

Step-Indexed Kripke Models over Recursive Worlds

Lars Birkedal

IT University of Copenhagen

Joint work with

Bernhard Reus, Jan Schwinghammer, Kristian Støvring, Jacob Thamsborg, Hongseok Yang

May, 2010

Introduction

- Simple semantic techniques for reasoning about realistic programming languages
 - higher-order store
 - dynamic allocation
 - recursive types
 - impredicative polymorphism

Application Areas

Unary:

- ML references
- logics for higher-order store (nested Hoare triples)
- capability calculi
- storable locks and concurrency

Relational:

- ctx. equivalence (relational parametricity, data abstraction)
- effect-based program transformations (a la Benton-Hofmann)
- compositional compiler correctness and soundness of optimizations (extending Benton's work to higher-order store)

Case Study: Unary model of ML refs

- Language: $F_{\mu, \text{ref}}$.
- Call-by-value operational semantics.
- Typing judgements:

$$\Xi; \Gamma; \Sigma \vdash M : \tau,$$

where

$$\begin{aligned}\Xi &= \alpha_1, \dots, \alpha_n \\ \Gamma &= X_1 : \tau_1, \dots, X_m : \tau_m \\ \Sigma &= l_1 : \tau_1, \dots, l_k : \tau_k\end{aligned}$$

Unary Model of ML refs — Ideas

- Impredicative polymorphism:
 - Types as predicates over some set V of values.
- Dynamic allocation of references:
 - Kripke Model, worlds capturing types of allocated locations
 - Types = predicates indexed over worlds
- In Summa: recursive equation:

$$\begin{aligned}V &= \text{set of values, including locations} \\ \mathcal{W} &= \mathbb{N} \rightarrow_{fin} \mathcal{T} \\ \mathcal{T} &= \mathcal{W} \rightarrow_{mon} \text{Pred}(V)\end{aligned}$$

- Our approach: solve equation in category of metric spaces.

Unification of Methods

- Idea first developed using domain-theoretic model of programming language [BST - FOSSACS'09, MSCS'10]
- Now show that it applies to operational semantics via step-indexing
 - pros: simpler, scales well to concurrency
 - high-level understanding of step-indexing
 - *essence* of step-indexing
 - generalizes Hobor et. al.'s Indirection Theory [POPL'10], which is aimed at giving general description of step-indexed models
 - has been formalized in Coq [C. Varming & LB]
- Denotational approach still useful
 - gives more abstract model for relational reasoning
 - reasoning in the model is at same abstraction level as modal logics for reasoning about step-indexed models, see [BST'10, Dreyer et. al., LICS'09, POPL'10]

Outline

- Background on metric spaces
- Step-indexed Model of ML refs
- Pointers to other applications

Recall:

- An *ultrametric space* is a metric space (D, d) that instead of triangle inequality satisfies the stronger *ultrametric inequality*:

$$d(x, z) \leq \max(d(x, y), d(y, z)).$$

- A function $f : D_1 \rightarrow D_2$ from a metric space (D_1, d_1) to a metric space (D_2, d_2) is *non-expansive* if $d_2(f(x), f(y)) \leq d_1(x, y)$ for all x and y in D_1 .
- A function $f : D_1 \rightarrow D_2$ from a metric space (D_1, d_1) to a metric space (D_2, d_2) is *contractive* if there exists $\delta < 1$ such that $d_2(f(x), f(y)) \leq \delta \cdot d_1(x, y)$ for all x and y in D_1 .
- CBUlt_{ne} is the category with complete, non-empty, 1-bounded ultrametric spaces and non-expansive functions.

- We'll work with *bisected* metric spaces: all non-zero distances are of the form 2^{-n} , for some natural number $n \geq 0$.
- Write $x \stackrel{n}{=} y$ to mean that $d(x, y) \leq 2^{-n}$.
- Fact: $\stackrel{n}{=}$ is an equivalence relation (since ultrametric).
- Fact: $x \stackrel{0}{=} y$ always holds (since space 1-bunded).
- Fact: $f : X \rightarrow Y$ is non-expansive iff, for all $n > 0$,
 $x \stackrel{n}{=} x' \Rightarrow f(x) \stackrel{n}{=} f(x')$

CBUIne, III

- CBUIt_{ne} is cartesian closed; the exponential $(D_1, d_1) \rightarrow (D_2, d_2)$ is the set of non-expansive maps with the “sup”-metric $d_{D_1 \rightarrow D_2}$ as distance function:

$$d_{D_1 \rightarrow D_2}(f, g) = \sup\{d_2(f(x), g(x)) \mid x \in D_1\}.$$

- Solutions to recursive domain equations for locally contractive functors (America-Rutten):
- A functor $F : \text{CBUIt}^{\text{op}} \times \text{CBUIt} \rightarrow \text{CBUIt}$ is *locally contractive* if there exists $\delta < 1$ such that

$$d(F(f, g), F(f', g')) \leq \delta \cdot \max(d(f, f'), d(g, g'))$$

for all non-expansive functions $f, f', g,$ and g' .

Uniform Predicates

- Uniform predicates:

$$UPred(Val) = \{p \subseteq \mathbb{N} \times Val \mid \forall (k, v) \in p. \forall j \leq k. (j, v) \in p\}$$

(“uniform” by analogy to complete uniform per’s in realizability models).

- For $p \in UPred(Val)$ and $k \in \mathbb{N}$, let

$$\bar{p}^k = \{(m, v) \in p \mid m < k\}$$

- Distance:

$$d(p, q) = \begin{cases} 2^{-\max\{k \mid \bar{p}^k = \bar{q}^k\}} & \text{if } p \neq q \\ 0 & \text{otherwise.} \end{cases}$$

- Lemma $(UPred(Val), d)$ is a well-defined object in $CBUlt_{ne}$.

Worlds

Lemma

Let $(D, d) \in \text{CBUlt}$. The set $\mathbb{N} \rightarrow_{\text{fin}} D$ with distance function:

$$d'(\Delta, \Delta') = \begin{cases} \max \{d(\Delta(I), \Delta'(I)) \mid I \in \text{dom}(\Delta)\} & \text{if } \text{dom}(\Delta) = \text{dom}(\Delta') \\ 1 & \text{otherwise.} \end{cases}$$

is in CBUlt .

Extension ordering: $\Delta \leq \Delta'$ iff

$$\text{dom}(\Delta) \subseteq \text{dom}(\Delta') \wedge \forall I \in \text{dom}(\Delta). \Delta(I) = \Delta'(I).$$

Space of Types, I

Lemma

$$F(D) = (\mathbb{N} \rightarrow_{fin} D) \rightarrow_{mon} UPred(Val)$$

(monotone, non-expansive maps) defines a functor
 $F : CBUltne^{op} \rightarrow CBUltne$.

Theorem

There exists $\hat{\mathcal{T}} \in CBUltne$ such that

$$\hat{\mathcal{T}} \cong \frac{1}{2} \cdot ((\mathbb{N} \rightarrow_{fin} \hat{\mathcal{T}}) \rightarrow_{mon} UPred(Val))$$

is an iso in $CBUltne$.

Example proof

Lemma

$$F(D) = (\mathbb{N} \rightarrow_{fin} D) \rightarrow_{mon} UPred(Val)$$

(monotone, non-expansive maps) defines a functor
 $F : CBUltne^{op} \rightarrow CBUltne$.

Proof.

SFTS: limit of Cauchy sequence of monotone maps is also monotone.
Let $(\nu_m)_{m \in \omega}$ be a Cauchy seq of monotone maps, with limit ν .

- TS: $\nu(w) \subseteq \nu(w')$, for all $w \sqsubseteq w'$.
- SFTS: $\forall n. \overline{\nu(w)}^n \subseteq \overline{\nu(w')}^n$.
- Fix n . By limit, there exists m , s.t. $d(\nu, \nu_m) \leq 2^{-n}$, so
- $\overline{\nu(w)}^n = \overline{\nu_m(w)}^n$, for all w .
- Then

$$\overline{\nu(w)}^n = \overline{\nu_m(w)}^n \subseteq \overline{\nu_m(w')}^n = \overline{\nu(w')}^n.$$

Space of Types, II

Definition

$$\begin{aligned}\mathcal{W} &= N \rightarrow_{fin} \widehat{\mathcal{T}} \\ \mathcal{T} &= \mathcal{W} \rightarrow_{mon} UPred(Val).\end{aligned}$$

Observe: $\widehat{\mathcal{T}} = \frac{1}{2}(\mathcal{W} \rightarrow_{mon} UPred(Val))$ and

$$\begin{aligned}& w \stackrel{n}{=}_{\mathcal{W}} w' \\ \Rightarrow w(I) & \stackrel{n}{=}_{\widehat{\mathcal{T}}} w'(I) \\ \Rightarrow w(I) & \stackrel{n-1}{=}_{\mathcal{W} \rightarrow_{mon} UPred(Val)} w'(I) \\ \Rightarrow \forall w_0 \in \mathcal{W}. w(I)(w_0) & \stackrel{n-1}{=}_{UPred(Val)} w'(I)(w_0)\end{aligned}$$

Interpretation of Types, I

Define non-expansive map

$$\llbracket \Xi \vdash \tau \rrbracket : \mathcal{T}^{|\Xi|} \rightarrow \mathcal{T}$$

by induction on τ :

$$\llbracket \Xi \vdash \tau \rrbracket_{\eta} : \mathcal{W} \rightarrow_{\text{mon}} \text{UPred}(\text{Val})$$

$$\llbracket \Xi \vdash \mathbf{1} \rrbracket_{\eta} \mathbf{w} = \{(k, ()) \mid k \in \mathbb{N}\}$$

$$\begin{aligned} \llbracket \Xi \vdash \text{ref } \tau \rrbracket_{\eta} \mathbf{w} = \{(k, l) \mid l \in \text{dom}(\mathbf{w}) \wedge \\ \forall \mathbf{w}' \sqsupseteq \mathbf{w}. i(\mathbf{w}(l))(\mathbf{w}') \stackrel{k}{=} \llbracket \Xi \vdash \tau \rrbracket_{\eta} \mathbf{w}'\} \end{aligned}$$

$$\llbracket \Xi \vdash \alpha \rrbracket_{\eta} \mathbf{w} = \eta(\alpha)(\mathbf{w})$$

$$\begin{aligned} \llbracket \Xi \vdash \forall \alpha. \tau \rrbracket_{\eta} \mathbf{w} = \{(k, v) \mid \forall \tau'. \forall r \in \mathcal{T}. \forall \mathbf{w}' \sqsupseteq \mathbf{w}. \\ \forall i \leq k. (i, v[\tau']) \in \mathcal{E}[\llbracket \Xi, \alpha \vdash \tau \rrbracket_{\eta[\alpha \mapsto r]} \mathbf{w}']\} \end{aligned}$$

Interpretation of Types, II

- Recursive Types:

$$\llbracket \Delta \vdash \mu\alpha.\tau \rrbracket_{\eta} = \text{fix}(\lambda r. \lambda w. \{(k, \text{fold } v) \mid k > 0 \Rightarrow (k - 1, v) \in \llbracket \Delta, \alpha \vdash \tau \rrbracket_{\eta[\alpha \mapsto r]} w\})$$

- Uses Banach's fixed point theorem.
- Contractiveness ensured by use of $k - 1$.

Interpretation of Types, III

$$\begin{aligned} \llbracket \Xi \vdash \tau \rightarrow \tau' \rrbracket_{\eta} \mathbf{w} &= \{(k, \nu) \mid \forall \nu' \in \mathit{Val}. \forall \mathbf{w}' \sqsupseteq \mathbf{w}. \forall i \leq k. \\ &\quad (i, \nu') \in \llbracket \Xi \vdash \tau \rrbracket_{\eta} \mathbf{w}' \Rightarrow (i, \nu \nu') \in \mathcal{E} \llbracket \Xi \vdash \tau' \rrbracket_{\eta} \mathbf{w}'\} \end{aligned}$$

$$\mathcal{E} \llbracket \Xi \vdash \tau \rrbracket_{\eta} : \mathcal{W} \rightarrow_{\text{mon}} \mathit{UPred}(\mathit{Exp})$$

$$\begin{aligned} \mathcal{E} \llbracket \tau \rrbracket_{\eta} \mathbf{w} &= \{(k, t) \mid \forall i \leq k. \forall h, h'. \forall \nu \in \mathit{Val}. \\ &\quad (h :_k \mathbf{w} \wedge (t \mid h) \mapsto^i (\nu \mid h')) \\ &\quad \Rightarrow (\exists \mathbf{w}' \sqsupseteq \mathbf{w}. h' :_{k-i} \mathbf{w}' \wedge (k-i, \nu) \in \llbracket \tau \rrbracket_{\eta} \mathbf{w}')\} \end{aligned}$$

$$\begin{aligned} h :_k \mathbf{w} &\iff \forall i < k. \mathit{dom}(h) = \mathit{dom}(\mathbf{w}) \wedge \\ &\quad \forall l \in \mathit{dom}(\mathbf{w}). (i, h(l)) \in \mathbf{w}(l)(\mathbf{w}) \end{aligned}$$

Interpretation of open expressions

$$\llbracket \Xi \vdash \Gamma \rrbracket_\eta : W \rightarrow \text{UPred}(\text{Val}^{|\Gamma|})$$

$$\llbracket \emptyset \rrbracket_\eta \mathbf{w} = \{()\}$$

$$\llbracket \Gamma, x : \tau \rrbracket_\eta \mathbf{w} = \{(k, \rho[x \mapsto v]) \mid (k, \rho) \in \llbracket \Gamma \rrbracket_\eta \mathbf{w} \wedge (k, v) \in \llbracket \tau \rrbracket_\eta \mathbf{w}\}$$

$$\llbracket \Sigma \rrbracket : \text{UPred}(W)$$

$$\llbracket \Sigma \rrbracket = \{(k, \mathbf{w}) \mid \forall (l : \tau) \in \Sigma. (k, l) \in \llbracket \emptyset \vdash \text{ref } \tau \rrbracket_{()} \mathbf{w}\}$$

$$\Xi; \Gamma; \Sigma \vdash t :^{\text{log}} \tau \iff$$

$$\exists \alpha_1, \dots, \alpha_n. \Xi = \alpha_1, \dots, \alpha_n \wedge$$

$$\forall \tau_1, \dots, \tau_n. \forall k \geq 0. \forall \eta. \forall \rho. \forall \mathbf{w}.$$

$$(\eta \in \mathcal{T}^{|\Xi|} \wedge (k, \rho) \in \llbracket \Xi \vdash \Gamma \rrbracket_\eta \mathbf{w} \wedge (k, \mathbf{w}) \in \llbracket \Sigma \rrbracket)$$

$$\Rightarrow ((k, (\rho(t))[\alpha_1 := \tau_1, \dots, \alpha_n := \tau_n]) \in \mathcal{E} \llbracket \Xi \vdash \tau \rrbracket_\eta \mathbf{w})$$

Well-definedness

- Metric setup tells you what you have to show:
 - non-expansiveness of $\llbracket \Xi \vdash \tau \rrbracket$
 - non-expansiveness of $\llbracket \Xi \vdash \tau \rrbracket_\eta$
 - contractiveness of map for recursive types.
- Simple calculations.

Example lemma

Lemma

If $s :_k w$ and $w \stackrel{n}{\equiv}_{\mathcal{W}} w'$ and $k < n$, then also $s :_k w'$.

Proof.

TS: $\forall j < k. \text{dom}(s) = \text{dom}(w') \wedge \forall l \in \text{dom}(w'). (j, s(l)) \in w'(l)(w')$.

Sps. $k > 0$; then $n > 0$. Let $j < k$. By $w \stackrel{n}{\equiv}_{\mathcal{W}} w'$, we get

$\text{dom}(w) = \text{dom}(w') \wedge \forall l \in \text{dom}(w). \forall w_0. w(l)(w_0) \stackrel{n-1}{\equiv} w'(l)(w_0)$. Since $\text{dom}(s) = \text{dom}(w)$ by the assumption that $s :_k w$ (using $k > 0$), we get $\text{dom}(s) = \text{dom}(w')$. Moreover,

$$w(l)(w) \stackrel{n}{\equiv} w(l)(w') \stackrel{n-1}{\equiv} w'(l)(w')$$

since $w(l)$ is non-expansive, and since $w \stackrel{n}{\equiv}_{\mathcal{W}} w'$. Thus, as $(j, s(l)) \in w(l)(w)$ by assumption, and since $j < k \leq n - 1$, we also get $(j, s(l)) \in w'(l)(w')$, as desired. \square

Soundness

Theorem

If $\Xi; \Gamma; \Sigma \vdash t : \tau$, then $\Xi; \Gamma; \Sigma \vdash t :^{\text{log}} \tau$.

Specialization to Indirection Theory

- Indirection Theory. Hobor et. al. POPL'10
 - General formulation of step-indexed models. Also observe cannot solve world-equation in sets. Instead describe approximate solutions and show how they can be used in many step-indexed models.
- We prove that one can derive an approx. solution a la Indirection Theory from one of our metric equations (see paper for detailed formulation and formal theorems).
- Corollary: applies to all the models described by indirection theory.

Advantages of metric approach

(some propaganda :-))

- Useful guiding framework.
- Supporting theory (e.g., recursive equations when spaces equipped with structure).
- Supports recursively-defined operations on worlds.
- Connection between step-indexing and metric spaces known from start of step-indexing (Appel-McAllester); but useful not to forget the connection!
- Also formalized in Coq (Varming, Birkedal).

Recursively defined operation on worlds

- For describing how to extend world-dependent invariants.
- Has been used for Nested Hoare Triples & Capability Calculus (stored code)
- Capability Calculus setup: $\mathcal{W} \cong \frac{1}{2}\mathcal{W} \rightarrow \mathit{UPred}^\uparrow(\mathit{Heap})$.
- Define non-expansive operation $\circ : \mathcal{W} \times \mathcal{W} \rightarrow \mathcal{W}$, s.t. for all $p, r, w \in \mathcal{W}$,

$$\iota^{-1}(p \circ r)(w) = \iota^{-1}(p)(r \circ w) * \iota^{-1}(r)(w).$$

- Intuition:
 - p and r world dependent invariants
 - world-dependency via application
 - $p \circ r$ is the extension of p with r : first extend r with w , and then apply p to that, in addition to “starring on” $r(w)$.
- Well-defined by Banach: intuitively because the \circ on the right is as an argument, below an unfolding via ι^{-1} .

Current / Future Work

- Storable locks and concurrency
- Metric model of Nakano's calculus with a Modality for Recursion.
- Extend capability calculus model with antiframe and fates/observations a la Pottier (more formal connection between recursive world extension operation and use of state transition diagrams by Dreyer et. al.)
- Extend capability calculus model to reason about *shared* data structures (most work so far focused on data abstraction qua separation).
- Semantic model of focusing with cyclic proofs.
- Refine basic setup: Formulations based on simple categories of (pre)sheaves. Generalize solution theory.
- Effect-based program transformations.
- Compiler correctness in presence of higher-order store.
- Extend HTT with better types for higher-order store.

References, I

Available at www.itu.dk/people/birkedal/papers

- Birkedal, Reus, Schwinghammer, Støvring, Thamsborg, Yang: Step-Indexed Kripke Models over Recursive Worlds. Submitted. [step-recworld-conf.pdf](#)
 - Today's material + model of capability calculus (recursive operation defined on worlds) + model of nested triples (recursive operation defined on worlds) + formal relationship to Indirection Theory.
- Birkedal, Støvring, Thamsborg: Realizability semantics of parametric polymorphism, general references, and recursive types. To Appear in MSCS, short version in FOSSACS'09. [parametricity-state-metric-journal.pdf](#)
 - Denotational approach, simple worlds, self-contained. Approximate locations in the denotational model.

References, II

- Schwinghammer, Birkedal, Reus, Yang: Nested Hoare Triples and Frame Rules for Higher-Order Store. CSL'09.
 - Higher-order Frame Rules: recursive operation defined on worlds. Denotational Approach. Unary. `nested-triples-conf.pdf`
- Schwinghammer, Yang, Birkedal, Reus, Pottier: A Semantic Foundation for Hidden State. FOSSACS'10.
 - Separation Logic for Higher-Order Store + Antiframe. Recursive operation defined on worlds. Unary. Denotational Approach. `infohiding-conf.pdf`

References, III

- Ahmed, Dreyer, Rossberg: State-Dependent Representation Independence. POPL'09.
 - Step-indexed relational model. Expressive worlds, many examples. Available at Dreyer's home page.
- Dreyer, Ahmed, Birkedal: Logical Step-Indexed Logical Relations. LICS'09.
 - Logic (LSLR) for step-indexing to get more abstract reasoning. Language with \forall , μ (no higher-order store). Relational. Journal submission: `lslr-journal.pdf`

References, IV

- Dreyer, Neis, Rossberg, Birkedal: A relational modal logic for higher-order stateful ADTs. POPL'10.
 - Logic (LADR) for step-indexing to get more abstract reasoning. Language: $F_{\mu, \text{ref}}$. Relational. `ladr-conf.pdf`
- Dreyer, Neis, Birkedal: The impact of higher-order state and control effects on local relational reasoning.
 - Relational. More expressive worlds via state transition diagrams (can prove all known examples). Call/cc. Several sub-languages; proof method exploiting that. Submitted. `stslr-conf.pdf`

References, V

- Birkedal, Støvring, Thamsborg: A relational realizability model for higher-order stateful ADTs. Submitted to journal.
 - Extending BTS model to LADR worlds, denotational approach. More abstract model than step-model. [relrmhoadt.pdf](#)
- Birkedal, Støvring, Thamsborg: The category-theoretic solution of recursive metric-space equations.
 - Supporting theory. M-categories. To Appear in TCS (after minor revisions). [ITU-TR-2009-119.pdf](#)

References, VI

- Varming and Birkedal: Ultrametric semantics and domain theory in Coq.
 - Coq formalization of solutions to recursive eqn's in M-categories. Application to ML-references (as in this talk). Manuscript. `metric-formalization.pdf`.