A Propositional Refinement Type System for Algebraic Effects

Danel Ahman

(joint work with Gordon Plotkin)

PLInG Meeting, 27 January 2014

informatics

lfcs | Laboratory for Foundations of Computer Science

THE UNIVERSITY *of* EDINBURGH

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

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

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

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

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

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

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

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

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

computational language (CBPV) with algebraic effects

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 $\mathcal{T}_{\mathsf{ND}}$ 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 $\mathcal{T}_{\mathsf{I/O}}$ of input/output (of bits, over a fixed channel)
  - three operations $\mathsf{receive} : 2$,

$$\mathsf{send}_0 : 1,$$

$$\mathsf{send}_1 : 1$$

  - 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

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

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

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

$\downarrow$

computational language (CBPV)
with algebraic effects

Call-by-Push-Value with algebraic effects

- Strict separation into values and computations

Levy '04

- 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:
  - $V ::= \star \mid \langle V_1, V_2 \rangle \mid \mathsf{inj}_i\, V \mid \mathsf{thunk}\, M$
  - $M ::= \mathsf{return}\, V \mid M\, \mathsf{to}\, x : A.\, N \mid \mathsf{op}_{\underline{B}}(M_1, \ldots, M_n) \mid$
    $\lambda x : A.M \mid MV \mid \mathsf{force}\, V \mid \langle M_1, M_2 \rangle \mid \mathsf{fst}\, M \mid \mathsf{snd}\, M \mid$
    $\mathsf{match}\, V\, \mathsf{as}\, (x_1 : A_1, x_2 : A_2).M \mid \mathsf{match}\, V\, \mathsf{as}\, \{\} \mid$
    $\mathsf{match}\, V\, \mathsf{as}\, \{\mathsf{inj}_1\, (x_1 : A_1) \mapsto M_1, \mathsf{inj}_2\, (x_2 : A_2) \mapsto M_2\}$
- Equational theory:
  - standard $\beta\eta$-equations $+$
    algebraic equations $\Gamma \vDash_c M \oplus N = N \oplus M : \underline{B}$

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

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

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

computational language (CBPV)
with algebraic effects

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$_{\mathrm{ref}}$

- Separated into value and computation refinement types

$$\sigma \quad ::= \quad 1 \mid 0 \mid \sigma_1 \times \sigma_2 \mid \sigma_1 + \sigma_2 \mid U^{\mathsf{ref}}\underline{\tau} \quad \leftarrow \boxed{\text{structural}}$$

$$\sigma_1 \wedge \sigma_2 \mid \sigma_1 \vee \sigma_2 \mid \bot_A \quad \leftarrow \boxed{\text{logical}}$$

$$\underline{\tau} \quad ::= \quad F^{\mathsf{ref}}\sigma \mid \underline{\tau}_1 \times \underline{\tau}_2 \mid \sigma \to \underline{\tau} \mid$$

$$\langle \mathsf{op} \rangle_{\underline{B}}(\underline{\tau}_1, \ldots, \underline{\tau}_n) \mid \quad \leftarrow \boxed{\text{operation modalities}}$$

$$X \mid \mu X : \mathsf{Ref}(\underline{B}).\underline{\tau} \mid \underline{\tau}_1 \wedge \underline{\tau}_2 \mid \underline{\tau}_1 \vee \underline{\tau}_2 \mid \bot_{\underline{B}}$$

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

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

- Prop.: Unique kinding

Logical reasoning

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

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

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

- and logical rules

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

CBPV_{ref} terms

- Separated into value and computation terms

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

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

- 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 . . .

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

- First exception: algebraic operations

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

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

- Second exception: sequential composition

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

- 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

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

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

- Third exception: subtyping

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

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

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

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

- and dual rules for values

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

- Interaction rules allow us to prove interesting derivations, e.g.:

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

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

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

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

  - refinement-typed, e.g.:

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

  and rest of the "interaction" equations

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$_{\mathrm{ref}}$

computational language (CBPV) with algebraic effects

Type-and-effect systems

- $\Gamma \vdash loc_2 := !(loc_1) \,;\, loc_1 := 0 : () \,!\, \{\mathsf{lookup}, \mathsf{update}\}$
  - 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_{ref}, we define $\sigma \,!\, \varepsilon \overset{\text{def}}{=} C_\varepsilon[F^{\mathsf{ref}}\sigma]$ where

$$C_\varepsilon \overset{\text{def}}{=} \mu X.([\,] \vee \langle \mathsf{op}_1 \rangle_{FA}(X, \ldots, X) \vee \ldots \vee \langle \mathsf{op}_m \rangle_{FA}(X, \ldots, X))$$

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

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

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

Type-and-effect systems + optimizations

- Can translate well-(ε-)kinded Kammar & Plotkin types:
  - $(F_\varepsilon A)^\circ_\varepsilon \stackrel{\mathsf{def}}{=} C_\varepsilon[F^{\mathsf{ref}} A^\circ]$, $(A \to \underline{B})^\circ_\varepsilon \stackrel{\mathsf{def}}{=} A^\circ \to \underline{B}^\circ_\varepsilon$ (, and $\times$)

  with translated types having expected properties, e.g.:
  - $\models^{\mathsf{r}}_{\mathsf{c}} C_\varepsilon[\underline{B}^\circ_\varepsilon] \sqsubseteq \underline{B}^\circ_\varepsilon$
  - $\models^{\mathsf{r}}_{\mathsf{c}} C_{\varepsilon_1}[F^{\mathsf{ref}} A^\circ] \sqsubseteq C_{\varepsilon_2}[F^{\mathsf{ref}} A^\circ]$ when $\varepsilon_1 \subseteq \varepsilon_2$
- Kammar & Plotkin's abstract optimizations, e.g. **Copy**

$$\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M : C_\varepsilon[F^{\mathsf{ref}}\sigma]$$

$$t(t(x_{11}, \ldots, x_{1n}), \ldots, t(x_{n1}, \ldots, x_{nn})) = t(x_{11}, \ldots, x_{nn})$$

holds in $\mathcal{T}_{\mathsf{eff}}$ for all terms t built from ε

$$\overline{\Gamma \models^{\mathsf{r}}_{\mathsf{c}} M \,\mathsf{to}\, x : \sigma.\, M \,\mathsf{to}\, y : \sigma.\, \mathsf{return}\, \langle x, y\rangle = M \,\mathsf{to}\, x.\, \mathsf{return}\, \langle x, x\rangle : C_\varepsilon[F^{\mathsf{ref}}(\sigma \times \sigma)]}$$

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

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

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

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

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

- 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:

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

- Using a file correctly repetitively:

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

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$_{\mathrm{ref}}$

computational language (CBPV) with algebraic effects

State + Hoare Logic

- Would like to annotate terms with Hoare triples $\{P\}\,\sigma\,\{Q\}$
- Hoare Logic in some ref. ty. sys. supporting only state effects:
  - using the refined state monad — Borgström et al. '11
  - using the Dijkstra monad — Swamy et al. '13
- 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

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

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

State + Hoare Logic

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

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

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

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

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

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

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

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

$\big\downarrow$ two-level semantics

computational language (CBPV) with algebraic effects

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

- The big picture

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

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:

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

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$_{\mathrm{ref}}$ 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

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

- Importantly: closed C's still have functorial flavor:

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

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

- Even more on the big picture

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

- 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 CBPV$_{ref}$ term all agree

Parametric effect monads (Katsumata,POPL'14)

- A proposal for a unifying categorical semantics of a monadic effect-annotated language

  Wadler & Thiemann '03

  - $A, B ::= 1 \mid 0 \mid A \times B \mid A \to B \mid T_\varepsilon A$

- Instead of taking a single monad $T : \mathcal{V} \to \mathcal{V}$
- Take a lax mon. functor $T : \mathbb{E} \to [\mathcal{V}, \mathcal{V}]$
  - $\mathbb{E}$ a preordered monoid of effects (a mon. category)
  - $\eta : \mathsf{Id} \to T_1$
  - $\mu_{\varepsilon_1, \varepsilon_2} : T_{\varepsilon_1} \circ T_{\varepsilon_2} \to T_{\varepsilon_1 \circ \varepsilon_2}$

- Parallels with our work:
  - can interpret closed C's as unary functors
    $$[\![C]\!] : \mathsf{Ref}(\mathsf{Alg}) \to \mathsf{Ref}(\mathsf{Alg})$$
  - closed C's form a preordered monoid $\mathbb{C}$
  - so could also interpret as $[\![-]\!] : \mathbb{C} \to [\mathsf{Ref}(\mathsf{Alg}), \mathsf{Ref}(\mathsf{Alg})]$

Summary

- Algebraic work for accommodating effects in ref. ty. systems
- Examples: Effect-and-type annotations, protocols, Hoare Logic

Future work:

- Operations with parameters and binding
- Local effects
- Effect handlers
- Optimizations making use of fine-grained refinements
- Combining theories vs. combining refinements
- More general categorical account of $\mathrm{CBPV}_{\mathrm{ref}}$