A fuller language: PCF

Simply typed call-by-value lambda calculus plus:

- a fixpoint operator (`rec`) to get needed expressiveness within simply typed framework
- a name binding form (`let`) for convenience
- some additional basic values (numbers, booleans), corresponding types (`nat`, `bool`), and operations (+, =, `not`, `if`)
- some additional value constructors (pairs, unions), corresponding type constructors (×, +), and operations

```
E   ::= I | λI:τ.E | E E
      | rec I:τ = E
      | let I = E in E
      | N | E+E | E=E | ...
      | true | false | not E | ...
      | if E then E else E
      | pair E E | first E | second E
      | mkfirst_{τ+τ} E | mksecond_{τ+τ} E
      | case E first(I) E second(I) E
τ   ::= nat | bool | τ→τ | τ×τ | τ+τ
```

`rec` is nicer syntax for `fix`:

`rec I:τ = E` $\Leftrightarrow$ `fix λI:τ.E`

Static semantics of PCF

Identifiers:

[var] $\Gamma \vdash \mathtt{I}{:}\tau$ if $\mathtt{I}{:}\tau \in \Gamma$

Functions:

[→ intro] $$\frac{\Gamma \cup \mathtt{I}{:}\tau_1 \vdash \mathtt{E}{:}\tau_2}{\Gamma \vdash (\lambda \mathtt{I}{:}\tau_1.\mathtt{E}){:}\tau_1 \to \tau_2}$$

[→ elim] $$\frac{\Gamma \vdash \mathtt{E}_1{:}\tau_1 \to \tau_2 \qquad \Gamma \vdash \mathtt{E}_2{:}\tau_1}{\Gamma \vdash (\mathtt{E}_1\ \mathtt{E}_2){:}\tau_2}$$

Recursion:

[rec] $$\frac{\Gamma \cup \mathtt{I}{:}\tau \vdash \mathtt{E}{:}\tau}{\Gamma \vdash (\mathtt{rec\ I}{:}\tau\ \mathtt{=\ E}){:}\tau}$$

Name binding:

[let] $$\frac{\Gamma \vdash \mathtt{E}_1{:}\tau_1 \qquad \Gamma \cup \mathtt{I}{:}\tau_1 \vdash \mathtt{E}_2{:}\tau_2}{\Gamma \vdash (\mathtt{let\ I\ =\ E}_1\ \mathtt{in\ E}_2){:}\tau_2}$$

Numbers:

[num] $\Gamma \vdash \mathtt{N}{:}\mathtt{nat}$

[+] $$\frac{\Gamma \vdash \mathtt{E}_1{:}\mathtt{nat} \qquad \Gamma \vdash \mathtt{E}_2{:}\mathtt{nat}}{\Gamma \vdash (\mathtt{E}_1\mathtt{+E}_2){:}\mathtt{nat}}$$

[=] $$\frac{\Gamma \vdash \mathtt{E}_1{:}\mathtt{nat} \qquad \Gamma \vdash \mathtt{E}_2{:}\mathtt{nat}}{\Gamma \vdash (\mathtt{E}_1\mathtt{=E}_2){:}\mathtt{bool}}$$

Booleans:

[true] $\Gamma \vdash \mathtt{true}{:}\mathtt{bool}$

[false] $\Gamma \vdash \mathtt{false}{:}\mathtt{bool}$

[not] $$\frac{\Gamma \vdash \mathtt{E}{:}\mathtt{bool}}{\Gamma \vdash (\mathtt{not\ E}){:}\mathtt{bool}}$$

[if] $$\frac{\Gamma \vdash \mathtt{E}_1{:}\mathtt{bool} \qquad \Gamma \vdash \mathtt{E}_2{:}\tau \qquad \Gamma \vdash \mathtt{E}_3{:}\tau}{\Gamma \vdash (\mathtt{if\ E}_1\ \mathtt{then\ E}_2\ \mathtt{else\ E}_3){:}\tau}$$

Pairs:

[× intro] $$\frac{\Gamma \vdash \mathtt{E}_1{:}\tau_1 \qquad \Gamma \vdash \mathtt{E}_2{:}\tau_2}{\Gamma \vdash (\mathtt{pair\ E}_1\ \mathtt{E}_2){:}\tau_1 \times \tau_2}$$

[× $\mathrm{elim}_{\mathrm{first}}$] $$\frac{\Gamma \vdash \mathtt{E}{:}\tau_1 \times \tau_2}{\Gamma \vdash (\mathtt{first\ E}){:}\tau_1}$$

[× $\mathrm{elim}_{\mathrm{second}}$] $$\frac{\Gamma \vdash \mathtt{E}{:}\tau_1 \times \tau_2}{\Gamma \vdash (\mathtt{second\ E}){:}\tau_2}$$

Unions:

[+ $\mathrm{intro}_{\mathrm{first}}$] $$\frac{\Gamma \vdash \mathtt{E}{:}\tau_1}{\Gamma \vdash \left(\mathtt{mkfirst}_{\tau_1+\tau_2}\ \mathtt{E}\right){:}\tau_1 + \tau_2}$$

[+ $\mathrm{intro}_{\mathrm{second}}$] $$\frac{\Gamma \vdash \mathtt{E}{:}\tau_2}{\Gamma \vdash \left(\mathtt{mksecond}_{\tau_1+\tau_2}\ \mathtt{E}\right){:}\tau_1 + \tau_2}$$

[+ elim] $$\frac{\Gamma \vdash \mathtt{E}{:}\tau_1 + \tau_2 \qquad \Gamma \cup \mathtt{I}_1{:}\tau_1 \vdash \mathtt{E}_1{:}\tau \qquad \Gamma \cup \mathtt{I}_2{:}\tau_2 \vdash \mathtt{E}_2{:}\tau}{\Gamma \vdash \begin{pmatrix}\mathtt{case\ E\ first(I}_1\mathtt{)\ E}_1 \\ \mathtt{second(I}_2\mathtt{)\ E}_2\end{pmatrix}{:}\tau}$$

Natural (operational) semantics of PCF

Evaluation judgments: $\rho \vdash \mathtt{E} \Downarrow \mathtt{V}$ if
in environment ρ the expression $\mathtt{E}$ evaluates to value $\mathtt{V}$

Domain of environment: $\rho \in \text{Env} \equiv \text{Id} \to \text{Val}$

Domain of values: $\mathtt{V} \in \text{Val} \equiv \bigcup_\tau \text{Val}_\tau$

$\text{Val}_{\mathtt{nat}} \equiv \mathrm{N}$

$\text{Val}_{\mathtt{bool}} \equiv \{\text{true},\text{false}\}$

$\text{Val}_{\tau1\to\tau2} \equiv [\mathtt{E},\text{Env}]$

$\text{Val}_{\tau1\times\tau2} \equiv (\text{Val},\text{Val})$

$\text{Val}_{\tau1+\tau2} \equiv \langle\text{first=Val}\rangle \cup \langle\text{second=Val}\rangle$

Identifiers:

$[\text{var}] \qquad \rho \vdash \mathtt{I} \Downarrow \mathtt{V} \qquad \text{if } \mathtt{I{:}V} \in \rho$

Functions:

$[\lambda] \qquad \rho \vdash (\lambda\mathtt{I.E}) \Downarrow [(\lambda\mathtt{I.E}),\rho]$

$$[\text{app}] \quad \frac{\rho \vdash \mathtt{E_1} \Downarrow [(\lambda\mathtt{I.E}),\rho'] \qquad \rho \vdash \mathtt{E_2} \Downarrow \mathtt{V_2} \qquad \rho' \cup \mathtt{I{:}V_2} \vdash \mathtt{E} \Downarrow \mathtt{V}}{\rho \vdash (\mathtt{E_1\ E_2}) \Downarrow \mathtt{V}}$$

Recursion:

$$[\text{rec}] \quad \frac{\rho \cup \mathtt{I{:}V} \vdash \mathtt{E} \Downarrow \mathtt{V}}{\rho \vdash (\mathtt{rec\ I{:}\tau\ =\ E}) \Downarrow \mathtt{V}}$$

- hmmm....
- implementable using side-effects
  if $\mathtt{E}$ is of the form $\lambda\mathtt{I'{:}\tau'.E'}$
- version in small-step semantics:
  $\mathtt{rec\ I{:}\tau{=}E} \Rightarrow [\mathtt{rec\ I{:}\tau{=}E/I}]\mathtt{E}$

Name binding:

$$[\text{let}] \quad \frac{\rho \vdash \mathtt{E_1} \Downarrow \mathtt{V_1} \qquad \rho \cup \mathtt{I{:}V_1} \vdash \mathtt{E_2} \Downarrow \mathtt{V_2}}{\rho \vdash (\mathtt{let\ I\ =\ E_1\ in\ E_2}) \Downarrow \mathtt{V_2}}$$

Numbers:

$[\text{num}] \qquad \rho \vdash \mathtt{N} \Downarrow \mathrm{N} \qquad \text{where N is semantic equivalent of } \mathtt{N}$

$$[+] \quad \frac{\rho \vdash \mathtt{E_1} \Downarrow \mathtt{V_1} \qquad \rho \vdash \mathtt{E_2} \Downarrow \mathtt{V_2}}{\rho \vdash (\mathtt{E_1{+}E_2}) \Downarrow (\mathtt{V_1} + \mathtt{V_2})}$$

$$[=] \quad \frac{\rho \vdash \mathtt{E_1} \Downarrow \mathtt{V_1} \qquad \rho \vdash \mathtt{E_2} \Downarrow \mathtt{V_2}}{\rho \vdash (\mathtt{E_1{=}E_2}) \Downarrow (\mathtt{V_1} = \mathtt{V_2})}$$

- difference between *syntactic* $\mathtt{N}$, $\mathtt{+}$, $\mathtt{=}$ and *semantic* N, $+$, $=$

Booleans:

$[\text{true}] \qquad \rho \vdash \mathtt{true} \Downarrow \text{true}$

$[\text{false}] \qquad \rho \vdash \mathtt{false} \Downarrow \text{false}$

$$[\text{not}_{\text{true}}] \quad \frac{\rho \vdash \mathtt{E} \Downarrow \text{true}}{\rho \vdash (\mathtt{not\ E}) \Downarrow \text{false}}$$

$$[\text{not}_{\text{false}}] \quad \frac{\rho \vdash \mathtt{E} \Downarrow \text{false}}{\rho \vdash (\mathtt{not\ E}) \Downarrow \text{true}}$$

$$[\text{if}_{\text{true}}] \quad \frac{\rho \vdash \mathtt{E_1} \Downarrow \text{true} \qquad \rho \vdash \mathtt{E_2} \Downarrow \mathtt{V}}{\rho \vdash (\mathtt{if\ E_1\ then\ E_2\ else\ E_3}) \Downarrow \mathtt{V}}$$

$$[\text{if}_{\text{false}}] \quad \frac{\rho \vdash \mathtt{E_1} \Downarrow \text{false} \qquad \rho \vdash \mathtt{E_3} \Downarrow \mathtt{V}}{\rho \vdash (\mathtt{if\ E_1\ then\ E_2\ else\ E_3}) \Downarrow \mathtt{V}}$$

Pairs:

$$[\times\ \text{intro}] \quad \frac{\rho \vdash \mathtt{E_1} \Downarrow \mathtt{V_1} \qquad \rho \vdash \mathtt{E_2} \Downarrow \mathtt{V_2}}{\rho \vdash (\mathtt{pair\ E_1\ E_2}) \Downarrow (\mathtt{V_1},\mathtt{V_2})}$$

$$[\times\ \text{elim}_{\text{first}}] \quad \frac{\rho \vdash \mathtt{E} \Downarrow (\mathtt{V_1},\mathtt{V_2})}{\rho \vdash (\mathtt{first\ E}) \Downarrow \mathtt{V_1}}$$

$$[\times\ \text{elim}_{\text{second}}] \quad \frac{\rho \vdash \mathtt{E} \Downarrow (\mathtt{V_1},\mathtt{V_2})}{\rho \vdash (\mathtt{second\ E}) \Downarrow \mathtt{V_2}}$$

Unions:

$$[+\ \text{intro}_{\text{first}}] \quad \frac{\rho \vdash \mathtt{E} \Downarrow \mathtt{V}}{\rho \vdash \left(\mathtt{mkfirst}_{\tau_1+\tau_2}\ \mathtt{E}\right) \Downarrow \langle\text{first=}\mathtt{V}\rangle}$$

$$[+\ \text{intro}_{\text{second}}] \quad \frac{\rho \vdash \mathtt{E} \Downarrow \mathtt{V}}{\rho \vdash \left(\mathtt{mksecond}_{\tau_1+\tau_2}\ \mathtt{E}\right) \Downarrow \langle\text{second=}\mathtt{V}\rangle}$$

$$[+\ \text{elim}_{\text{first}}] \quad \frac{\rho \vdash \mathtt{E} \Downarrow \langle\text{first=}\mathtt{V}\rangle \qquad \rho \cup \mathtt{I_1{:}V} \vdash \mathtt{E_1} \Downarrow \mathtt{V_1}}{\rho \vdash \begin{pmatrix}\mathtt{case\ E\ first(I_1)\ E_1} \\ \mathtt{second(I_2)\ E_2}\end{pmatrix} \Downarrow \mathtt{V_1}}$$

$$[+\ \text{elim}_{\text{second}}] \quad \frac{\rho \vdash \mathtt{E} \Downarrow \langle\text{second=}\mathtt{V}\rangle \qquad \rho \cup \mathtt{I_2{:}V} \vdash \mathtt{E_2} \Downarrow \mathtt{V_2}}{\rho \vdash \begin{pmatrix}\mathtt{case\ E\ first(I_1)\ E_1} \\ \mathtt{second(I_2)\ E_2}\end{pmatrix} \Downarrow \mathtt{V_2}}$$

Denotational semantics of PCF

Meaning function: $\mathcal{E}$: E → Env → Val
$\mathcal{E}[\![E]\!]\ \rho = v$ if
E denotes a function that,
given an environment ρ, yields a result value *v*

Domain of environment: $\rho \in \text{Env} \equiv \text{Id} \to \text{Val}$

Domain of values: $V \in \text{Val} \equiv (\bigcup_\tau \text{Val}_\tau) \cup \{\bot\}$
$\text{Val}_{\texttt{nat}} \equiv \mathrm{N}$
$\text{Val}_{\texttt{bool}} \equiv \{\mathrm{true}, \mathrm{false}\}$
$\text{Val}_{\tau1\to\tau2} \equiv \text{Val}_{\tau1} \to \text{Val}_{\tau2}$
$\text{Val}_{\tau1\times\tau2} \equiv (\text{Val}, \text{Val})$
$\text{Val}_{\tau1+\tau2} \equiv \langle\mathrm{first}{=}\text{Val}\rangle \cup \langle\mathrm{second}{=}\text{Val}\rangle$

All computations have a meaning;
⊥ is the meaning of a nonterminating computation
- need checks so that
  if subexpression means ⊥
  then enclosing expression also means ⊥
  - enclosing expression is **strict** in such subexpressions
  - all PCF expressions are strict in all their subexpressions except λ, `if`, and `case` (≈ the special forms of PCF)

Identifiers:
$\mathcal{E}[\![I]\!]\ \rho \equiv \rho(I)$

Functions:
$\mathcal{E}[\![\lambda I.E]\!]\ \rho \equiv \lambda v.\ \mathcal{E}[\![E]\!]\ (\rho \cup \{I{:}v\})$
$\mathcal{E}[\![E_1\ E_2]\!]\ \rho \equiv$
let $v_1 = \mathcal{E}[\![E_1]\!]\ \rho$, $v_2 = \mathcal{E}[\![E_2]\!]\ \rho$ in
if $v_1 = \bot$ or $v_2 = \bot$ then ⊥ else $v_1(v_2)$

Recursion:
$\mathcal{E}[\![\texttt{rec } I{:}\tau \texttt{ = } E]\!]\ \rho \equiv v$ where $v = \mathcal{E}[\![E]\!]\ (\rho \cup \{I{:}v\})$
$\equiv \mathrm{fix}(\lambda v.\ \mathcal{E}[\![E]\!]\ (\rho \cup \{I{:}v\}))$
- fix yields ⊥ if E is nonterminating

Name binding:
$\mathcal{E}[\![\texttt{let } I \texttt{ = } E_1 \texttt{ in } E_2]\!]\ \rho \equiv$
let $v = \mathcal{E}[\![E_1]\!]\ \rho$ in
if $v = \bot$ then ⊥ else $\mathcal{E}[\![E_2]\!]\ (\rho \cup \{I{:}v\})$

Numbers:
$\mathcal{E}[\![N]\!]\ \rho \equiv \mathrm{N}$, where N is semantic equivalent of N
$\mathcal{E}[\![E_1\texttt{+}E_2]\!]\ \rho \equiv (\mathcal{E}[\![E_1]\!]\ \rho) +_\bot (\mathcal{E}[\![E_2]\!]\ \rho)$
$\mathcal{E}[\![E_1\texttt{=}E_2]\!]\ \rho \equiv$ if $(\mathcal{E}[\![E_1]\!]\ \rho) =_\bot (\mathcal{E}[\![E_2]\!]\ \rho)$ then true else false

Booleans:
$\mathcal{E}[\![\texttt{true}]\!]\ \rho \equiv \mathrm{true}$
$\mathcal{E}[\![\texttt{false}]\!]\ \rho \equiv \mathrm{false}$
$\mathcal{E}[\![\texttt{not } E]\!]\ \rho \equiv$ if $(\mathcal{E}[\![E]\!]\ \rho) =_\bot \mathrm{true}$ then false else true
$\mathcal{E}[\![\texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3]\!]\ \rho \equiv$
if $(\mathcal{E}[\![E_1]\!]\ \rho) =_\bot \mathrm{true}$ then $\mathcal{E}[\![E_2]\!]\ \rho$ else $\mathcal{E}[\![E_3]\!]\ \rho$

Pairs:
$\mathcal{E}[\![\texttt{pair } E_1\ E_2]\!]\ \rho \equiv$
let $v_1 = \mathcal{E}[\![E_1]\!]\ \rho$, $v_2 = \mathcal{E}[\![E_2]\!]\ \rho$ in
if $v_1 = \bot$ or $v_2 = \bot$ then ⊥ else (v_1,v_2)
$\mathcal{E}[\![\texttt{first } E]\!]\ \rho \equiv$
let $v = \mathcal{E}[\![E]\!]\ \rho$ in
case *v* of ⊥: ⊥
of (v_1,v_2): v_1
$\mathcal{E}[\![\texttt{second } E]\!]\ \rho \equiv$
let $v = \mathcal{E}[\![E]\!]\ \rho$ in
case *v* of ⊥: ⊥
of (v_1,v_2): v_2

Unions:
$\mathcal{E}[\![\texttt{mkfirst}_{\tau1+\tau2}\ E]\!]\ \rho \equiv$
let $v = \mathcal{E}[\![E]\!]\ \rho$ in
if $v = \bot$ then ⊥ else $\langle\mathrm{first}{=}v\rangle$
$\mathcal{E}[\![\texttt{mksecond}_{\tau1+\tau2}\ E]\!]\ \rho \equiv$
let $v = \mathcal{E}[\![E]\!]\ \rho$ in
if $v = \bot$ then ⊥ else $\langle\mathrm{second}{=}v\rangle$
$\mathcal{E}[\![\texttt{case } E \texttt{ first(}I_1\texttt{) } E_1 \texttt{ second(}I_2\texttt{) } E_2]\!]\ \rho \equiv$
let $v = \mathcal{E}[\![E]\!]\ \rho$ in
case *v* of ⊥: ⊥
of $\langle\mathrm{first}{=}v_1\rangle$: $\mathcal{E}[\![E_1]\!]\ (\rho \cup \{I_1{:}v_1\})$
of $\langle\mathrm{second}{=}v_2\rangle$: $\mathcal{E}[\![E_2]\!]\ (\rho \cup \{I_2{:}v_2\})$

Meaning of fix

fix $\equiv$ limit$_{i \to \infty}$ fix$_i$

fix$_0$ F = $\perp$

fix$_i$ F = F(fix$_{i-1}$)

fix$_i$ F is the version of F that can call itself recursively i-1 times before failing (yielding $\perp$)

- fix$_i$ is a better approximation of the fixpoint of F than fix$_{i-1}$

fix F is the version of F that can call itself recursively an unbounded but finite number of times, yielding $\perp$ only if calling itself an infinite number of times

Domain theory [Dana Scott 69]

- gives a rigorous foundation based on lattices for fix and other aspects of denotation metalanguage

Adding side-effects to PCF

Add ML-style refs to PCF

```
E    ::= ...
       | ref E | ! E | E := E
       | E ; E
τ    ::= ... | ref τ
```

Typechecking rules:

[ref intro] $\dfrac{\Gamma \vdash \texttt{E}:\tau}{\Gamma \vdash \texttt{(ref E)}:\texttt{ref }\tau}$

[ref elim] $\dfrac{\Gamma \vdash \texttt{E}:\texttt{ref }\tau}{\Gamma \vdash \texttt{(! E)}:\tau}$

[ref assign] $\dfrac{\Gamma \vdash \texttt{E}_1:\texttt{ref }\tau \qquad \Gamma \vdash \texttt{E}_2:\tau}{\Gamma \vdash (\texttt{E}_1 \texttt{ := } \texttt{E}_2):\tau}$

[sequence] $\dfrac{\Gamma \vdash \texttt{E}_1:\tau_1 \qquad \Gamma \vdash \texttt{E}_2:\tau_2}{\Gamma \vdash (\texttt{E}_1 \texttt{ ; } \texttt{E}_2):\tau_2}$

Operational semantics, first try

Extend the Val domain to include values of type `ref` τ:

Val$_{\texttt{ref}\ \tau} \equiv \langle$Val$\rangle$

Semantic rules for references:

[ref intro] $\dfrac{\rho \vdash \texttt{E} \Rightarrow \texttt{V}}{\rho \vdash \texttt{(ref E)} \Rightarrow \langle \texttt{V} \rangle}$

[ref elim] $\dfrac{\rho \vdash \texttt{E} \Rightarrow \langle \texttt{V} \rangle}{\rho \vdash \texttt{(! E)} \Rightarrow \texttt{V}}$

[ref assign] $\dfrac{\rho \vdash \texttt{E}_1 \Rightarrow \langle \texttt{V}_1 \rangle \qquad \rho \vdash \texttt{E}_2 \Rightarrow \texttt{V}_2}{\rho \vdash (\texttt{E}_1 \texttt{ := } \texttt{E}_2) \Rightarrow \texttt{V}_2}$

[sequence] $\dfrac{\rho \vdash \texttt{E}_1 \Rightarrow \texttt{V}_1 \qquad \rho \vdash \texttt{E}_2 \Rightarrow \texttt{V}_2}{\rho \vdash (\texttt{E}_1 \texttt{ ; } \texttt{E}_2) \Rightarrow \texttt{V}_2}$

Where did the update go in [ref assign]?

Stores

To record side-effects that flow through the program as it runs, introduce a **store** $\sigma \in$ Store:Loc $\to$ Val, indexed by locations $\iota \in$ Loc

References now evaluate to locations: Val$_{\texttt{ref}\ \tau} \equiv$ Loc

Evaluation judgments now take a store as context and produce a new store as result

$\rho,\sigma \vdash \texttt{E} \Rightarrow \texttt{V},\sigma'$ means
in environment ρ and store σ, the expression `E` evaluates to value `V` producing new store σ'

Operational semantics, final try

Semantic rules for references:

$$[\text{ref intro}]\quad \frac{\rho, \sigma \vdash \mathtt{E} \Rightarrow \mathtt{V}, \sigma'}{\rho, \sigma \vdash (\mathtt{ref\ E}) \Rightarrow \iota, \sigma''}$$

where $\iota \notin \mathrm{dom}(\sigma') \wedge \sigma'' = \sigma' \cup \{\iota{:}\mathtt{V}\}$

$$[\text{ref elim}]\quad \frac{\rho, \sigma \vdash \mathtt{E} \Rightarrow \iota, \sigma'}{\rho, \sigma \vdash (\mathtt{!\ E}) \Rightarrow \mathtt{V}, \sigma'} \quad \text{if } \iota{:}\mathtt{V} \in \sigma'$$

$$[\text{ref assign}]\quad \frac{\rho, \sigma \vdash \mathtt{E}_1 \Rightarrow \iota, \sigma' \qquad \rho, \sigma' \vdash \mathtt{E}_2 \Rightarrow \mathtt{V}_2, \sigma''}{\rho, \sigma \vdash (\mathtt{E}_1\ \mathtt{:=}\ \mathtt{E}_2) \Rightarrow \mathtt{V}_2, \sigma'''}$$

where $\sigma''' = \sigma'' \cup \{\iota{:}\mathtt{V}_2\}$

$$[\text{sequence}]\quad \frac{\rho, \sigma \vdash \mathtt{E}_1 \Rightarrow \mathtt{V}_1, \sigma' \qquad \rho, \sigma' \vdash \mathtt{E}_2 \Rightarrow \mathtt{V}_2, \sigma''}{\rho, \sigma \vdash (\mathtt{E}_1\ \mathtt{;}\ \mathtt{E}_2) \Rightarrow \mathtt{V}_2, \sigma''}$$

Note that “threading” of store through hypotheses
specifies evaluation order

Some other rules, extended with store-passing

Identifiers:

$[\text{var}]\quad \rho, \sigma \vdash \mathtt{I} \Rightarrow \mathtt{V}, \sigma \qquad \text{if } \mathtt{I{:}V} \in \rho$

Functions:

$[\lambda]\quad \rho, \sigma \vdash (\lambda\mathtt{I.E}) \Rightarrow [(\lambda\mathtt{I.E}), \rho], \sigma$

- lambda captures environment ρ, but not store σ

$$[\text{app}]\quad \frac{\rho, \sigma \vdash \mathtt{E}_1 \Rightarrow [(\lambda\mathtt{I.E}), \rho'], \sigma' \qquad \rho, \sigma' \vdash \mathtt{E}_2 \Rightarrow \mathtt{V}_2, \sigma'' \qquad \rho' \cup \mathtt{I{:}V}_2, \sigma'' \vdash \mathtt{E} \Rightarrow \mathtt{V}, \sigma'''}{\rho, \sigma \vdash (\mathtt{E}_1\ \mathtt{E}_2) \Rightarrow \mathtt{V}, \sigma'''}$$

- when evaluating function body, use lambda’s captured env ρ’ but caller’s store σ’’

Recursion:

$$[\text{rec}]\quad \frac{\rho \cup \mathtt{I{:}V}, \sigma \vdash \mathtt{E} \Rightarrow \mathtt{V}, \sigma'}{\rho, \sigma \vdash (\mathtt{rec\ I{:}\tau\ =\ E}) \Rightarrow \mathtt{V}, \sigma'}$$

Denotational semantics

Meaning function: $\mathcal{E}: \mathtt{E} \to \text{Env} \to \text{Store} \to \text{Val}\times\text{Store}$

$\mathcal{E}[\![\mathtt{E}]\!]\ \rho\ \sigma = (v, \sigma')$ if
E denotes a function that,
given an environment ρ and a store σ,
yields a result value *v* and a new store σ’

Semantic rules for references:

$\mathcal{E}[\![\mathtt{ref\ E}]\!]\ \rho\ \sigma \equiv$
let $(v,\sigma') = \mathcal{E}[\![\mathtt{E}]\!]\ \rho\ \sigma$ in
if $v = \bot$ then $(\bot,\sigma')$ else $(\iota, \sigma' \cup \{\iota{:}v\})$
where $\iota \notin \mathrm{dom}(\sigma')$

$\mathcal{E}[\![\mathtt{!\ E}]\!]\ \rho\ \sigma \equiv$
let $(\iota,\sigma') = \mathcal{E}[\![\mathtt{E}]\!]\ \rho\ \sigma$ in $(\sigma(\iota), \sigma')$

$\mathcal{E}[\![\mathtt{E}_1\ \mathtt{:=}\ \mathtt{E}_2]\!]\ \rho\ \sigma \equiv$
let $(\iota,\sigma') = \mathcal{E}[\![\mathtt{E}_1]\!]\ \rho\ \sigma$, $(v,\sigma'') = \mathcal{E}[\![\mathtt{E}_2]\!]\ \rho\ \sigma'$ in
if $\iota = \bot$ or $v = \bot$ then $(\bot,\sigma'')$ else $(v, \sigma'' \cup \{\iota{:}v\})$

$\mathcal{E}[\![\mathtt{E}_1\ \mathtt{;}\ \mathtt{E}_2]\!]\ \rho\ \sigma \equiv$
let $(v,\sigma') = \mathcal{E}[\![\mathtt{E}_1]\!]\ \rho\ \sigma$ in $\mathcal{E}[\![\mathtt{E}_2]\!]\ \rho\ \sigma'$

Some other rules, extended with store-passing

Identifiers:

$\mathcal{E}[\![\mathtt{I}]\!]\ \rho\ \sigma \equiv (\rho(\mathtt{I}), \sigma)$

Functions:

$\mathcal{E}[\![\lambda\mathtt{I.E}]\!]\ \rho\ \sigma \equiv$
let $f = \lambda v.\ \lambda\sigma'.\ \mathcal{E}[\![\mathtt{E}]\!]\ (\rho \cup \{\mathtt{I}{:}v\})\ \sigma'$ in (f, σ)

$\mathcal{E}[\![\mathtt{E}_1\ \mathtt{E}_2]\!]\ \rho\ \sigma \equiv$
let $(v_1,\sigma') = \mathcal{E}[\![\mathtt{E}_1]\!]\ \rho\ \sigma$, $(v_2,\sigma'') = \mathcal{E}[\![\mathtt{E}_2]\!]\ \rho\ \sigma'$ in
if $v_1 = \bot$ or $v_2 = \bot$ then $(\bot,\sigma'')$ else $v_1(v_2)(\sigma'')$

Recursion:

$\mathcal{E}[\![\mathtt{rec\ I{:}\tau\ =\ E}]\!]\ \rho\ \sigma$
$\equiv (v,\sigma')$ where $(v,\sigma') = \mathcal{E}[\![\mathtt{E}]\!]\ (\rho \cup \{\mathtt{I}{:}v\})\ \sigma$
$\equiv \mathrm{fix}(\lambda v.\ \lambda\sigma'.\ \mathcal{E}[\![\mathtt{E}]\!]\ (\rho \cup \{\mathtt{I}{:}v\})\ \sigma')\ \sigma$

- the fixpoint produces a function of type Store→(Val×Store), which is applied to the initial store σ to get the result of $\mathcal{E}$