Proper function calls (i.e. use space for argument eval, not the call itself)

SECD-machine that models improper calls
↳ Dump ⟶ the previous state before the last function call

Control

(0 (λx.x)) — trying to call b as fun
(/ 1 0) — δ is undefined on input
(+ 1 (λx.x)) — call prim on fun

$eval(M) = b$ if $M \mapsto^* b$
or 'fun if $M \mapsto \lambda X.N$

{ eval undefined
1. divergence
2. stuckness

"Stuck" means no rule applies (β and δ) AND not ∈ V

Error-ISWIM

$M ::= \ldots \mid err_\ell$ $\quad \ell \in$ some set

Add error rules

$(b \; V) \mapsto err_\ell$ (partial)

old: $\delta : b \ldots \not\mapsto b$

new: $\delta : b \ldots \to b \cup err_\ell$ (total)

$(c^n \; V \ldots (\lambda X.M) \; K \ldots) \mapsto err_\ell$

$err_\ell \in V$? — if Y, then $(err_\ell \; (\lambda X.X))$
— if N, then how to reduce ↺

New context rule

$E[err_\ell] \mapsto err_\ell$

all old rules

$E[(\lambda X.M) \; v] \mapsto E[M[X \leftarrow V]]$
↑ unchanged

$\langle err_\ell, E, K \rangle \mapsto \langle err_\ell, \emptyset, ret \rangle$

$$eval(M) = \begin{cases} b & \text{if } M \mapsto^* b \\ \text{'fun} & \text{if } M \mapsto^* \lambda X.M \\ err_\ell & \text{if } M \mapsto^* err_\ell \end{cases}$$

Why is Error-ISWIM "right"?

$\forall M \in$ ISWIM,

1. $\text{eval}_v(M) \overset{A}{\Rightarrow} \quad \text{eval}_e(M) = A$
2. $\text{eval}_e(M) = A$ and $A \neq \text{err}_e \Rightarrow \text{eval}_v(M) = A$
3. $\text{eval}_e(M) = \text{err}_e$ then $\text{eval}_v(M)$ is undefined
   $(\exists N.\ M \to^* N$ and $!\exists L.\ N \mapsto L$ and $N \notin V)$
4. $\text{eval}_e(M)$ is undefined then $\text{eval}_v(M)$ is undefined
   $(\forall N.\ M \to^* N \Rightarrow \exists L.\ N \mapsto L)$

Error has more observations (ie $\simeq$ has smaller equiv classes)

If M and N are obs equiv $(M \simeq_v N)$

but not in error $(M \not\simeq_e N)$

iff $\exists C \in$ Context of error-iswim

1. $\exists L, k.\ \text{err}_L \neq \text{err}_K$ and $\text{eval}_e(C[M]) = \text{err}_L$ and $\text{eval}_e(C[N]) = \text{err}_K$

isomorphic

2. M is an err and N diverges
3. N diverges and M is an error

ex 1. $M = (0\ (\lambda X.X))$ $\quad N = (+\ 0\ (\lambda X.X))$

$M \simeq_v N$ but in error $C = \square$

$C[M] \mapsto \text{err}_\beta \quad C[N] \mapsto \text{err}_\delta$

$\Rightarrow M \not\simeq_e N$

ex 2. $M = \Omega$ $\quad N = (0\ (\lambda X.X))$

$M \simeq_v N$ but in error $C = \square$

diverges $\quad \text{err}_\beta$

$\Rightarrow M \not\simeq_e N$

Exn - ISWIM

throw exn (the value b)

$M ::= \ldots \mid (\text{throw } b)$

$\mid (\text{catch } M_1 \text{ with } (\lambda X. M_2))$ — catch

run M_1 if it errors run M_2 where X is the b that was thrown

Error-ISWIM : $E[(b\ V)] \mapsto E[err_\beta]$

Exn-ISWIM : $E[(b\ V)] \mapsto E[\text{throw } err_\beta]$

$E[\text{throw } b] \mapsto b$ (we lose the context)

does not appear

$E[\text{throw } b] \mapsto E'[b]$ where $E' = \text{search}(E)$

Define F as a "catchless" eval-ctxt

$F = \square \mid (F\ M) \mid (V\ F) \mid (o^n\ V \ldots F\ M \ldots)$

$E = \text{catch } E \text{ with } (\lambda X. M) \mid F$

lose

$E[\text{catch } F[\text{throw } b] \text{ with } (\lambda X. M)]$

$\mapsto E[(\lambda X. M)\ b]$

save

$F[\text{throw } b] \mapsto b$

$E[\text{catch } V \text{ with } (\lambda X. M)] \mapsto E[V]$

$K = \ldots \mid \text{catch}(E, \lambda X. M, K)$

$\langle \text{catch } M_1 \text{ with } (\lambda X. M_2), E, K \rangle \mapsto \langle M_1, E, \text{catch}(E, \lambda X. M_2, K) \rangle$

$\langle V, E, \text{catch}(E, \lambda X. M, K) \rangle \mapsto \langle V, E, K \rangle$

$\langle \text{throw } b, E, K \rangle \mapsto \langle b, \emptyset, \text{fun}(\text{clo}(\lambda X. M, E'), K') \rangle$

where $\text{catch}(E', \lambda X. M, K') = \text{search}(K)$

$\text{search}(\text{fun}(V, K)) = \text{search}(K)$ $\quad \text{search}(\text{arg}(E, N, K)) = \text{search}(K)$

$\text{search}(\text{catch}(E, \lambda X. M, K)) = \text{catch}(E, \lambda X. M, K)$

$\text{search}(\text{ret}) = \text{catch}(\emptyset, \lambda X. X, \text{ret})$