Evaluation Contexts

(Yet another way to avoid search rules)

$e ::= x \mid () \mid \lambda x.e \mid e\ e \mid (e,e) \mid \text{fst}\ e \mid \text{snd}\ e \qquad v ::= () \mid \lambda x.e \mid (v,v)$

$\tau ::= \text{unit} \mid \tau \to \tau \mid \tau \times \tau$

$\mathcal{E} ::= \underset{[\,]}{\circ} \mid \mathcal{E}\ e \mid v\ \mathcal{E} \mid (\mathcal{E}, e) \mid (v, \mathcal{E}) \mid \text{fst}\ \mathcal{E} \mid \text{snd}\ \mathcal{E}$

Evaluation Contexts
- exprs w/ one "hole"

$\underbrace{(5, (6, (\lambda x.x)\ \underbrace{(\text{fst}\ (7,8))}_{\text{the part that can step}}))}_{\text{everything else}}$

$\mathcal{E}[\text{fst}\ (7,8)]$

$\mathcal{E}[e]$ – fill the hole w/ e

$\circ[e] = e$

$(\mathcal{E}\ e)[e'] = (\mathcal{E}[e'])\ e$

$(v\ \mathcal{E})[e] = v\ \mathcal{E}[e]$

$(\mathcal{E}, e)[e'] = (\mathcal{E}[e'], e)$

$(v, \mathcal{E})[e'] = (v, \mathcal{E}[e'])$

$(\text{fst}\ \mathcal{E})[e] = \text{fst}\ (\mathcal{E}[e])$

$(\text{snd}\ \mathcal{E})[e] = \text{snd}\ (\mathcal{E}[e])$

$(5, (6, (\lambda x.x)\ \circ))[\text{fst}\ (7,8)]$
$= (5, (6, (\lambda x.x)\ (\text{fst}\ (7,8))))$

$$\frac{e \mapsto e'}{\mathcal{E}[e] \mapsto \mathcal{E}[e']}$$

– One big rule!

(still need to define this judgment)

$$\frac{}{(\lambda x.e)\ v \to [v/x]e} \qquad \frac{}{\text{fst}\ (v_1, v_2) \to v_1} \qquad \frac{}{\text{snd}\ (v_1, v_2) \to v_2}$$

Contexts (("Expression contexts" / ("Program contexts")

$C ::= \circ \mid \lambda x.C \mid e\,C \mid C\,e \mid (C,e) \mid (e,C) \mid \mathrm{fst}\ C \mid \mathrm{snd}\ C$

- Any expression w/ a hole; hole doesn't need to be where we're evaluating

$(\circ, (6, (\lambda x.x)(\mathrm{fst}(7,8))))[5] = (5, (6, (\lambda x.x)(\mathrm{fst}\ (7,8))))$

Type of a context?
First guess: $\tau \rightsquigarrow \tau'$
But:
$\lambda x.\circ$ exp can have free x!
Need a context
$(\lambda x.\circ)[y]$ Can also use in a diff. context!

$C : (\Gamma \triangleright \tau) \rightsquigarrow (\Gamma' \triangleright \tau') \Leftrightarrow$ If $\Gamma \vdash e : \tau$, then $\Gamma' \vdash C[e] : \tau'$

$\lambda x.\circ : (\Gamma, x:\tau \triangleright \tau') \rightsquigarrow (\Gamma \triangleright \tau \to \tau')$

Program context: Complete program w/ a hole
$C : (\Gamma \triangleright \tau) \rightsquigarrow (\bullet \triangleright B)$
Base type: unit, int, bool, ...

Take a base type, say int. "Kleene" ($e \simeq e'$)
Suppose $\bullet \vdash e : \mathrm{int}$ and $\bullet \vdash e' : \mathrm{int}$. e and e' are equal iff $\exists n$ s.t. $e \mapsto^* \bar{n}$ and $e' \mapsto^* \bar{n}$.

What does it mean for exps of other types to be equal?

$\lambda x.x \stackrel{?}{=} \lambda y.\mathrm{fst}(y,y)$
$x \stackrel{?}{=} \mathrm{fst}\ (x,x)$

Observational/Contextual Equivalence

Suppose $\Gamma \vdash e : \tau$ and $\Gamma \vdash e : \tau'$. $\Gamma \vdash e \cong e' : \tau$ if
for all $C : (\Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright nat)$, $C[e] \simeq C[e']$

$\lambda x. x \cong \lambda y.\ fst\ (y, y)$? Yes.

Consider a context C.

If it doesn't use $\circ$ in an interesting way (e.g. $fst\ (7, \circ)$)
then $C[\lambda x.x] \simeq C[\lambda y. fst\ (y,y)]$

If it does, it must use it by applying it to an (eventual) int.

$$\cdots (\lambda x. x)\ \bar{n} \cdots \cong \cdots (\lambda y. fst(y,y))\ \bar{n} \cdots$$

$$\Leftrightarrow \cdots \bar{n} \cdots \cong \cdots fst\ (\bar{n}, \bar{n}) \cdots$$

$$\cong \cdots \bar{n} \cdots \checkmark$$

formal proof: more complicated, usually don't use ctx. eq. directly
But based on the above intuition