Adequacy for Algebraic Effects Revisited

Alex Kavvos

OOPSLA 2025, Singapore, 17 October 2025

What is adequacy?

Two main kinds of program semantics:

- **Operational Semantics** $M \longrightarrow N$
  Usually syntactic and intuitive. **Hard to reason about**.
  Induces an **operational equivalence** $M \simeq N$

- **Denotational Semantics** $[\![M]\!]$
  A translation of programs to mathematics. **Easy to reason about**.
  Induces a **denotational equivalence** $[\![M]\!] = [\![N]\!]$

Adequacy (Milner 1975):

$$[\![M]\!] = [\![N]\!] \quad \Longrightarrow \quad M \simeq N$$

Every equivalence proven **mathematically** also holds **operationally**.

Previous work proves this only for **finitary algebraic effects** (Plotkin and Power 2001).
The proof is complicated. Can we do it with **logical relations** instead?

What are algebraic effects?

Functional programming

$\Longrightarrow$ Programs produce **values**

Algebraic effects

$\Longrightarrow$ Programs produce **interaction trees**

A **signature** Σ of **operations** ($\approx$ system calls)

Ops have a **sending type** and a **receiving type**:

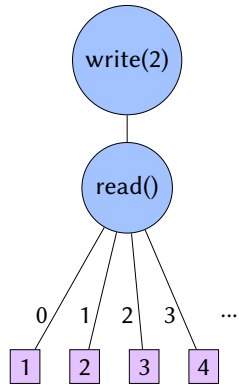
$$\mathrm{op} : !\,\sigma\,?\,\tau \quad \in \quad \Sigma$$

These types must be **first-order** ($\approx$ serializable)

For a global store with a single cell holding a Nat:

$$\Sigma_{\mathrm{store}} \stackrel{\mathrm{def}}{=} \{\ \mathrm{write} : !\,\mathrm{Nat}\,?\,\mathbf{1},\ \ \mathrm{read} : !\,\mathbf{1}\,?\,\mathrm{Nat}\ \}$$

Example: $\mathrm{write}(\overline{2};\ \mathrm{read}(x.\ \mathrm{return}\ \mathrm{succ}(x)))$



Call-by-Push-Value (CBPV)

value types

$$
\begin{array}{rcll}
A & ::= & \mathbf{1} & \text{unit type} \\
 & & \mathsf{Nat} & \text{natural numbers} \\
 & & A_1 + A_2 & \text{sum types} \\
 & & A_1 \times A_2 & \text{product types} \\
 & & U\underline{C} & \text{thunks}
\end{array}
$$

first-order types

$$\sigma, \tau ::= \mathbf{0} \mid \mathbf{1} \mid \mathsf{Nat} \mid \sigma \times \tau \mid \sigma + \tau$$

computation types

$$
\begin{array}{rcll}
\underline{C} & ::= & FA & \text{returners} \\
 & & A \to \underline{C} & \text{functions}
\end{array}
$$

$\Gamma \vdash^{\mathrm{v}} V : A$ "V is a value of type A"

$\Gamma \vdash^{\mathrm{c}} M : \underline{C}$ "M is a computation of type $\underline{C}$"

$$\frac{}{\Gamma, x : A, \Gamma' \vdash^{\mathrm{v}} x : A} \qquad \frac{\Gamma \vdash^{\mathrm{v}} V : A}{\Gamma \vdash^{\mathrm{c}} \mathsf{return}\ V : FA} \qquad \frac{\Gamma \vdash^{\mathrm{c}} M : FA \qquad \Gamma, x : A \vdash^{\mathrm{c}} N : \underline{C}}{\Gamma \vdash^{\mathrm{c}} M\ \mathsf{to}\ x.\ N : \underline{C}}$$

$$\frac{\Gamma, x : A \vdash^{\mathrm{c}} M : \underline{C}}{\Gamma \vdash^{\mathrm{c}} \lambda x : A.\ M : A \to \underline{C}} \qquad \frac{\mathrm{op} : !\,\sigma\,?\,\tau \in \Sigma \qquad \Gamma \vdash^{\mathrm{v}} V : \sigma \qquad \Gamma, x : \tau \vdash^{\mathrm{c}} M : \underline{C}}{\Gamma \vdash^{\mathrm{c}} \mathrm{op}(V; x.\ M) : \underline{C}}$$

Small-step operational semantics

Two relations: $M \longrightarrow N$ and $M \xrightarrow{\mathrm{op}\,!\,V\,?\,W} N$.

$$\mathcal{E}[-] \quad ::= \quad [-] \mid \mathcal{E}[-] \text{ to } x.\, N \mid \mathcal{E}[-](V)$$

$$\frac{}{(\lambda x : A.\, M)(V) \longrightarrow M[V/x]} \qquad \frac{M \longrightarrow N}{\mathcal{E}[M] \longrightarrow \mathcal{E}[N]}$$

$$\frac{\mathrm{op} : !\,\sigma\,?\,\tau \qquad W \in \mathrm{Val}(\tau)}{\mathrm{op}(V; x.\, M) \xrightarrow{\mathrm{op}\,!\,V\,?\,W} M[W/x]} \qquad \frac{M \xrightarrow{\mathrm{op}\,!\,V\,?\,W} N}{\mathcal{E}[M] \xrightarrow{\mathrm{op}\,!\,V\,?\,W} \mathcal{E}[N]}$$

Example:

$$\mathrm{write}(\overline{2}; \mathrm{read}(x.\, \mathrm{return}\ \mathrm{succ}(x))) \xrightarrow{\mathrm{write}\,!\,\overline{2}\,?\,()} \mathrm{read}(x.\, \mathrm{return}\ \mathrm{succ}(x)) \xrightarrow{\mathrm{read}\,!\,()\,?\,\overline{3}} \mathrm{return}\ \overline{4}$$

Use these to define $\|-\| : \mathrm{Comp}(FA) \longrightarrow \mathrm{IT}_\Sigma(\mathrm{Val}(A))$

Denotational Semantics

Let T be a monad with $\eta : X \to TX$ and $(\gg\!=) : TX \to [X \to TY] \to TY$.

An **algebraic operation** $\mathrm{op} : !\,\sigma\,?\,\tau$ over T consists of a family

$$\mathrm{op}_X : \mathfrak{U}_\sigma \times (\mathfrak{U}_\tau \to TX) \to TX$$

satisfying the **fundamental equation of algebraic effects**:

$$\mathrm{op}_X(v, k) \gg\!= f = \mathrm{op}_Y\big(v, x \mapsto k(x) \gg\!= f\big)$$

T **supports** Σ if it comes with an algebraic operation for each $\mathrm{op} : !\,\sigma\,?\,\tau$.

Example: $\mathrm{St}(X) \stackrel{\text{def}}{=} \mathbb{N} \to \mathbb{N} \times X$ supports $\Sigma_{\text{store}} \stackrel{\text{def}}{=} \{\ \mathrm{write} : !\,\mathrm{Nat}\,?\,\mathbf{1},\ \mathrm{read} : !\,\mathbf{1}\,?\,\mathrm{Nat}\ \}$

Map value types to **cpo's**, computation types to **monad algebras over cppo's**.
Values and computations are mapped to **continuous functions**:

$$[\![V]\!] : [\![\Gamma]\!] \to [\![A]\!] \qquad\qquad [\![M]\!] : [\![\Gamma]\!] \to [\![\underline{C}]\!]$$

Operational Equivalence

Observational equivalence for a pure language (Morris 1969):

$$M \simeq N \stackrel{\text{def}}{\equiv} \forall \text{ ground } C[-] : \sigma.\ C[M] \Downarrow V \iff C[N] \Downarrow V$$

A ground context $C[-]$ of type σ is an **observable testing bench**.

…how to adapt to interaction trees?

Idea Define what it means to have **the same observations** instead.

Define (natural) functions $\varepsilon_X : \mathrm{IT}_\Sigma(X) \longrightarrow TX$, interpreting ops in T.

Suppose that for each $R \subseteq A \times B$ we have a $\mathbb{T}(R) \subseteq TA \times TB$. Then

$$M \simeq_{\mathbb{T}} N \stackrel{\text{def}}{\equiv} \forall \text{ ground } C[-] : \sigma.\ \varepsilon_{\mathrm{Val}(\sigma)}(\|C[M]\|)\ \mathbb{T}(=_{\mathrm{Val}(\sigma)})\ \varepsilon_{\mathrm{Val}(\sigma)}(\|C[N]\|)$$

where $=_{\mathrm{Val}(\sigma)} \stackrel{\text{def}}{=} \{(V, V) \mid V \in \mathrm{Val}(\sigma)\} \subseteq \mathrm{Val}(\sigma) \times \mathrm{Val}(\sigma)$ is the **equality** relation.

We will do this for any monad T that comes with a compatible **relator** (Gavazzo 2019).

Relators

Let $F : \mathbf{Set} \longrightarrow \mathbf{Set}$. A **relator** $\mathbb{F}$ for F is an assignment

$$\mathcal{R} \subseteq X \times Y \quad \rightsquigarrow \quad \mathbb{F}\mathcal{R} \subseteq FX \times FY$$

such that, defining the **graph** $f^\bullet \stackrel{\text{def}}{=} \{(x, f(x)) \mid x \in A\} \subseteq A \times B$ of $f : A \to B$,

$$=_{FX} \subseteq \mathbb{F}(=_X) \tag{r_1}$$
$$\mathbb{F}\mathcal{R}_1 \,;\, \mathbb{F}\mathcal{R}_2 \subseteq \mathbb{F}(\mathcal{R}_1 \,;\, \mathcal{R}_2) \tag{r_2}$$
$$\mathcal{R}_1 \subseteq \mathcal{R}_2 \Rightarrow \mathbb{F}\mathcal{R}_1 \subseteq \mathbb{F}\mathcal{R}_2 \tag{r_3}$$
$$(Ff)^\bullet \subseteq \mathbb{F}(f^\bullet) \tag{r_4}$$
$$((Ff)^\bullet)^{\mathrm{op}} \subseteq \mathbb{F}(f^\bullet)^{\mathrm{op}} \tag{r_5}$$

Example: $\zeta_1 \; \mathbb{ST}(\mathcal{R}) \; \zeta_2 \stackrel{\text{def}}{\equiv} \forall n \in \mathbb{N}.\;\; \zeta_1(n).1 = \zeta_2(n).1$ and $\zeta_1(n).2 \; \mathcal{R} \; \zeta_2(n).2$

Our relators need to be **compatible with** Σ and **compatible with recursion**.

The logical relation

Define

$$\mathcal{R}_A \subseteq [\![A]\!] \times \mathrm{Val}(A)$$
$$\mathcal{R}_{\underline{C}} \subseteq [\![\underline{C}]\!] \times \mathrm{Comp}(\underline{C})$$

by induction on types.

$$\begin{aligned}
n\ \mathcal{R}_{\mathrm{Nat}}\ V &\stackrel{\mathrm{def}}{\equiv} V = \overline{n} \\
(d_1, d_2)\ \mathcal{R}_{A_1 \times A_2}\ (V_1, V_2) &\stackrel{\mathrm{def}}{\equiv} d_1\ \mathcal{R}_{A_1}\ V_1 \wedge d_2\ \mathcal{R}_{A_2}\ V_2 \\
d\ \mathcal{R}_{U\underline{C}}\ \mathrm{thunk}\ M &\stackrel{\mathrm{def}}{\equiv} d\ \mathcal{R}_{\underline{C}}\ M \\
d\ \mathcal{R}_{FA}\ M &\stackrel{\mathrm{def}}{\equiv} d\ \mathbb{T}\mathcal{R}_A\ \varepsilon(\|M\|) \\
f\ \mathcal{R}_{A \to \underline{C}}\ M &\stackrel{\mathrm{def}}{\equiv} \forall d.\ d\ \mathcal{R}_A\ V \Longrightarrow f(d)\ \mathcal{R}_{\underline{C}}\ M(V)
\end{aligned}$$

Lemma (Graph Lemma)

$v\ \mathcal{R}_\sigma\ V$ if and only if $v = [\![V]\!]$

for any first-order type σ

Lemma (Fundamental Lemma)

- $\vdash^v W : A \Longrightarrow [\![W]\!]\ \mathcal{R}_A\ W$.
- $\vdash^c M : \underline{C} \Longrightarrow [\![M]\!]\ \mathcal{R}_{\underline{C}}\ M$.

Corollary

$\vdash^c M : F\sigma \Longrightarrow [\![M]\!]\ \mathbb{T}\mathcal{R}_\sigma\ \varepsilon(\|M\|)$

The smoking gun

Lemma (Soundness through relators)

For any M : *FA we have* $\varepsilon(\|M\|)\ \mathbb{T}(\mathcal{R}_\sigma{}^{op})\ [\![M]\!]$.

Theorem (Adequacy)

$[\![M]\!] \sqsubseteq [\![N]\!] \implies M \sqsubseteq_{\mathbb{T}} N$

Proof.

Let $C[-]$ be ground, and let $[\![M]\!] \sqsubseteq [\![N]\!]$. By soundness, compositionality, and the fundamental lemma

$$\varepsilon(\|C[M]\|)\ \mathbb{T}(\mathcal{R}_\sigma{}^{\mathrm{op}})\ [\![C[M]]\!] \sqsubseteq [\![C[N]]\!]\ \mathbb{T}\mathcal{R}_\sigma\ \varepsilon(\|C[N]\|)$$

Then, by (r_2), the right module property, and the graph lemma,

$$\mathbb{T}(\mathcal{R}_\sigma{}^{\mathrm{op}})\ ;\ \mathbb{T}\mathcal{R}_\sigma \subseteq \mathbb{T}(\mathcal{R}_\sigma{}^{\mathrm{op}}\ ;\ \mathcal{R}_\sigma) = \mathbb{T}(=_{\mathrm{Val}(\sigma)})$$

Hence $\varepsilon(\|C[M]\|)\ \mathbb{T}(\mathcal{R}_\sigma{}^{\mathrm{op}})\ \varepsilon(\|C[N]\|)$, i.e. $M \sqsubseteq_{\mathbb{T}} N$. Rinse and repeat! □

Future work

- Recursive types (see Jiang, Xue, New, OOPSLA 2025); minimal invariants? (Pitts 1996)
- Other notions of relation lifting, e.g.
  - Free lifting? (Kammar 2014)
  - `TT`-lifting? (Katsumata 2005, 2013)
  - Codensity lifting? (Katsumata, Sato, and Uustalu 2018)

  Not relators!
- Formalisation

Future work

- Recursive types (see Jiang, Xue, New, OOPSLA 2025); minimal invariants? (Pitts 1996)
- Other notions of relation lifting, e.g.
  - Free lifting? (Kammar 2014)
  - $\mathbb{TT}$-lifting? (Katsumata 2005, 2013)
  - Codensity lifting? (Katsumata, Sato, and Uustalu 2018)

  Not relators!
- Formalisation

Thank you!