Verifying an Open Compiler from System F to Assembly

James T. Perconti & Amal Ahmed

Northeastern University

Compiler Correctness

$$s \leadsto t \quad \Longrightarrow \quad s \approx t$$

compiles to

same meaning

Semantics-preserving compilation

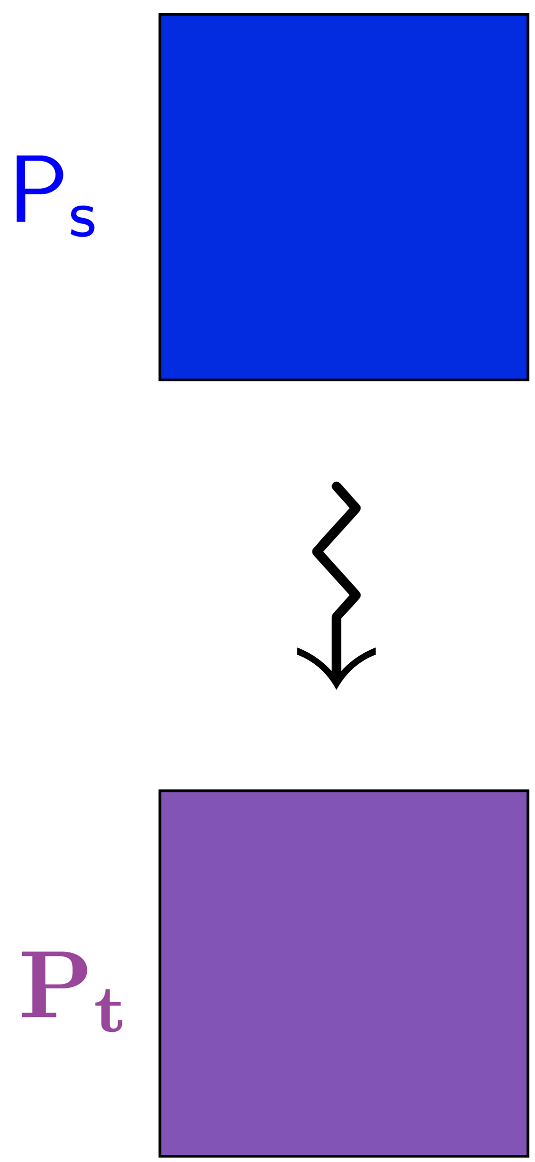
$$s \leadsto t \implies s \approx t$$

compiles to

same meaning

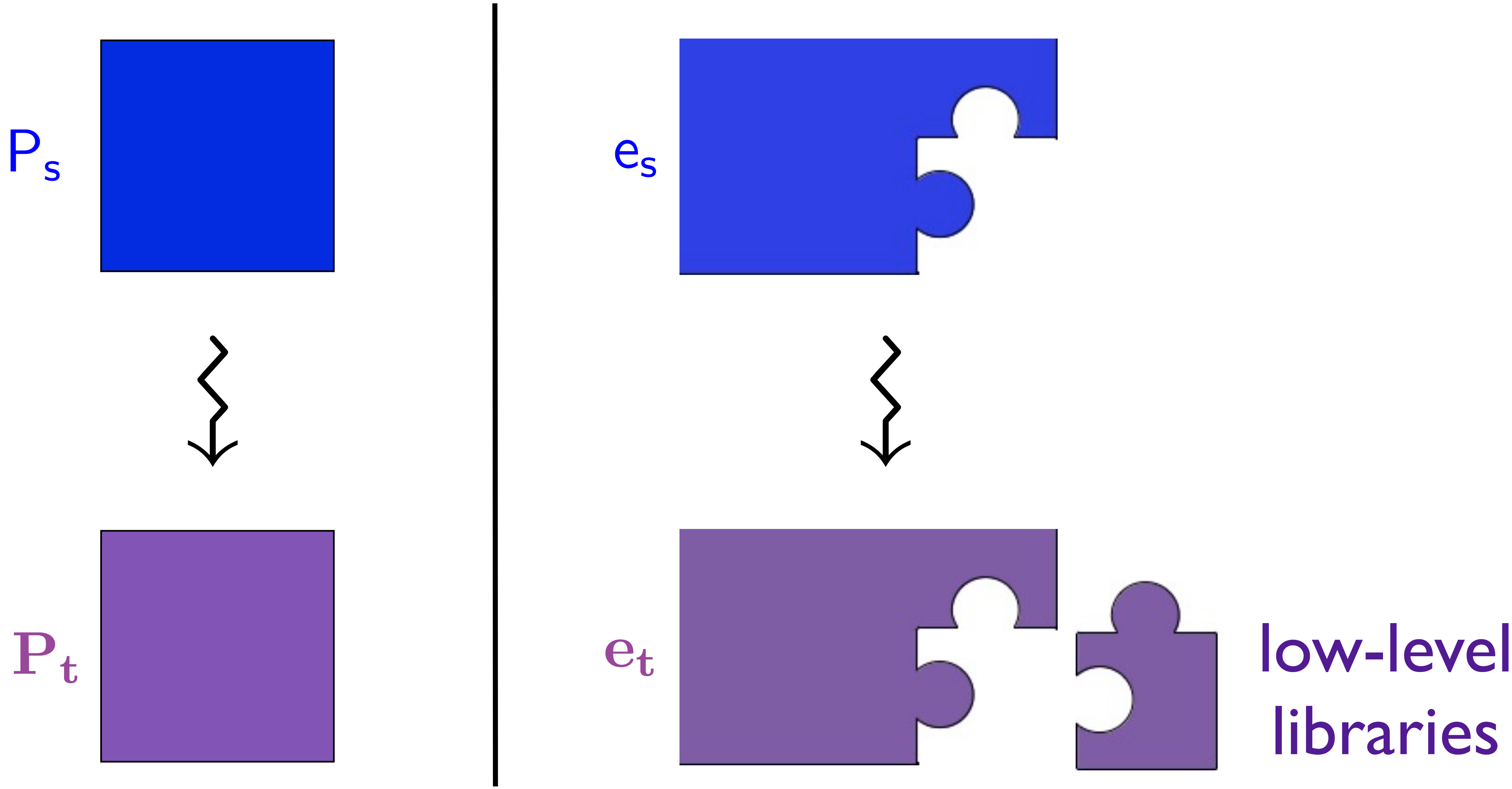
Problem: Closed-World Assumption

Correct compilation guarantee only applies to **whole** programs!



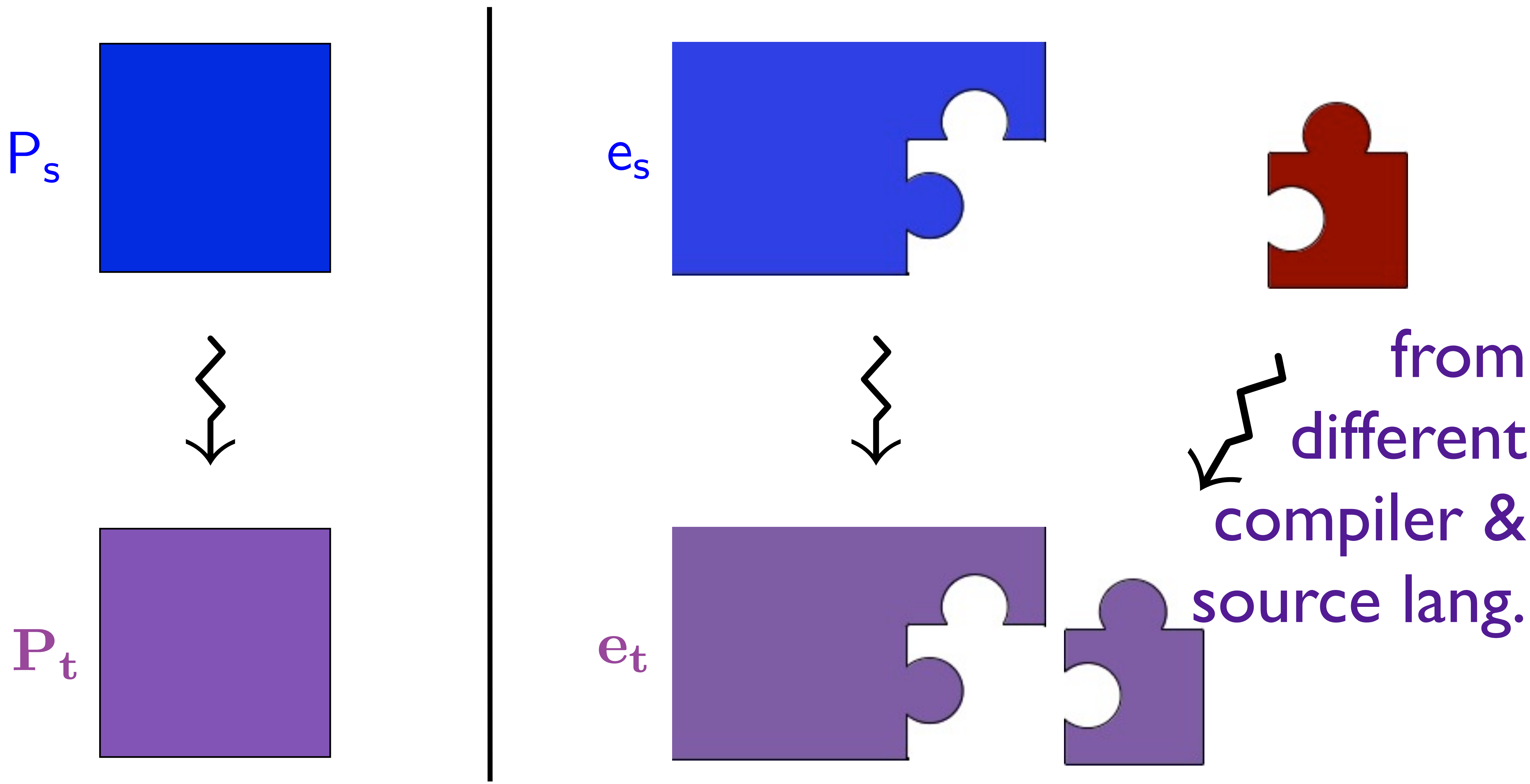
Problem: Closed-World Assumption

Correct compilation guarantee only applies to whole programs!



Problem: Closed-World Assumption

Correct compilation guarantee only applies to **whole** programs!



Why Whole Programs?

$$s \rightsquigarrow t \qquad \Longrightarrow \qquad s \approx t$$

expressed how?

Why Whole Programs?

$$P_s \rightsquigarrow P_t \quad \Longrightarrow \quad P_s \approx P_t$$

expressed how?

CompCert

$$P_s \longmapsto \ldots \longmapsto P_s^i \longmapsto P_s^{i+1} \longmapsto \ldots$$

$$P_t \longmapsto \ldots \longmapsto P_t^j \longmapsto^* P_t^{j+n} \longmapsto \ldots$$

Verifying Open Compilers: Benton-Hur

$$\mathsf{x} : \tau' \vdash \mathsf{e_s} : \tau \rightsquigarrow \mathbf{e_t} \quad \Longrightarrow \quad \mathsf{x} : \tau' \vdash \mathsf{e_s} \simeq \mathbf{e_t} : \tau$$

[Benton-Hur, ICFP'09, MSR'10]

[Hur-Dreyer POPL'11]

Verifying Open Compilers: Benton-Hur

$$x : \tau' \vdash e_s : \tau \rightsquigarrow \mathbf{e_t} \implies x : \tau' \vdash e_s \simeq \mathbf{e_t} : \tau$$

cross-language logical relation

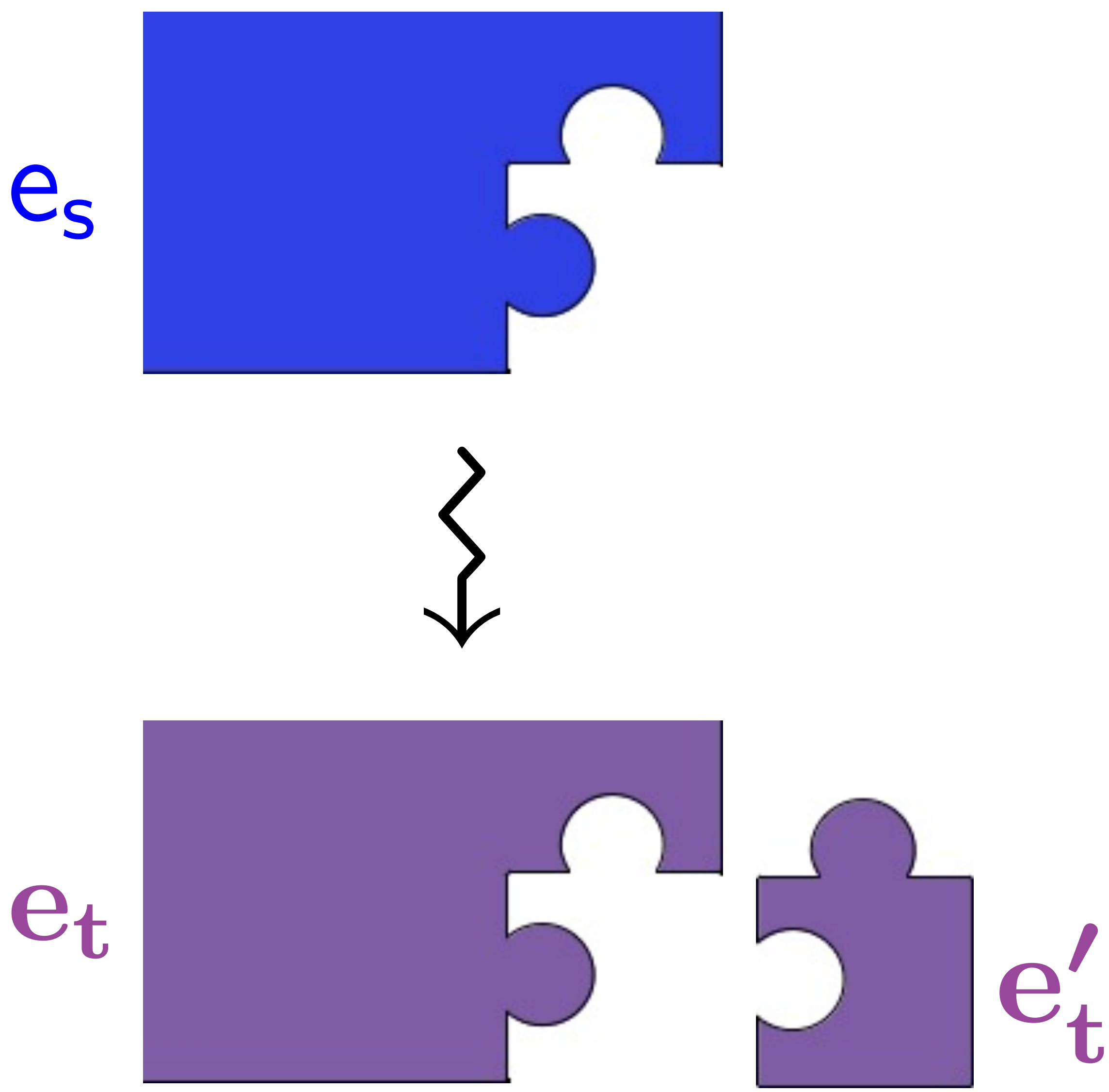
$$\forall v_s, \mathbf{v_s}.\ \vdash v_s \simeq \mathbf{v_t} : \tau' \implies \vdash e_s[v_s/x] \simeq \mathbf{e_t} \ddagger \mathbf{v_t} : \tau$$

[Benton-Hur, ICFP'09, MSR'10]

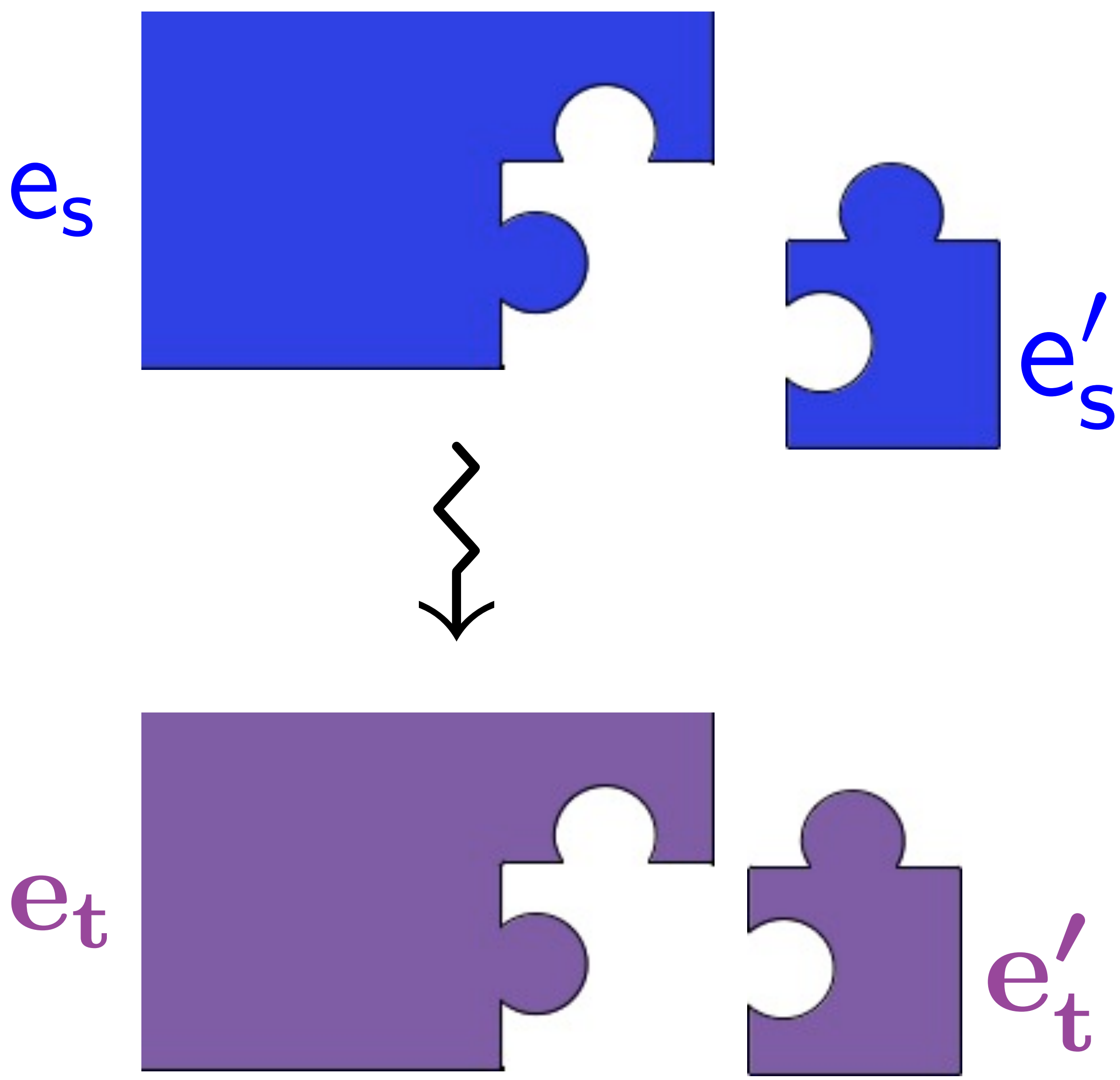
[Hur-Dreyer POPL'11]

Benton-Hur: Problem 1

Have $x : \tau' \vdash e_s \simeq e_t : \tau$

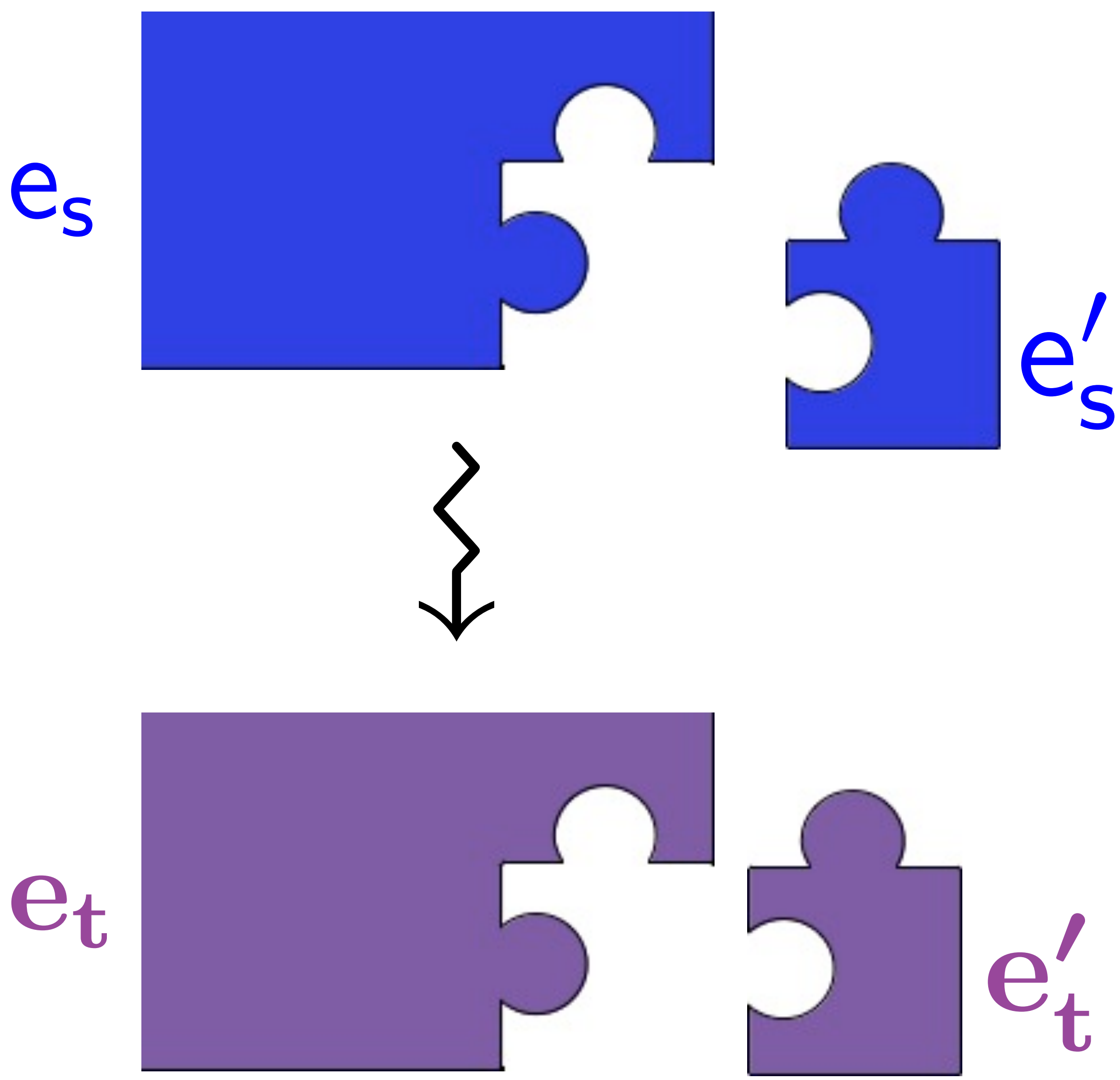


Benton-Hur: Problem 1



Have $x : \tau' \vdash e_s \simeq \mathbf{e_t} : \tau$

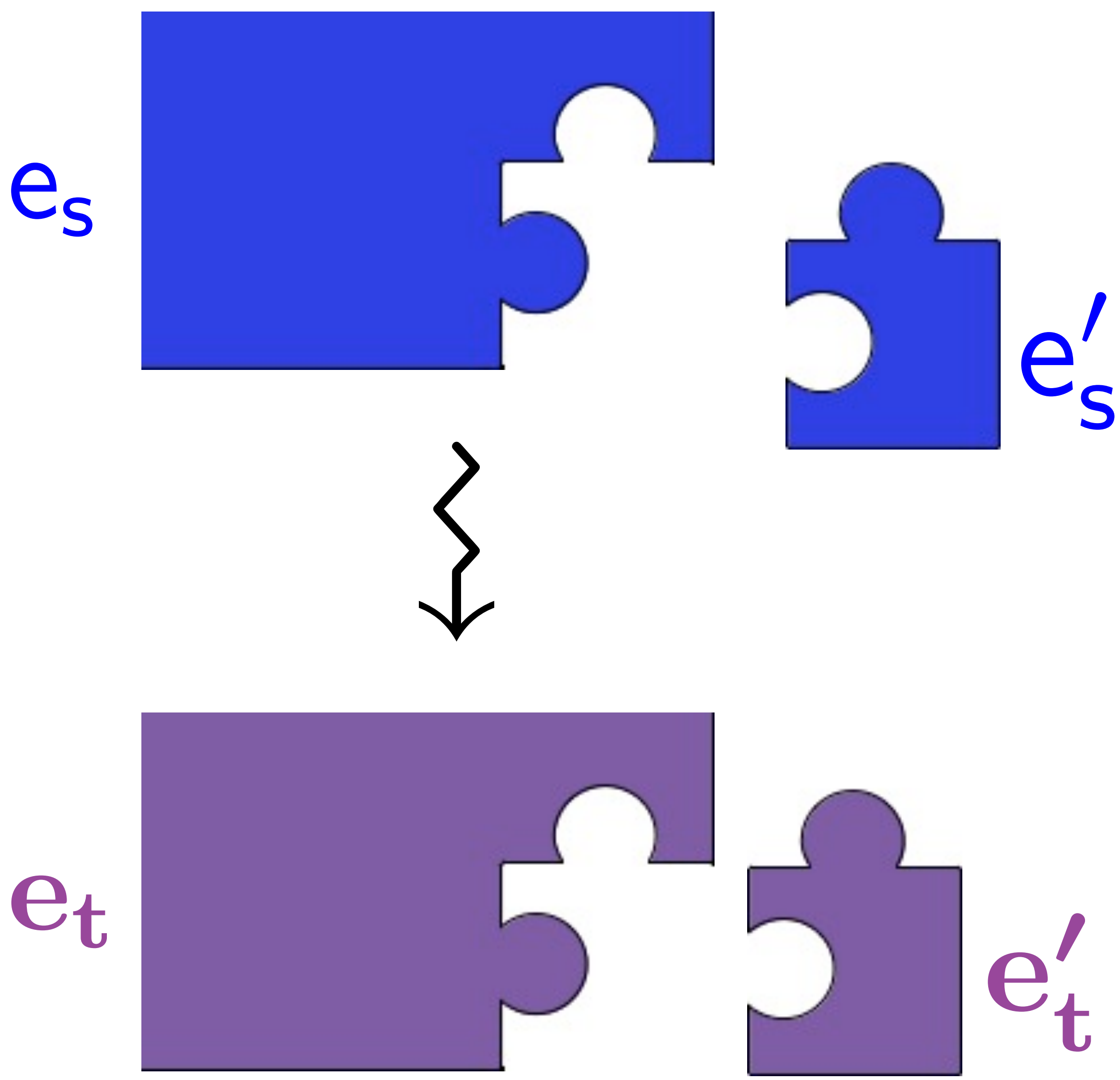
Benton-Hur: Problem 1



Have $x : \tau' \vdash e_s \simeq \mathbf{e_t} : \tau$

$\vdash e'_s \simeq \mathbf{e'_t} : \tau'$

Benton-Hur: Problem 1

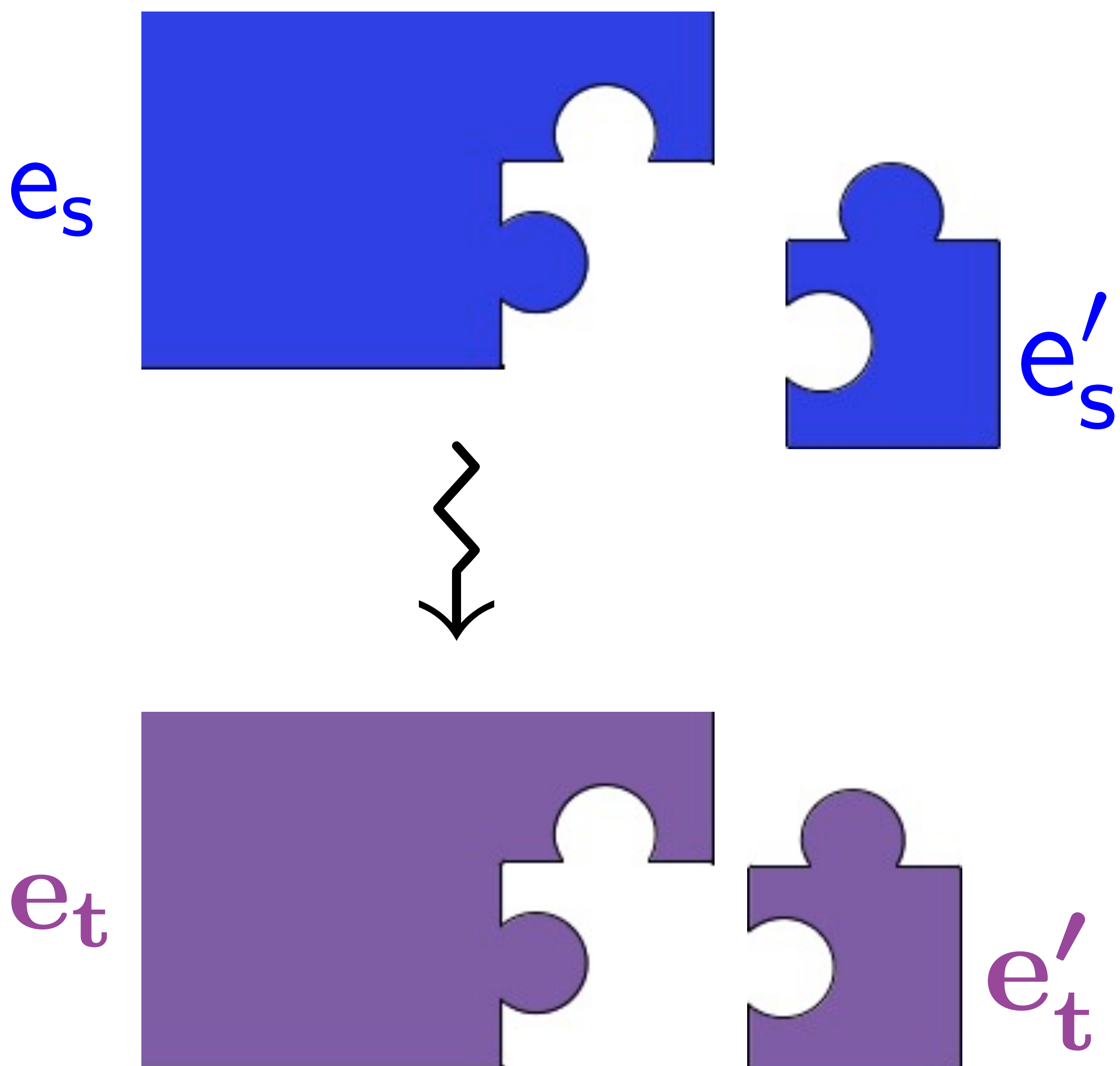


Have $x : \tau' \vdash e_s \simeq e_t : \tau$

$\vdash e'_s \simeq e'_t : \tau'$

$\therefore \; \vdash e_s[e'_s/x] \simeq e_t \ddagger e'_t : \tau$

Benton-Hur: Problem 1



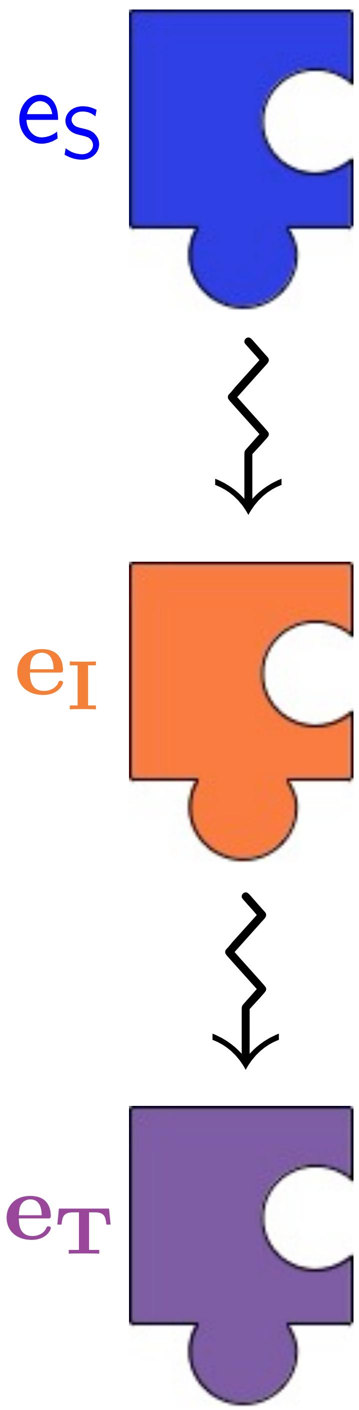
Have $x : \tau' \vdash e_s \simeq e_t : \tau$

$\vdash e'_s \simeq e'_t : \tau'$

$\therefore \ \vdash e_s[e'_s/x] \simeq e_t \ddagger e'_t : \tau$

Need to come up with e'_s
-- not feasible in practice!

Benton-Hur: Problem 2



$$\left.\begin{array}{l} e_S \simeq e_I \\ \\ e_I \simeq e_T \end{array}\right\} \overset{?}{\Longrightarrow} e_S \simeq e_T$$

Transitivity for single-lang. logical relation?

$$\left.\begin{array}{l} e_1 \approx e_2 \\ \\ e_2 \approx e_3 \end{array}\right\} \overset{?}{\Longrightarrow} \quad e_1 \approx e_3$$

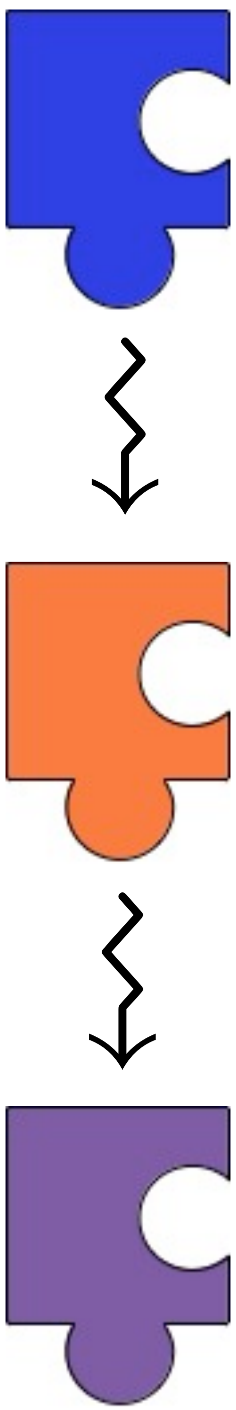
Transitivity for single-lang. logical relation?

$$\left.\begin{array}{l} e_1 \approx e_2 \\ \\ e_2 \approx e_3 \end{array}\right\} \overset{?}{\Longrightarrow} \quad e_1 \approx e_3$$

- Prove: $\quad e \approx e' \iff e \approx^{ctx} e'$

- $\approx^{ctx}$ is transitive, $\therefore$ $\approx$ is transitive

Benton-Hur: Problem 2



e_S

e_I

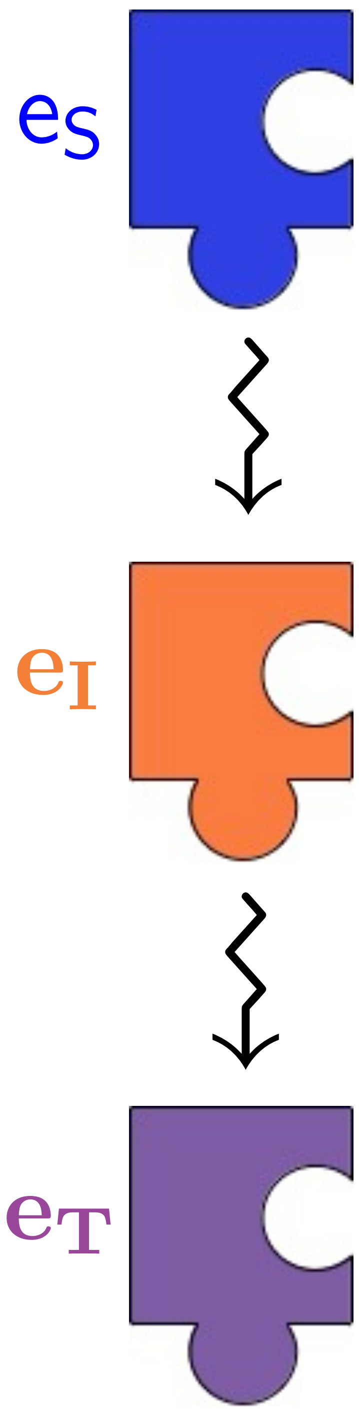
e_T

$$\left.\begin{array}{l} e_S \simeq e_I \\ \\ e_I \simeq e_T \end{array}\right\} \overset{?}{\Longrightarrow} e_S \simeq e_T$$

Transitivity for single-lang. logical relation:

$$e \approx e' \iff e \approx^{ctx} e'$$

Benton-Hur: Problem 2



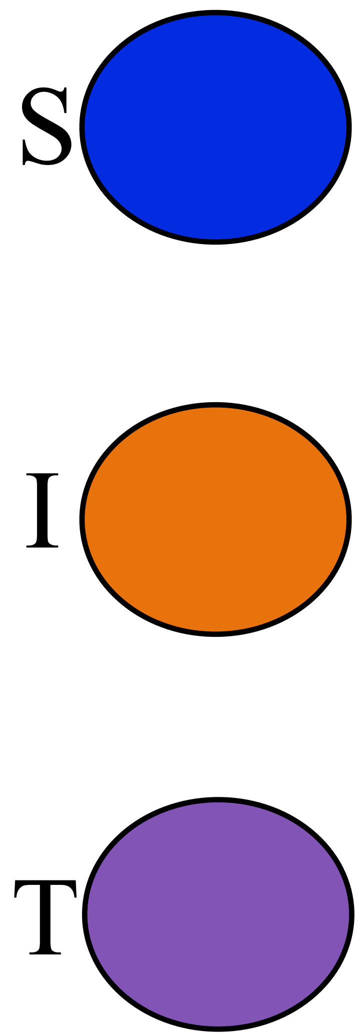
$$\left.\begin{array}{l} e_S \simeq e_I \\ \\ e_I \simeq e_T \end{array}\right\} \overset{?}{\Longrightarrow} e_S \simeq e_T$$

cross-language relation;
no definition of ctx. equiv

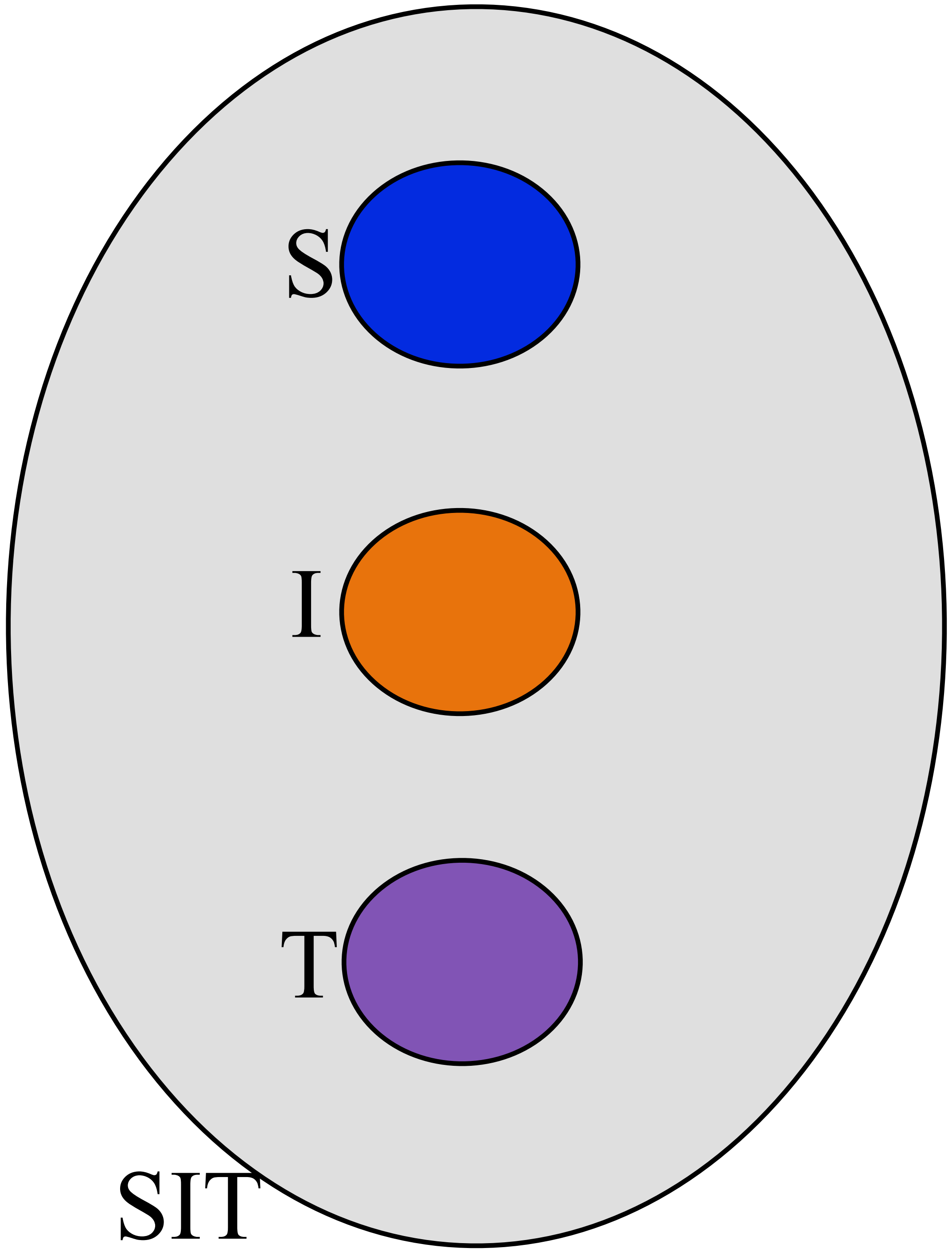
Transitivity for single-lang. logical relation:

$$e \approx e' \iff e \approx^{ctx} e'$$

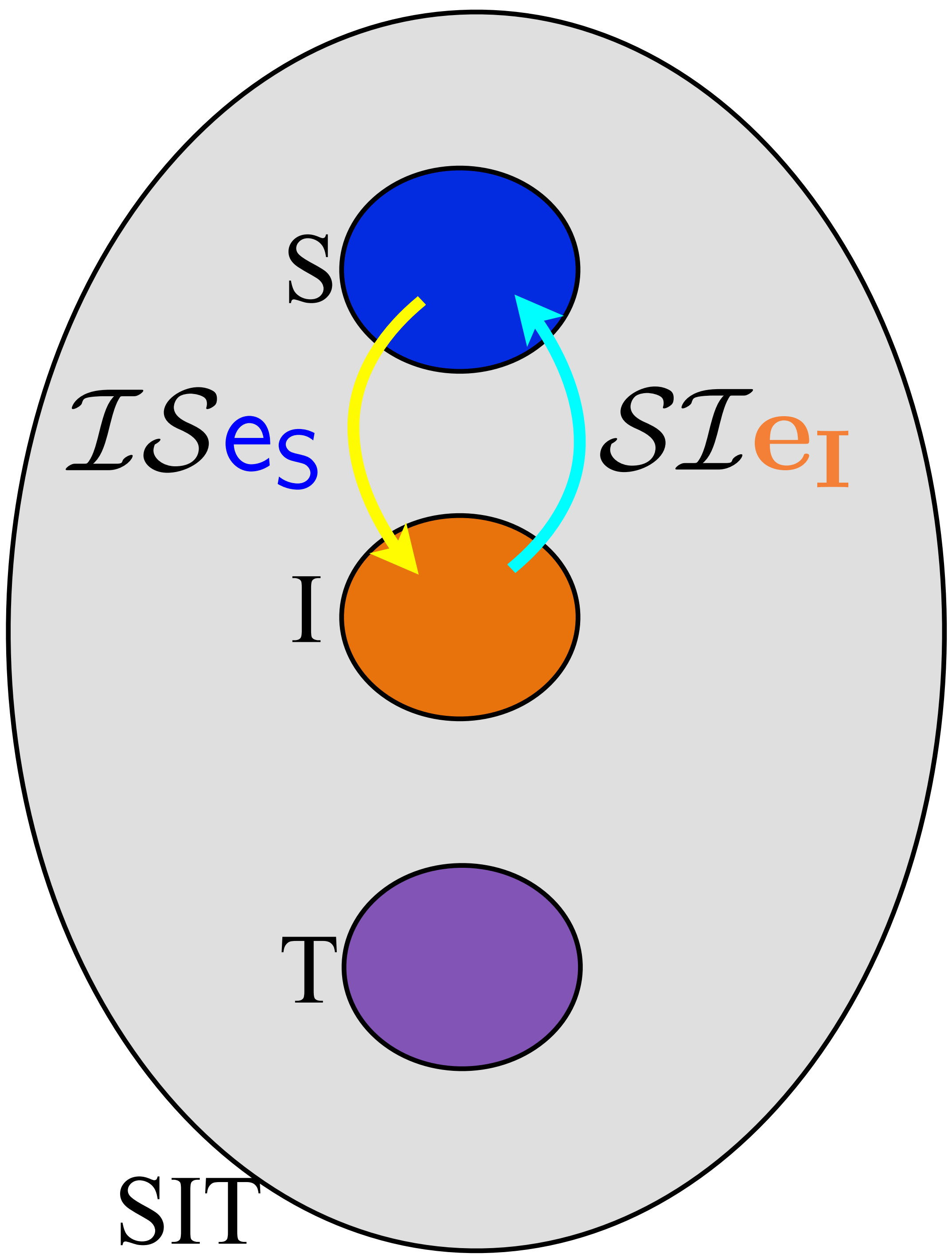
Our Approach



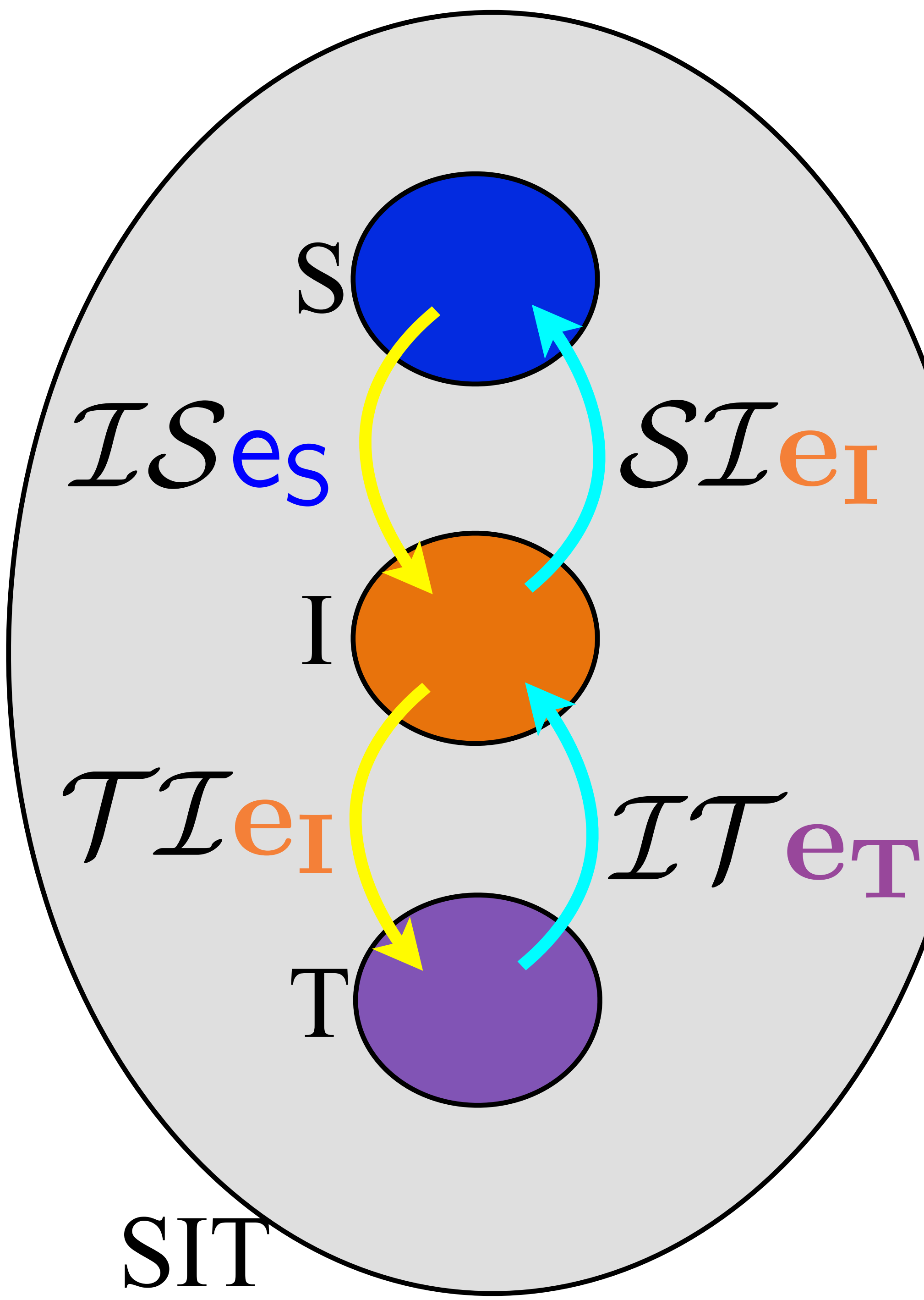
Our Approach



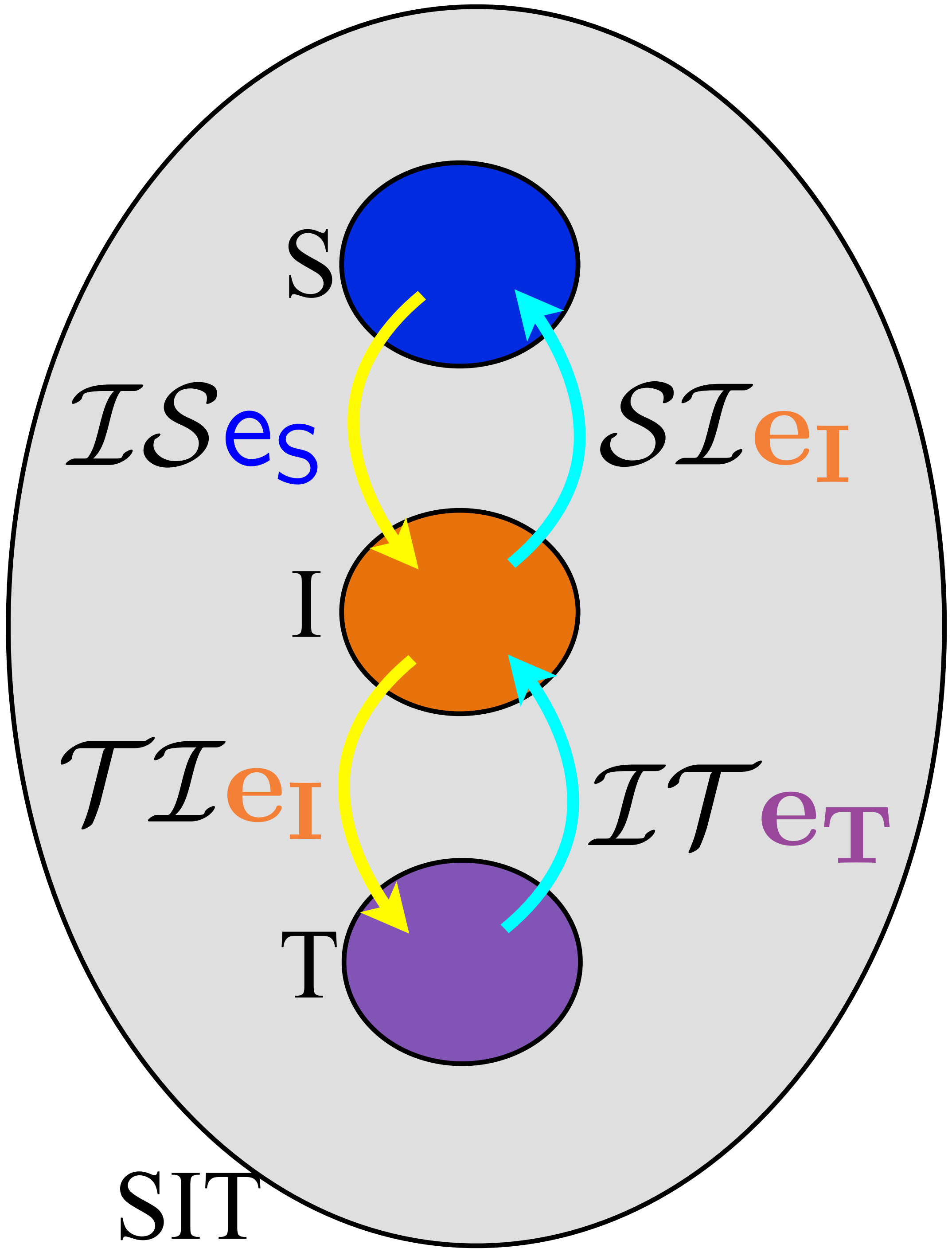
Our Approach



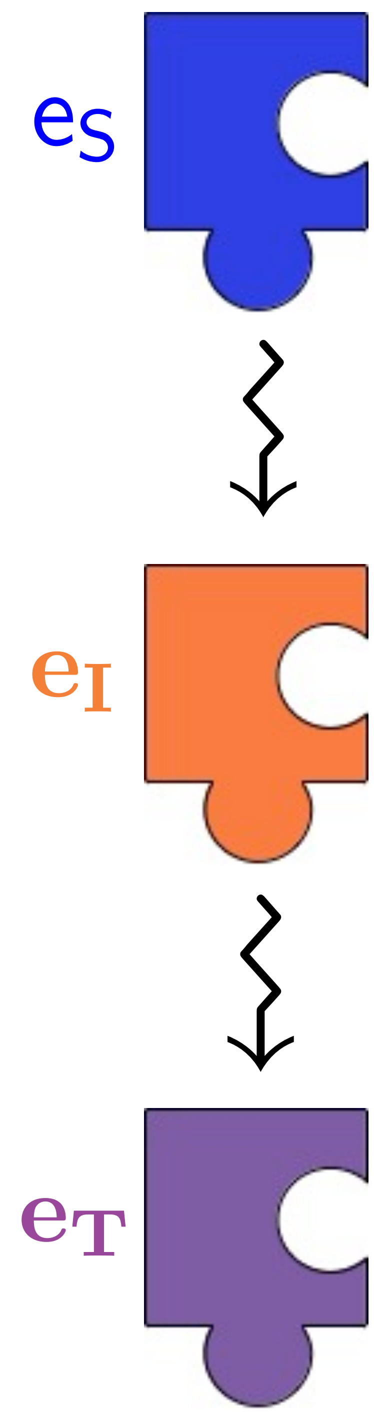
Our Approach



Our Approach



Compiler Correctness

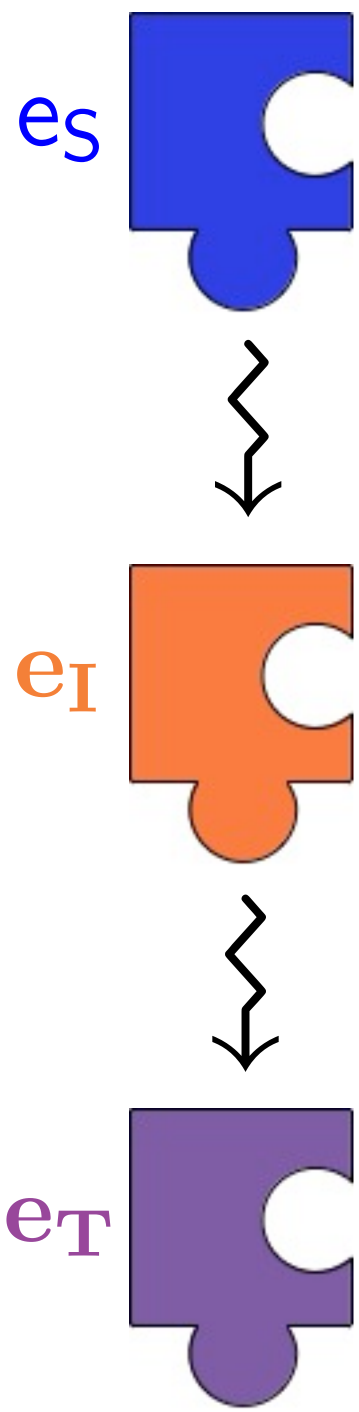


$$e_S \approx^{ctx} \mathcal{SI}\,e_I$$

$$e_I \approx^{ctx} \mathcal{IT}\,e_T$$

Our Approach: Fixes Problem 2

Compiler Correctness

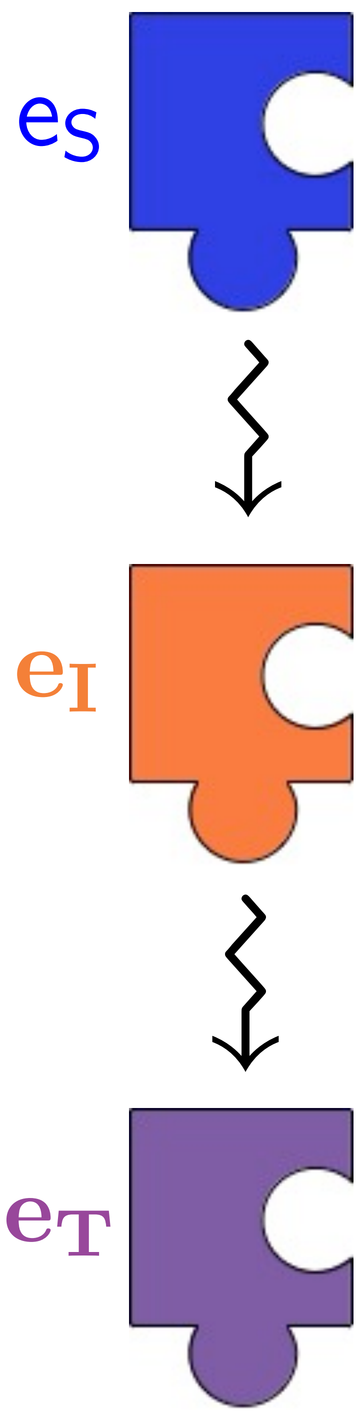


$$\mathsf{e_S} \approx^{ctx} \mathcal{SI}\, \mathbf{e_I}$$

$$\mathbf{e_I} \approx^{ctx} \mathcal{IT}\, \mathbf{e_T}$$

Our Approach: Fixes Problem 2

Compiler Correctness



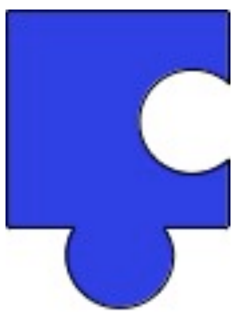
$$\mathsf{e_S} \approx^{ctx} \mathcal{SI}\mathbf{e_I}$$

$$\mathcal{SI}\mathbf{e_I} \approx^{ctx} \mathcal{SI}(\mathcal{IT}\mathbf{e_T})$$

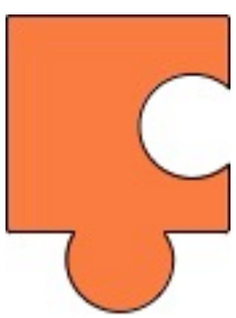
Our Approach: Fixes Problem 2

Compiler Correctness

$\mathsf{e_S}$



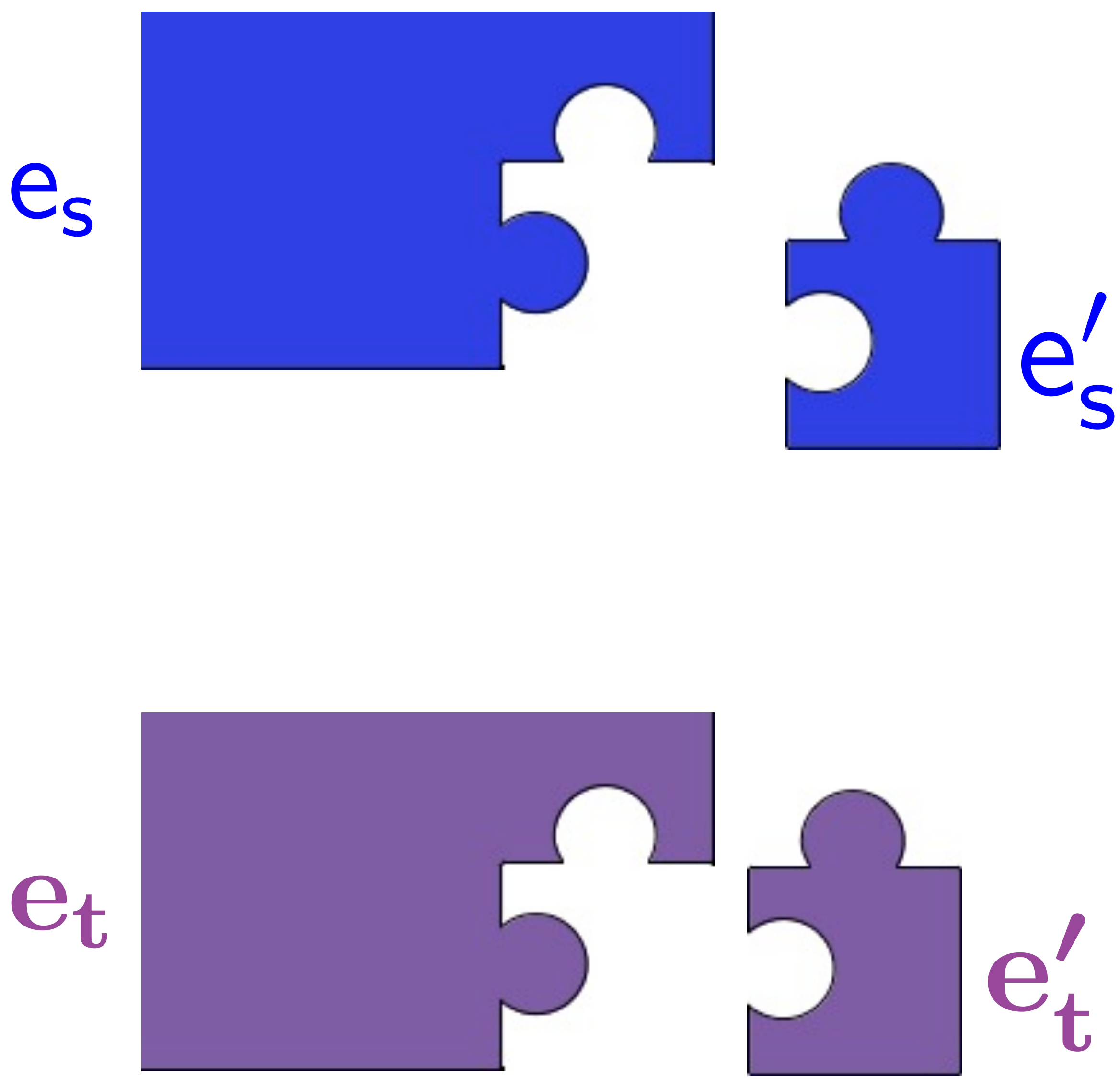
$\mathbf{e_I}$



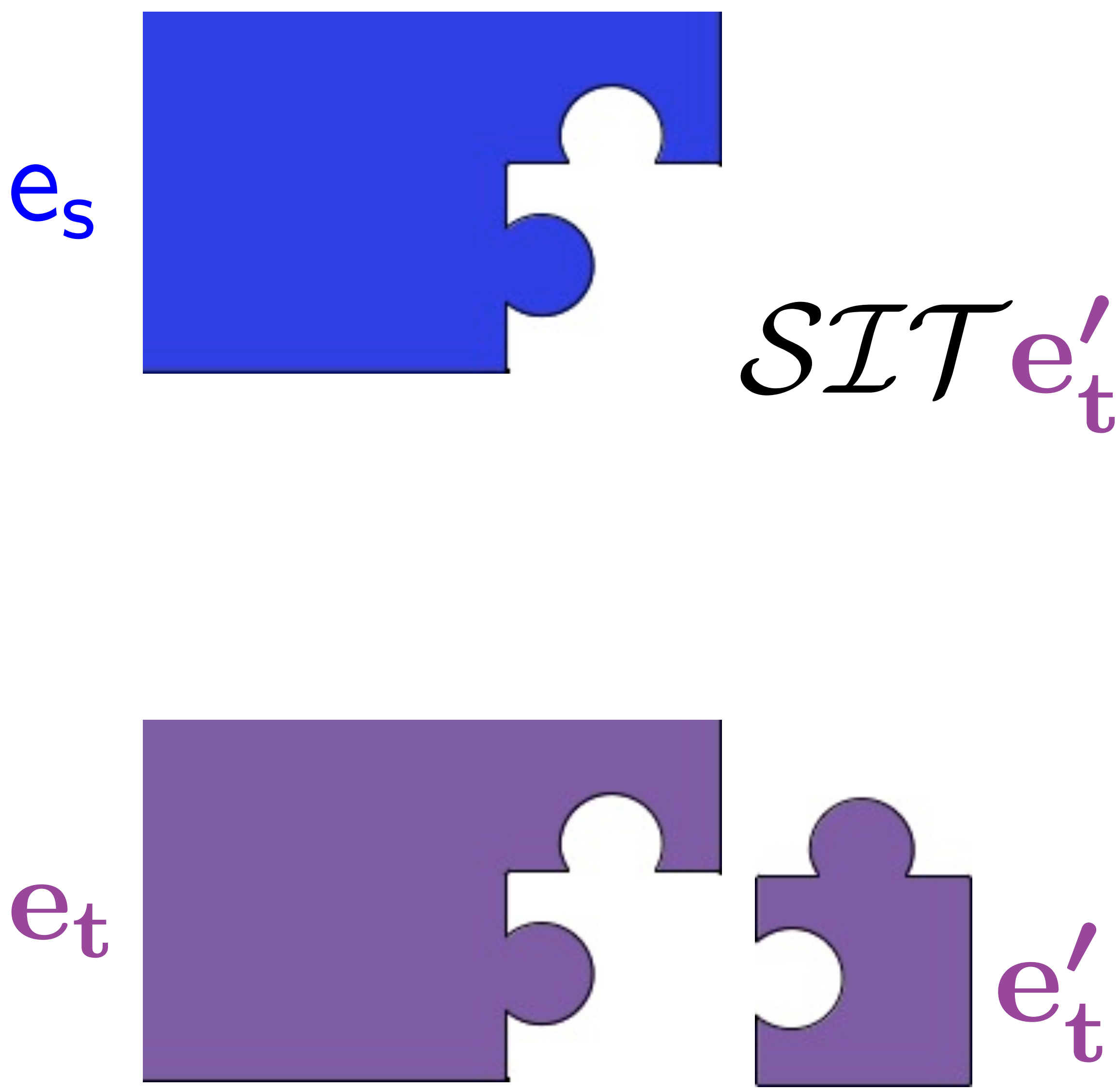
$\mathbf{e_T}$

$$\left.\begin{array}{l} \mathsf{e_S} \approx^{ctx} \mathcal{SI}\mathbf{e_I} \\[2em] \mathcal{SI}\mathbf{e_I} \approx^{ctx} \mathcal{SI}(\mathcal{IT}\mathbf{e_T}) \end{array}\right\} \mathsf{e_S} \approx^{ctx} \mathcal{SIT}\mathbf{e_T}$$

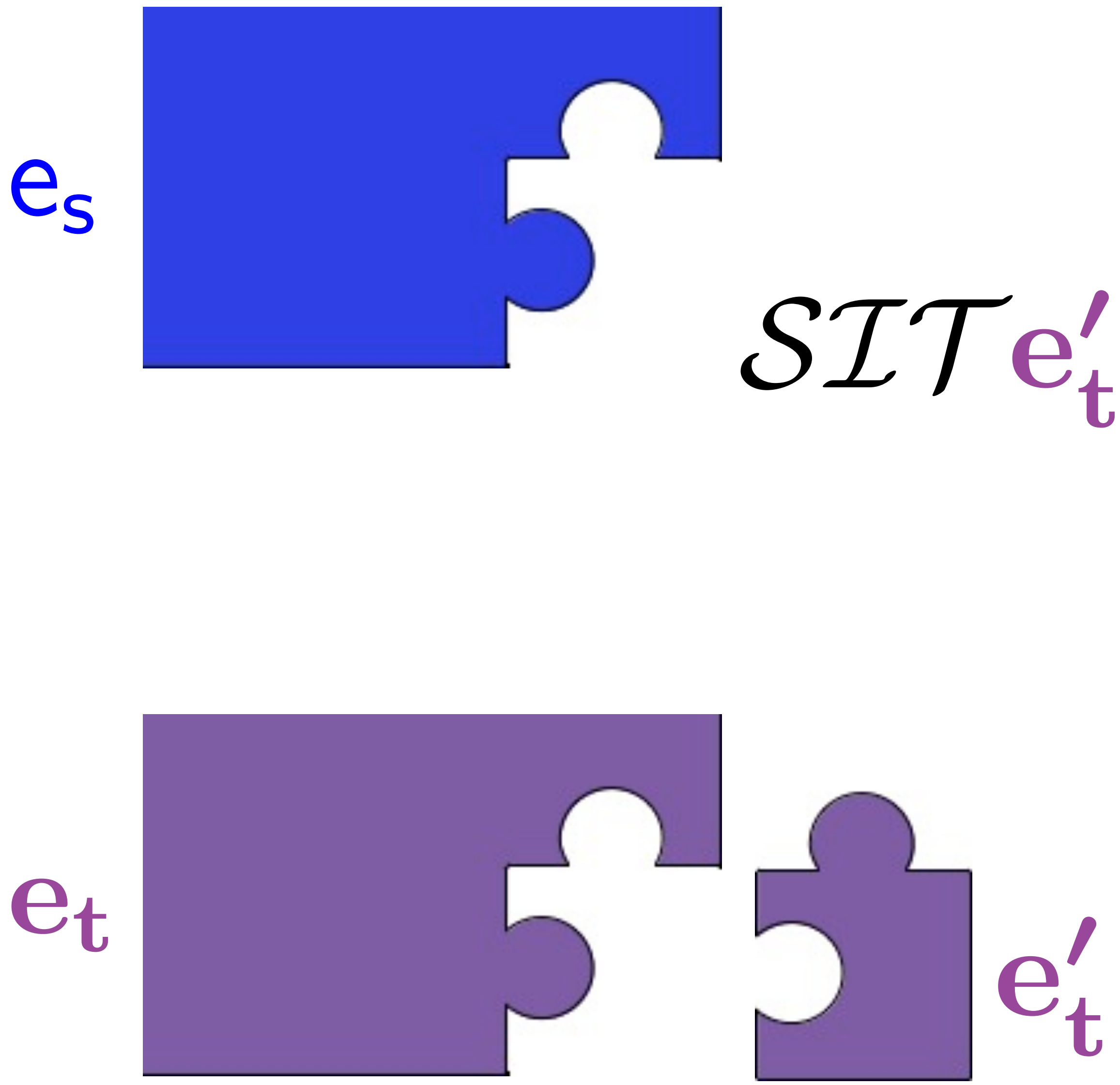
Our Approach: Fixes Problem 1



Our Approach: Fixes Problem 1

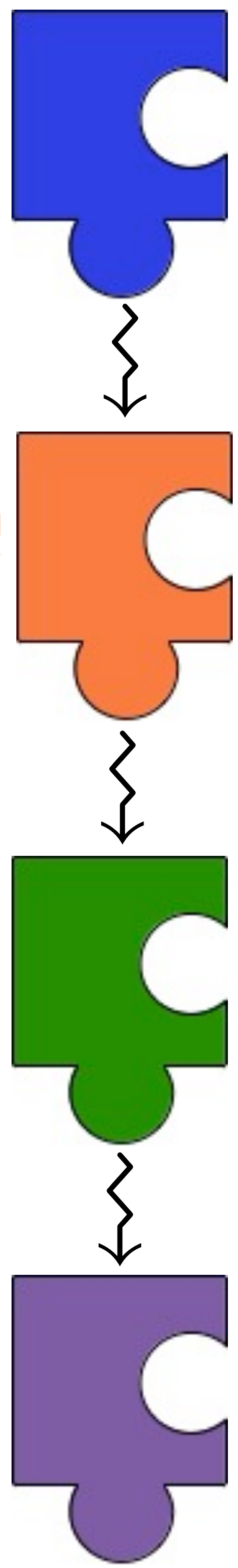


Our Approach: Fixes Problem 1



$$\mathcal{TIS}(\mathsf{e_s}\ (\mathcal{SIT}\,\mathbf{e'_t})) \approx^{ctx} \mathbf{e_t}\ \mathbf{e'_t}$$

Our Compiler: System F to TAL



e_F

Closure Conversion $\tau^{\mathcal{C}}$

e_C

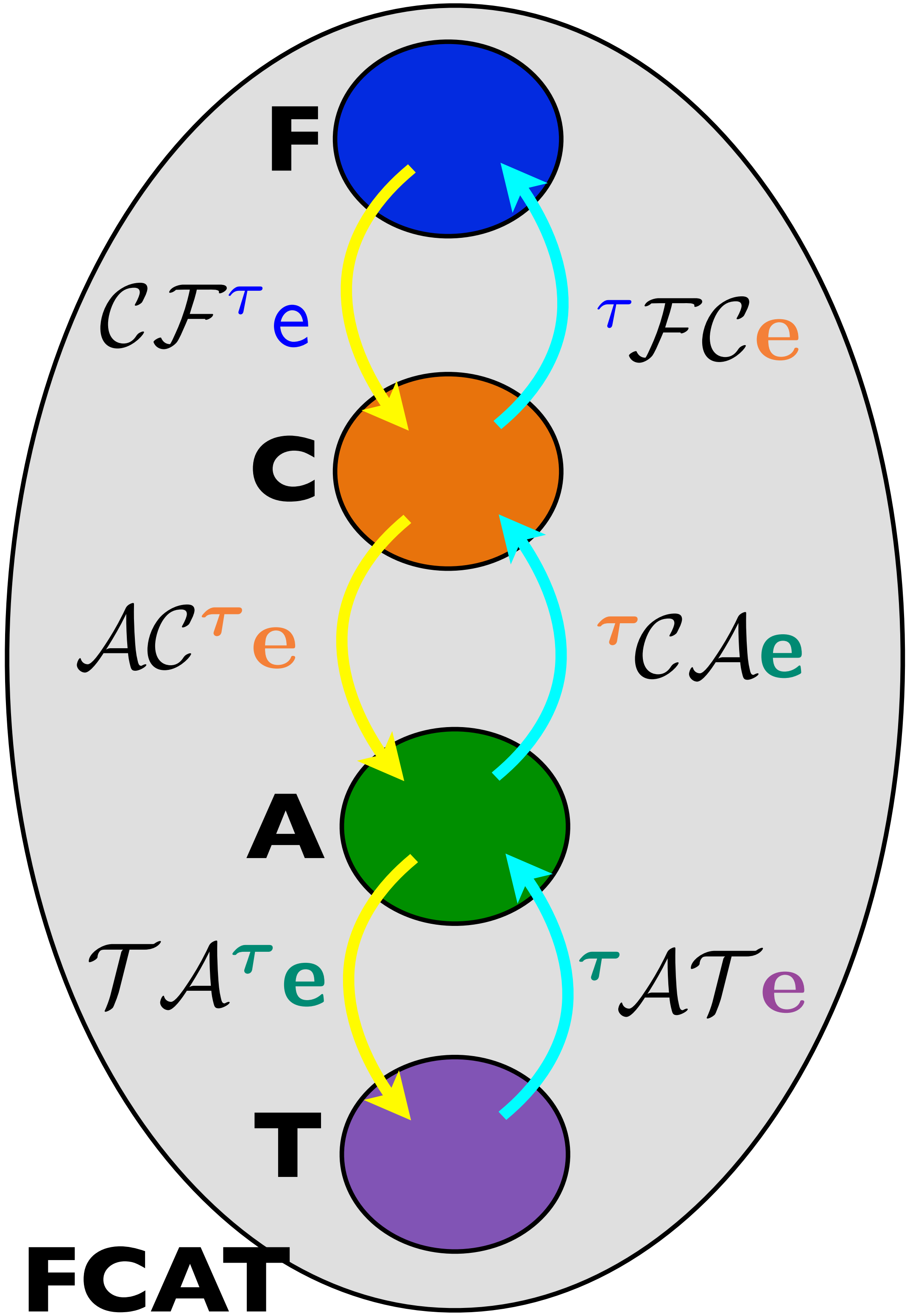
Allocation $\tau^{\mathcal{A}}$

e_A

Code Generation $\tau^{\mathcal{T}}$

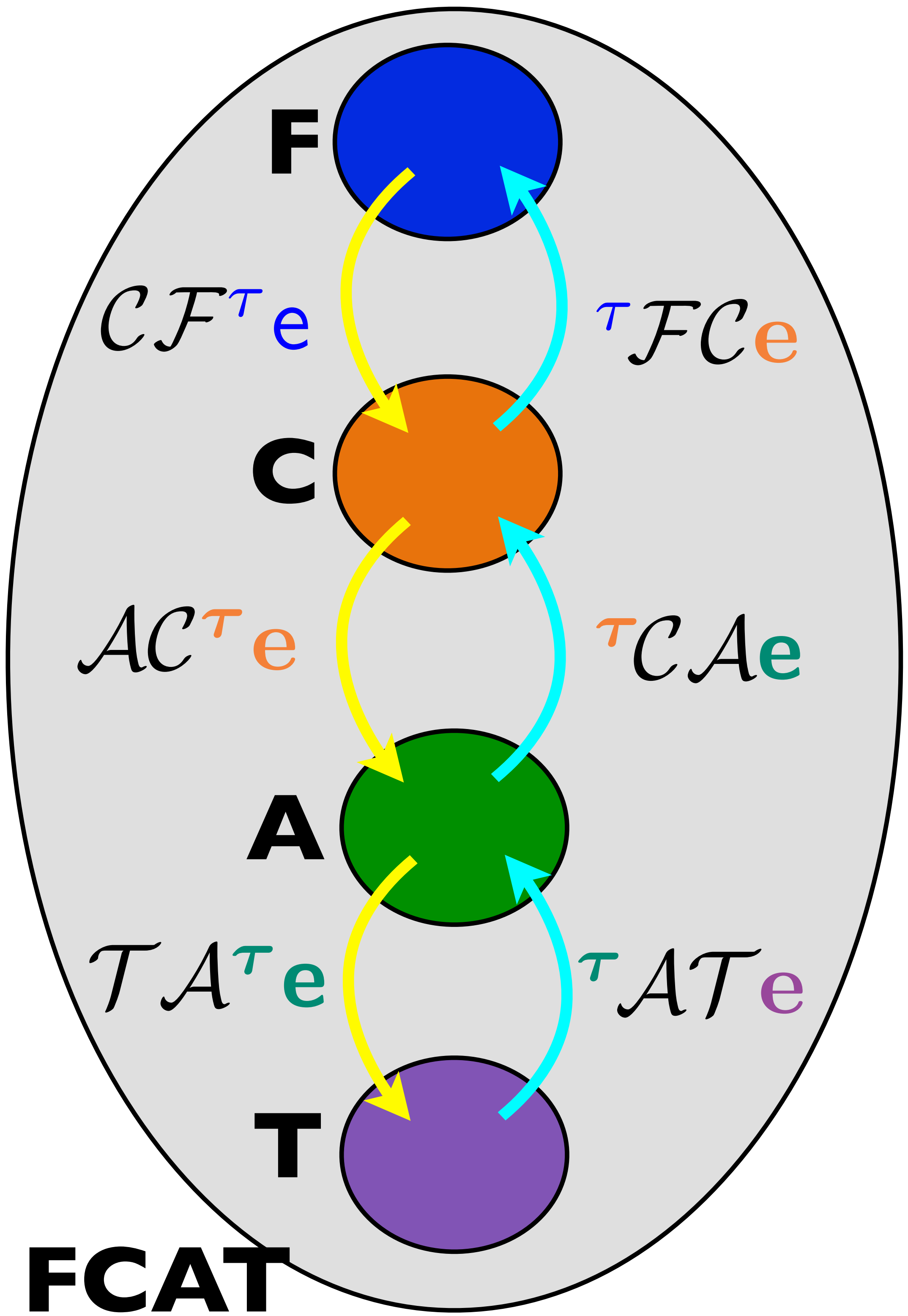
e_T

Combined language **FCAT**



- Boundaries mediate between
  - τ & $\tau^{\mathcal{C}}$ τ & $\tau^{\mathcal{A}}$ τ & $\tau^{\mathcal{T}}$

Combined language **FCAT**



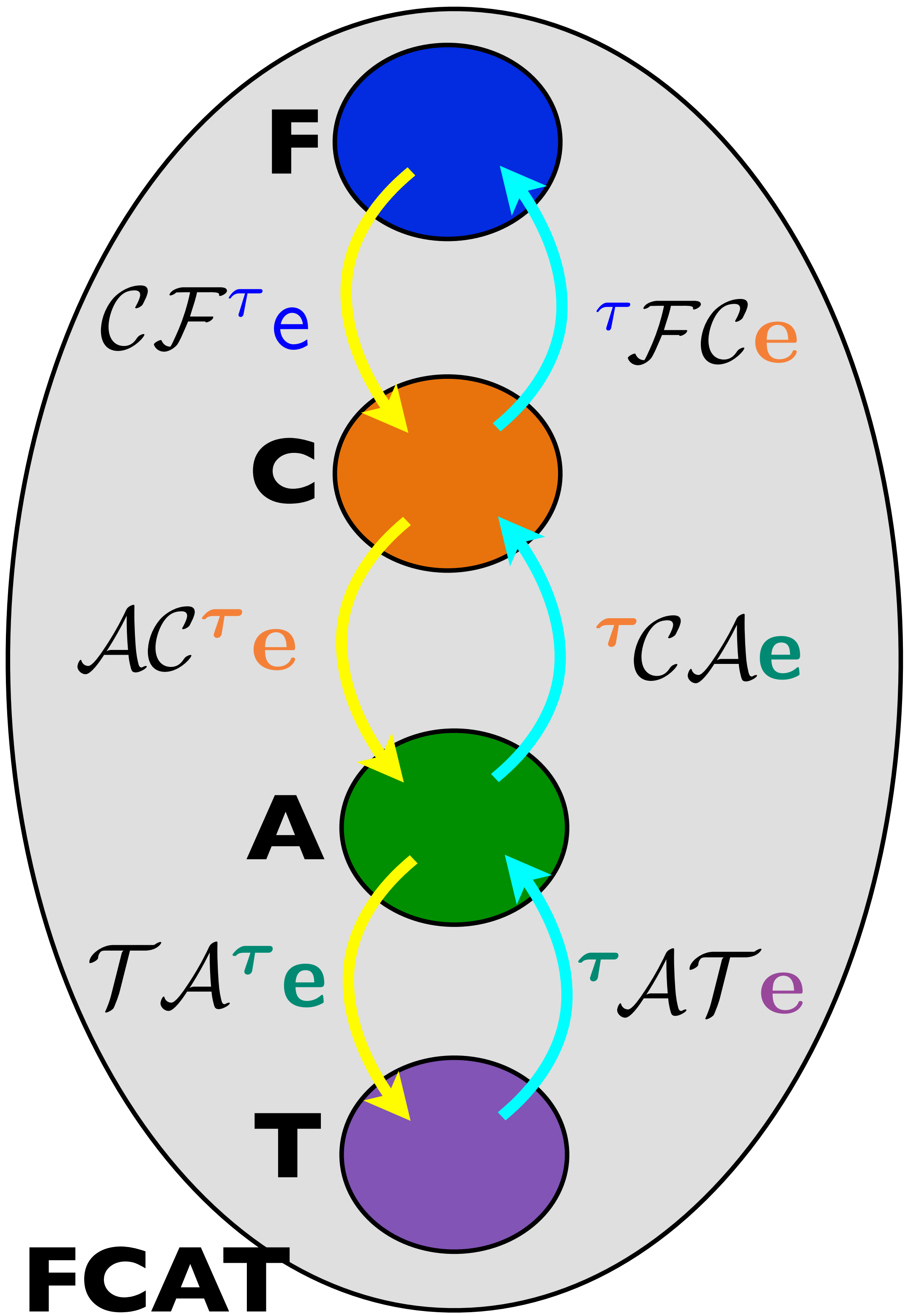
- Boundaries mediate between
  - τ & $\tau^{\mathcal{C}}$ $\quad \tau$ & $\tau^{\mathcal{A}}$ $\quad \tau$ & $\tau^{\mathcal{T}}$

- Operational semantics

$$\mathcal{CF}^{\tau}\mathsf{e} \longmapsto^{*} \mathcal{CF}^{\tau}\mathsf{v} \longmapsto \mathbf{v}$$

$$^{\tau}\mathcal{FC}\mathbf{e} \longmapsto^{*} {}^{\tau}\mathcal{FC}\mathbf{v} \longmapsto \mathsf{v}$$

Combined language **FCAT**



- Boundaries mediate between
  - τ & $\tau^{\mathcal{C}}$ τ & $\tau^{\mathcal{A}}$ τ & $\tau^{\mathcal{T}}$
- Operational semantics

$$\mathcal{CF}^{\tau}\mathsf{e} \longmapsto^* \mathcal{CF}^{\tau}\mathsf{v} \longmapsto \mathbf{v}$$

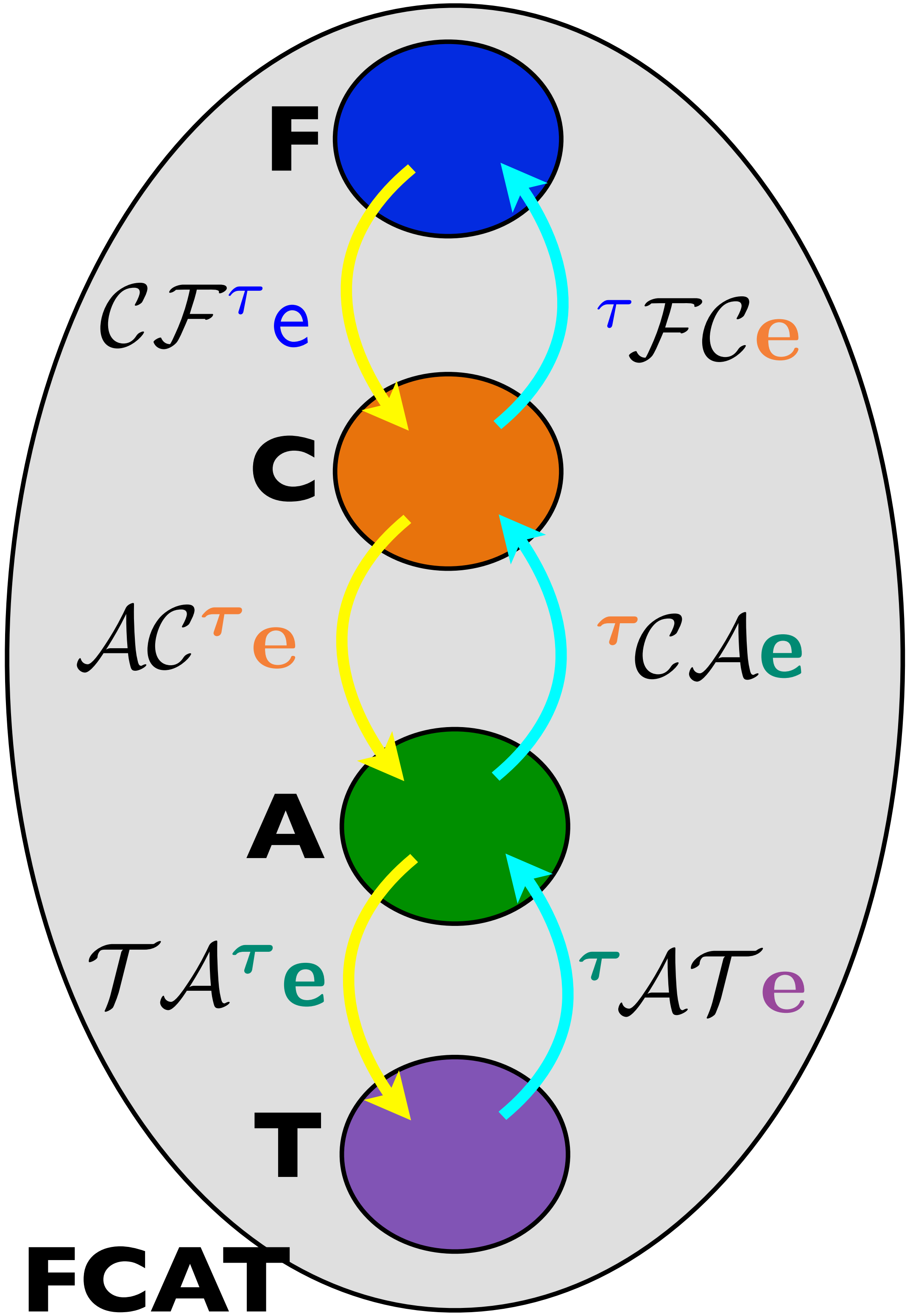
$$^{\tau}\mathcal{FC}\mathbf{e} \longmapsto^* {}^{\tau}\mathcal{FC}\mathbf{v} \longmapsto \mathsf{v}$$

- Boundary cancellation

$$^{\tau}\mathcal{FCCF}^{\tau}\mathsf{e} \approx^{ctx} \mathsf{e} : \tau$$

$$\mathcal{CF}^{\tau}\,{}^{\tau}\mathcal{FC}\mathbf{e} \approx^{ctx} \mathbf{e} : \tau^{\mathcal{C}}$$

Challenges / Roadmap for rest of talk

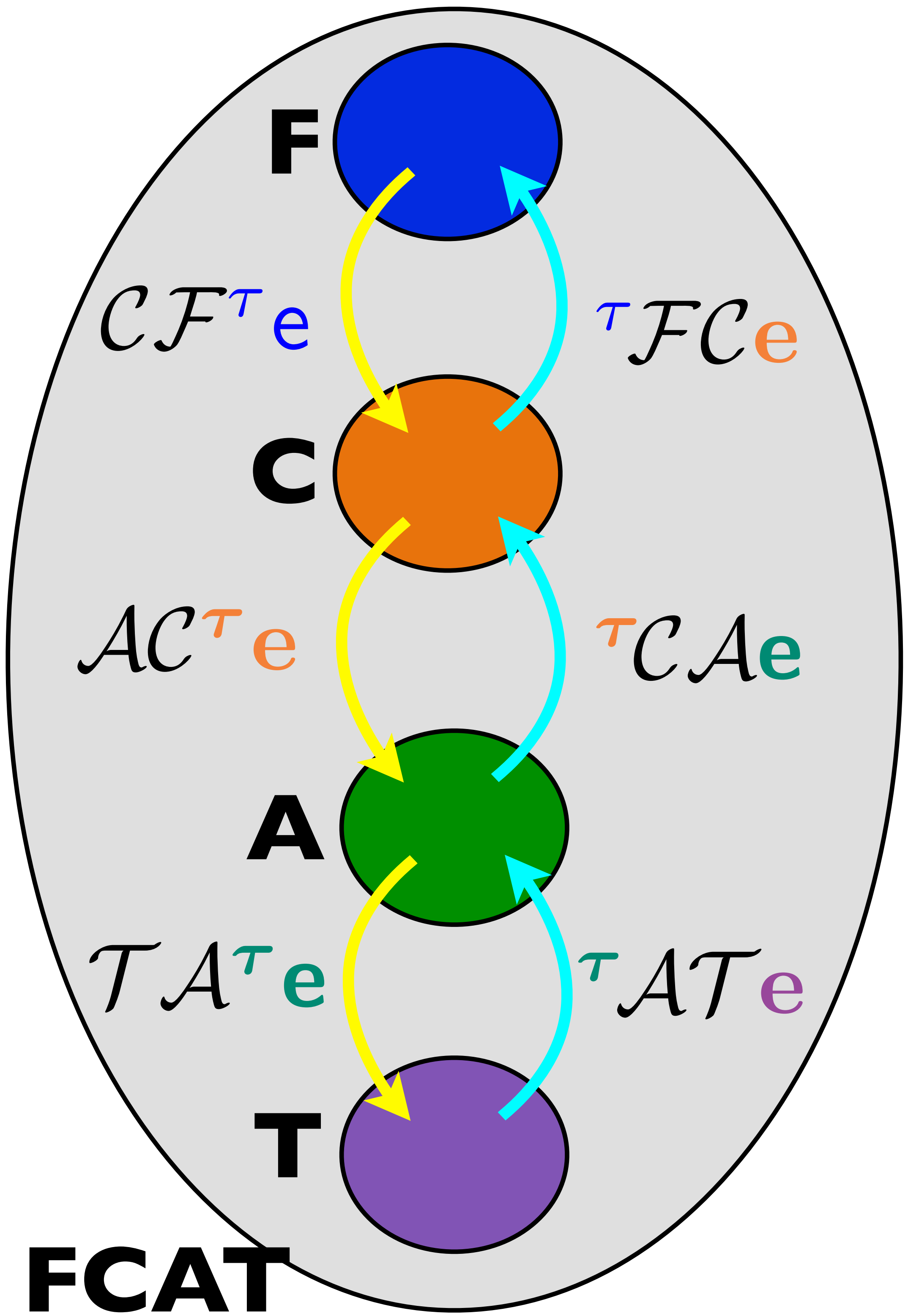


F+C: Interoperability semantics with type abstraction in both languages

C+A: Interoperability when compiler pass allocates code & tuples on heap

A+T: What is e? What is $\mathbf{v}$? How to define contextual equiv. for TAL *components*? How to define logical relation?

Challenges / Roadmap for rest of talk



F+C: Interoperability semantics with type abstraction in both languages

C+A: Interoperability when compiler pass allocates code & tuples on heap

A+T: What is e? What is $\mathbf{v}$? How to define contextual equiv. for TAL *components*? How to define logical relation?

F

$$
\begin{array}{rcl}
\tau & ::= & \alpha \mid \mathsf{unit} \mid \mathsf{int} \mid \forall[\overline{\alpha}].(\overline{\tau}) \to \tau \mid \exists\alpha.\tau \mid \mu\alpha.\tau \mid \langle\overline{\tau}\rangle \\
\mathsf{e} & ::= & \mathsf{t} \\
\mathsf{t} & ::= & \mathsf{x} \mid () \mid \mathsf{n} \mid \mathsf{t\;p\;t} \mid \mathsf{if0\;t\;t\;t} \mid \lambda[\overline{\alpha}](\overline{\mathsf{x}:\tau}).\mathsf{t} \mid \mathsf{t}\,[\overline{\tau}]\,\overline{\mathsf{t}} \\
& & \mid \mathsf{pack}\langle\tau,\mathsf{t}\rangle\;\mathsf{as}\;\exists\alpha.\tau \mid \mathsf{unpack}\;\langle\alpha,\mathsf{x}\rangle = \mathsf{t}\;\mathsf{in}\;\mathsf{t} \mid \mathsf{fold}_{\mu\alpha.\tau}\;\mathsf{t} \\
& & \mid \mathsf{unfold}\;\mathsf{t} \mid \langle\overline{\mathsf{t}}\rangle \mid \pi_{\mathsf{i}}(\mathsf{t}) \\
\mathsf{p} & ::= & + \mid - \mid * \\
\mathsf{v} & ::= & () \mid \mathsf{n} \mid \lambda[\overline{\alpha}](\overline{\mathsf{x}:\tau}).\mathsf{t} \mid \mathsf{pack}\langle\tau,\mathsf{v}\rangle\;\mathsf{as}\;\exists\alpha.\tau \mid \mathsf{fold}_{\mu\alpha.\tau}\;\mathsf{v} \mid \langle\overline{\mathsf{v}}\rangle
\end{array}
$$

$\boxed{\Delta;\Gamma \vdash \mathsf{e} : \tau}$ where $\Delta ::= \cdot \mid \Delta, \alpha$ and $\Gamma ::= \cdot \mid \Gamma, \mathsf{x} : \tau$

C

$$
\begin{array}{rcl}
\tau & ::= & \alpha \mid \mathbf{unit} \mid \mathbf{int} \mid \forall[\overline{\alpha}].(\overline{\tau}) \rightarrow \tau \mid \exists \alpha.\tau \mid \mu\alpha.\tau \mid \langle \overline{\tau} \rangle \\
\mathrm{e} & ::= & \mathrm{t} \\
\mathrm{t} & ::= & \mathrm{x} \mid () \mid \mathrm{n} \mid \mathrm{t\ p\ t} \mid \mathbf{if0}\ \mathrm{t\ t\ t} \mid \lambda[\overline{\alpha}](\overline{\mathrm{x}:\tau}).\mathrm{t} \mid \mathrm{t}\,[]\,\overline{\mathrm{t}} \\
 & \mid & \mathrm{t}[\tau] \mid \mathbf{pack}\langle\tau,\mathrm{t}\rangle\ \mathbf{as}\ \exists\alpha.\tau \mid \mathbf{unpack}\ \langle\alpha,\mathrm{x}\rangle = \mathrm{t}\ \mathbf{in}\ \mathrm{t} \\
 & \mid & \mathbf{fold}_{\mu\alpha.\tau}\ \mathrm{t} \mid \mathbf{unfold}\ \mathrm{t} \mid \langle\overline{\mathrm{t}}\rangle \mid \pi_{\mathrm{i}}(\mathrm{t}) \\
\mathrm{p} & ::= & + \mid - \mid * \\
\mathrm{v} & ::= & () \mid \mathrm{n} \mid \lambda[\overline{\alpha}](\overline{\mathrm{x}:\tau}).\mathrm{t} \mid \mathbf{pack}\langle\tau,\mathrm{v}\rangle\ \mathbf{as}\ \exists\alpha.\tau \\
 & \mid & \mathbf{fold}_{\mu\alpha.\tau}\ \mathrm{v} \mid \langle\overline{\mathrm{v}}\rangle \mid \mathrm{v}[\tau]
\end{array}
$$

$$\boxed{\Delta;\Gamma \vdash \mathrm{e} : \tau}$$

$$\frac{\overline{\alpha};\overline{\mathrm{x}:\tau} \vdash \mathrm{t} : \tau'}{\Delta;\Gamma \vdash \lambda[\overline{\alpha}](\overline{\mathrm{x}:\tau}).\mathrm{t} : \forall[\overline{\alpha}].(\overline{\tau}) \rightarrow \tau'}$$

Closure Conversion: **F** to **C**

$\boxed{\tau^{\mathcal{C}}}$ Type Translation

$$\alpha^{\mathcal{C}} = \alpha \qquad \forall[\overline{\alpha}].(\overline{\tau}) \to \tau'^{\mathcal{C}} = \exists\beta.\langle(\forall[\overline{\alpha}].(\beta, \overline{\tau^{\mathcal{C}}}) \to \tau'^{\mathcal{C}}), \beta\rangle$$

$$\mathsf{unit}^{\mathcal{C}} = \mathbf{unit} \qquad \exists\alpha.\tau^{\mathcal{C}} = \exists\alpha.\tau^{\mathcal{C}}$$

$$\mathsf{int}^{\mathcal{C}} = \mathbf{int} \qquad \mu\alpha.\tau^{\mathcal{C}} = \mu\alpha.\tau^{\mathcal{C}}$$

$$\langle\tau_1, \ldots, \tau_{\mathsf{n}}\rangle^{\mathcal{C}} = \langle{\tau_1}^{\mathcal{C}}, \ldots, {\tau_{\mathsf{n}}}^{\mathcal{C}}\rangle$$

$\boxed{\Delta; \Gamma \vdash \mathsf{e} : \tau \rightsquigarrow \mathbf{e}}$ where $\Delta^{\mathcal{C}}; \Gamma^{\mathcal{C}} \vdash \mathbf{e} : \tau^{\mathcal{C}}$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}(\mathsf{v}) = \mathbf{v}}$$ Value Translation

$$\mathbf{CF}^{\mathsf{int}}(\mathsf{n}) = \mathbf{n}$$

$$\boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$^{\mathsf{int}}\mathbf{FC}(\mathbf{n}) = \mathsf{n}$$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}(\mathsf{v}) = \mathbf{v}} \qquad \boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$(\tau \to \tau')^{\mathcal{C}} = \exists \beta. \langle ((\beta, \tau^{\mathcal{C}}) \to \tau'^{\mathcal{C}}), \beta \rangle$$

$$\mathbf{CF}^{\tau \to \tau'}(\mathsf{v}) = \mathbf{pack} \langle \mathbf{unit}, \langle \mathbf{v}, () \rangle \rangle \ \mathbf{as} \ \exists \beta. \langle ((\beta, \tau^{\mathcal{C}}) \to \tau'^{\mathcal{C}}), \beta \rangle$$

$$\mathbf{v} = \boldsymbol{\lambda}(\mathbf{z} : \mathbf{unit}, \mathbf{x} : \tau^{\mathcal{C}}). \mathcal{CF}^{\tau'}(\mathsf{v} \ {}^{\tau}\mathcal{FC} \ \mathbf{x})$$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}(\mathsf{v}) = \mathbf{v}} \qquad \boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$(\tau \to \tau')^{\mathcal{C}} = \exists\beta.\langle((\beta, \tau^{\mathcal{C}}) \to \tau'^{\mathcal{C}}), \beta\rangle$$

$$\mathbf{CF}^{\tau \to \tau'}(\mathsf{v}) =$$
$$\mathbf{pack}\langle\mathbf{unit}, \langle\mathbf{v}, ()\rangle\rangle\ \mathbf{as}\ \exists\beta.\langle((\beta, \tau^{\mathcal{C}}) \to \tau'^{\mathcal{C}}), \beta\rangle$$

$$\mathbf{v} = \lambda(\mathbf{z} : \mathbf{unit}, \mathbf{x} : \tau^{\mathcal{C}}).\mathcal{CF}^{\tau'}(\mathsf{v}\ {}^{\tau}\mathcal{FC}\ \mathbf{x})$$

$${}^{\tau \to \tau'}\mathbf{FC}(\mathbf{v}) = \lambda(\mathsf{x} : \tau).{}^{\tau'}\mathcal{FC}(\mathbf{unpack}\ \langle\beta, \mathbf{y}\rangle = \mathbf{v}$$
$$\mathbf{in}\ \pi_1(\mathbf{y})\ \pi_2(\mathbf{y})\ \mathcal{CF}^{\tau}\mathsf{x})$$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}(\mathsf{v}) = \mathbf{v}} \qquad \boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$(\forall[\alpha].(\alpha) \to \alpha)^{\mathcal{C}} = \exists\beta.\langle(\forall[\alpha].(\beta, \alpha) \to \alpha), \beta\rangle$$

$$\alpha^{\mathcal{C}} = \alpha$$

$$\mathbf{CF}^{\forall[\alpha].(\alpha) \to \alpha}(\mathsf{v}) = \mathbf{pack}\langle\mathbf{unit}, \langle\mathbf{v}, ()\rangle\rangle \ \mathbf{as}\ (\forall[\alpha].(\alpha) \to \alpha)^{\mathcal{C}}$$

$$\mathbf{v} = \boldsymbol{\lambda}[\alpha](\mathbf{z}:\mathbf{unit}, \mathbf{x}:\alpha).\mathcal{CF}^{\alpha}(\mathsf{v}\,[\alpha]\ {}^{\alpha}\mathcal{FC}\mathbf{x})$$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}(\mathsf{v}) = \mathbf{v}} \qquad \boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$(\forall[\alpha].(\alpha) \to \alpha)^{\mathcal{C}} = \exists\beta.\langle(\forall[\alpha].(\beta,\alpha) \to \alpha),\beta\rangle$$

$$\alpha^{\mathcal{C}} = \alpha$$

$$\mathbf{CF}^{\forall[\alpha].(\alpha) \to \alpha}(\mathsf{v}) = \mathbf{pack}\langle\mathbf{unit},\langle\mathbf{v},()\rangle\rangle\ \mathbf{as}\ (\forall[\alpha].(\alpha) \to \alpha)^{\mathcal{C}}$$

$$\mathbf{v} = \lambda[\alpha](\mathbf{z}:\mathbf{unit},\mathbf{x}:\alpha).\mathcal{CF}^{\alpha}(\mathsf{v}\,[\alpha]\,{}^{\alpha}\mathcal{FC}\mathbf{x})$$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}(\mathsf{v}) = \mathbf{v}} \qquad \boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$(\forall[\alpha].(\alpha) \to \alpha)^{\mathcal{C}} = \exists\beta.\langle(\forall[\alpha].(\beta, \alpha) \to \alpha), \beta\rangle$$

$$\alpha^{\mathcal{C}} = \alpha \qquad \boxed{\mathsf{L}\langle\tau\rangle^{\mathcal{C}} = \tau}$$

$$\mathbf{CF}^{\forall[\alpha].(\alpha) \to \alpha}(\mathsf{v}) = \mathbf{pack}\langle\mathbf{unit}, \langle\mathbf{v}, ()\rangle\rangle\ \mathbf{as}\ (\forall[\alpha].(\alpha) \to \alpha)^{\mathcal{C}}$$

$$\mathbf{v} = \boldsymbol{\lambda}[\alpha](\mathbf{z}: \mathbf{unit}, \mathbf{x}: \alpha).\mathcal{CF}^{\mathsf{L}\langle\alpha\rangle}(\mathsf{v}\,[\mathsf{L}\langle\alpha\rangle]\ {}^{\mathsf{L}\langle\alpha\rangle}\mathcal{FC}\mathbf{x})$$

Add new type $\mathsf{L}\langle\tau\rangle$ & new value form ${}^{\mathsf{L}\langle\tau\rangle}\mathcal{FC}\mathbf{v}$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}(\mathsf{v}) = \mathbf{v}} \qquad \boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$(\forall[\alpha].(\alpha) \rightarrow \alpha)^{\langle \mathcal{C} \rangle} = \exists \beta. \langle (\forall[\alpha].(\beta, \alpha) \rightarrow \alpha), \beta \rangle$$

$$\alpha^{\langle \mathcal{C} \rangle} = \alpha \qquad \mathsf{L}\langle \tau \rangle^{\langle \mathcal{C} \rangle} = \tau$$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}\,(\mathsf{v}) = \mathbf{v}} \qquad \boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$(\forall[\alpha].(\alpha) \to \alpha)^{\langle\mathcal{C}\rangle} = \exists\beta.\langle(\forall[\alpha].(\beta,\alpha) \to \alpha),\beta\rangle$$

$$\alpha^{\langle\mathcal{C}\rangle} = \alpha \qquad\qquad \mathsf{L}\langle\tau\rangle^{\langle\mathcal{C}\rangle} = \tau$$

$$^{\forall[\alpha].(\alpha)\to\alpha}\mathbf{FC}(\mathbf{v}) = \lambda[\alpha](\mathsf{x}:\alpha).{}^{\alpha}\mathcal{FC}(\mathbf{unpack}\ \langle\beta,\mathbf{y}\rangle = \mathbf{v}$$
$$\mathbf{in}\ \pi_1(\mathbf{y})\ [\alpha^{\mathcal{C}}]\ \pi_2(\mathbf{y})\ \mathcal{CF}^{\alpha}\mathsf{x})$$

Interoperability: **F** and **C**

$$\boxed{\mathbf{CF}^{\tau}(\mathsf{v}) = \mathbf{v}} \qquad \boxed{{}^{\tau}\mathbf{FC}(\mathbf{v}) = \mathsf{v}}$$

$$(\forall[\alpha].(\alpha) \to \alpha)^{\langle \mathcal{C} \rangle} = \exists \beta.\langle (\forall[\alpha].(\beta, \alpha) \to \alpha), \beta \rangle$$

$$\boxed{\alpha^{\mathcal{C}} = \lceil \alpha \rceil} \qquad \mathsf{L}\langle \tau \rangle^{\langle \mathcal{C} \rangle} = \tau$$

$$^{\forall[\alpha].(\alpha) \to \alpha}\mathbf{FC}(\mathbf{v}) = \lambda[\alpha](\mathsf{x} : \alpha).{}^{\alpha}\mathcal{FC}(\mathbf{unpack}\ \langle \beta, \mathbf{y} \rangle = \mathbf{v}$$
$$\mathbf{in}\ \pi_1(\mathbf{y})[\lceil \alpha \rceil]\pi_2(\mathbf{y})\ \mathcal{CF}^{\alpha}\mathsf{x})$$

$$\boxed{\text{Add new type } \lceil \alpha \rceil \ \text{ \& define } \lceil \alpha \rceil[\tau/\alpha] = \tau^{\langle \mathcal{C} \rangle}}$$

Interoperability: **F** and **C**

$\boxed{\tau^{\langle \mathcal{C} \rangle}}$ Operational Type Translation

$$\forall[\overline{\alpha}].(\overline{\tau}) \to \tau'^{\langle \mathcal{C} \rangle} = \exists\beta.\langle \left(\forall[\overline{\alpha}].(\beta, \overline{\tau^{\langle \mathcal{C} \rangle}\overline{[\alpha/\lceil\alpha\rceil]}}) \to \tau'^{\langle \mathcal{C} \rangle}\overline{[\alpha/\lceil\alpha\rceil]}\right), \beta\rangle$$

$$\alpha^{\langle \mathcal{C} \rangle} = \lceil\alpha\rceil \qquad \exists\alpha.\tau^{\langle \mathcal{C} \rangle} = \exists\alpha.(\tau^{\langle \mathcal{C} \rangle}[\alpha/\lceil\alpha\rceil])$$

$$\mathsf{unit}^{\langle \mathcal{C} \rangle} = \mathbf{unit} \qquad \mu\alpha.\tau^{\langle \mathcal{C} \rangle} = \mu\alpha.(\tau^{\langle \mathcal{C} \rangle}[\alpha/\lceil\alpha\rceil])$$

$$\mathsf{int}^{\langle \mathcal{C} \rangle} = \mathbf{int} \qquad \langle\tau_1, \ldots, \tau_{\mathsf{n}}\rangle^{\langle \mathcal{C} \rangle} = \langle\tau_1{}^{\langle \mathcal{C} \rangle}, \ldots, \tau_{\mathsf{n}}{}^{\langle \mathcal{C} \rangle}\rangle$$

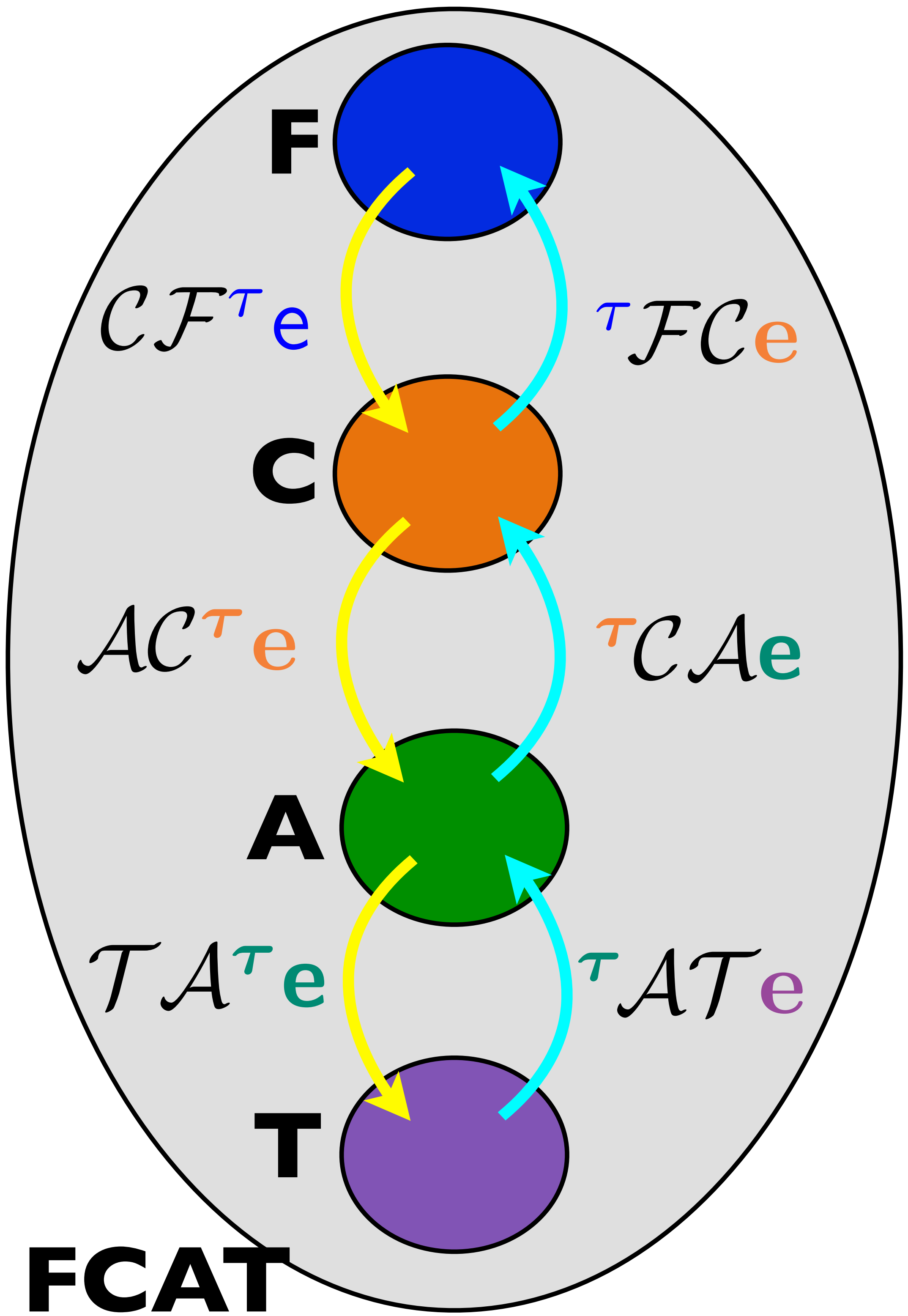
$$\mathsf{L}\langle\tau\rangle^{\langle \mathcal{C} \rangle} = \tau$$

Type Substitution: $\lceil\alpha\rceil[\tau/\alpha] = \tau^{\langle \mathcal{C} \rangle}$

$\boxed{\Delta; \Gamma \vdash e : \tau}$ Include **F** and **C** rules, with environments replaced by $\Delta; \Gamma$

$$\frac{\Delta; \Gamma \vdash \mathbf{e} : \tau^{\langle \mathcal{C} \rangle}}{\Delta; \Gamma \vdash {}^{\tau}\mathcal{F}\mathcal{C}\,\mathbf{e} : \tau} \qquad \frac{\Delta; \Gamma \vdash \mathsf{e} : \tau}{\Delta; \Gamma \vdash \mathcal{C}\mathcal{F}^{\tau}\,\mathsf{e} : \tau^{\langle \mathcal{C} \rangle}}$$

Challenges / Roadmap



F+C: Interoperability semantics with type abstraction in both languages

C+A: Interoperability when compiler pass allocates code & tuples on heap

A+T: What is e? What is v? How to define contextual equiv. for TAL *components*? How to define logical relation?

A

$$\tau ::= \alpha \mid \mathsf{unit} \mid \mathsf{int} \mid \exists\alpha.\tau \mid \mu\alpha.\tau \mid \mathsf{box}\,\psi$$

$$\psi ::= \forall[\overline{\alpha}].(\overline{\tau}) \to \tau \mid \langle \tau, \ldots, \tau \rangle$$

$$\mathsf{e} ::= (\mathsf{t}, \mathsf{H}) \mid \mathsf{t}$$

$$\mathsf{t} ::= \mathsf{x} \mid () \mid \mathsf{n} \mid \mathsf{t}\ \mathsf{p}\ \mathsf{t} \mid \mathsf{if0}\ \mathsf{t}\ \mathsf{t}\ \mathsf{t} \mid \ell \mid \mathsf{t}\,[]\,\bar{\mathsf{t}} \mid \mathsf{t}[\tau]$$
$$\mid \mathsf{pack}\langle \tau, \mathsf{t} \rangle\ \mathsf{as}\ \exists\alpha.\tau \mid \mathsf{unpack}\ \langle \alpha, \mathsf{x} \rangle = \mathsf{t}\ \mathsf{in}\ \mathsf{t} \mid \mathsf{fold}_{\mu\alpha.\tau}\ \mathsf{t}$$
$$\mid \mathsf{unfold}\ \mathsf{t} \mid \mathsf{balloc}\ \langle \bar{\mathsf{t}} \rangle \mid \mathsf{read}[\mathsf{i}]\ \mathsf{t}$$

$$\mathsf{p} ::= + \mid - \mid *$$

$$\mathsf{v} ::= () \mid \mathsf{n} \mid \mathsf{pack}\langle \tau, \mathsf{v} \rangle\ \mathsf{as}\ \exists\alpha.\tau \mid \mathsf{fold}_{\mu\alpha.\tau}\ \mathsf{v} \mid \ell \mid \mathsf{v}[\tau]$$

$$\mathsf{H} ::= \cdot \mid \mathsf{H}, \ell \mapsto \mathsf{h}$$

$$\mathsf{h} ::= \lambda[\overline{\alpha}]\,(\overline{\mathsf{x} : \tau}).\mathsf{t} \mid \langle \mathsf{v}, \ldots, \mathsf{v} \rangle$$

$\boxed{\langle \mathsf{H} \mid \mathsf{e} \rangle \longmapsto \langle \mathsf{H}' \mid \mathsf{e}' \rangle}$ Reduction Relation (selected cases)

$$\langle \mathsf{H} \mid (\mathsf{t}, \mathsf{H}') \rangle \longmapsto \langle (\mathsf{H}, \mathsf{H}') \mid \mathsf{t} \rangle \qquad \mathrm{dom}(\mathsf{H}) \cap \mathrm{dom}(\mathsf{H}') = \emptyset$$

$$\langle \mathsf{H} \mid \mathsf{E}[\ell\ [\overline{\tau'}]\ \overline{\mathsf{v}}] \rangle \longmapsto \langle \mathsf{H} \mid \mathsf{E}[\mathsf{t}[\overline{\tau'}/\overline{\alpha}][\overline{\mathsf{v}}/\overline{\mathsf{x}}]] \rangle \quad \mathsf{H}(\ell) = \lambda[\overline{\alpha}]\,(\overline{\mathsf{x} : \tau}).\mathsf{t}$$

Allocation: **C** to **A**

$\boxed{\tau^{\mathcal{A}}}$ Type Translation

$$\alpha^{\mathcal{A}} = \alpha \qquad \forall[\overline{\alpha}].(\overline{\tau}) \to \tau'^{\mathcal{A}} = \mathsf{box}\,\forall[\overline{\alpha}].(\overline{\tau^{\mathcal{A}}}) \to \tau'^{\mathcal{A}}$$

$$\mathrm{unit}^{\mathcal{A}} = \mathsf{unit} \qquad \exists\alpha.\tau^{\mathcal{A}} = \exists\alpha.\tau^{\mathcal{A}}$$

$$\mathrm{int}^{\mathcal{A}} = \mathsf{int} \qquad \mu\alpha.\tau^{\mathcal{A}} = \mu\alpha.\tau^{\mathcal{A}}$$

$$\langle\tau_1, \ldots, \tau_n\rangle^{\mathcal{A}} = \mathsf{box}\,\langle(\tau_1{}^{\mathcal{A}}), \ldots (\tau_n{}^{\mathcal{A}})\rangle$$

$\boxed{\Delta; \Gamma; \vdash \mathrm{e} : \tau \rightsquigarrow (\mathsf{t}, \mathsf{H} : \Psi)}$ where $\Delta; \Gamma \vdash \mathrm{e} : \tau$, $\cdot \vdash \mathsf{H} : \Psi$, and
$\cdot; \Delta^{\mathcal{A}}; \Gamma^{\mathcal{A}} \vdash (\mathsf{t}, \mathsf{H}) : \tau^{\mathcal{A}}$

Interoperability: **C** and **A**

$$\tau ::= \cdots \mid \mathbf{L}\langle \tau \rangle$$
$$\tau ::= \cdots \mid \lceil \alpha \rceil \mid \lceil \alpha \rceil$$

$$\lceil \alpha \rceil [\tau / \alpha] = (\tau^{\langle \mathcal{C} \rangle})^{\langle \mathcal{A} \rangle}$$
$$\lceil \alpha \rceil [\tau / \alpha] = \tau^{\langle \mathcal{A} \rangle}$$

$$\frac{\Psi; \Delta; \Gamma \vdash \mathbf{e} : \tau^{\langle \mathcal{A} \rangle}}{\Psi; \Delta; \Gamma \vdash {}^{\tau}\mathcal{C}\mathcal{A}\mathbf{e} : \tau} \qquad \frac{\Psi; \Delta; \Gamma \vdash \mathbf{e} : \tau}{\Psi; \Delta; \Gamma \vdash \mathcal{A}\mathcal{C}^{\tau}\,\mathbf{e} : \tau^{\langle \mathcal{A} \rangle}}$$

Interoperability: C and A

$$\boxed{\mathbf{AC}^{\tau}(\mathbf{v}, M) = (\mathbf{v}, M')}$$

$$\mathbf{AC}^{\langle\overline{\tau}\rangle}(\langle\overline{\mathbf{v}}\rangle, M_1) = (\ell, (M_{n+1}, \ell \mapsto \langle\overline{\mathbf{v}}\rangle))$$

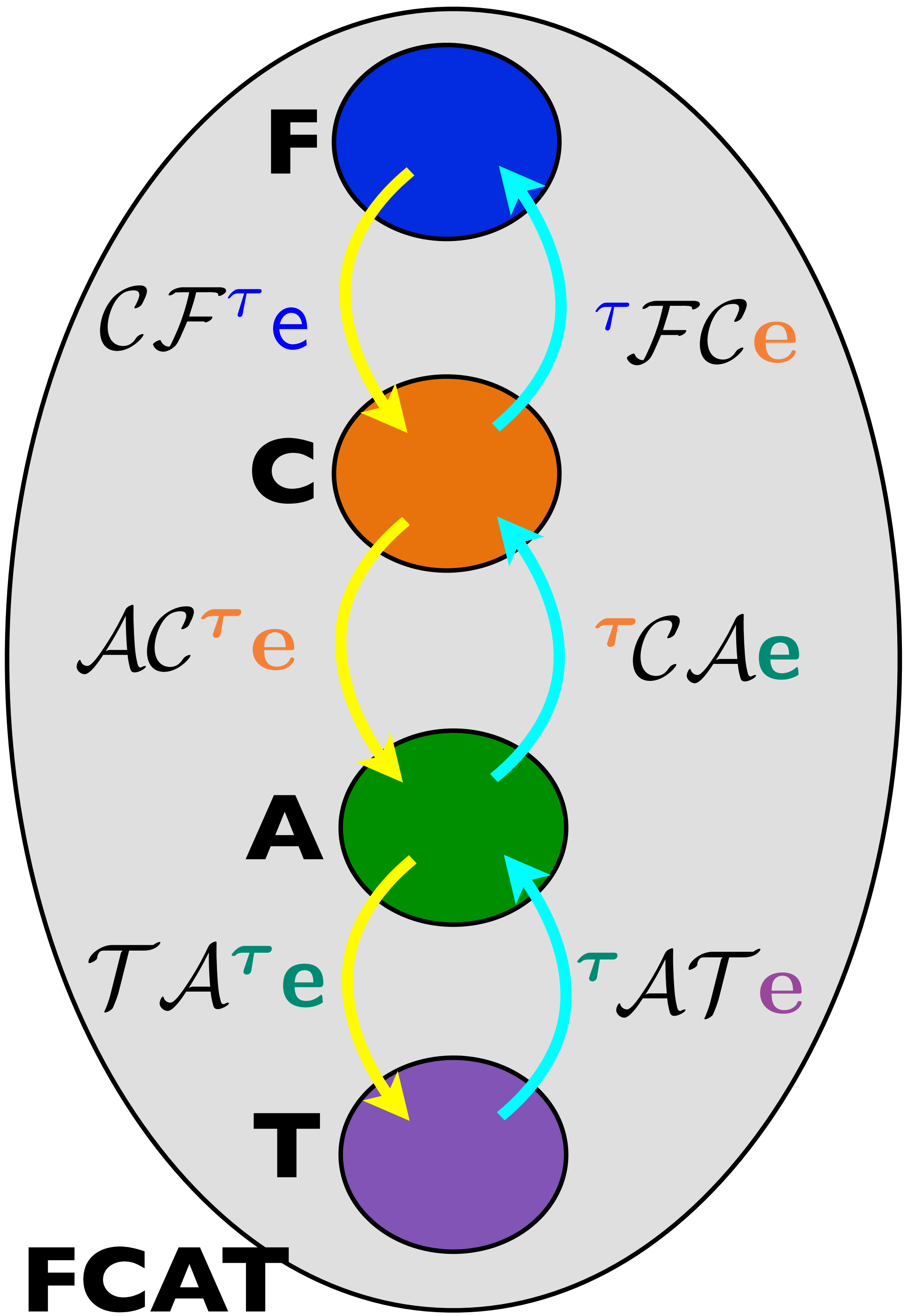
$$\text{where } \mathbf{AC}^{\tau_{\mathbf{i}}}(\mathbf{v}_{\mathbf{i}}, M_i) = (\mathbf{v}_{\mathbf{i}}, M_{i+1})$$

$$\boxed{{}^{\tau}\mathbf{CA}(\mathbf{v}, M) = (\mathbf{v}, M')}$$

$$^{\langle\overline{\tau}\rangle}\mathbf{CA}(\ell, M_1) = (\langle\overline{\mathbf{v}}\rangle, M_{n+1})$$

$$\text{where } M_1(\ell) = \langle\overline{\mathbf{v}}\rangle \text{ and } {}^{\tau_{\mathbf{i}}}\mathbf{CA}(\mathbf{v}_{\mathbf{i}}, M_i) = (\mathbf{v}_{\mathbf{i}}, M_{i+1})$$

Challenges / Roadmap



F+C: Interoperability semantics with type abstraction in both languages

C+A: Interoperability when compiler pass allocates code & tuples on heap

A+T: What is e? What is v? How to define contextual equiv. for TAL *components*? How to define logical relation?

T

$$
\begin{array}{lcll}
\tau & ::= & \alpha \mid \mathbf{unit} \mid \mathbf{int} \mid \exists\alpha.\tau \mid \mu\alpha.\tau & \textit{Type} \\
 & \mid & \mathbf{ref}\ \langle\tau,\ldots,\tau\rangle \mid \mathbf{box}\ \psi & \\
\psi & ::= & \forall[\Delta].\{\chi;\sigma\}^{\mathrm{q}} \mid \langle\tau,\ldots,\tau\rangle & \textit{Heap value type} \\
\chi & ::= & \cdot \mid \chi, \mathrm{r} : \tau & \textit{Register file type} \\
\sigma & ::= & \zeta \mid \bullet \mid \tau :: \sigma & \textit{Stack type} \\
\mathrm{q} & ::= & \epsilon \mid \mathrm{r} \mid \mathrm{i} \mid \mathbf{end}[\tau;\sigma] & \textit{Return marker} \\
\Delta & ::= & \cdot \mid \Delta, \alpha \mid \Delta, \zeta \mid \Delta, \epsilon & \textit{Type variable environment} \\
\omega & ::= & \tau \mid \sigma \mid \mathrm{q} & \textit{Instantiation of type variable} \\
\mathrm{r} & ::= & \mathtt{r1} \mid \mathtt{r2} \mid \cdots \mid \mathtt{r7} \mid \mathtt{ra} & \textit{Register} \\
\mathrm{h} & ::= & \mathbf{code}[\Delta]\{\chi;\sigma\}^{\mathrm{q}}.\mathrm{I} \mid \langle \mathrm{w},\ldots,\mathrm{w}\rangle & \textit{Heap value} \\
\mathrm{w} & ::= & () \mid \mathrm{n} \mid \ell \mid \mathbf{pack}\langle\tau,\mathrm{w}\rangle\ \mathbf{as}\ \exists\alpha.\tau & \textit{Word value} \\
 & \mid & \mathbf{fold}_{\mu\alpha.\tau}\ \mathrm{w} \mid \mathrm{w}[\omega] & \\
\mathrm{u} & ::= & \mathrm{w} \mid \mathrm{r} \mid \mathbf{pack}\langle\tau,\mathrm{u}\rangle\ \mathbf{as}\ \exists\alpha.\tau & \textit{Small value} \\
 & \mid & \mathbf{fold}_{\mu\alpha.\tau}\ \mathrm{u} \mid \mathrm{u}[\omega] & \\
\mathrm{I} & ::= & \iota;\mathrm{I} \mid \mathtt{jmp}\ \mathrm{u} \mid \mathtt{ret}\ \mathrm{q}, \mathrm{r} & \textit{Instruction sequence}
\end{array}
$$

T

$$
\begin{array}{rcll}
\iota & ::= & \mathbf{aop}\ \mathbf{r_d}, \mathbf{r_s}, \mathbf{u} \mid \mathtt{bnz}\ \mathbf{r}, \mathbf{u} \mid \mathtt{mv}\ \mathbf{r_d}, \mathbf{u} & \textit{Instruction} \\
& \mid & \mathtt{ralloc}\ \mathbf{r_d}, \mathbf{n} \mid \mathtt{balloc}\ \mathbf{r_d}, \mathbf{n} \mid \mathtt{ld}\ \mathbf{r_d}, \mathbf{r_s}[\mathbf{i}] \mid \mathtt{st}\ \mathbf{r_d}[\mathbf{i}], \mathbf{r_s} & \\
& \mid & \mathtt{unpack}\ \langle \alpha, \mathbf{r_d} \rangle\ \mathbf{u} \mid \mathtt{unfold}\ \mathbf{r_d}, \mathbf{u} \mid \mathtt{salloc}\ \mathbf{n} \mid \mathtt{sfree}\ \mathbf{n} & \\
& \mid & \mathtt{sld}\ \mathbf{r_d}, \mathbf{i} \mid \mathtt{sst}\ \mathbf{i}, \mathbf{r_s} & \\
\mathbf{aop} & ::= & \mathtt{add} \mid \mathtt{sub} \mid \mathtt{mult} & \textit{Arithmetic operation} \\
\mathbf{e} & ::= & (\mathbf{I}, \mathbf{H}) \mid \mathbf{I} & \textit{Component} \\
\mathbf{v} & ::= & \mathtt{ret}\ \mathbf{q}, \mathbf{r} & \textit{Term value} \\
\mathbf{E} & ::= & (\mathbf{E_I}, \cdot) & \textit{Evaluation context} \\
\mathbf{E_I} & ::= & [\cdot] & \textit{Instruction evaluation context} \\
\mathbf{H} & ::= & \cdot \mid \mathbf{H}, \ell \mapsto \mathbf{h} & \textit{Heap or Heap fragment} \\
\mathbf{R} & ::= & \cdot \mid \mathbf{R}, \mathbf{r} \mapsto \mathbf{w} & \textit{Register file} \\
\mathbf{S} & ::= & \mathbf{nil} \mid \mathbf{w} :: \mathbf{S} & \textit{Stack} \\
\mathbf{M} & ::= & (\mathbf{H}, \mathbf{R}, \mathbf{S} : \sigma) & \textit{Memory}
\end{array}
$$

$$\boxed{\langle \mathbf{M} \mid \mathbf{e} \rangle \longmapsto \langle \mathbf{M'} \mid \mathbf{e'} \rangle}$$

Well-typed Components in $\mathbf{T}$

$$\boxed{\Psi; \Delta; \chi; \sigma; \mathbf{q} \vdash \mathbf{e} : \tau; \sigma'}$$

$$\frac{\begin{array}{cc} \Psi \vdash \mathbf{H} : \Psi_{\mathbf{e}} & \text{boxheap}(\Psi_{\mathbf{e}}) \\ \text{ret-type}(\mathbf{q}, \chi, \sigma) = \tau; \sigma' & (\Psi, \Psi_{\mathbf{e}}); \Delta; \chi; \sigma; \mathbf{q} \vdash \mathbf{I} \end{array}}{\Psi; \Delta; \chi; \sigma; \mathbf{q} \vdash (\mathbf{I}, \mathbf{H}) : \tau; \sigma'}$$

Equivalence of **T** Components: Tricky!

Logical relations: related inputs to related outputs

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{(W, \lambda \mathsf{x}.\mathsf{e}_1, \lambda \mathsf{x}.\mathsf{e}_1) \mid \ldots\}$$

$$\mathcal{HV}[\![\forall[\boldsymbol{\Delta}].\{\boldsymbol{\chi}; \boldsymbol{\sigma}\}^{\mathrm{q}}]\!] = \{(W, \mathbf{code}[\boldsymbol{\Delta}]\{\boldsymbol{\chi}; \boldsymbol{\sigma}\}^{\mathrm{q}}.\mathbf{I_1}, \mathbf{code}[\boldsymbol{\Delta}]\{\boldsymbol{\chi}; \boldsymbol{\sigma}\}^{\mathrm{q}}.\mathbf{I_2}) \mid \ldots\}$$

Equivalence of **T** Components: Tricky!

Logical relations: related inputs to related outputs

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{(W, \lambda \mathsf{x}.\mathsf{e}_1, \lambda \mathsf{x}.\mathsf{e}_1) \mid \ldots\}$$

$$\mathcal{HV}[\![\forall[\boldsymbol{\Delta}].\{\chi;\sigma\}^{\mathrm{q}}]\!] = \{(W, \mathbf{code}[\boldsymbol{\Delta}]\{\chi;\sigma\}^{\mathrm{q}}.\mathbf{I}_1, \mathbf{code}[\boldsymbol{\Delta}]\{\chi;\sigma\}^{\mathrm{q}}.\mathbf{I}_2) \mid \ldots\}$$

Equivalence of **T** Components: Tricky!

Logical relations: related inputs to related outputs

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{(W, \lambda \mathsf{x}.\mathsf{e}_1, \lambda \mathsf{x}.\mathsf{e}_1) \mid \ldots\}$$

$$\mathcal{HV}[\![\forall[\boldsymbol{\Delta}].\{\boldsymbol{\chi};\boldsymbol{\sigma}\}^{\mathbf{q}}]\!] = \{(W, \mathbf{code}[\boldsymbol{\Delta}]\{\boldsymbol{\chi};\boldsymbol{\sigma}\}^{\mathbf{q}}.\mathbf{I}_1, \mathbf{code}[\boldsymbol{\Delta}]\{\boldsymbol{\chi};\boldsymbol{\sigma}\}^{\mathbf{q}}.\mathbf{I}_2) \mid \ldots\}$$

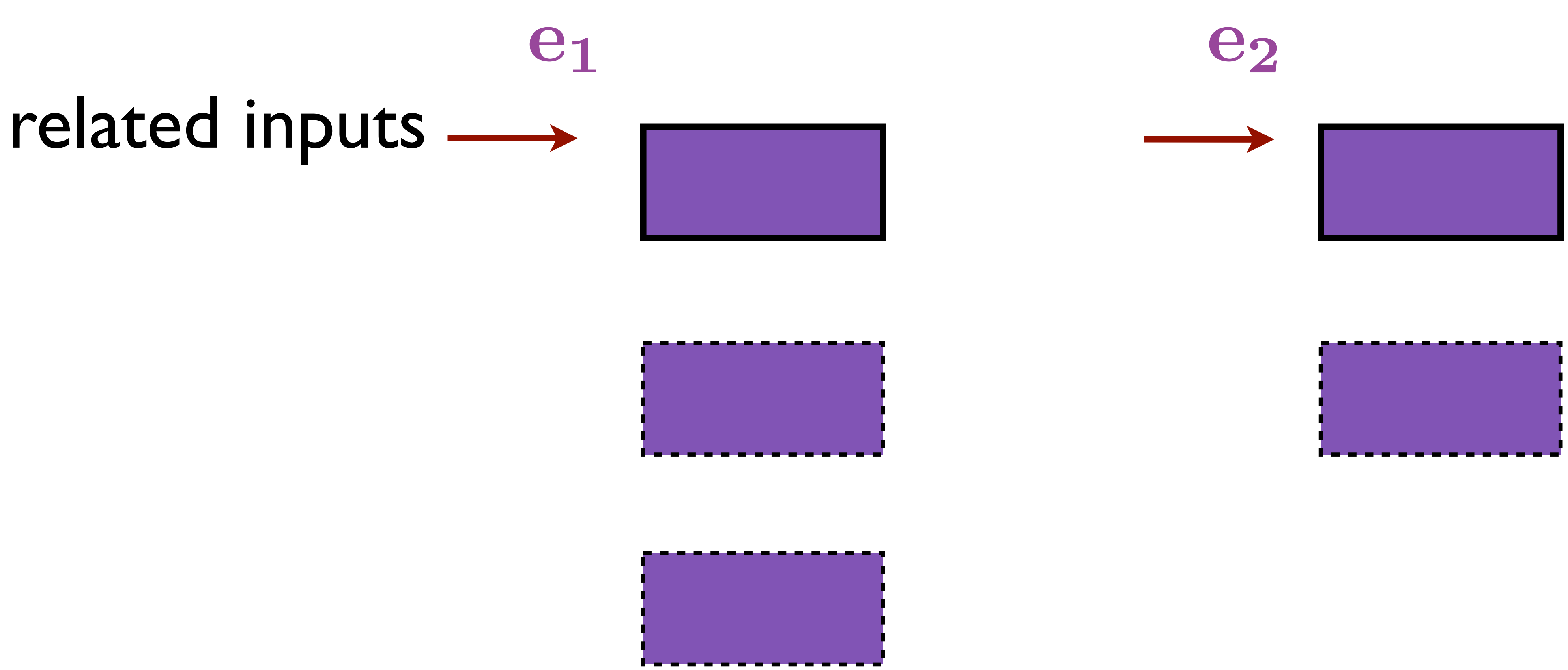
 = $\mathbf{code}[\boldsymbol{\Delta}]\{\boldsymbol{\chi};\boldsymbol{\sigma}\}^{\mathbf{q}}.\mathbf{I}$

Equivalence of **T** Components: Tricky!

Logical relations: related inputs to related outputs

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{(W, \lambda \mathsf{x}.\mathsf{e}_1, \lambda \mathsf{x}.\mathsf{e}_1) \mid \ldots\}$$

$$\mathcal{HV}[\![\forall[\boldsymbol{\Delta}].\{\chi;\sigma\}^{\mathrm{q}}]\!] = \{(W, \mathbf{code}[\boldsymbol{\Delta}]\{\chi;\sigma\}^{\mathrm{q}}.\mathbf{I}_1, \mathbf{code}[\boldsymbol{\Delta}]\{\chi;\sigma\}^{\mathrm{q}}.\mathbf{I}_2) \mid \ldots\}$$

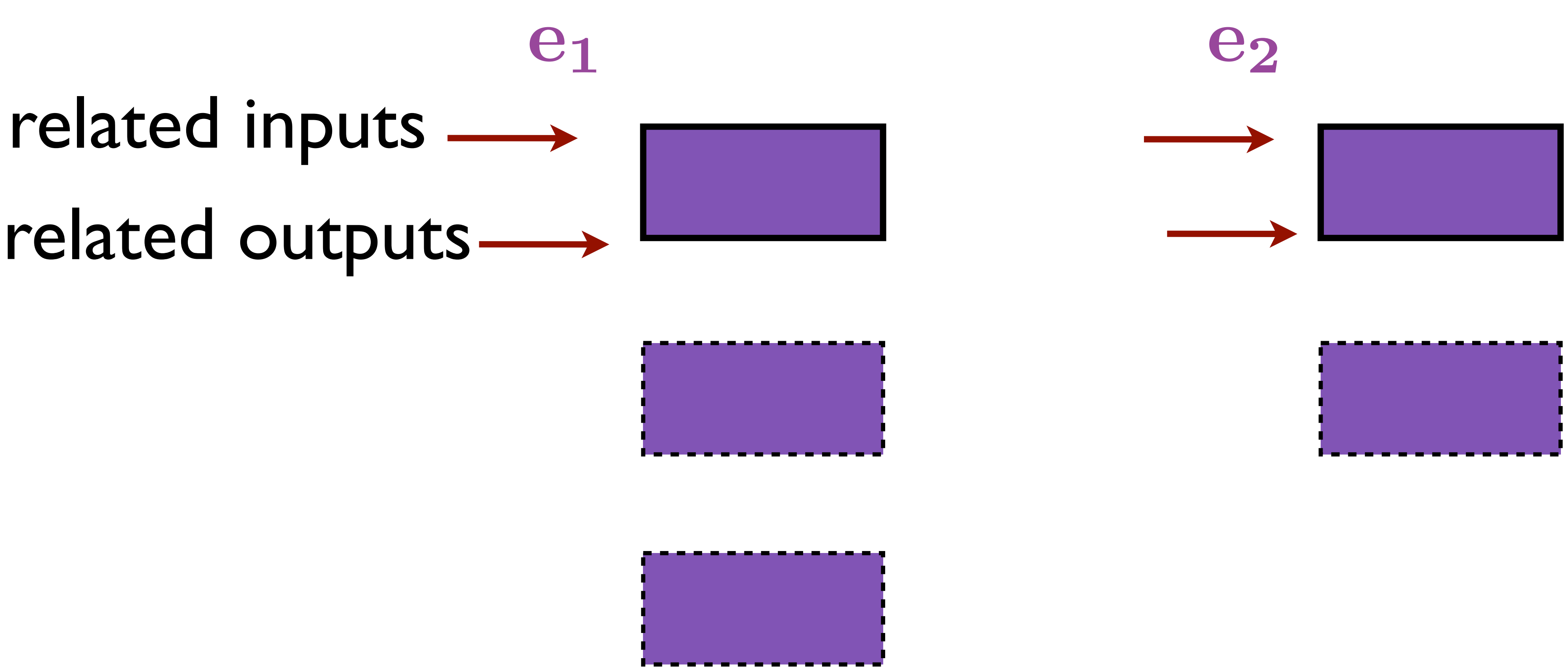


Equivalence of **T** Components: Tricky!

Logical relations: related inputs to related outputs

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{(W, \lambda \mathsf{x}.\mathsf{e}_1, \lambda \mathsf{x}.\mathsf{e}_1) \mid \ldots\}$$

$$\mathcal{HV}[\![\forall[\boldsymbol{\Delta}].\{\chi; \sigma\}^{\mathbf{q}}]\!] = \{(W, \mathbf{code}[\boldsymbol{\Delta}]\{\chi; \sigma\}^{\mathbf{q}}.\mathbf{I}_1, \mathbf{code}[\boldsymbol{\Delta}]\{\chi; \sigma\}^{\mathbf{q}}.\mathbf{I}_2) \mid \ldots\}$$

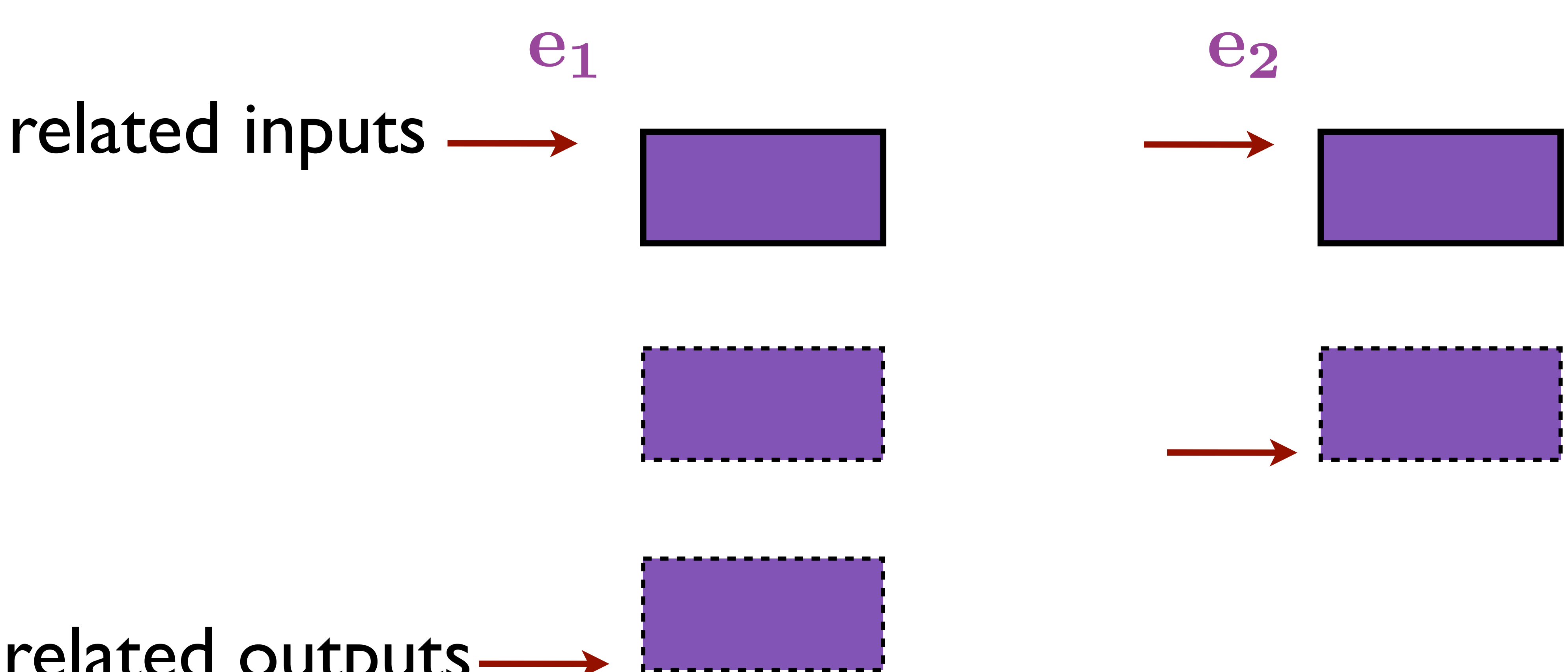


Equivalence of $\mathbf{T}$ Components: Tricky!

Logical relations: related inputs to related outputs

$$\mathcal{V}[\![\tau_1 \to \tau_2]\!] = \{(W, \lambda\mathsf{x}.\mathsf{e}_1, \lambda\mathsf{x}.\mathsf{e}_1) \mid \ldots\}$$

$$\mathcal{HV}[\![\forall[\Delta].\{\chi;\sigma\}^{\mathrm{q}}]\!] = \{(W, \mathbf{code}[\Delta]\{\chi;\sigma\}^{\mathrm{q}}.\mathbf{I}_1, \mathbf{code}[\Delta]\{\chi;\sigma\}^{\mathrm{q}}.\mathbf{I}_2) \mid \ldots\}$$



Code Generation: **A** to **T**

$\boxed{\tau^{\mathcal{T}}}$ Type translation

$$
\begin{aligned}
&\mathbf{box}\,\forall[\overline{\alpha}].(\tau_1,\ldots,\tau_n) \to \tau'^{\mathcal{T}} \\
&\quad = \mathbf{box}\,\forall[\overline{\alpha},\zeta,\epsilon]. \\
&\qquad\qquad \{\mathtt{ra} : \mathbf{box}\,\forall[].\{\mathtt{r1} : \tau'^{\mathcal{T}};\zeta\}^{\epsilon}; \\
&\qquad\qquad\ \ \tau_n{}^{\mathcal{T}} :: \cdots :: \tau_1{}^{\mathcal{T}} :: \zeta\}^{\mathtt{ra}}
\end{aligned}
$$

Code Generation: **A** to **T**

$\boxed{\tau^{\mathcal{T}}}$ Type translation

$$\begin{aligned}
&\mathbf{box}\,\forall[\overline{\alpha}].(\tau_1,\ldots,\tau_n)\rightarrow\tau'^{\mathcal{T}}\\
&\quad=\mathbf{box}\,\forall[\overline{\alpha},\zeta,\epsilon].\\
&\qquad\qquad\{\mathtt{ra}:\mathbf{box}\,\forall[].\{\mathtt{r1}:\tau'^{\mathcal{T}};\zeta\}^{\epsilon};\\
&\qquad\qquad\ \tau_n{}^{\mathcal{T}}::\cdots::\tau_1{}^{\mathcal{T}}::\zeta\}^{\mathtt{ra}}
\end{aligned}$$

$\boxed{\Psi;\Delta;\Gamma\vdash \mathbf{e}:\tau\rightsquigarrow \mathbf{e}}$ where $\Psi^{\mathcal{T}};(\Delta^{\mathcal{T}},\zeta,\epsilon);\chi;\sigma;\mathtt{ra}\vdash \mathtt{e}:\tau^{\mathcal{T}};\sigma$

for $\chi=\mathtt{ra}:\forall[].\{\mathtt{r1}:\tau^{\mathcal{T}};\sigma\}^{\epsilon}$ and $\sigma=\mathrm{order}(\Gamma,\zeta)^{\mathcal{T}}$

Interoperability: **A** and **T**

$$\frac{\Psi; \Delta; \Gamma; \cdot; \boldsymbol{\sigma}; \mathbf{end}[\boldsymbol{\tau}^{\langle \mathcal{T} \rangle}; \boldsymbol{\sigma}'] \vdash \mathrm{e} : \boldsymbol{\tau}^{\langle \mathcal{T} \rangle}; \boldsymbol{\sigma}'}{\Psi; \Delta; \Gamma; \chi; \boldsymbol{\sigma}; \mathbf{out} \vdash {}^{\boldsymbol{\tau}}\mathcal{A}\mathcal{T}\mathrm{e} : \boldsymbol{\tau}; \boldsymbol{\sigma}'}$$

$$\frac{{}^{\boldsymbol{\tau}}\mathbf{AT}(M.\mathbf{M.R(r)}, M) = (\mathsf{v}, M')}{\langle M \mid E[{}^{\boldsymbol{\tau}}\mathcal{A}\mathcal{T}\mathtt{ret}\,\mathbf{end}[\boldsymbol{\tau}^{\langle \mathcal{T} \rangle}; \boldsymbol{\sigma}], \mathbf{r}]\rangle \longmapsto \langle M' \mid E[\mathsf{v}]\rangle}$$

Interoperability: **A** and **T**

$$\iota \quad ::= \cdots \mid \texttt{import}\ \mathbf{r_d}, {}^{\sigma}\mathcal{T}\mathcal{A}^{\tau}\,\mathbf{e}$$

$$\frac{\mathbf{TA}^{\tau}(\mathtt{v}, M) = (\mathbf{w}, M')}{\langle M \mid E[\texttt{import}\ \mathbf{r_d}, {}^{\sigma'}\mathcal{T}\mathcal{A}^{\tau}\,\mathtt{v}; \mathbf{I}]\rangle \longmapsto \langle M' \mid E[\texttt{mv}\ \mathbf{r_d}, \mathbf{w}; \mathbf{I}]\rangle}$$

Conclusions

- Compiler verification methodology that
  - guarantees correct compilation of components, not just whole programs
  - works for multi-pass compilers
  - supports reasoning about whole programs produced by linking with arbitrary target code
- Interoperability semantics provides specification of when source and target code are related
  - easier to understand compiler correctness theorem
  - but, have to get all the languages to fit together!

Questions?