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

  $\Xi = \alpha_1, \ldots, \alpha_n$

  $\Gamma = x_1 : \tau_1, \ldots, x_m : \tau_m$

  $\Sigma = l_1 : \tau_1, \ldots, l_k : \tau_k$
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:

  \[
  V = \text{set of values, including locations} \\
  W = \mathbb{N} \rightarrow_{\text{fin}} T \\
  T = \forall W \rightarrow_{\text{mon}} \text{Pred}(V)
  \]

- 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 \(D_1\).

- \(\text{CBUIt}_{\text{ne}}\) 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 $x \equiv^n y$ to mean that $d(x, y) \leq 2^{-n}$.

Fact: $\equiv^n$ is an equivalence relation (since ultrametric).

Fact: $x \equiv^0 y$ always holds (since space 1-bunded).

Fact: $f : X \to Y$ is non-expansive iff, for all $n > 0$, $x \equiv^n x' \Rightarrow f(x) \equiv^n f(x')$
CBUlnè, III

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

\[
d_{D_1 \to D_2}(f, g) = \sup\{d_2(f(x), g(x)) \mid x \in D_1\}.
\]

- Solutions to recursive domain equations for locally contractive functors (America-Rutten):

- A functor \( F : \text{CBUlnè}^{\text{op}} \times \text{CBUlnè} \to \text{CBUlnè} \) is \textit{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:

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

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

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

\[ \overline{p}^k = \{(m, v) \in p \mid m < k\} \]

- Distance:

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

- Lemma \((\text{UPred}(\text{Val}), d)\) is a well-defined object in CBUlt\text{ne}. 

Lars Birkedal (ITU)  
Kripke Models over Recursive Worlds  
Paris 2010 11 / 32
Lemma

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

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

is in \(\text{CBUlt}\).

Extension ordering: \(\Delta \leq \Delta'\) iff

\[
\text{dom}(\Delta) \subseteq \text{dom}(\Delta') \land \forall l \in \text{dom}(\Delta). \Delta(l) = \Delta'(l).
\]
Lemma

\[ F(D) = (\mathbb{N} \to_{\text{fin}} D) \to_{\text{mon}} \text{UPred}(\text{Val}) \]

(monotone, non-expansive maps) defines a functor

\[ F : \text{CBUlt}_{\text{ne}}^{\text{op}} \to \text{CBUlt}_{\text{ne}}. \]

Theorem

There exists \( \hat{T} \in \text{CBUlt}_{\text{ne}} \) such that

\[ \hat{T} \simeq \frac{1}{2} \cdot ((\mathbb{N} \to_{\text{fin}} \hat{T}) \to_{\text{mon}} \text{UPred}(\text{Val})) \]

is an iso in \( \text{CBUlt}_{\text{ne}} \).
Example proof

Lemma

\[ F(D) = (\mathbb{N} \rightarrow_{\text{fin}} D) \rightarrow_{\text{mon}} \text{UPred}(\text{Val}) \]

(monotone, non-expansive maps) defines a functor

\[ F : \text{CBUltne}^{\text{op}} \rightarrow \text{CBUltne}. \]

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 \(\nu\).

- TS: \(\nu(w) \subseteq \nu(w')\), for all \(w \subseteq w'\).
- 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
  \[ \overline{\nu(w)^n} = \overline{\nu_m(w)^n}, \text{ for all } w. \]
- 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**

\[ \mathcal{W} = N \rightarrow_{\text{fin}} \hat{T} \]
\[ \mathcal{T} = \mathcal{W} \rightarrow_{\text{mon}} \text{UPred}(\text{Val}). \]

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

\[ w \models^n_{\mathcal{W}} w' \]
\[ \Rightarrow w(l) \models^n_{\hat{T}} w'(l) \]
\[ \Rightarrow w(l) \models^{n-1}_{\mathcal{W} \rightarrow_{\text{mon}} \text{UPred}(\text{Val})} w'(l) \]
\[ \Rightarrow \forall w_0 \in \mathcal{W}. w(l)(w_0) \models^{n-1}_{\text{UPred}(\text{Val})} w'(l)(w_0) \]
Interpretation of Types, I

Define non-expansive map

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

by induction on $\tau$:

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

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

$$\llbracket \Xi \vdash \text{ref} \tau \rrbracket_\eta w = \{(k, l) \mid l \in \text{dom}(w) \land \forall w' \sqsubseteq w. i(w(l))(w') = k \llbracket \Xi \vdash \tau \rrbracket_\eta w'\}$$

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

$$\llbracket \Xi \vdash \forall \alpha.\tau \rrbracket_\eta w = \{(k, v) \mid \forall \tau'. \forall r \in \mathcal{T}. \forall w' \sqsubseteq w. \forall i \leq k. (i, v[\tau']) \in E\llbracket \Xi, \alpha \vdash \tau \rrbracket_{\eta[\alpha \mapsto r]} w'\}$$
Interpretation of Types, II

- **Recursive Types:**

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

- Uses Banach’s fixed point theorem.
- Contractiveness ensured by use of \(k - 1\).
Interpretation of Types, III

$$\forall \eta. \forall w. \forall i \leq k. (i, v') \in [\Xi \vdash \tau]_\eta w' \Rightarrow (i, v \cdot v') \in \mathcal{E}[\Xi \vdash \tau']_\eta w'$$

$$\mathcal{E}[\Xi \vdash \tau]_\eta : \forall \rightarrow_{mon} UPred(\text{Exp})$$

$$\forall \eta. \forall w. \forall i \leq k. \forall h, h'. \forall v \in \text{Val}. (h :_k w \land (t \mid h) \xrightarrow{i} (v \mid h')) \Rightarrow (\exists w' \supset w. h' :_{k-i} w' \land (k - i, v) \in [\tau]_\eta w')$$

$$h :_k w \iff \forall i < k. \text{dom}(h) = \text{dom}(w) \land \forall l \in \text{dom}(w). (i, h(l)) \in w(l)(w)$$
Interpretation of open expressions

\[ [\Xi \vdash \Gamma]_\eta : W \rightarrow \text{UPred}(\text{Val}^{\lvert \Gamma \rvert}) \]

\[ [\emptyset]_\eta w = \{(\)\} \]

\[ [\Gamma, x : \tau]_\eta w = \{(k, \rho[x \mapsto v]) \mid (k, \rho) \in [\Gamma]_\eta w \land (k, v) \in [\tau]_\eta w\} \]

\[ [\Sigma] : \text{UPred}(W) \]

\[ [\Sigma] = \{(k, w) \mid \forall (l : \tau) \in \Sigma. (k, l) \in [\emptyset \vdash \text{ref } \tau]() w\} \]

\(\Xi; \Gamma; \Sigma \vdash t : \text{log } \tau \iff \exists \alpha_1, \ldots, \alpha_n. \exists (k, \rho) \in [\Xi \vdash \Gamma]_\eta w \land (k, w) \in [\Sigma]) \]

\(\Rightarrow (k, (\rho(t))[\alpha_1:=\tau_1, \ldots, \alpha_n:=\tau_n]) \in \mathcal{E}[\Xi \vdash \tau]_\eta w) \)
Well-definedness

- Metric setup tells you what you have to show:
  - non-expansiveness of $\left\lfloor \exists \vdash \tau \right\rfloor$
  - non-expansiveness of $\left\lfloor \exists \vdash \tau \right\rfloor_\eta$
  - contractiveness of map for recursive types.

- Simple calculations.
Example lemma

Lemma

If \( s :_k w \) and \( w \overset{n}{=} w' \) and \( k < n \), then also \( s :_k w' \).

Proof.

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

Sps. \( k > 0 \); then \( n > 0 \). Let \( j < k \). By \( w \overset{n}{=} w' \), we get

\( \text{dom}(w) = \text{dom}(w') \land \forall l \in \text{dom}(w). \forall w_0. w(l)(w_0) \overset{n-1}{=} w'(l)(w_0) \).

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,

\[
 w(l)(w) \overset{n}{=} w(l)(w') \overset{n-1}{=} w'(l)(w')
\]

since \( w(l) \) is non-expansive, and since \( w \overset{n}{=} w' \). 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.
Soundness

Theorem

If $\Xi; \Gamma; \Sigma \vdash t : \tau$, then $\Xi; \Gamma; \Sigma \vdash t :^{\log} \tau$. 
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: $\mathcal{W} \cong \frac{1}{2} \mathcal{W} \rightarrow UPred^\uparrow(Heap)$.
- Define non-expansive operation $\circ: \mathcal{W} \times \mathcal{W} \rightarrow \mathcal{W}$, s.t. for all $p, r, w \in \mathcal{W}$,

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

- 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 $\iota^{-1}$.
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.
Available at www.itu.dk/people/birkedal/papers

  - 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.

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 ∀, μ (no higher-order store). Relational. Journal submission: lslr-journal.pdf
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
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.

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.