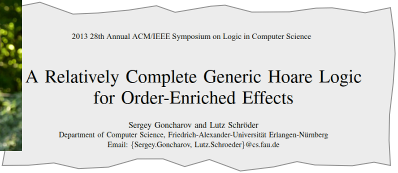
From KAT through Monad-Based Hoare Logic to Stone Duality

Sergey Goncharov

February 11, 2026

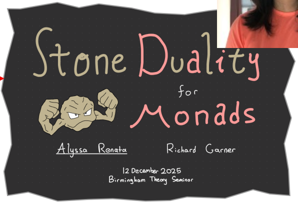
University of Birmingham

This Talk


2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science

A Relatively Complete Generic Hoare Logic for Order-Enriched Effects

Sergey Goncharov and Lutz Schröder
Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg
Email: {Sergey.Goncharov, Lutz.Schroeder}@cs.fau.de



This talk

(work in progress)

Plan

- Recap of Kleene algebra with Tests (KAT)
- How good is it for generic reasoning about programs?
- Categories and Monads
- Monad-based Hoare Logic
- Representing monads and Stone duality

Kleene Algebra with Tests

Kleene Algebra

Kleene algebra is

- Idempotent semiring $(S, 0, 1, +, ;)$
  - $(S, 0, +)$ is commutative and idempotent monoid
  - $(S, 1, ;)$ is monoid
  - distributive laws:

$$p; (q + r) = p; q + p; r \qquad\qquad p; 0 = 0$$
$$(p + q); r = p; r + q; r \qquad\qquad 0; p = 0$$

  (thus, S is partially ordered: $x \leq y$ iff $x + y = y$)

- ... plus, Kleene iteration p^*, such that

$$\boxed{p^*; q} = q + p; \boxed{p^*; q} \qquad \text{and} \qquad \boxed{q; p^*} = q + \boxed{q; p^*}; p$$

  are least solutions, in particular: $p^* = 1 + p; p^* = 1 + p^*; p$

... with Tests

- Programming view: algebra elements = programs
  - 0 – divergence and/or deadlock, 1 – neutral program, etc.
- Kleene algebra with tests (KAT) adds control via tests:
  - Kleene sub-algebra B
  - B is Boolean algebra under $(0, 1, ;, +)$
  - alternatively (and non-trivially!), B supports complementation $\overline{(-)}: B \to B$, such that

$$a; \bar{a} = 0 \qquad\qquad a + \bar{a} = 1$$

- This enables encodings:
  - Branching (**if** b **then** p **else** q) as $b; p + \overline{b}; q$
  - Looping (**while** b **do** p) as $(b; p)^*; \overline{b}$
  - Hoare triples $\{a\}\, p\, \{b\}$ as $a; p; b = a; p$

Example: while b **do** p = **if** b **then** p **else** (**while** b **do** p)

Kleene Algebra: Use

- Regular expressions
- Algebraic language of finite state machines and beyond
- Relational semantics of programs
- Relational reasoning and verification, e.g. via dynamic logic
- Plenty of extensions:
  - modal $\Rightarrow$ modal Kleene algebra (Struth et al.)
  - stateful $\Rightarrow$ KAT + B! (Grathwohl, Kozen, Mamouras)
  - concurrent $\Rightarrow$ concurrent Kleene algebra (Hoare et al.)
  - nominal $\Rightarrow$ nominal Kleene algebra (Kozen et al.)
  - differential equations $\Rightarrow$ differential dynamic logic (Platzer et al.)
  - network primitives $\Rightarrow$ NetKAT (Foster et al.)
  - etc., etc., etc.
- decidability and completeness (most famously w.r.t. language interpretation and relational interpretation)

KAT: Relational Model

Fix set X

- Programs: relations $R \subseteq X \times X$
- $+ =$ set-theoretic union
- $; =$ relational composition
- 0 – empty relation $\emptyset$, 1 – identity relation $\{(x, x) \mid x \in X\}$
- $R^\star =$ reflexive transitive closure $1 \cup R \cup R; R \cup \ldots$

Tests:

- predicates $b \subseteq X$
- identified with relations $\{(x, x) \mid x \in b\}$

Soundness of Hoare Logic in KAT

Check soundness of rule

$$\frac{\{a\}\ p\ \{b\} \qquad \{b\}\ q\ \{c\}}{\{a\}\ p; q\ \{c\}}$$

Recall encoding:

$$\{x\}\, r\, \{y\} \quad \equiv \quad x; r; y = x; r$$

Assume:

$$a; p; b = a; p \qquad b; q; c = b; q$$

Thus:

$$\begin{aligned} a; p; q; c &= a; p; b; q; c && \text{(since } a; p = a; p; b\text{)} \\ &= a; p; b; q && \text{(since } b; q; c = b; q\text{)} \\ &= a; p; q \end{aligned}$$

KAT for Semantics?

- KAT incorporates many general and robust semantic idioms:
    - Tests as well-behaved programs
    - Encoding of **if** and **while** through tests
    - Equational encoding of Hoare triples
- It also incorporates many specific design choices, not ubiquitous in semantics
    1. Nondeterminism
    2. Identification of tests and assertions
        - b in ($\mathbf{if}\, b\, \mathbf{then}\, p\, \mathbf{else}\, q$) is decision (decidable predicate)
        - b in $\{a\}\, p\, \{b\}$ is assertion (possibly undecidable predicate)
    3. Restrictive axioms, e.g. right strictness $p; 0 = 0$ — ensuing analysis:
        - Goncharov, *Shades of Iteration: From Elgot to Kleene*
        - Goncharov, Uustalu, *A Unifying Categorical View of Nondeterministic Iteration and Tests*

Here: Dwell on 1 & 2!

KAT View:

- Programs form a monoid $(S, 1, ;)$
- 1 is "skip" program, ; models sequencing

Categorical View:

- A category is a many-object generalization of a monoid
- Objects = types / state spaces
- Morphisms $A \to B$ = programs from A to B
- Semirings $\rightsquigarrow$ categories enriched in pointed semilattices

Example:

- Relational model $\rightsquigarrow$ category of relations

Monad-Based Hoare Logic

Monads for Effects

- Ambient categories: $\mathbf{C} = \mathbf{Set}$, $\mathbf{C}$ – domains, nominal sets, ...
- Monads T on $\mathbf{C}$, to model effects, such as
  - Nondeterminism: $TX = \mathcal{P}X$
  - Store: $TX = S \to S \times X$
  - Exceptions: $TX = X + E$
  - Probability: $TX =$ probability distributions

Definition (Monad = Kleisli Triple): $(T,\ \eta,\ (\_)^\sharp)$ where for $f : A \to TB$ we have $f^\sharp : TA \to TB$ and

$$\eta^\sharp = \mathrm{id} \qquad \eta; f^\sharp = f \qquad (g; f^\sharp)^\sharp = g^\sharp; f^\sharp$$

Definition (Kleisli category): $\mathbf{C}_T$: same objects as $\mathbf{C}$, morphisms $A \to B$ are $\mathbf{C}(A, TB)$, $\eta\colon A \to TA$ – identity, Kleisli composition: $f, g \mapsto (f\colon A \to TB); (g\colon B \to TC)^\sharp$

Example: Category of relations $= \mathbf{Set}_{\mathcal{P}}$

Strong Monads and Do-Notation

- Strong monads also support strength (**C** must have products):

$$\tau_{A,B} : A \times TB \to T(A \times B)$$

- Then we can generalize sequencing in $\mathbf{C_T}$:

$$p : \Gamma \to TA, \quad f : \Gamma \times A \to TB \quad \mapsto \quad \mathbf{do}\, x \leftarrow p; f(x)$$

- Strong monad laws:

$$\mathbf{do}\, x \leftarrow p; \eta(x) = p$$
$$\mathbf{do}\, x \leftarrow \eta(t); f(x) = f(t)$$
$$\mathbf{do}\, x \leftarrow (\mathbf{do}\, y \leftarrow p; q); r = \mathbf{do}\, y \leftarrow p; x \leftarrow q; r$$

Examples:

- State monad: do $x \leftarrow get; put(x+1)$
- Nondeterminism: do $x \leftarrow \{1,2\}; y \leftarrow \{3,4\}; \eta(x,y) = \{(1,3),(1,4),(2,3),(2,4)\}$

Enrichment

- Enrichment in $\mathbf{V}$ means: every $\mathrm{Hom}(X, Y)$ is in $\mathbf{V}$, and compositions $f; (-)$, $(-); f$ are $\mathbf{V}$-morphisms
- Recall: KAT corresponds to enrichment in pointed semilattices
- Our design choice – à la domain theory: enrichment of $\mathbf{C_T}$ in $\mathbf{bdCpo}_\perp$ =
  - complete partial orders,
  - .. with least element $\perp$,
  - .. and upper bounded sets have least upper bounds

  Additionally, $\perp$ and $\sqcup$ are substitution-stable and

$$\mathbf{do}\, x \leftarrow \perp; p = \perp \qquad \mathbf{do}\, x \leftarrow (p \sqcup q); r = \mathbf{do}\, x \leftarrow p; r \sqcup \mathbf{do}\, x \leftarrow p; r$$

  (not e.g. $\mathbf{do}\, x \leftarrow p; \perp = \perp$ – e.g. failed by exception monad)

Note: Our $\mathbf{bdCpo}_\perp$-monads $\neq$ $\mathbf{bdCpo}_\perp$-enriched monads!

Examples of $\mathbf{bdCpo}_\perp$-Monads

Examples of $\mathbf{bdCpo}_\perp$-monads:

- **Powerset monad** $\mathcal{P}$ (non-deterministic functions)
- **Partiality monad** $X_\perp = X + 1$ (partial functions)
- **Partial store monad** $TX = S \to (X \times S)_\perp$ (reading/writing from store)
- **Countable subdistribution monad** $TX = \left\{d \colon X \to [0,1] \mid \sum d \leq 1\right\}$

Non-Examples

- Non-empty powerset/distributions/store (no $\perp$)
- Finite powerset/finite distributions (no directed joins)

Tests as Decisions

- Standard semantics of if-then-else: given $b\colon \Gamma \to 1+1 = 2$ and $p, q\colon \Gamma \to A$

$$\textbf{if}\, b\, \textbf{then}\, p\, \textbf{else}\, q = \textbf{case}\, b\, \textbf{of}\, \textbf{inl}\, \_ \mapsto p;\ \textbf{inr}\, \_ \mapsto q$$

- This is general enough: for $b\colon X \to T2$, we can define

$$\textbf{if}\, b\, \textbf{then}\, p\, \textbf{else}\, q = \textbf{do}\, x \leftarrow b;\ \textbf{if}\, x\, \textbf{then}\, p\, \textbf{else}\, q$$

Tests as Assertions

- Define $? : 2 \to T1$

$$b? = \textbf{if}\, b\, \textbf{then}\, \eta(\star)\, \textbf{else}\, \bot$$

- It can be shown that

$$\textbf{if}\, b\, \textbf{then}\, p\, \textbf{else}\, q = \textbf{do}\, b?; p \sqcup \textbf{do}\, \bar{b}?; q,$$

  like in KAT!

- This uses bounded completeness

While-Loops

- Given $b\colon \Gamma \to 2$, $p\colon \Gamma \to T1$, $\mathbf{while}\, b\, \mathbf{do}\, p$ is the least solution of equation

$$\mathbf{while}\, b\, \mathbf{do}\, p = \mathbf{if}\, b\, \mathbf{then}\, (\mathbf{do}\, p;\ \mathbf{while}\, b\, \mathbf{do}\, p)\, \mathbf{else}\, \eta(\star)$$

- This can be computed thanks to enrichment by Kleene fixpoint theorem

Innocent Monads for Assertions

- We have conversion $? \colon 2 \to T1$ from tests to assertions
- But not all programs in $T1$ may be assertions,

  e.g. for partial store monad $T1 = S \to S \times 2$
- So, how can we specify assertions?

Definition (Innocent Monads): $\mathbf{bdCpo}_\perp$-monad P is innocent if it is

- commutative: $\mathbf{do}\, x \leftarrow p; y \leftarrow q; \eta(x, y) = \mathbf{do}\, y \leftarrow q; x \leftarrow p; \eta(x, y)$
- copy-monad: $\mathbf{do}\, x \leftarrow p; y \leftarrow p; \eta(x, y) = \mathbf{do}\, x \leftarrow p; \eta(x, x)$
- weakly discardable: $\mathbf{do}\, x \leftarrow p; \eta(\star) \sqsubseteq \eta(\star)$

One consequence: $p \sqcap q = \mathbf{do}\, p; q$ (like in KAT!)

Example: Partial reader monad $PX = S \to X + 1$

Frame of Assertions

Definition (Assertions): Assertions are morphisms $\Gamma \to P1$ for innocent monad P

Recall that set F is called frame if

- F is lattice
- F has all joins
- F validates frame distributively: $a \wedge \bigvee_{i \in I} b_i = \bigvee_{i \in I}(a \wedge b_i)$

Theorem: Assertions $\Gamma \to P1$ form a frame, in particular internal Heyting algebra

Thus, we can interpret over $P1$: logical connectives, quantifiers, fixpoints of predicates

Example: For partial reader monad, $P1 = S \to 2 \cong \mathcal{P}(S)$ – Boolean algebra of predicates on S

Hoare Triples

- Given $\mathbf{bdCpo}_\perp$-monad T with innocent submonad P, we interpret Hoare triples

$$\{\phi\}\; x \leftarrow p\; \{\psi(x)\} \quad \equiv \quad \mathbf{do}\, \phi; x \leftarrow p; \psi(x); \eta(x) \;=\; \mathbf{do}\, \phi; p$$

- This allows defining generic Hoare calculus
- Main result of our LICS 2013 paper: soundness and relative completeness
- Proof idea: show expressibility of weakest (liberal) preconditions

$$\mathbf{wp}(x \leftarrow p, \psi(x)) = \bigsqcup \{\phi \mid \{\phi\}\; x \leftarrow p\; \{\psi(x)\}\}$$

For example: $\mathbf{wp}(\mathbf{while}\, b\, \mathbf{do}\, p, \psi) = \nu\gamma.\, \mathbf{if}\, b\, \mathbf{then}\, \mathbf{wp}(p, \gamma)\, \mathbf{else}\, \psi$

Representing Innocent Monads

Innocent Monads on Set

$\mathbf{bdCpo}_{\perp}$-monads are very general, but innocent monads tend to be specific

Examples:

- Starting from subdistributions $TX = \{d \colon X \to [0,1] \mid \sum d \leq 1\}$, the largest copy-submonad is maybe-monad $PX = X + 1$
- Largest weekly discardable submonad of partial store monad $TX = S \to S \times X + 1$ is partial reader monad $PX = S \to X + 1$

We can very generally define "largest copy submonad" and "largest weekly discardable submonad" by equalizers, e.g.

$$PX \hookrightarrow TX \underset{\psi \circ \Delta}{\overset{T\Delta}{\rightrightarrows}} T(X \times X)$$

Not "largest commutative submonad"! But weakly-discardable copy-monads tend to be commutative

Representability Question

Question: Is every innocent monad on **Set** submonad of partial reader monad

$$PX = S \to X + 1$$

for some S?

Frame Monad

Given frame F, $TX = F^X$ extends to monad on **Set**:

- $\eta(x)(x') = \begin{cases} \top & (x = x') \\ \bot & (x \neq x') \end{cases}$
- $\mathbf{do}\, x \leftarrow p; f(x) = y \mapsto \bigsqcup_x p(x) \sqcap f(x)(y)$

Frame monads are almost innocent, but fail to be copy

Example: If $F = 2$, $TX = 2^X$ – powerset monad:

$$\mathbf{do}\, x \leftarrow p; f(x) = \bigcup_x \{y \mid x \in p \wedge y \in f(x)\} = \bigcup_{x \in p} f(x)$$

Representability in Frame Monads

Theorem: Let P be innocent monad. Then P is isomorphic to largest copy-submonad of $P1^{(-)}$, and isomorphism preserves order

Isomorphism $\alpha\colon P \to P1^{(-)}$:

$$\alpha_X(p \in PX)(x \in X) = (P\,!)(p \sqcap \eta(x))$$

Copy submonad is identified by condition

$$x \neq y \implies p(x) \sqcap p(y) = \bot \qquad (p\colon X \to P1)$$

Example: Start with $PX = S \to X + 1 \quad \rightsquigarrow \quad P1^X = (2^S)^X \cong S \to \mathcal{P}(X)$

Copy condition for $p, q\colon S \to \mathcal{P}(X)$: $p(s) \cap q(s) = \emptyset \quad \rightsquigarrow \quad$ largest copy-submonad is P

Topological State Monad

- Let S be any topological space, and $\mathcal{O}(S)$ be its frame of opens
- Topological state monad (on **Set**!): $TX = S \to_{cont} X_\perp$ (continuous functions) to one-point compactification of discrete space X
- $P1 = 1_\perp^S \cong \mathcal{O}(S)$ where $1_\perp$ – Sierpiński space
- By restricting to copy-submonad of $(\mathcal{O}(S))^{(-)}$, we thus identify S as points of frame of opens $\mathcal{O}(S)$!

Stone Duality

Generally, Stone dualities:

$$
\begin{tikzcd}
\mathit{Spaces} \arrow[r, bend left, "\mathcal{O}"] & \mathit{Algebras} \arrow[l, bend left, "pt"]
\end{tikzcd}
$$

Examples:

- Spaces = Stone spaces, Algebras = Boolean algebras
- Spaces = Sets, Algebras = Complete atomic Boolean algebras
- $\boxed{\text{Spaces} = \text{Sober spaces, Algebras} = \text{Spatial frames}}$

Spatial frames = frames with "enough points" = isomorphic to $\mathcal{O}(S)$ of some space = those F, for which any $p, q \in F$ can be separated by some frame morphism $F \to 2$

Spatiality

Definition: Call Innocent monad P spatial if $P1$ is spatial frame

Theorem: Every spatial innocent monad embeds into a partial reader monad

Proof Idea: State space S = frame morphisms $P1 \to 2$ = completely prime filters

Monad morphism $\alpha_X \colon PX \to (S \to X + 1)$:

$$\alpha_X(p \in PX)(s \colon P1 \to 2) = \begin{cases} \mathbf{inl}\, x & \text{if } s(\delta_x^\sharp(p)) = \top \\ \mathbf{inr}\, \star & \text{otherwise} \end{cases}$$

where $\delta_x(x) = \eta(\star)$, $\delta_x(x') = \bot$

Current State

Conjecture: For non-spatial P there is no embedding of P to partial state monad, preserving both meets and joins

- Potential example: largest copy submonad of $F^{(-)}$ with F – frame (actually, Boolean algebra) of regular opens of $[0, 1]$

  (Regular opens $=$ opens that are equal to interiors of their closure)
- Regular opens embed to all opens $\rightsquigarrow$ possibly, we can embed to partial state monad, if not insist on join preservation

Further Work

- When exactly embedding holds?
- How good/bad can it be (Meet-preserving? Order-preserving?)
- Any impact of accessibility (=rank)?
- Representation for complete semiring module monads $S^{(-)}$ (S – complete semiring)
- Related: Representation for innocence, without copy