}|_{\mathsf{erase}} : \dot{\tau}$.
(ii) If $\dot{\Gamma} \vdash \dot{e} \Leftarrow \dot{\tau}$ then $\dot{\Gamma} \vdash |\dot{e}|_{\mathsf{erase}} : \dot{\tau}$.

Conjecture (Substitution).
If $\dot{\Gamma}, x : \tau_x \vdash \dot{e} : \dot{\tau}$
and $\dot{e}'$ final (do we actually need this condition?)
and $\dot{\Gamma} \vdash \dot{e}' : \dot{\tau}_x$
then $\dot{\Gamma} \vdash \dot{e}[\dot{e}'/x] : \dot{\tau}$

Conjecture (Canonical forms)
If $\cdot \vdash \dot{e} : \dot{\tau}$
and $\dot{e}$ final then

- if $\dot{e} = \lceil \dot{e}' \rceil$ then $\dot{e}'$ indet and $\cdot \vdash \dot{e}' : \dot{\tau}$
- else:
  - if $\dot{\tau} = \mathtt{num}$ then exists $\underline{n}$ such that $\dot{e} = \underline{n}$.
  - else if $\dot{\tau} = (\dot{\tau}_1 \to \dot{\tau}_2)$ then exists x and $\dot{e}'$ such that $\dot{e} = (\lambda x.\dot{e}')$
  - else if $\dot{\tau} = (\!|\!)$ then either:
    - $\dot{e} = (\!|\!)$, or
    - exists $\dot{\tau}'$ and $\dot{e}'$ such that $\dot{e} = (\!|\dot{e}'|\!)$, $\dot{e}'$ final and $\cdot \vdash \dot{e}' : \dot{\tau}'$

Conjecture (Progresss).
If $\cdot \vdash \dot{e}_1 : \dot{\tau}$
then either $\dot{e}_1$ final
or exists $\dot{e}_2$ such that $\dot{e}_1 \longrightarrow \dot{e}_2$.

Conjecture (Preservation).
If $\cdot \vdash \dot{e}_1 : \dot{\tau}$
and $\dot{e}_1 \longrightarrow \dot{e}_2$
then $\cdot \vdash \dot{e}_2 : \dot{\tau}$