A Propositional Refinement Type System for Algebraic Effects

(with linear refinement modalities)

Danel Ahman

(joint work with Gordon Plotkin)

TSEM, 10 March 2014

informatics

lfcs Laboratory for Foundations of Computer Science

THE UNIVERSITY of EDINBURGH

Overview

- This work is about:
  - programming languages with computational effects
  - type systems for specifying program behavior
- What happens quite often:
  - a computational effect $\Longrightarrow$ a language
  - a specification style $\Longrightarrow$ a language
  - a language $\Longrightarrow$ a paper
- This talk:
  - modular in the computational effects at hand
  - modularly covering a wide range of specification styles
  - a modal spec. syntax, induced by comp. effects at hand
  - an instance of Katsumata/Tate effect annotation monoids

  - (but has some surprising non-linearity issues)

Single system - different effects - different specs.

- Consider a (fragment of a) simple computational language:

$$\frac{\Gamma \vDash_{\mathsf{v}} V : A}{\Gamma \vDash_{\mathsf{c}} \mathsf{return}\, V : FA} \qquad \frac{\Gamma \vDash_{\mathsf{c}} M : FA \quad \Gamma, x : A \vDash_{\mathsf{c}} N : FB}{\Gamma \vDash_{\mathsf{c}} M\, \mathsf{to}\, x.\, N : FB}$$

$$\frac{\Gamma \vDash_{\mathsf{c}} M_1 : FA \quad \ldots \quad \Gamma \vDash_{\mathsf{c}} M_n : FA}{\Gamma \vDash_{\mathsf{c}} \mathsf{op}(M_1, \ldots, M_n) : FA} \ (\mathsf{op} : n)$$

- Effect annotations (as in effect-and-type systems)
  - $\varepsilon \in \mathcal{P}(\{\mathsf{op}_1, \ldots, \mathsf{op}_m\})$

Single system - different effects - different specs.

- Consider a (fragment of a) simple computational language:

$$\frac{\Gamma \vDash_{v} V : A}{\Gamma \vDash_{c} \mathsf{return}\, V : A \,!\, \emptyset} \qquad \frac{\Gamma \vDash_{c} M : A \,!\, \varepsilon_1 \qquad \Gamma, x : A \vDash_{c} N : B \,!\, \varepsilon_2}{\Gamma \vDash_{c} M \,\mathsf{to}\, x.\, N : B \,!\, \varepsilon_1 \cup \varepsilon_2}$$

$$\frac{\Gamma \vDash_{c} M_1 : A \,!\, \varepsilon_1 \quad \ldots \quad \Gamma \vDash_{c} M_n : A \,!\, \varepsilon_n}{\Gamma \vDash_{c} \mathsf{op}(M_1, \ldots, M_n) : A \,!\, \varepsilon_1 \cup \cdots \cup \varepsilon_n \cup \{\mathsf{op}\}} \ (\mathsf{op} : n)$$

- Effect annotations (as in effect-and-type systems)
  - $\varepsilon \in \mathcal{P}(\{\mathsf{op}_1, \ldots, \mathsf{op}_m\})$

Lucassen & Gifford '88

Wadler & Thiemann '03

Benton et al. '06, '07, '09, ...

Kammar & Plotkin '12

Tate '13

Katsumata '14

Single system - different effects - different specs.

- Consider a (fragment of a) simple communication language:

$$\frac{\Gamma \vDash_{\mathsf{v}} V : A}{\Gamma \vDash_{\mathsf{c}} \mathsf{return}\, V : FA} \qquad \frac{\Gamma \vDash_{\mathsf{c}} M : FA \quad \Gamma, x : A \vDash_{\mathsf{c}} N : FB}{\Gamma \vDash_{\mathsf{c}} M\, \mathsf{to}\, x.\, N : FB}$$

$$\frac{\Gamma \vDash_{\mathsf{c}} M : FA \quad \Gamma \vDash_{\mathsf{c}} N : FA}{\Gamma \vDash_{\mathsf{c}} \mathsf{receive}(M, N) : FA} \qquad \frac{\Gamma \vDash_{\mathsf{c}} M : FA}{\Gamma \vDash_{\mathsf{c}} \mathsf{send}_0(M) : FA} \qquad \frac{\Gamma \vDash_{\mathsf{c}} M : FA}{\Gamma \vDash_{\mathsf{c}} \mathsf{send}_1(M) : FA}$$

- Session refinements (inspired by session types)
  - $S_A ::= end_A \mid !(0).S_A \mid !(1).S_A \mid !(0 \vee 1).S_A \mid ?(S_A; S'_A)$

Single system - different effects - different specs.

- Consider a (fragment of a) simple communication language:

$$\frac{\Gamma \models_{v} V : A}{\Gamma \models_{c} \mathsf{return}\, V : end_A} \qquad \frac{\Gamma \models_{c} M : S_A \quad \Gamma, x : A \models_{c} N : S_B}{\Gamma \models_{c} M \,\mathsf{to}\, x.\, N : S_A; S_B}$$

$$\frac{\Gamma \models_{c} M : S_A \quad \Gamma \models_{c} N : S'_A}{\Gamma \models_{c} \mathsf{receive}(M, N) : ?(S_A, S'_A)} \qquad \frac{\Gamma \models_{c} M : S_A}{\Gamma \models_{c} \mathsf{send}_0(M) : !(0).S_A} \qquad \frac{\Gamma \models_{c} M : S_A}{\Gamma \models_{c} \mathsf{send}_1(M) : !(1).S_A}$$

- Session refinements (inspired by session types)
  - $S_A ::= end_A \mid !(0).S_A \mid !(1).S_A \mid !(0 \vee 1).S_A \mid ?(S_A; S'_A)$

Honda '93

Kobayashi '03,'07

Yoshida & Vasconcelos '07

Wadler '12

Single system - different effects - different specs.

- Consider a (fragment of a) simple stateful language:

$$\frac{\Gamma \models_{\mathsf{v}} V : A}{\Gamma \models_{\mathsf{c}} \mathsf{return}\, V : FA} \qquad \frac{\Gamma \models_{\mathsf{c}} M : FA \quad \Gamma, x : A \models_{\mathsf{c}} N : FB}{\Gamma \models_{\mathsf{c}} M\, \mathsf{to}\, x.\, N : FB}$$

$$\frac{\Gamma \models_{\mathsf{c}} M : FA \quad \Gamma \models_{\mathsf{c}} N : FA}{\Gamma \models_{\mathsf{c}} \mathsf{lookup}(M, N) : FA} \qquad \frac{\Gamma \models_{\mathsf{c}} M : FA}{\Gamma \models_{\mathsf{c}} \mathsf{update}_0(M) : FA} \qquad \frac{\Gamma \models_{\mathsf{c}} M : FA}{\Gamma \models_{\mathsf{c}} \mathsf{update}_1(M) : FA}$$

- Hoare refinements (inspired by Hoare monad in F7, HTT)
  - $\{P\}\, A\, \{Q\}$
  - P, Q predicates on state
  - here $P, Q \in \mathcal{P}(\{0, 1\})$

Single system - different effects - different specs.

- Consider a (fragment of a) simple stateful language:

$$\frac{\Gamma \vDash_{\mathsf{v}} V : A}{\Gamma \vDash_{\mathsf{c}} \mathsf{return}\, V : \{P\}\, A\, \{P\}} \qquad \frac{\Gamma \vDash_{\mathsf{c}} M : \{P\}\, A\, \{Q\} \quad \Gamma, x : A \vDash_{\mathsf{c}} N : \{Q\}\, A\, \{R\}}{\Gamma \vDash_{\mathsf{c}} M\, \mathsf{to}\, x.\, N : \{P\}\, A\, \{R\}}$$

$$\frac{\Gamma \vDash_{\mathsf{c}} M : \{P \cap \{0\}\}\, A\, \{Q\} \quad \Gamma \vDash_{\mathsf{c}} N : \{P \cap \{1\}\}\, A\, \{Q\}}{\Gamma \vDash_{\mathsf{c}} \mathsf{lookup}(M, N) : \{P\}\, A\, \{Q\}}$$

$$\frac{\Gamma \vDash_{\mathsf{c}} M : \{\{0\}\}\, A\, \{Q\}}{\Gamma \vDash_{\mathsf{c}} \mathsf{update}_0(M) : \{P\}\, A\, \{Q\}} \qquad \frac{\Gamma \vDash_{\mathsf{c}} M : \{\{1\}\}\, A\, \{Q\}}{\Gamma \vDash_{\mathsf{c}} \mathsf{update}_1(M) : \{P\}\, A\, \{Q\}}$$

- Hoare refinements (refined state monad, Dijkstra monad, HTT)
  - $\{P\}\, A\, \{Q\}$ Borgström et al. '11 Swamy et al. '13 Nanevski et al. '08
  - P, Q predicates on state
  - here $P, Q \in \mathcal{P}(\{0, 1\})$

Single system - different effects - different specs.

type-and-effect systems
+ optimizations

input/output
+ sessions and protocols

state
+ Hoare logic

$\subseteq$ $\qquad$ $\subseteq$ $\qquad$ $\subseteq$

CBPV$_{\mathrm{ref}}$

specifications on values
(even/odd numbers)

computational language (CBPV)
with algebraic effects

Single system - different effects - different specs.

type-and-effect systems
+ optimizations

$\Gamma \vdash M : \sigma \,!\, \varepsilon$

input/output
+ sessions and protocols

$\Gamma \vdash M : ?(!(1).end; end)$

state
+ Hoare logic

$\Gamma \vdash M : \{P\}\sigma\{Q\}$

$\subseteq$ $\subseteq$ $\subseteq$

$\boxed{\text{CBPV}_{\text{ref}}}$

$\downarrow$

computational language (CBPV) with algebraic effects

Algebraic effects

- Assume: an algebraic theory $\mathcal{T}_{\text{eff}}$ of computational effects
  - collection of operation symbols
  - collection of equations

Plotkin & Power '02,'03

- Theory $\mathcal{T}_{\text{ND}}$ of non-determinism
  - one binary operation $\oplus : 2$
  - equations $x \oplus x = x$,
    $$x \oplus y = y \oplus x,$$
    $$(x \oplus y) \oplus z = x \oplus (y \oplus z)$$

- Theory $\mathcal{T}_{\text{I/O}}$ of input/output (of bits, over a fixed channel)
  - three operations receive $: 2$,
    $$\text{send}_0 : 1,$$
    $$\text{send}_1 : 1$$
  - no equations

Algebraic effects

- Assume: an algebraic theory $\mathcal{T}_{\mathsf{eff}}$ of computational effects
  - collection of operation symbols
  - collection of equations

Plotkin & Power '02,'03

- Theory $\mathcal{T}_{\mathsf{S}}$ of global state (of bits)
  - three operations $\mathsf{lookup} : 2$,

    $\mathsf{update}_0 : 1$,

    $\mathsf{update}_1 : 1$
  - equations

$$x = \mathsf{lookup}(\mathsf{update}_0(x), \mathsf{update}_1(x))$$

$$\mathsf{update}_0(\mathsf{lookup}(x, y)) = \mathsf{update}_0(x)$$

$$\mathsf{update}_1(\mathsf{lookup}(x, y)) = \mathsf{update}_1(y)$$

$$\mathsf{update}_i(\mathsf{update}_j(x)) = \mathsf{update}_j(x)$$

Single system - different effects - different specs.

type-and-effect systems
+ optimizations

$\Gamma \vdash M : \sigma \, ! \, \varepsilon$

input/output
+ sessions and protocols

$\Gamma \vdash M : ?(!(1).end; end)$

state
+ Hoare logic

$\Gamma \vdash M : \{P\}\sigma\{Q\}$

$\subseteq \quad \subseteq \quad \supseteq$

CBPV$_{\text{ref}}$

computational language (CBPV) with algebraic effects

Call-by-Push-Value with algebraic effects

Levy '04

- Strict separation into values and computations
- Types:
  - $A ::= 1 \mid 0 \mid A_1 \times A_2 \mid A_1 + A_2 \mid U\underline{B}$
  - $\underline{B} ::= FA \mid \underline{B}_1 \times \underline{B}_2 \mid A \to \underline{B}$
- Terms:
  - $V ::= \star \mid \langle V_1, V_2 \rangle \mid \mathsf{inj}_i^{A_1+A_2} V \mid \mathsf{thunk}\, M$
  - $M ::= \mathsf{return}\, V \mid M\, \mathsf{to}\, x : A.\, N \mid \mathsf{op}_{\underline{B}}(M_1, \ldots, M_n) \mid$
    $\lambda x : A.M \mid MV \mid \mathsf{force}\, V \mid \langle M_1, M_2 \rangle \mid \mathsf{fst}\, M \mid \mathsf{snd}\, M \mid$
    $\mathsf{match}\, V\, \mathsf{as}\, (x_1 : A_1, x_2 : A_2).M \mid \mathsf{match}\, V\, \mathsf{as}\, \{\} \mid$
    $\mathsf{match}\, V\, \mathsf{as}\, \{\mathsf{inj}_1\, (x_1 : A_1) \mapsto M_1, \mathsf{inj}_2\, (x_2 : A_2) \mapsto M_2\}$
- Equational theory:
  - standard $\beta\eta$-equations +
    algebraic equations $\Gamma \vDash_{c} M \oplus N = N \oplus M : \underline{B}$

Call-by-Push-Value with algebraic effects

- Models of CBPV with algebraic effects:

$$\begin{tikzcd} \mathsf{Set} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathsf{Alg} \arrow[ll, bend left, "U"] \end{tikzcd}$$

  where Alg is the category of $\mathcal{T}_{\mathsf{eff}}$-algebras in Set

- $[\![A]\!]$: Set
- $[\![\underline{B}]\!]$: Alg
- $[\![\Gamma \vdash_{\mathsf{v}} V : A]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]$
- $[\![\Gamma \vdash_{\mathsf{c}} M : \underline{B}]\!] : [\![\Gamma]\!] \longrightarrow U[\![\underline{B}]\!]$

- $[\![\Gamma \vdash_{\mathsf{c}} \mathsf{op}_{\underline{B}}(M_1, \ldots, M_n) : \underline{B}]\!](\gamma) = [\![\mathsf{op}]\!]_{[\![\underline{B}]\!]}([\![M_1]\!](\gamma), \ldots, [\![M_n]\!](\gamma))$
  - where $[\![\mathsf{op}]\!]_{[\![\underline{B}]\!]} : U[\![\underline{B}]\!] \times \cdots \times U[\![\underline{B}]\!] \longrightarrow U[\![\underline{B}]\!]$

Single system - different effects - different specs.

type-and-effect systems
+ optimizations

$\Gamma \vdash M : \sigma \,!\, \varepsilon$

input/output
+ sessions and protocols

$\Gamma \vdash M : ?(!(1).end; end)$

state
+ Hoare logic

$\Gamma \vdash M : \{P\}\sigma\{Q\}$

$\subseteq \quad \subseteq \quad \subseteq$

$\boxed{\text{CBPV}_{\text{ref}}}$

$\downarrow$

two-level semantics

computational language (CBPV)
with algebraic effects

Two-level CBPV$_{\mathrm{ref}}$ semantics

- The big picture

$$
\begin{tikzcd}
 & & \boxed{\langle \mathcal{A} : \mathcal{T}_{\mathsf{eff}}\text{-}\mathsf{Alg}, \mathcal{R} \subseteq U\mathcal{A} \times U\mathcal{A} \rangle} \arrow[d] \\
\boxed{\langle X : \mathsf{Set}, \mathcal{R} \subseteq X \times X \rangle} \arrow[r] & \mathsf{Rel}(\mathsf{Set}) \arrow[r, bend left, "F^{\mathsf{ref}}"] \arrow[r, phantom, "\perp"] \arrow[d, "|-|_v"'] & \mathsf{Ref}(\mathsf{Alg}) \arrow[l, bend left, "U^{\mathsf{ref}}"] \arrow[d, "|-|_c"] \\
 & \mathsf{Set} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathsf{Alg} \arrow[l, bend left, "U"]
\end{tikzcd}
$$

where Alg is the category of $\mathcal{T}_{\mathsf{eff}}$-algebras in Set

Refinement types

- To accommodate logical properties in your favorite type system
- Intersection types $\sigma_1 \wedge \sigma_2$, union types $\sigma_1 \vee \sigma_2$

Freeman & Pfenning '91, ...

  - practical propositional reasoning
  - ref. types not explicitly connected to underlying system
- First-order logic in types $\{x : \sigma \mid \varphi\}$

Denney '98, ...

  - reasoning in full first-order logic
  - ref. types explicitly connected to underlying system
    - $\vdash \sigma : \mathsf{Ref}(A)$ by using indexed kinds
- $\mathsf{CBPV}_{\mathrm{ref}}$ is in-between these two approaches

Refinement types in CBPV$_{\text{ref}}$

- Separated into value and computation refinement types

$$
\begin{array}{rcll}
\sigma & ::= & 1 \mid 0 \mid \sigma_1 \times \sigma_2 \mid \sigma_1 + \sigma_2 \mid U^{\mathsf{ref}}\underline{\tau} & \leftarrow \boxed{\text{structural}} \\
& & \sigma_1 \wedge \sigma_2 \mid \sigma_1 \vee \sigma_2 \mid \bot_A & \leftarrow \boxed{\text{logical}} \\
\underline{\tau} & ::= & F^{\mathsf{ref}}\sigma \mid \underline{\tau}_1 \times \underline{\tau}_2 \mid \sigma \to \underline{\tau} \mid & \\
& & \langle \mathsf{op} \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_n) \mid & \leftarrow \boxed{\text{operation modalities}} \\
& & X \mid \mu X : \mathsf{Ref}(\underline{B}).\underline{\tau} \mid \underline{\tau}_1 \wedge \underline{\tau}_2 \mid \underline{\tau}_1 \vee \underline{\tau}_2 \mid \bot_{\underline{B}} &
\end{array}
$$

- Well-kinded ref. types over CBPV types, e.g.:

$$
\frac{\vdash \sigma : \mathsf{Ref}(A) \quad \Delta \vdash \underline{\tau} : \mathsf{Ref}(\underline{B})}{\Delta \vdash \sigma \to \underline{\tau} : \mathsf{Ref}(A \to \underline{B})}
\qquad
\frac{\Delta \vdash \sigma_1 : \mathsf{Ref}(A) \quad \Delta \vdash \sigma_2 : \mathsf{Ref}(A)}{\Delta \vdash \sigma_1 \wedge \sigma_2 : \mathsf{Ref}(A)}
$$

Logical reasoning

- Main tool: subtyping
- Two subtyping relations
  - $\Delta \models^{\mathsf{r}}_{\mathsf{v}} \sigma_1 \sqsubseteq_A \sigma_2$
  - $\Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau}_1 \sqsubseteq_{\underline{B}} \underline{\tau}_2$
- Again, structural

$$\frac{\models^{\mathsf{r}}_{\mathsf{v}} \sigma_2 \sqsubseteq_A \sigma_1 \quad \Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau}_1 \sqsubseteq_{\underline{B}} \underline{\tau}_2}{\Delta \models^{\mathsf{r}}_{\mathsf{c}} \sigma_1 \to \underline{\tau}_1 \sqsubseteq_{A \to \underline{B}} \sigma_2 \to \underline{\tau}_2} \qquad \frac{\Delta \models^{\mathsf{r}}_{\mathsf{v}} \sigma_1 \sqsubseteq_A \sigma_2}{\Delta \models^{\mathsf{r}}_{\mathsf{c}} F^{\mathsf{ref}}\sigma_1 \sqsubseteq_{FA} F^{\mathsf{ref}}\sigma_2}$$

$$\Delta \models^{\mathsf{r}}_{\mathsf{c}} \langle \mathsf{op} \rangle_{A \to \underline{B}}(\sigma \to \underline{\tau}_1, \ldots, \sigma \to \underline{\tau}_n) = \sigma \to \langle \mathsf{op} \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_n)$$

- and logical rules

$$\frac{\Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau} \sqsubseteq \underline{\tau}'[\mu X : \mathsf{Ref}(\underline{B}).\underline{\tau}'/X]}{\Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau} \sqsubseteq \mu X : \mathsf{Ref}(\underline{B}).\underline{\tau}'} \qquad \frac{\Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau}[\underline{\tau}'/X] \sqsubseteq \underline{\tau}'}{\Delta \models^{\mathsf{r}}_{\mathsf{c}} \mu X : \mathsf{Ref}(\underline{B}).\underline{\tau} \sqsubseteq \underline{\tau}'}$$

Logical reasoning

- But there is a small problem:
  - equations in $\mathcal{T}_{\mathsf{eff}}$ might not induce valid subtyping inequalities
  - consider $x_1, \ldots, x_n \vdash x = x \oplus x$
  - only $\Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau} \sqsubseteq_{\underline{B}} \langle \mathsf{op} \rangle_{\underline{B}}(\underline{\tau}, \underline{\tau})$ is valid
  - $\Delta \models^{\mathsf{r}}_{\mathsf{c}} \langle \mathsf{op} \rangle_{\underline{B}}(\underline{\tau}, \underline{\tau}) \sqsubseteq_{\underline{B}} \underline{\tau}$ is not valid!
- So we introduce a linearity constraint:

$$\frac{x_1, \ldots, x_n \vdash t = u \qquad t \text{ linear} \qquad \Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau}_i \sqsubseteq_{\underline{B}} \underline{\tau}'_i \quad (i \in \{1, \ldots, n\})}{\Delta \models^{\mathsf{r}}_{\mathsf{c}} t[\underline{\tau}_1/x_1, \ldots, \underline{\tau}_n/x_n]^{\underline{B}} \sqsubseteq_{\underline{B}} u[\underline{\tau}'_1/x_1, \ldots, \underline{\tau}'_n/x_n]^{\underline{B}}}$$

- We further require $\mathcal{T}_{\mathsf{eff}}$ to have either linear or semi-linear axioms

CBPV$_{\text{ref}}$ terms

- Separated into value and computation terms

$$V ::= x \mid \star \mid \langle V_1, V_2 \rangle \mid \mathsf{inj}_i^{\sigma_1+\sigma_2}\, V \mid \mathsf{thunk}\, M$$

$$\begin{aligned} M, N ::= {} & \mathsf{return}\, V \mid M\, \mathsf{to}\, x : \sigma.\, N \mid \mathsf{op}_{\underline{B}}(M_1, \ldots, M_n) \mid \\ & \lambda x : \sigma.M \mid M V \mid \mathsf{force}\, V \mid \langle M_1, M_2 \rangle \mid \mathsf{fst}\, M \mid \mathsf{snd}\, M \\ & \mathsf{match}\, V\, \mathsf{as}\, (x_1 : \sigma_1, x_2 : \sigma_2).M \mid \mathsf{match}\, V\, \mathsf{as}\, \{\} \mid \\ & \mathsf{match}\, V\, \mathsf{as}\, \{\mathsf{inj}_1\, (x_1 : \sigma_1) \mapsto M_1, \mathsf{inj}_2\, (x_2 : \sigma_2) \mapsto M_2\} \mid \end{aligned}$$

- Typing rules mostly standard from CBPV, e.g.:

$$\frac{\Gamma \vDash^{\mathsf{r}}_{\mathsf{v}} V : \sigma}{\Gamma \vDash^{\mathsf{r}}_{\mathsf{c}} \mathsf{return}\, V : F^{\mathsf{ref}} \sigma} \qquad \frac{\Gamma, x : \sigma \vDash^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}}{\Gamma \vDash^{\mathsf{r}}_{\mathsf{c}} \lambda x : \sigma.M : \sigma \to \underline{\tau}}$$

- except for ...

$\mathsf{CBPV}_{\mathrm{ref}}$ terms

- First exception: algebraic operations

$$\frac{\Gamma \vDash^{\mathsf{r}}_{\mathsf{c}} M_1 : \underline{\tau}_1 \quad \ldots \quad \Gamma \vDash^{\mathsf{r}}_{\mathsf{c}} M_n : \underline{\tau}_n}{\Gamma \vDash^{\mathsf{r}}_{\mathsf{c}} \mathsf{op}_{\underline{B}}(M_1, \ldots, M_n) : \langle \mathsf{op} \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_n)} \; (\mathsf{op} : n)$$

CBPV$_{\text{ref}}$ terms

- Second exception: sequential composition
  - Idea: spec. on composition ought to be a composition of specs.

$$\frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : C[F^{\mathsf{ref}}\sigma] \quad \Gamma, x : \sigma \models^{\mathsf{r}}_{\mathsf{c}} N : \underline{\tau}}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M \,\mathsf{to}\, x : \sigma.\, N : C[\underline{\tau}]}$$

- Uses computation refinement contexts C

$$C ::= [\,] \mid \langle \mathsf{op} \rangle (C_1, \ldots, C_n) \mid X \mid \mu X.C \mid C_1 \wedge C_2 \mid C_1 \vee C_2 \mid \bot$$

- Hole filling $C[\underline{\tau}]$ defined by struct. recursion

- For any $\vdash C$ we have:
  - $\Delta \models^{\mathsf{r}}_{\mathsf{c}} C[\underline{\tau}_1 \times \underline{\tau}_2] \sqsubseteq_{\underline{B}_1 \times \underline{B}_2} C[\underline{\tau}_1] \times C[\underline{\tau}_2]$
  - $\Delta \models^{\mathsf{r}}_{\mathsf{c}} C[\sigma \to \underline{\tau}] \sqsubseteq_{A \to \underline{B}} \sigma \to C[\underline{\tau}]$

CBPV$_{\mathrm{ref}}$ terms

- Third exception: subtyping

$$\frac{\Gamma \models^{\mathsf{r}}_{\mathsf{v}} V : \sigma_1 \quad \models^{\mathsf{r}}_{\mathsf{v}} \sigma_1 \sqsubseteq \sigma_2}{\Gamma \models^{\mathsf{r}}_{\mathsf{v}} V : \sigma_2} \qquad \frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}_1 \quad \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau}_1 \sqsubseteq \underline{\tau}_2}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}_2}$$

- Fourth exception: interaction between local and global, e.g.:

$$\frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}}{\Gamma_{\perp} \models^{\mathsf{r}}_{\mathsf{c}} M : \perp_{|\underline{\tau}|}} \qquad \frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}_1 \quad \Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}_2}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}_1 \wedge \underline{\tau}_2}$$

$$\frac{\Gamma, x : \sigma_1 \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau} \quad \Gamma, x : \sigma_2 \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau} \quad \models^{\mathsf{r}}_{\mathsf{v}} \sigma \sqsubseteq \sigma_1 \vee \sigma_2}{\Gamma, x : \sigma \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}}$$

- and dual rules for values

CBPV_{ref} terms

- Interaction rules allow us to check interesting specifications, e.g.:

$$\frac{\Gamma \Vdash^{\mathsf{r}}_{\mathsf{v}} V : \bot_1 + 1 \quad \Gamma, x_1 : 1 \Vdash^{\mathsf{r}}_{\mathsf{c}} \mathsf{update}_1(\mathsf{return}\,\star) : \langle \mathsf{update}_1 \rangle_{\underline{B}}(F^{\mathsf{ref}}1) \quad \Gamma, x_2 : 1 \Vdash^{\mathsf{r}}_{\mathsf{c}} \mathsf{return}\,\star : F^{\mathsf{ref}}1}{\Gamma \Vdash^{\mathsf{r}}_{\mathsf{c}} \mathsf{match}\, V \,\mathsf{as}\, \{\mathsf{inj}_1\,(x_1) \mapsto \mathsf{update}_1(\mathsf{return}\,\star), \mathsf{inj}_2\,(x_2) \mapsto \mathsf{return}\,\star\} : F^{\mathsf{ref}}1}$$

What CBPV$_{\mathrm{ref}}$ can't do right now

- A problem: we can't check all semantically valid refinements
- Example: from $\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}$ want to show $\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M \oplus M : \underline{\tau}$
- Can't use subtyping: l.h.s. of $\Delta \models^{\mathsf{r}}_{\mathsf{c}} \langle \mathsf{op} \rangle_{\underline{B}}(\underline{\tau}, \underline{\tau}) \sqsubseteq_{\underline{B}} \underline{\tau}$ not linear!
- Could naïvely extend CBPV$_{\mathrm{ref}}$ with the following typing rule:

$$\frac{x_1, \ldots, x_n \vdash t = u \qquad \Gamma \models^{\mathsf{r}}_{\mathsf{c}} M_i : \underline{\tau}_i \quad (i \in \{1, \ldots, n\})}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} t[M_1/x_1, \ldots, M_n/x_n]^{\underline{B}} : u[\underline{\tau}_1/x_1, \ldots, \underline{\tau}_n/x_n]^{\underline{B}}}$$

- But the non-linearity problem does not go away:
    - Still can not show: $\Gamma \models^{\mathsf{r}}_{\mathsf{c}} (\lambda x : 1.M)() \oplus M : \underline{\tau}$
- ...

CBPV$_{\mathrm{ref}}$ equational theory

- For equational reasoning and program optimizations
- Collection of equations $\Gamma \models^{\mathsf{r}}_{\mathsf{v}} V_1 = V_2 : \sigma$ and $\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M_1 = M_2 : \underline{\tau}$
- Divided into:
  - basic (congruence relation + subtyping)
  - structural ($\beta\eta$-equations from CBPV)
  - algebraic, e.g.:

$$\Gamma \models_{\mathsf{c}} M \oplus N = N \oplus M : \langle \oplus \rangle_{\underline{B}}(\underline{\tau}_N, \underline{\tau}_M)$$

  - refinement-typed, e.g.:

$$\frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : F^{\mathsf{ref}}\sigma \quad \Gamma \models^{\mathsf{r}}_{\mathsf{c}} N : \underline{\tau}}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M \,\mathsf{to}\, x : \sigma.\, N = N : \underline{\tau}} \qquad \frac{\Gamma \models^{\mathsf{r}}_{\mathsf{v}} V_1 : \sigma \quad \Gamma \models^{\mathsf{r}}_{\mathsf{v}} V_2 : \sigma}{\Gamma_{\perp} \models^{\mathsf{r}}_{\mathsf{v}} V_1 = V_2 : \perp_{|\sigma|}}$$

and rest of the "interaction" equations

$\mathsf{CBPV}_{\mathrm{ref}}$ equational theory

- Interaction equations allow us to prove interesting optimizations, e.g.:

$$\frac{\Gamma \vdash^{\mathsf{r}}_{\mathsf{v}} V : \bot_1 + 1 \quad \Gamma, x_1 : 1 \vdash^{\mathsf{r}}_{\mathsf{c}} \mathsf{update}_1(\mathsf{return}\,\star) : \langle \mathsf{update}_1 \rangle_{\underline{B}}(F^{\mathsf{ref}}\,1) \quad \Gamma, x_2 : 1 \vdash^{\mathsf{r}}_{\mathsf{c}} \mathsf{return}\,\star : F^{\mathsf{ref}}\,1}{\begin{array}{c}\Gamma \vdash^{\mathsf{r}}_{\mathsf{c}} \mathsf{match}\, V \,\mathsf{as}\, \{\mathsf{inj}_1\,(x_1) \mapsto \mathsf{update}_1(\mathsf{return}\,\star), \mathsf{inj}_2\,(x_2) \mapsto \mathsf{return}\,\star\} \\ = \\ \mathsf{return}\,\star : F^{\mathsf{ref}}\,1\end{array}}$$

Single system - different effects - different specs.

type-and-effect systems
+ optimizations

$\Gamma \vdash M : \sigma \,!\, \varepsilon$

input/output
+ sessions and protocols

$\Gamma \vdash M : ?(!(1).end; end)$

state
+ Hoare logic

$\Gamma \vdash M : \{P\}\sigma\{Q\}$

$\subseteq$ $\subseteq$ $\subseteq$

$\boxed{\text{CBPV}_{\text{ref}}}$

$\downarrow$

computational language (CBPV)
with algebraic effects

Type-and-effect systems

- $\Gamma \vdash loc_2 := !(loc_1)\,;\, loc_1 := 0 : ()\,!\,\{\mathsf{lookup}_1, \mathsf{update}_1, \mathsf{update}_2\}$
    - typing rules $\Gamma \vdash M : \sigma\,!\,\varepsilon$ with extra effect annotations
- Algebraic idea: let ε be an algebraic signature Kammar & Plotkin '12
- We use a collection of equiv. classes of trees built from ε
- In CBPV$_{\mathrm{ref}}$, we define $\sigma\,!\,\varepsilon \stackrel{\mathsf{def}}{=} C_\varepsilon[F^{\mathsf{ref}}\sigma]$ where

$$C_\varepsilon \stackrel{\mathsf{def}}{=} \mu X.([\,] \vee \langle\mathsf{op}_1\rangle(X, \ldots, X) \vee \ldots \vee \langle\mathsf{op}_m\rangle(X, \ldots, X))$$

for $\varepsilon = \{\mathsf{op}_1 : n_1, \ldots, \mathsf{op}_m : n_m\}$

- Another example: (update_0 not observable and thus not in ε)

$$\Gamma \stackrel{\mathsf{r}}{\underset{\mathsf{c}}{\vdash}} \mathsf{lookup}(\mathsf{update}_0(\mathsf{return}\,\star), \mathsf{return}\,\star) : 1\,!\,\{\mathsf{lookup}\}$$

Type-and-effect systems + optimizations

- Can validate effect-dependent optimizations, e.g. **Copy**

$$\dfrac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : C_\varepsilon[F^{\mathsf{ref}}\sigma] \qquad \begin{array}{r} t(t(x_{11},\ldots,x_{1n}),\ldots,t(x_{n1},\ldots,x_{nn})) = t(x_{11},\ldots,x_{nn}) \\ \text{holds in } \mathcal{T}_{\mathsf{eff}} \text{ for all terms } t \text{ built from } \varepsilon \end{array}}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M \,\mathsf{to}\, x : \sigma.\, M \,\mathsf{to}\, y : \sigma.\, \mathsf{return}\, \langle x, y\rangle = M \,\mathsf{to}\, x.\, \mathsf{return}\, \langle x, x\rangle : C_\varepsilon[F^{\mathsf{ref}}(\sigma \times \sigma)]}$$

- or **Discard**

$$\dfrac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : F^{\mathsf{ref}}\sigma \qquad \Gamma \models^{\mathsf{r}}_{\mathsf{c}} N : \underline{\tau}}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M \,\mathsf{to}\, x : \sigma.\, N = N : \underline{\tau}}$$

Single system - different effects - different specs.

type-and-effect systems
+ optimizations

$\Gamma \vdash M : \sigma \,!\, \varepsilon$

input/output
+ sessions and protocols

$\Gamma \vdash M : ?(!(1).end; end)$

state
+ Hoare logic

$\Gamma \vdash M : \{P\}\sigma\{Q\}$

$\subseteq$

CBPV_{ref}

$\downarrow$

computational language (CBPV)
with algebraic effects

Input/output + sessions

- Want to specify how a process should use I/O channel(s)
- Interested in following specs., inspired by session types

$$S \ ::= \ !(0).S \ | \ !(1).S \ | \ !(0 \vee 1).S \ | \ ?(S_1, S_2) \ | \ end$$

- Easy to define using operation modalities and C's

$$\begin{aligned}
C_{!(0).S} &\stackrel{\text{def}}{=} \langle \mathsf{send}_0 \rangle (C_S) \\
C_{!(1).S} &\stackrel{\text{def}}{=} \langle \mathsf{send}_1 \rangle (C_S) \\
C_{!(0\vee 1).S} &\stackrel{\text{def}}{=} \langle \mathsf{send}_0 \rangle (C_S) \vee \langle \mathsf{send}_1 \rangle (C_S) \\
C_{?(S_1,S_2)} &\stackrel{\text{def}}{=} \langle \mathsf{receive} \rangle (C_{S_1}, C_{S_2}) \\
end &\stackrel{\text{def}}{=} [\,]
\end{aligned}$$

Input/output + protocols

- Any use for fixed point refinements?
- Protocols for correctly using files (talking to a server, etc.)
- Using a file correctly once:

$$C_{\textsf{files}} ::= \langle \textsf{open} \rangle (\mu X.(\langle \textsf{close} \rangle ([\,]) \vee \langle \textsf{write}_i \rangle (X) \vee \langle \textsf{read} \rangle (X, X)))$$

- Using a file correctly repetitively:

$$C_{\textsf{rep-files}} ::= \mu Y. ([\,] \vee C_{\textsf{files}}[Y])$$

Single system - different effects - different specs.

type-and-effect systems
+ optimizations

$\Gamma \vdash M : \sigma \,!\, \varepsilon$

input/output
+ sessions and protocols

$\Gamma \vdash M : ?(!(1).end; end)$

state
+ Hoare logic

$\Gamma \vdash M : \{P\}\sigma\{Q\}$

CBPV_{ref}

computational language (CBPV)
with algebraic effects

State + Hoare Logic

- Would like to annotate terms with Hoare triples $\{P\}\,\sigma\,\{Q\}$
- Idea similar to refined state monad and HTT
- In this (propositionally clumsy) example, we take $P, Q \subseteq \{0,1\}$
- Define $\{P\}\,\sigma\,\{Q\} \stackrel{\text{def}}{=} \{P\}\,[F^{\text{ref}}\sigma]\,\{Q\}$ by case analysis on P

$$
\begin{aligned}
\{\emptyset\}\,[\,]\,\{Q\} &\stackrel{\text{def}}{=} \langle\mathsf{lookup}\rangle(\textstyle\bigvee_i\langle\mathsf{update}_i\rangle([\,]), \bigvee_i\langle\mathsf{update}_i\rangle([\,]))\\
\{\{0\}\}\,[\,]\,\{Q\} &\stackrel{\text{def}}{=} \langle\mathsf{lookup}\rangle(\textstyle\bigvee_q\langle\mathsf{update}_q\rangle([\,]), \bigvee_i\langle\mathsf{update}_i\rangle([\,]))\\
\{\{1\}\}\,[\,]\,\{Q\} &\stackrel{\text{def}}{=} \langle\mathsf{lookup}\rangle(\textstyle\bigvee_i\langle\mathsf{update}_i\rangle([\,]), \bigvee_q\langle\mathsf{update}_q\rangle([\,]))\\
\{\{0,1\}\}\,[\,]\,\{Q\} &\stackrel{\text{def}}{=} \langle\mathsf{lookup}\rangle(\textstyle\bigvee_q\langle\mathsf{update}_q\rangle([\,]), \bigvee_q\langle\mathsf{update}_q\rangle([\,]))
\end{aligned}
$$

where $i \in \{0,1\}$ and $q \in Q$

State + Hoare Logic

- Prop.: With this def. of $\{P\}\,\sigma\,\{Q\}$ following rules are admissible

$$\frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \{P \cap \{0\}\}\,\sigma\,\{Q\} \qquad \Gamma \models^{\mathsf{r}}_{\mathsf{c}} N : \{P \cap \{1\}\}\,\sigma\,\{Q\}}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} \mathsf{lookup}(M, N) : \{P\}\,\sigma\,\{Q\}}$$

$$\frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \{\{0\}\}\,\sigma\,\{Q\}}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} \mathsf{update}_0(M) : \{P\}\,\sigma\,\{Q\}} \qquad \frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \{\{1\}\}\,\sigma\,\{Q\}}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} \mathsf{update}_1(M) : \{P\}\,\sigma\,\{Q\}}$$

$$\frac{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \{P\}\,\sigma_1\,\{Q\} \qquad \Gamma, x : \sigma_1 \models^{\mathsf{r}}_{\mathsf{c}} N : \{Q\}\,\sigma_2\,\{R\}}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M \,\mathsf{to}\, x : \sigma_1.\, N : \{P\}\,\sigma_2\,\{R\}}$$

$$\frac{P \subseteq P' \qquad \Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \{P'\}\,\sigma\,\{Q'\} \qquad Q' \subseteq Q}{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : \{P\}\,\sigma\,\{Q\}}$$

Single system - different effects - different specs.

type-and-effect systems
+ optimizations

$\Gamma \vdash M : \sigma\,!\,\varepsilon$

input/output
+ sessions and protocols

$\Gamma \vdash M : ?(!(1).end; end)$

state
+ Hoare logic

$\Gamma \vdash M : \{P\}\sigma\{Q\}$

$\boxed{\mathrm{CBPV}_{\mathrm{ref}}}$

$\downarrow$

two-level semantics

computational language (CBPV) with algebraic effects

Two-level CBPV$_{\mathrm{ref}}$ semantics

- The big picture

$$
\begin{tikzcd}[column sep=huge]
\mathsf{Rel}(\mathsf{Set}) \arrow[r, bend left, "F^{\mathsf{ref}}"] \arrow[r, phantom, "\perp"] \arrow[d, "|-|_v"'] & \mathsf{Ref}(\mathsf{Alg}) \arrow[l, bend left, "U^{\mathsf{ref}}"] \arrow[d, "|-|_c"] \\
\mathsf{Set} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathsf{Alg} \arrow[l, bend left, "U"]
\end{tikzcd}
$$

where Alg is the category of $\mathcal{T}_{\mathsf{eff}}$-algebras in Set

Two-level CBPV$_{\mathrm{ref}}$ semantics

- The big picture

$$\begin{tikzcd}
 & & & \boxed{\begin{array}{l}\langle \mathcal{A} : \mathcal{T}_{\mathsf{eff}}\text{-}\mathsf{Alg}, \\ \mathcal{R} \subseteq U\mathcal{A} \times U\mathcal{A} \rangle\end{array}} \arrow[d] \\
\boxed{\begin{array}{l}\langle X : \mathsf{Set}, \\ \mathcal{R} \subseteq X \times X \rangle\end{array}} \arrow[r] & \mathsf{Rel}(\mathsf{Set}) \arrow[rr, bend left, "F^{\mathsf{ref}}"] \arrow[rr, phantom, "\perp"] \arrow[d, "|-|_v"'] & & \mathsf{Ref}(\mathsf{Alg}) \arrow[ll, bend left, "U^{\mathsf{ref}}"] \arrow[d, "|-|_c"] \\
 & \mathsf{Set} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathsf{Alg} \arrow[ll, bend left, "U"]
\end{tikzcd}$$

where Alg is the category of $\mathcal{T}_{\mathsf{eff}}$-algebras in Set

Two-level CBPV$_{\mathrm{ref}}$ semantics

- How one constructs this quadruple:

$$
\begin{tikzcd}
\mathsf{Sub}(\mathsf{Set}) \arrow[d, "p_{subset}"'] & \mathsf{Rel}(\mathsf{Set}) \arrow[l] \arrow[d, "|-|_v"] \arrow[r, bend left, "F^{\mathsf{ref}}"] \arrow[r, phantom, "\perp"] & \mathsf{Ref}(\mathsf{Alg}) \arrow[l, bend left, "U^{\mathsf{ref}}"] \arrow[d, "|-|_c"] \\
\mathsf{Set} & \mathsf{Set} \arrow[l, "(-)\times(-)\circ\delta"] \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathsf{Alg} \arrow[l, bend left, "U"]
\end{tikzcd}
$$

where

- $U^{\mathsf{ref}} \stackrel{\mathsf{def}}{=} (|-|_v)^*(U)$
- $F^{\mathsf{ref}}\sigma \stackrel{\mathsf{def}}{=} \langle F|\sigma|_v, (\eta_{|\sigma|_v})_!(\sigma)\rangle$

Hermida '93

Two-level CBPV$_{\text{ref}}$ semantics

- Back to the big picture

$$\begin{tikzcd}
\mathsf{Rel}(\mathsf{Set}) \arrow[rr, bend left, "F^{\mathsf{ref}}"] \arrow[rr, phantom, "\perp"] \arrow[d, "|-|_v"'] & & \mathsf{Ref}(\mathsf{Alg}) \arrow[ll, bend left, "U^{\mathsf{ref}}"] \arrow[d, "|-|_c"] \\
\mathsf{Set} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathsf{Alg} \arrow[ll, bend left, "U"]
\end{tikzcd}$$

- Refinement types interpreted as functors

$$[\![\vec{X} : \mathsf{Ref}(\underline{\vec{B}}) \vdash \sigma : \mathsf{Ref}(A)]\!] : \mathsf{Ref}_{[\![\underline{B}_1]\!]}(\mathsf{Alg}) \times \ldots \times \mathsf{Ref}_{[\![\underline{B}_n]\!]}(\mathsf{Alg}) \to \mathsf{Rel}_{[\![A]\!]}(\mathsf{Set})$$

$$[\![\vec{X} : \mathsf{Ref}(\underline{\vec{B}}) \vdash \underline{\tau} : \mathsf{Ref}(\underline{B})]\!] : \mathsf{Ref}_{[\![\underline{B}_1]\!]}(\mathsf{Alg}) \times \ldots \times \mathsf{Ref}_{[\![\underline{B}_n]\!]}(\mathsf{Alg}) \to \mathsf{Ref}_{[\![\underline{B}]\!]}(\mathsf{Alg})$$

Two-level CBPV$_{\mathrm{ref}}$ semantics

- More on the big picture

$$\begin{tikzcd} \mathsf{Rel}(\mathsf{Set}) \arrow[rr, bend left, "F^{\mathsf{ref}}"] \arrow[rr, phantom, "\perp"] \arrow[d, "|-|_v"'] & & \mathsf{Ref}(\mathsf{Alg}) \arrow[ll, bend left, "U^{\mathsf{ref}}"] \arrow[d, "|-|_c"] \\ \mathsf{Set} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathsf{Alg} \arrow[ll, bend left, "U"] \end{tikzcd}$$

- Computation refinement contexts interpreted as fam. of functors

$$[\![\vec{X} \vdash C]\!]_{\underline{B}} : \mathsf{Ref}_{[\![\underline{B}]\!]}(\mathsf{Alg}) \times \ldots \times \mathsf{Ref}_{[\![\underline{B}]\!]}(\mathsf{Alg}) \times \underbrace{\mathsf{Ref}_{[\![\underline{B}]\!]}(\mathsf{Alg})}_{\text{the hole } [\,]} \to \mathsf{Ref}_{[\![\underline{B}]\!]}(\mathsf{Alg})$$

- Importantly: closed C's are still functorial:

$$[\![\vdash \underline{\tau}_1]\!] \to [\![\vdash \underline{\tau}_2]\!] \implies [\![\vdash C]\!]_{\underline{B}_1}([\![\vdash \underline{\tau}_1]\!]) \to [\![\vdash C]\!]_{\underline{B}_2}([\![\vdash \underline{\tau}_2]\!])$$

Two-level CBPV$_{\mathrm{ref}}$ semantics

- Even more on the big picture

$$\begin{tikzcd}
\mathsf{Rel}(\mathsf{Set}) \arrow[rr, bend left, "F^{\mathsf{ref}}"] \arrow[rr, phantom, "\perp"] \arrow[d, "|-|_v"'] & & \mathsf{Ref}(\mathsf{Alg}) \arrow[ll, bend left, "U^{\mathsf{ref}}"] \arrow[d, "|-|_c"] \\
\mathsf{Set} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathsf{Alg} \arrow[ll, bend left, "U"]
\end{tikzcd}$$

- Terms interpreted as morphisms in $\mathsf{Rel}(\mathsf{Set})$

$$[\![\Gamma \vdash^{\mathsf{r}}_{\mathsf{v}} V : \sigma]\!] : [\![\Gamma]\!] \longrightarrow [\![\sigma]\!]$$

$$[\![\Gamma \vdash^{\mathsf{r}}_{\mathsf{c}} M : \underline{\tau}]\!] : [\![\Gamma]\!] \longrightarrow U^{\mathsf{ref}}[\![\underline{\tau}]\!]$$

Two-level $\mathrm{CBPV}_{\mathrm{ref}}$ semantics

- Even more on the big picture

$$
\begin{tikzcd}[column sep=huge]
\mathsf{Rel}(\mathsf{Set}) \arrow[r, bend left, "F^{\mathsf{ref}}"] \arrow[r, phantom, "\perp"] \arrow[d, "|-|_v"'] & \mathsf{Ref}(\mathsf{Alg}) \arrow[l, bend left, "U^{\mathsf{ref}}"] \arrow[d, "|-|_c"] \\
\mathsf{Set} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathsf{Alg} \arrow[l, bend left, "U"]
\end{tikzcd}
$$

- Prop.: The interpretation is sound

$$\Gamma \vdash^{\mathsf{r}}_{\mathsf{v}} V_1 = V_2 : \sigma \implies [\![V_1]\!] = [\![V_2]\!] \quad \text{and} \quad \Gamma \vdash^{\mathsf{r}}_{\mathsf{c}} M_1 = M_2 : \underline{\tau} \implies [\![M_1]\!] = [\![M_2]\!]$$

- Prop.: The interpretation is coherent

  interpretations of different derivations of a $\mathit{CBPV}_{\mathit{ref}}$ term all agree

Summary

A refinement type system for algebraic effects

- accommodating a wide range of computational effects
- accommodating a wide range of specification styles, eg.:
  - effect-and-type annotations
  - protocols and sessions
  - file access
  - Hoare Logic
- a concrete algebraic and modal syntax to construct Katsumata- and Tate-style effect annotation monoids

Some future work:

- Operations with parameters and binding
- Local effects and effect handlers

Alternative approach with algebraic modalities

- $\underline{\tau} ::= \ldots \quad | \quad \langle (x_1, \ldots, x_n).t \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_n)$
- Then all equations in $\mathcal{T}_{\mathsf{eff}}$ induce subtyping inequalities:

$$\frac{x_1, \ldots, x_n \vdash t = u \qquad \Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau}_i \sqsubseteq_{\underline{B}} \underline{\tau}'_i \quad (i \in \{1, \ldots, n\})}{\Delta \models^{\mathsf{r}}_{\mathsf{c}} \langle t \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_n) \sqsubseteq_{\underline{B}} \langle u \rangle_{\underline{B}}(\underline{\tau}'_1, \ldots, \underline{\tau}'_n)}$$

- Characterization: colax and linearly lax action of $\mathcal{T}_{\mathsf{eff}}$ term clone (before we had the action of $\mathcal{T}_{\mathsf{eff}}$ term operad)

$$\Delta \models^{\mathsf{r}}_{\mathsf{c}} \langle t(t_1, \ldots, t_n) \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_m) \sqsubseteq_{\underline{B}} \langle t(y_1, \ldots, y_n) \rangle_{\underline{B}}(\langle t_1 \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_m), \ldots, \langle t_n \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_m))$$

$$\Delta \models^{\mathsf{r}}_{\mathsf{c}} \langle x_i \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_n) \sqsubseteq_{\underline{B}} \underline{\tau}_i \qquad \Delta \models^{\mathsf{r}}_{\mathsf{c}} \underline{\tau} \sqsubseteq_{\underline{B}} \langle x \rangle_{\underline{B}}(\underline{\tau})$$

$$\Delta \models^{\mathsf{r}}_{\mathsf{c}} \langle t(y_1, \ldots, y_n) \rangle_{\underline{B}}(\langle t_1 \rangle_{\underline{B}}(\underline{\tau}_{11}, \ldots, \underline{\tau}_{1m_1}), \ldots, \langle t_n \rangle_{\underline{B}}(\underline{\tau}_{n1}, \ldots, \underline{\tau}_{nm_n})) \sqsubseteq_{\underline{B}} \langle t(t_1, \ldots, t_n) \rangle_{\underline{B}}(\underline{\tau}_{11}, \ldots, \underline{\tau}_{1m_1}, \ldots, \underline{\tau}_{n1} \cdots \underline{\tau}_{nm_n})$$

- The non-linearity problems with ref. typing still there