THE OPERATIONAL SEMANTICS
OF
MULTI-LANGUAGE SYSTEMS

OUTLINE

0. MOTIVATION: HOW DO WE INTEROP?

1. APPROACHING SEMANTICS
   (MATTHEWS + FINDLER '07, '09)

2. A SURPRISING APPLICATION
   (AHMED + BLUME '11)

3. A MAJOR STRESS TEST
   (PATTERSON ET AL. '17)

↓ "DISTANCE" BETWEEN INTEROP-ING LANGUAGES

HOW DO WE INTEROP?

- LANGUAGES ARE DIFFERENT (EVEN UNDER THE HOOD)
  - > CALLING CONVENTIONS
  - > DATA REPRESENTATIONS
  - > MEMORY LAYOUTS, STRATEGIES

- SO HOW DO WE USE THEM TOGETHER?
  - > "PROTOCOLS"
  - > FOREIGN INTERFACES
  - > COMMON RUNTIMES

- WHAT ARE THE DRAWBACKS?
  - > GLUE CODE
  - > LOSSY TRANSLATIONS (COARSE-GRAINED)
  - > BROKEN ABSTRACTIONS (unsafe{...})
  - > BAD TOOLING

PROBLEM: HOW DO WE REASON
ABOUT INTEROP?

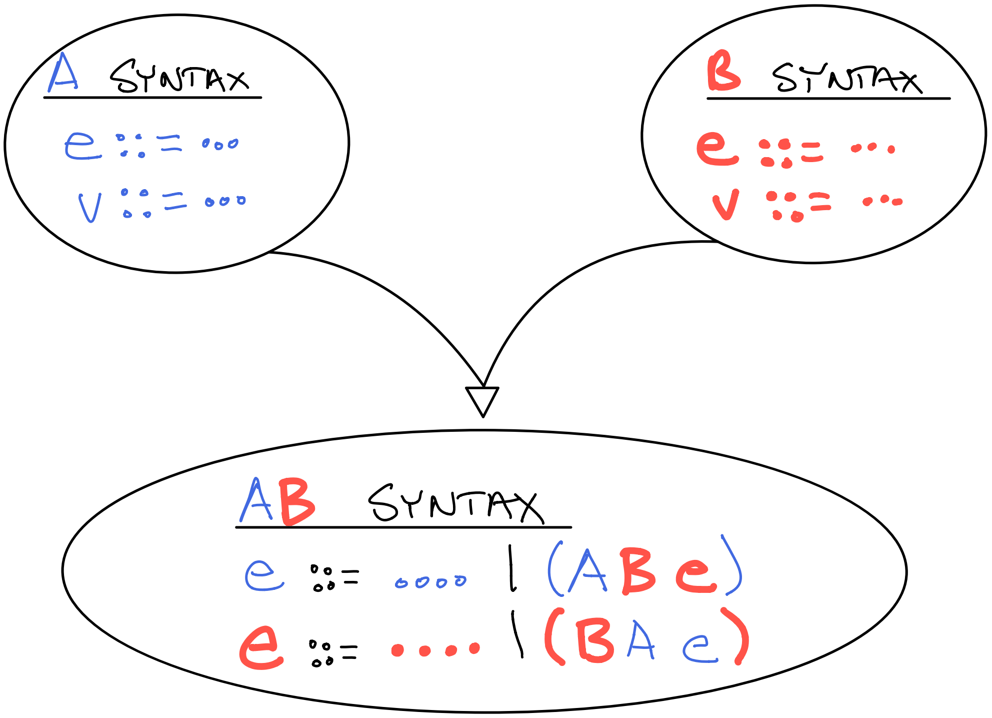
¯_(ツ)_/¯

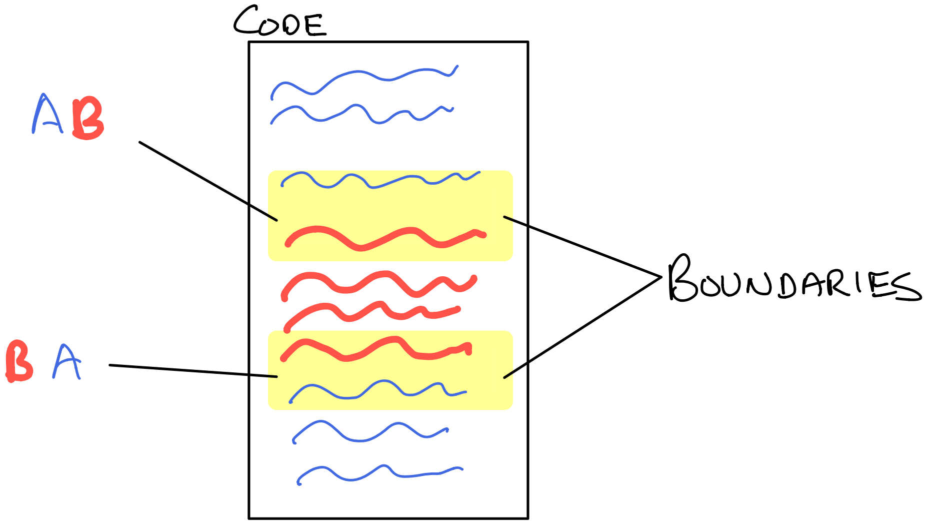
ONE APPROACH:

"OPERATIONAL SEMANTICS
FOR
MULTI-LANGUAGE PROGRAMS"
— MATTHEWS + FINDLER '07, '09

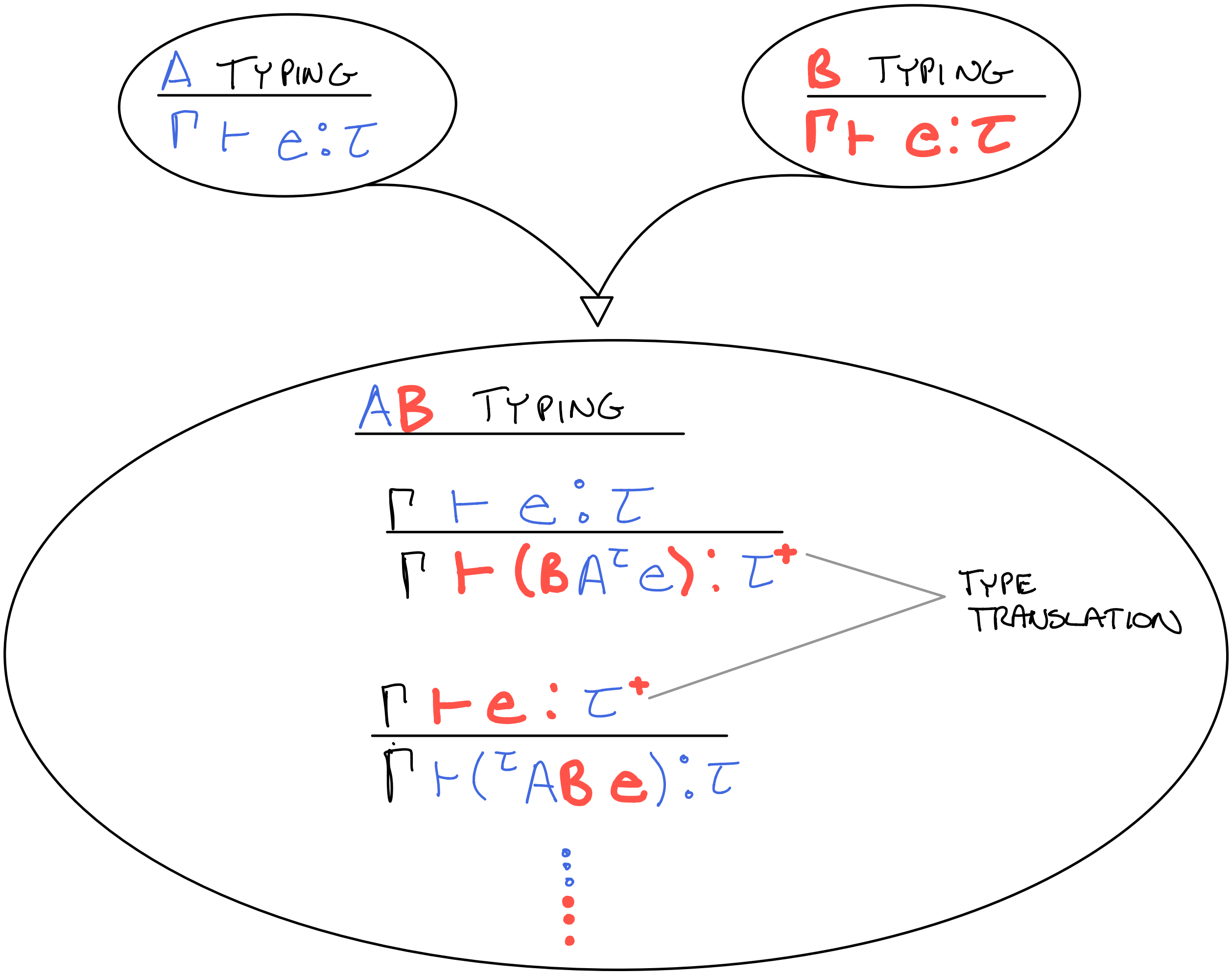
MF07: THE BIG PICTURE

TO INTEROPERATE LANGUAGES A + B,

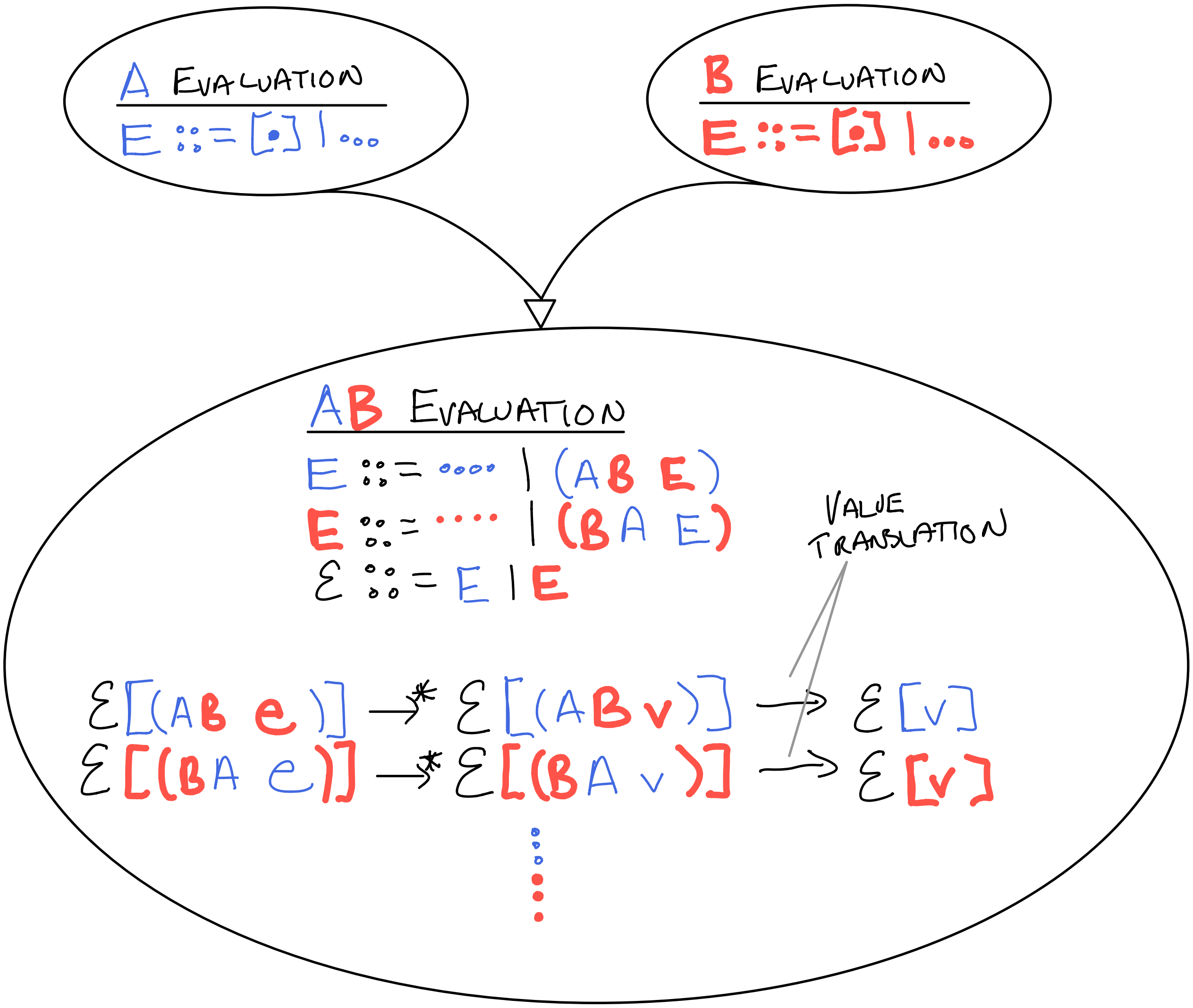




BOUNDARIES ≈ "CROSS-LANGUAGE CASTS"



"EVALUATE UNDER BOUNDARY, THEN TRANSLATE"



A EVALUATION

$E ::= [\bullet] \mid \dots$

B EVALUATION

$\mathbf{E} ::= [\bullet] \mid \dots$

A**B** EVALUATION

$E ::= \dots \mid (\mathrm{A}\mathbf{B}\ \mathbf{E})$

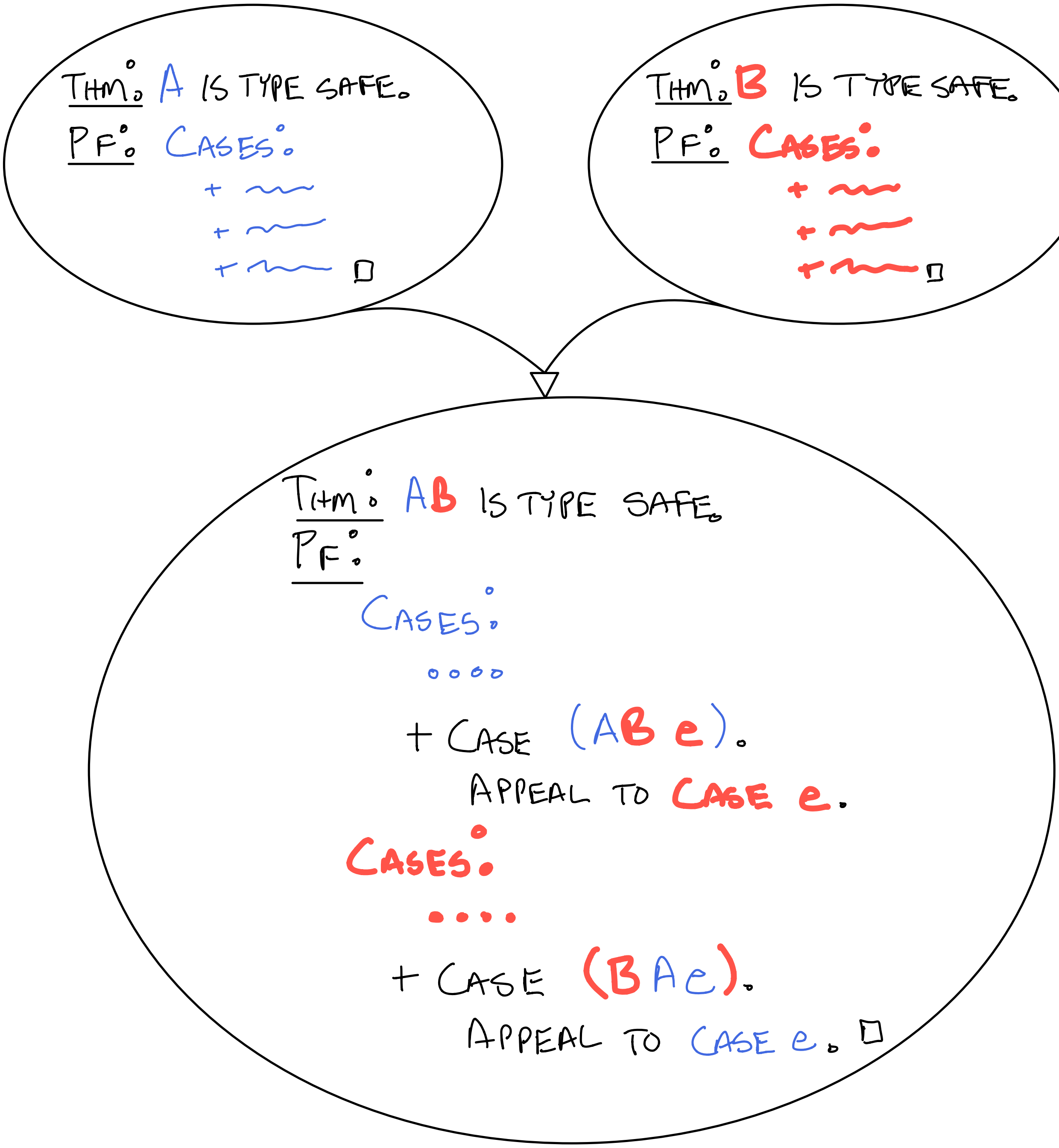
$\mathbf{E} ::= \dots \mid (\mathbf{B}\mathrm{A}\ E)$

$\mathcal{E} ::= E \mid \mathbf{E}$

$\mathcal{E}[(\mathrm{A}\mathbf{B}\ \mathbf{e})] \to^{*} \mathcal{E}[(\mathrm{A}\mathbf{B}\ \mathbf{v})] \to \mathcal{E}[v]$

$\mathcal{E}\mathbf{[}(\mathbf{B}\mathrm{A}\ e)\mathbf{]} \to^{*} \mathcal{E}\mathbf{[}(\mathbf{B}\mathrm{A}\ v)\mathbf{]} \to \mathcal{E}\mathbf{[v]}$

$\vdots$



- REUSE/REPURPOSE EXISTING META-THEORETIC TOOLS
  - > SUBJECT REDUCTION, LOGICAL RELATIONS, EQUIVALENCE, ETC
  - > DISCLAIMER: NOT ALWAYS STRAIGHTFORWARD!

MF07: ML-SCHEME (MORE LIKE STLC-λ)

$\tau^{+} \triangleq TST$ ("THE SCHEME TYPE")

$$\frac{\Gamma \vdash e : TST}{\Gamma \vdash ({}^{\tau}MS\ e) : \tau}$$

$$\frac{\Gamma \vdash e : \tau}{\Gamma \vdash (SM^{\tau} e) : TST}$$

RECALL: GRADUAL TYPING (OLEK)

MF07 WIP TALK

TOBIN-HOCHSDAT + FELLEISEN '06

MF07 PUBLISHED

MF09 PUBLISHED

WHAT HAPPENS OPERATIONALLY?

↳ MANY OPTIONS!

MF07: THE LUMP EMBEDDING

A FOREIGN VALUE IS A BLACKBOX LUMP

$\mathcal{E}[({}^{L}MS\ v)]$ M CAN'T TOUCH v

$\mathcal{E}[(SM^{L}\ v)]$ S CAN'T TOUCH v

UNLESS IT IS ITSELF A BOUNDARY

$$\mathcal{E}[({}^{\tau}MS\ (SM^{\tau}\ v))] \longrightarrow \mathcal{E}[v]$$

$$\mathcal{E}[(SM^{L}({}^{L}MS\ v))] \longrightarrow \mathcal{E}[v]$$

IN WHICH CASE BOUNDARIES MAY CANCEL

MF07: FOREIGN APPLY w/ LUMPS

$$\text{fapp} : L \to L \to L$$

$$\text{fapp}\ f\ x = {}^{L}MS\ \{\underbrace{(SM^{L} f)}_{\text{"}f\text{"}}\ \underbrace{(SM^{L} x)}_{\text{"}x\text{"}}\}$$

Ex:

(β)

$\text{fapp}\ ({}^{L}MS\ \text{add1})\ ({}^{L}MS\ 42)$

↳ ${}^{L}MS\ \{(SM^{L}({}^{L}MS\ \text{add1}))$
$(SM^{L}({}^{L}MS\ 42))\}$

(CANCEL)

↳ ${}^{L}MS\ \{\text{add1}$
$(SM^{L}({}^{L}MS\ 42))\}$

(CANCEL)

↳ ${}^{L}MS\ (\text{add1}\ 42)$

(β)

↳ ${}^{L}MS\ 43$

ACTUALLY CAN GET REALLY FAR (THEORETICALLY)
JUST WITH THIS!

MF07: THE NATURAL EMBEDDING

CAN WE USE FOREIGN VALUES NATIVELY?

$$SM^{N}\ 42 \longrightarrow 42$$
$${}^{N\to N}MS\ \mathrm{add1} \longrightarrow \text{"}\mathrm{add1}\text{"}$$

FIRST ATTEMPT: "ETA EXPANSION + BOUNDARIES"

$$\mathcal{E}[{}^{\tau_1\to\tau_2}MS\ f]$$
$$\hookrightarrow \mathcal{E}[\lambda(x:\tau_1).\ {}^{\tau_2}MS\ \{f\ \underbrace{(SM^{\tau_1}\ x)}_{\text{"}x\text{"}}\}]$$

PROBLEM: TOO TRUSTING! WHERE ARE THE CHECKS?

RECALL: HIGHER-ORDER CONTRACTS (CAMERON)
GRADUAL TYPING (OLEK)

MF07: GUARDS - FIRST ORDER

LITERAL NUMBERS

$$\mathcal{E}[\,{}^{N}M\,S\,G\,\bar{n}\,] \longrightarrow \mathcal{E}[\,\bar{n}\,]$$

OTHERWISE, $\mathcal{E}[\,{}^{N}M\,S\,G\,v\,] \to$ ERROR!

MF07: GUARDS - HIGHER ORDER

$$\mathcal{E}[{}^{\tau_1\to\tau_2}M\,S\,G\ \lambda x.e]$$

FIRST-ORDER TAG CHECK

$$\hookrightarrow \mathcal{E}[\lambda(x:\tau_1).\ \underbrace{({}^{\tau_2}M\,S\,G(\underbrace{(\lambda x.e)\ (G\,S\,M^{\tau_1}x)}_{\text{COMPUTE } v}))}_{\text{GUARD ENSURES } v \rightsquigarrow v \text{ BEHAVES LIKE } \tau_2}]$$

OTHERWISE, $\mathcal{E}[{}^{\tau_1\to\tau_2}M\,S\,G\ v] \to$ ERROR!

RECALL: HIGHER-ORDER CONTRACTS (CAMERON)

MF07: GUARDS IN ACTION

- $\mathcal{E}[{}^{N \to N}\mathrm{MSG}(\lambda x.\#t)]$

$\hookrightarrow \mathcal{E}[\lambda(x:N).\ {}^{N}\mathrm{MSG}\{(\lambda x.\#t)\ (\mathrm{GSM}^{N}\ x)\}]$

AS ABOVE

- $(f\ 42)$

(β)

$\hookrightarrow {}^{N}\mathrm{MSG}\{(\lambda x.\#t)\ (\mathrm{GSM}^{N}\ 42)\}$

(GSM^{N})

$\hookrightarrow {}^{N}\mathrm{MSG}\{(\lambda x.\#t)\ 42\}$

(β)

$\hookrightarrow {}^{N}\mathrm{MSG}\ \#t$

$({}^{N}\mathrm{MSG})$

$\hookrightarrow$ ERROR!

MF07: OTHER CONVERSION STRATEGIES

- SO FAR, TYPE-DIRECTED (E.G., ${}^{\tau}MS$)
- CONVERSION STRATEGY CAN BE DECOUPLED FROM TYPES!

EX: HANDLE S EXCEPTIONS FROM M

$$\mathcal{E}[{}^{\mathbb{N}!}MS\ \bar{n}] \longrightarrow \mathcal{E}[\bar{n}]$$

$$\mathcal{E}[{}^{\mathbb{N}!}MS\ \text{ERR!}] \longrightarrow \mathcal{E}[0]$$

SENTINEL

"CONVERSION STRATEGY"

MORE GENERALLY, DEFINE κ, $\lfloor\cdot\rfloor : \kappa \to \tau$

$$\kappa ::= \tau \mid \mathbb{N}! \mid \ldots$$

$$\lfloor \mathbb{N}! \rfloor = \mathbb{N}$$

$$\lfloor \mathbb{N} \rfloor = \mathbb{N}$$

$$\vdots$$

$$\frac{\Gamma \vdash e : T_{ST}}{\Gamma \vdash ({}^{\kappa}MS\ e) : \lfloor \kappa \rfloor} \qquad \frac{\Gamma \vdash e : \lfloor \kappa \rfloor}{\Gamma \vdash (SM^{\kappa}\ e) : T_{ST}}$$

MF07: RECAP

BOUNDARY TERMS

$$\mathcal{E}[({}^{\tau}AB\ e)] \to^{*} \mathcal{E}[({}^{\tau}AB\ v)] \to \mathcal{E}[v]$$

- INFLUENCED GRADUAL TYPING

- STRONG CONNECTION WITH CONTRACTS:
  - DECOUPLE GUARDS FROM BOUNDARIES
    - ↳ REPLACE W/ CONTRACTS (SHOWN IN PAPER)
  - C.F. GRAY ET AL. '05

- LOTS OF CHOICES FOR BOUNDARIES
  - WHAT/HOW DO WE TRANSLATE?
  - WHAT/WHEN DO WE CHECK?

- CASE STUDY: ML - SCHEME (STLC - λ)
  - TYPING: STATIC - DYNAMIC
  - TWO HIGH-LEVEL, SURFACE LANGUAGES
  - BOTH USE DIRECT-STYLE CONTROL FLOW

"AN EQUIVALENCE PRESERVING
CPS TRANSLATION
VIA MULTI-LANGUAGE SEMANTICS"
— AHMED + BLUME '11

AB11: CPS

"AN EQUIVALENCE-PRESERVING CPS TRANSLATION VIA MULTI-LANGUAGE SEMANTICS"

$$(\tau_1 \to \tau_2)^+ \triangleq \tau_1^+ \times (\tau_2^+ \to \text{ANS}) \to \text{ANS}$$

CONTINUATION

Ex:

$$\begin{array}{l}
\lambda(f : \mathbb{1} \to \mathbb{N},\ g : \mathbb{1} \to \mathbb{N}).\\
\quad f() + g()
\end{array}$$

$$\Big\downarrow \llbracket \cdot \rrbracket_{\text{CPS}}$$

$$\begin{array}{l}
\lambda(f, g : \mathbb{1} \times (\mathbb{N} \to \text{ANS}) \to \text{ANS},\ k : \mathbb{N} \to \text{ANS}).\\
\quad (f()\ (\lambda(x : \mathbb{N}).\\
\qquad (g()\ (\lambda(y : \mathbb{N}).\\
\qquad\quad \text{let } z = x + y\\
\qquad\quad \text{in } k\ z))))
\end{array}$$

"RETURN" WITH CONTINUATION APPLIED TO VALUES

AB11: EQUIVALENCE PRESERVATION

"AN EQUIVALENCE-PRESERVING CPS TRANSLATION VIA MULTI-LANGUAGE SEMANTICS"

$$e_1 \approx e_2 \;\Rightarrow\; \underbrace{[\![e_1]\!]^S_T \approx [\![e_2]\!]^S_T}_{\text{COMPILED TERMS}}$$

WHY? PROGRAMMER CAN SAFELY REASON IN S
WITHOUT KNOWING T, $[\![\cdot]\!]^S_T$

EX: REFACTORING e SHOULDN'T BREAK $[\![e]\!]^S_T$

$$\begin{tikzcd}
\lambda(f:\mathbb{1}\to\mathbb{N},\, g:\mathbb{1}\to\mathbb{N}).\; f() + g() \arrow[rr, dotted, bend left, "\text{REFACTOR}"] \arrow[d, squiggly, "{[\![\cdot]\!]_{CPS}}"'] & \approx_{STLC} & \lambda(f:\mathbb{1}\to\mathbb{N},\, g:\mathbb{1}\to\mathbb{N}).\; g() + f() \arrow[d, squiggly, "{[\![\cdot]\!]_{CPS}}"'] \\
e_1 & \approx & e_2
\end{tikzcd}$$

(HOPEFULLY!)

AB11: THE PROBLEM

$$e_1 \triangleq \lambda(f, g).\ f() + g() \quad \simeq_{STLC} \quad e_2 \triangleq \lambda(f, g).\ g() + f()$$

$$\llbracket \cdot \rrbracket_{CPS} \rightsquigarrow \qquad\qquad\qquad \llbracket \cdot \rrbracket_{CPS} \rightsquigarrow$$

$$\lambda(f, g, k).\ (f\ ()\ (\lambda x.\ (g\ ()\ (\lambda y.\ \ldots)))) \quad \not\simeq \quad \lambda(f, g, k).\ (g\ ()\ (\lambda x.\ (f\ ()\ (\lambda y.\ \ldots))))$$

$$C_{BAD} \triangleq \lambda k.\ \begin{array}{l} \text{let } f = \lambda(\_,\_).\ k\ 1 \\ \phantom{\text{let }} g = \lambda(\_,\_).\ k\ 2 \\ \text{in } [\cdot]\ f\ g\ k \end{array}$$

IGNORE
"THEIR OWN"
CONTINUATIONS

$$C_{BAD}[\llbracket e_1 \rrbracket](id) \rightarrow^* 1 \not\simeq C_{BAD}[\llbracket e_2 \rrbracket](id) \rightarrow^* 2$$

AB11: THE PROBLEM, MORE BROADLY

GOAL: $e_1 \approx e_2 \Rightarrow [\![e_1]\!] \boldsymbol{\approx} [\![e_2]\!]$

MORE EXPLICITLY,

$$\boxed{\forall C.\ C[e_1] \approx C[e_2] \Rightarrow \forall \mathbf{C}.\ \mathbf{C}\boldsymbol{[}[\![e_1]\!]\boldsymbol{]} \boldsymbol{\approx} \mathbf{C}\boldsymbol{[}[\![e_2]\!]\boldsymbol{]}}$$

UNIVERSALS ARE HARD, SO USUALLY WE DO THIS:

$$\boxed{\exists \mathbf{C}.\ \mathbf{C}\boldsymbol{[}[\![e_1]\!]\boldsymbol{]} \boldsymbol{\not\approx} \mathbf{C}\boldsymbol{[}[\![e_2]\!]\boldsymbol{]} \Rightarrow \exists C.\ C[e_1] \not\approx C[e_2]}$$

TO DO SO, DEFINE BACK TRANSLATION $\mathbf{C} \twoheadrightarrow C$

PROBLEM: IF $\mathbf{T}$ IS MORE POWERFUL THAN S, BACK TRANSLATION ISN'T ALWAYS POSSIBLE!

Ex: $\mathbf{C_{BAD}}$ CAN'T BE BACK TRANSLATED TO STLC

AB11: THE SOLUTION

IDEA: CHANGE TYPE TRANSLATION τ^+ SO THAT $[\![e]\!]$ DOESN'T TYPE CHECK IN BAD CONTEXTS C

$$(\tau_1 \to \tau_2)^+ \triangleq \forall\alpha.\, \tau_1^+ \times (\tau_2^+ \to \alpha) \to \alpha$$

INTUITION: ENSURE THAT CONTINUATION $\tau_2^+ \to \alpha$ IS ACTUALLY USED — NO OTHER WAY TO RETURN α!

$$\begin{array}{l} C_{BAD} \triangleq \lambda k. \\ \quad \text{let } f = \lambda(\_,\_).\ k\ 1 \\ \quad\quad\;\; g = \lambda(\_,\_).\ k\ 2 \\ \quad \text{in } [\cdot]\ f\ g\ k \end{array}$$

DON'T HAVE TYPE $(\mathbb{1} \to \mathbb{N})^+$

AB11: PROOF BY MULTI-LANGUAGE

"AN EQUIVALENCE-PRESERVING CPS TRANSLATION VIA MULTI-LANGUAGE SEMANTICS"

$$\begin{tikzcd}
e_1 \approx e_2 : \tau \arrow[r, Rightarrow, "\textcircled{1}"] \arrow[d, Rightarrow, dotted, "\text{TOO HARD!}"'] & e_1 \approx_{ST} e_2 : \tau \arrow[d, Rightarrow, "\textcircled{2}"] \\
{[\![e_1]\!] \approx [\![e_2]\!] : \tau^{+}} & {[\![e_1]\!] \approx_{ST} [\![e_2]\!] : \tau^{+}} \arrow[l, Rightarrow, "\textcircled{3}"']
\end{tikzcd}$$

① BACK TRANSLATION
② VIA "COMPILER CORRECTNESS" $(e \approx_{ST} ({}^{\tau}ST\,[\![e]\!]))$
③ "EASY" BECAUSE $T \subset ST$

- ST LOGICAL RELATION SIMPLER THAN CROSS-LANGUAGE
- ISOLATING ① + ② CRITICAL
- BOUNDARY CANCELLATION REQUIRED

AB11: TS BOUNDARY

$$\frac{\Gamma \vdash e : \tau}{\Gamma \vdash (TS^{\tau}\, e) : \tau^{+}}$$

RECALL: $(\tau_1 \to \tau_2)^{+} \triangleq \forall \alpha.\, \tau_1^{+} \times (\tau_2^{+} \to \alpha) \to \alpha$

SYNTACTIC RESTR.
ENFORCES CTRL FLOW

$$\mathcal{E}[\text{let } y = TS^{\tau_1 \to \tau_2} f \text{ in } e]$$

$$\hookrightarrow \mathcal{E}[e[y \mapsto f]]$$

WHERE $f = \lambda[\alpha](x : \tau_1^{+}, k : \tau_2^{+} \to \alpha).$

$\quad \text{let } z : \tau_2^{+} = TS^{\tau_2}(f\ ({}^{\tau_1}ST\ x))$

$\quad \text{in } k\ z$

AB11: ST BOUNDARY

$$\frac{\Gamma \vdash e : \tau^{+}}{\Gamma \vdash ({}^{\tau}ST\, e) : \tau}$$

RECALL: $\forall \alpha.\, \tau_1^{+} \times (\tau_2^{+} \to \alpha) \to \alpha \;=\; (\tau_1 \to \tau_2)^{+}$

$$\mathcal{E}\left[\, {}^{\tau_1 \to \tau_2}ST\; f \,\right]$$

$$\hookrightarrow \mathcal{E}\left[\, \lambda(x:\tau_1).\, {}^{\tau_2}ST\, (\text{let } x : \tau_1^{+} = TS^{\tau_1}\, x \;\; \text{in } f\, [\tau_2^{+}]\; x\; \text{id}) \,\right]$$

"JUST RETURN RESULT"

RECAP: AB11

USE MULTI-LANGUAGE TO RELATE e + $[\![e]\!]^{S}_{T}$

- MF07 TECHNIQUE HELPS WHEN $T > S$ IN POWER

- CASE STUDY: WIDER GAP BETWEEN LANGUAGES
  > DIFF. CONTROL FLOW: DIRECT - 1ST CLASS CONTINUATIONS
  > NONTRIVIAL TYPE TRANSLATION

- PAVED WAY FOR OPEN-WORLD COMPILER CORRECTNESS PROOFS

> THM: $e : \tau \rightsquigarrow e : \tau^{+} \Rightarrow e \approx_{ST} ({}^{\tau}ST\, e) : \tau$

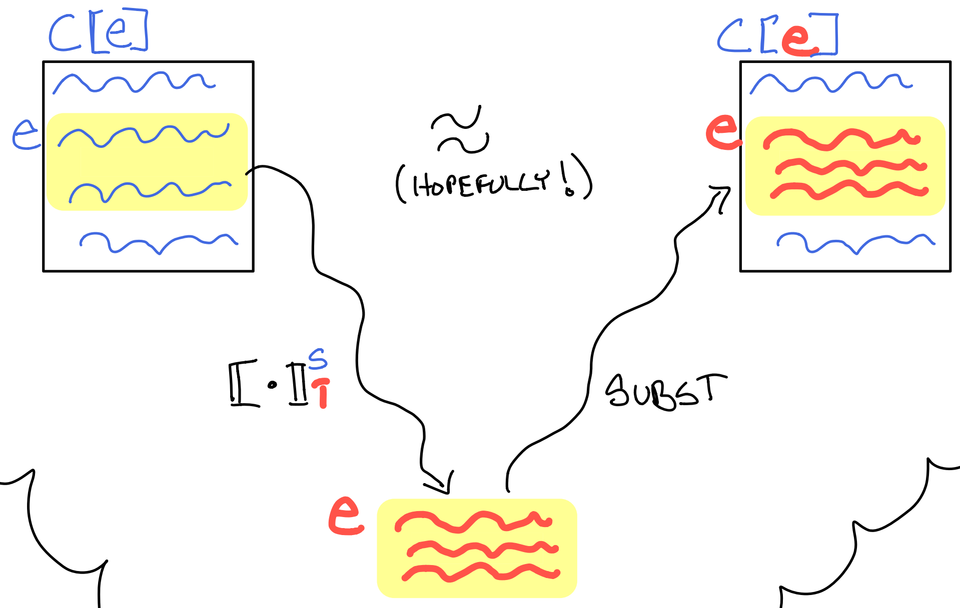
> COR: $e : \tau \rightsquigarrow e : \tau^{+} \Rightarrow e \approx_{ST} (TS^{\tau}\, e) : \tau^{+}$
PF: BY BOUNDARY CANCELLATION. □

> COR: $e_1 \approx_{ST} e_2 : \tau \Rightarrow [\![e_1]\!] \approx_{ST} [\![e_2]\!] : \tau^{+}$
PF: EASY ☺ □

"FunTAL: Reasonably Mixing
A
Functional Language
with
[Typed] Assembly"
— Patterson et al. '17

PPDA 17: MOTIVATION

RECALL: JIT COMPILATION (EMILY)



USE MULTI-LANG SEMANTICS TO REASON!

MAYBE,

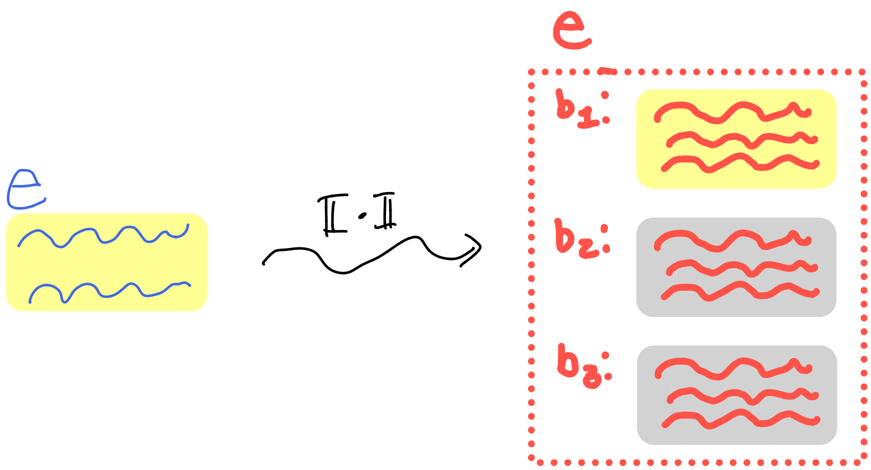
$$e \rightsquigarrow e \Rightarrow e \approx_{ST} (ST\,e)$$

PPDA 17: WHAT IF T = ASSEMBLY?

FOLLOWING RECIPE:

$$\mathcal{E}[F\,T\,e] \longrightarrow^{*} \mathcal{E}[F\,T\,v] \longrightarrow \mathcal{E}[v]$$

WHAT EVEN ARE THESE?

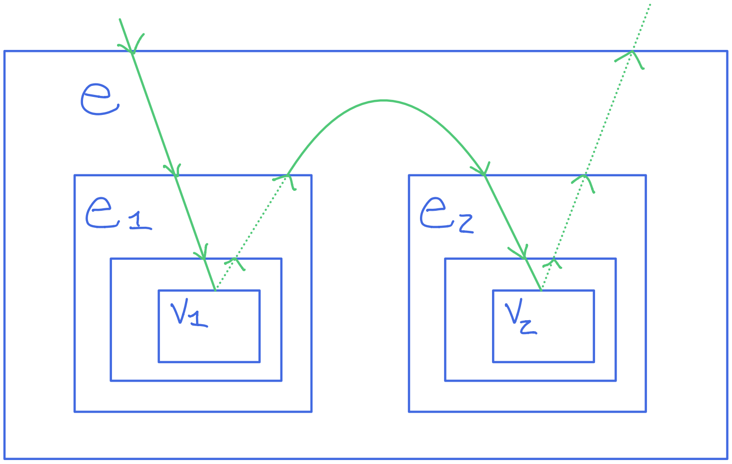


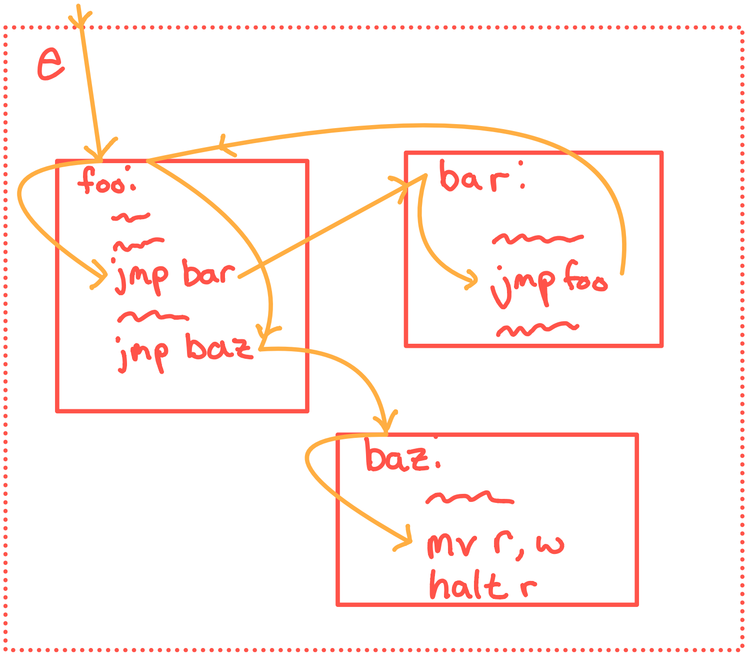
COMPONENT $\quad e ::= (I, H)$

INITIAL INSTRUCTION SEQUENCE

HEAP FRAGMENT WITH OTHER BASIC BLOCKS

PPDA17: CONTROL FLOW





- SERIOUS MISMATCH!
- DON'T WANT TO CHANGE CONTROL FLOW OF EITHER
- USE RICH TYPE SYSTEM TO GUIDE BOUNDARIES

PPDA 17: A TASTE OF TAL

BUILDING ON TAL WORK (MORRISETT ET AL. '98, '02)

$$\boxed{\underbrace{\Psi}_{\text{HEAP TYPING}} ; \overbrace{\Delta}^{\text{TYPE ENV (STANDARD)}} ; \underbrace{\chi}_{\text{REG TYPING}} ; \overbrace{\sigma}^{\text{STACK TYPING}} ; q \vdash e : \overbrace{\tau}^{\text{"RESULT" TYPE}} ; \underbrace{\sigma'}_{\text{NEW STACK TYPING}}}$$

RETURN MARKER

WHAT DO WE DO WHEN e IS DONE?

$$q ::= \overbrace{\underbrace{r}_{\text{AT REGISTER}} \mid \underbrace{i}_{\text{AT STACK INDEX}}}^{\text{CALL CONTINUATION}} \mid \overbrace{\text{end}\{\tau ; \sigma'\}}^{\text{HALT}}$$

PPDA17: FT BOUNDARY (SIMPLIFIED)

$$v ::= \text{halt}\ \tau\ \{r\}$$

TYPE OF "RESULT"

WHERE IT'S LOCATED

CURRENT STATE OF MEMORY

$$\langle M \mid E[\,{}^{\tau}FT(\text{halt}\ \tau^{+}\{r\}, \_)\,] \rangle$$

$$\hookrightarrow \langle M' \mid E[v] \rangle$$

WHERE $^{\tau}FT(M.R(r), M) = (v, M')$

LOOK UP VALUE IN REGISTER r

TRANSLATION OF $M.R(r)$

MEMORY CAN CHANGE DURING TRANSLATION!

PPDA17: TF BOUNDARY

$$\langle M \mid \mathcal{E}[\text{import } r, {}^{\sigma}\mathrm{TF}^{\tau}\, v; I] \rangle$$

REGISTER TO IMPORT INTO

VALUE TO BE IMPORTED

$$\hookrightarrow \langle M' \mid \mathcal{E}[\text{mv } r, v; I] \rangle$$

LOAD v INTO r

$$\text{IF } \mathrm{TF}^{\tau}(v, M) = (v, M')$$

TRANSLATION OF v

AGAIN, MEMORY CAN CHANGE DURING TRANSLATION!

PPDA17: RECAP

TAL COMPONENTS (l, H) + RETURN MARKERS &
$\Rightarrow$ **COMPATABILITY AT BOUNDARIES**

- CASE STUDY: WIDER GAP BETWEEN LANGUAGES
  > CONTROL FLOW: DIRECT — UNSTRUCTURED
  > MUTABILITY: IMMUTABLE — MUTABLE

- DIDN'T DEFINE $[\![\cdot]\!]^{F}_{T}$

- NO CORRECTNESS PROP LIKE

$$e \rightsquigarrow \mathbf{e} \not\Rightarrow e \approx_{FT} (FT\, \mathbf{e})$$

  > MAY NOT ALWAYS WANT THIS!
  INTUITIVELY: T CAN'T MAKE USE OF EXTRA POWER

SUMMARY

BOUNDARIES

$$\mathcal{E}[({}^{\tau}AB\ e)] \rightarrow^{*} \mathcal{E}[({}^{\tau}AB\ v)] \rightarrow \mathcal{E}[v]$$

	MF07	AB11	PPDA17
LANG	"SCHEME" "ML"	STLC SYS F	STLC^{+} TAL
TYPES	DYNAMIC STATIC	STATIC STATIC	STATIC STATIC
CONTROL FLOW	DIRECT DIRECT	DIRECT CPS	DIRECT ??

- DESIGN CHOICES
  - > WHAT / HOW DO WE TRANSLATE?
  - > WHAT / WHEN DO WE CHECK?
  - > WHAT EQUIVALENCES DO WE BREAK / PRESERVE?

- INFLUENCED BY CONTRACTS
- LEGACY IN
  - > GRADUAL TYPING
  - > SECURE COMPILATION

WHAT NOW?

- OTHER MF07 APPS
  - DEPENDENT TYPES (OSERA ET AL. '12)
  - LINEAR TYPES (SCHERER ET AL. '18)
- NEVER IMPLEMENTED AT SCALE
- MF07 REQUIRES BESPOKE SPEC FOR EVERY MIX
  - WITH n LANGUAGES, n^2 SPECS
  - DOESN'T SCALE TO FULL ECOSYSTEM
- LINKING TYPES? (DANIEL + AMAL)
  - EACH LANG SPECIFIES ONLY ITS SIDE OF BOUNDARY
  - EXPLICITLY STATE WHERE EQV. CAN BREAK

END
OF
TALK

~~~

OVERFLOW SLIDES
BELOW
⇓
~~~

MF07: PARAMETRIC POLYMORPHISM

PROBLEM: SCHEME MAY NOT RESPECT PARAMETRICITY

Ex: $\forall\alpha.\alpha\to\alpha$ MSG(λx.

```
(if (nat? x)
    (add1 x)
    x))
```

SOLUTION: LUMPS + CONVERSION STRATEGY

$$K ::= \tau \mid \underbrace{\langle \beta ; \tau \rangle}_{\text{“MASK” } \tau \text{ AS } \beta} \mid \ldots$$

$$\lfloor \langle \beta ; \tau \rangle \rfloor = \tau$$

$$\vdots$$

$$\mathcal{E}[(\Lambda\alpha.e)\ \tau\ ] \to \mathcal{E}[e[\alpha \mapsto \langle \beta ; \tau \rangle]] \quad (\text{FRESH } \beta)$$

Ex: $\mathrm{GSM}^{\langle \beta ; \mathbb{N} \rangle}\ v$ IS NOT nat?, BUT

$$^{\langle \beta ; \mathbb{N} \rangle}\mathrm{MSG}(\mathrm{GSM}^{\langle \beta ; \mathbb{N} \rangle}\ v) \to v : \lfloor \langle \beta ; \mathbb{N} \rangle \rfloor = \mathbb{N}$$

AB11: OPEN WORLD CC ⇒ EQU PRES

THM: $e : \tau \rightsquigarrow e : \tau^{+} \Rightarrow e \approx_{ST} ({}^{\tau}ST\, e) : \tau$

COR: $e_1 \approx_{ST} e_2 : \tau \Rightarrow [\![e_1]\!] \approx_{ST} [\![e_2]\!] : \tau^{+}$

PF:

FIRST,

$$\begin{tikzcd}
e_1 \arrow[r, phantom, "\overset{\text{ASUM.}}{\approx}"] \arrow[d, phantom, "\text{THM.}\ \wr\wr"] & e_2 \arrow[d, phantom, "\wr\wr\ \text{THM.}"] \\
({}^{\tau}ST\, [\![e_1]\!]) \arrow[r, phantom, "\overset{\text{TRANS.}}{\approx}"] & ({}^{\tau}ST\, [\![e_2]\!])
\end{tikzcd}$$

THEN,

$$({}^{\tau}ST\, [\![e_1]\!]) \approx_{ST} ({}^{\tau}ST\, [\![e_2]\!])$$

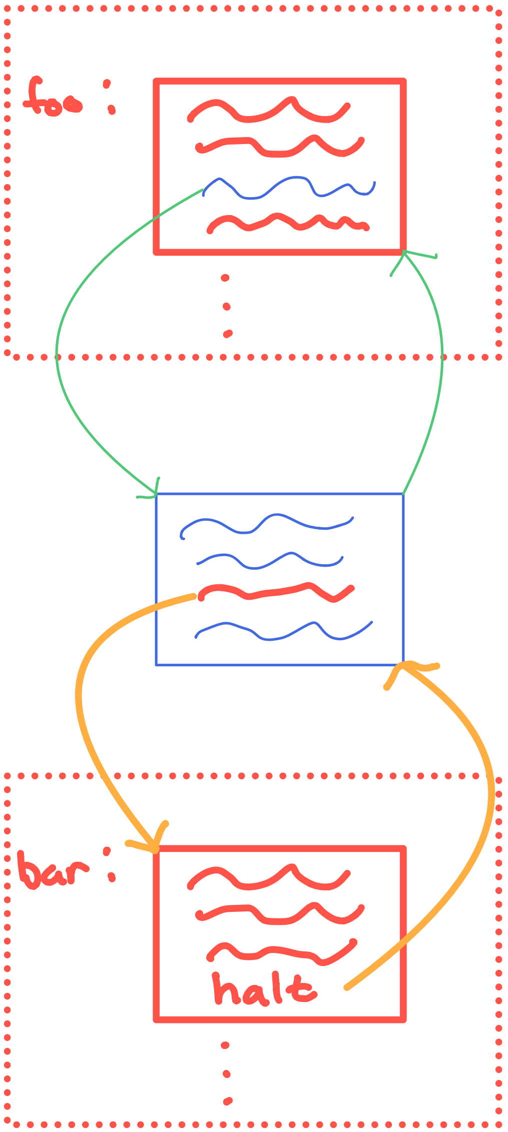
DEF EQU $\Lsh$ $$(TS^{\tau}({}^{\tau}ST\, [\![e_1]\!])) \approx_{ST} (TS^{\tau}({}^{\tau}ST\, [\![e_1]\!]))$$

BOUNDARY CANCELLATION $\Lsh$ $$[\![e_1]\!] \approx_{ST} [\![e_2]\!]$$

$\square$

PPDA17: STACK PROTECTION

RECALL: CALLER-SAVED VS. CALLEE-SAVED



HOW DO WE ENSURE THAT **bar** DOESN'T CLOBBER **foo**'S DATA?