Kripke Models over Recursive Worlds (The Joy of Ultrametric Spaces)

Lars Birkedal

IT University of Copenhagen
Joint work with Kristian Støvring, Jacob Thamsborg, Jan Schwinghammer, Bernhard Reus, Hongseok Yang, Francois Pottier, Carsten Varming

Aug, 2010

Plan

- Intro and overview of recent work on using Kripke models over recursively defined worlds
  - to model type systems and logics for dynamically allocated, often higher order, recursive structures (e.g., higher-order store / storable locks)
  - where recursive worlds defined in category of ultrametric spaces
- Talk: partly technical intro to such uses via an example, partly an overview with pointers to literature (more details in other talks by collaborators this week).
- Will focus on the core issue of recursive worlds, using *simple* notions of worlds, both in denotational and operational settings — for useful applications necessary to
  1. use more sophisticated worlds (e.g., for reasoning about local state)
  2. also use recursively defined operations on worlds (e.g., for higher-order frame rules)

  Will skip 1 almost entirely (intro to such in Derek's talk), will only touch briefly upon 2 (more about this in Jan's talk)

Papers can be found at `www.itu.dk/people/birkedal/papers`.

Case Study — Model of $F_{\mu,\text{ref}}$

[BST - FOSSACS'09, MSCS'10]

Slogan: one domain equation for each of $\forall$, ref, μ.

- $\forall$ impredicative polymorphism: choose to model types as relations *UARel*(V) over a recursively defined predomain V.
- ref general references with dynamic allocation: use Kripke model with recursively defined worlds, approximately of the form:

$$\begin{aligned}\mathcal{T} &= \mathcal{W} \to \mathit{UARel}(V)\\ \mathcal{W} &= \mathbb{N} \to \mathcal{T}\end{aligned}$$

  Solve in CBUlt.
- μ recursive types: relations interpreting types also recursively defined,
  - non-trivial for reference types, leads to novel modeling of locations involving some approximation information.

Predomain V of values

Proposition. There exists a uniform cpo $(V, (\pi_n)_{n\in\omega})$ satisfying:
In pCpo:

$$V \;\cong\; \mathbb{Z} + Loc + 1 + (V \times V) + (V + V) + V + TV + (V \to TV) \quad (1)$$

where

$$\begin{aligned} TV &= (V \to S \to Ans) \to S \to Ans \\ S &= \mathbb{N} \rightharpoonup_{fin} V \\ Ans &= (\mathbb{Z} + Err)_\perp \end{aligned}$$

and

$$\begin{aligned} Loc &= \mathbb{N} \times \overline{\omega} \\ Err &= 1\,. \end{aligned}$$

The functions $\pi_n : V \to V_\perp$ satisfy (and are determined by)

$$\pi_0 = \lambda v.\bot$$

$$\pi_{n+1}(in_{\mathbb{Z}}(k)) = \lfloor in_{\mathbb{Z}}(k) \rfloor$$

$$\pi_{n+1}(in_\times(v_1, v_2)) = \begin{cases} \lfloor in_\times(v_1', v_2') \rfloor & \text{if } \pi_n\, v_1 = \lfloor v_1' \rfloor \text{ and } \pi_n\, v_2 = \lfloor v_2' \rfloor \\ \bot & \text{otherwise} \end{cases}$$

...etc. as you'd expect, except:

$$\pi_{n+1}(in_{Loc}(l, m)) = \lfloor in_{Loc}(l, \min(n+1, m)) \rfloor$$

Untyped Semantics of Terms, I

$[\![t]\!]_X : V^X \to TV$ by induction on t:
Mostly standard, e.g.,

$$[\![\lambda x.\, t]\!]_X\, \rho = \eta(in_{\to}(\lambda v.\ [\![t]\!]_{X,x}\, (\rho[x \mapsto v])))$$

$$[\![t_1\ t_2]\!]_X\, \rho = [\![t_1]\!]_X\, \rho \star \lambda v_1.\ [\![t_2]\!]_X\, \rho \star \lambda v_2. \begin{cases} f\ v_2 & \text{if } v_1 = in_{\to}\, f \\ error & \text{otherwise} \end{cases}$$

Untyped Semantics of Terms, II

- For lookup and assignment we need to consider semantic locations:

$$[\![!\, t]\!]_X\, \rho = [\![t]\!]_X\, \rho \star \lambda v.\, \mathit{lookup}\, v$$

- where $\mathit{lookup}\, v =$

$$\lambda k\, \lambda s. \begin{cases} k\, s(l)\, s & \text{if } v = \lambda_l \text{ and } l \in \mathrm{dom}(s) \\ k\, v'\, s & \text{if } v = \lambda_l^{n+1},\ l \in \mathrm{dom}(s), \text{ and } \pi_n(s(l)) = \lfloor v' \rfloor \\ \bot_{Ans} & \text{if } v = \lambda_l^{n+1},\ l \in \mathrm{dom}(s), \text{ and } \pi_n(s(l)) = \bot \\ \mathit{error}_{Ans} & \text{otherwise} \end{cases}$$

Untyped Semantics of Terms, III

- Adequacy wrt. standard operational semantics can be shown using recursively defined logical relation.
  - Non-trivial, but not too hard using Pitts' technique (with a function-space lattice to deal with nested recursive types), since suffices to consider only *closed* types for adequacy [BST, TLDI'09].
- Now on to typed semantics, i.e., definition of logical relations over the untyped semantics. First define *space of types* using ultrametric spaces.

CBUlt

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 $f : D_1 \to D_2$ 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 $f : D_1 \to D_2$ 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 is the category with complete 1-bounded ultrametric spaces and non-expansive functions.

CBUlt, II

- CBUlt 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:

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

- Thm [America-Rutten]: Solutions to recursive domain equations for locally contractive functors exist.
- A functor $F : \mathrm{CBUlt}^{\mathrm{op}} \times \mathrm{CBUlt} \to \mathrm{CBUlt}$ 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'.

$UARel(V) \in$ CBUlt

Recall [Amadio, Abadi-Plotkin]:

- $UARel(V)$ is the set of admissible relations that are *unifom*: $\varpi_n \in R \to R_\perp$, for all n.
- Such relations are determined by its elements of the form $(\varpi_n\, e, \varpi_n\, e')$.
- $UARel(V) \in$ CBUlt, distance function:

$$d(R,S) = \begin{cases} 2^{-\max\{\, n\in\omega \mid \varpi_n \in R\to S \,\wedge\, \varpi_n \in S \to R \,\}} & \text{if } R \neq S \\ 0 & \text{if } R = S. \end{cases}$$

Worlds

- **Proposition.** Let $(D, d) \in$ CBUlt. The set $\mathbb{N} \rightharpoonup_{fin} D$ with distance function:

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

  is in CBUlt.
- Extension ordering: $\Delta \leq \Delta'$ iff

$$\mathrm{dom}(\Delta) \subseteq \mathrm{dom}(\Delta') \wedge \forall l \in \mathrm{dom}(\Delta).\, \Delta(l) = \Delta'(l)\,.$$

Space of types

- **Proposition.**

$$F(D) = (\mathbb{N} \rightharpoonup_{fin} D) \to_{mon} \mathit{UARel}(V)$$

  (monotone, non-expansive maps) defines a functor $F : \mathit{CBUlt}^{\mathrm{op}} \to \mathit{CBUlt}$.

- **Theorem.** There exists $\widehat{\mathcal{T}} \in$ CBUlt such that the isomorphism

$$\widehat{\mathcal{T}} \cong \tfrac{1}{2}((\mathbb{N} \rightharpoonup_{fin} \widehat{\mathcal{T}}) \to_{mon} \mathit{UARel}(V)) \tag{2}$$

  holds in CBUlt.

Space of Types, II

Define:

- Worlds: $\mathcal{W} = \mathbb{N} \rightharpoonup_{fin} \widehat{\mathcal{T}}$
- Types: $\mathcal{T} = \mathcal{W} \to_{mon} \mathit{UARel}(V)$
- Computations: $\mathcal{T}_T = \mathcal{W} \to_{mon} \mathit{UARel}(TV)$
- Continuations: $\mathcal{T}_K = \mathcal{W} \to_{mon} \mathit{UARel}(K)$
- States: $\mathcal{T}_S = \mathcal{W} \to \mathit{UARel}(S)$ (note: not monotone)

Semantics of Types

For every $\Xi \vdash \tau$, define the non-expansive $[\![\tau]\!]_\Xi : \mathcal{T}^\Xi \to \mathcal{T}$ by induction on τ:

$$
\begin{aligned}
[\![\alpha]\!]_\Xi\,\varphi &= \varphi(\alpha) \\
[\![\mathrm{int}]\!]_\Xi\,\varphi &= \lambda\Delta.\,\{\,(in_{\mathbb{Z}}\,k, in_{\mathbb{Z}}\,k) \mid k \in \mathbb{Z}\,\} \\
[\![1]\!]_\Xi\,\varphi &= \lambda\Delta.\,\{\,(in_1\,*, in_1\,*)\,\} \\
[\![\tau_1 \times \tau_2]\!]_\Xi\,\varphi &= [\![\tau_1]\!]_\Xi\,\varphi \times [\![\tau_2]\!]_\Xi\,\varphi \\
[\![0]\!]_\Xi\,\varphi &= \lambda\Delta.\,\emptyset \\
[\![\tau_1 + \tau_2]\!]_\Xi\,\varphi &= [\![\tau_1]\!]_\Xi\,\varphi + [\![\tau_2]\!]_\Xi\,\varphi \\
[\![\mathrm{ref}\,\tau]\!]_\Xi\,\varphi &= ref([\![\tau]\!]_\Xi\,\varphi) \\
[\![\forall\alpha.\tau]\!]_\Xi\,\varphi &= \lambda\Delta.\,\{\,(in_\forall\,c, in_\forall\,c') \mid \forall\nu \in \mathcal{T}.\,(c, c') \in \\
&= \qquad comp([\![\tau]\!]_{\Xi,\alpha}\,\varphi[\alpha \mapsto \nu])(\Delta)\,\} \\
[\![\mu\alpha.\tau]\!]_\Xi\,\varphi &= {\color{red}fix}\left(\lambda\nu.\,\lambda\Delta.\,\{\,(in_\mu\,v, in_\mu\,v') \mid (v, v') \in [\![\tau]\!]_{\Xi,\alpha}\,\varphi[\alpha \mapsto \nu]\,\Delta\,\}\right) \\
[\![\tau_1 \to \tau_2]\!]_\Xi\,\varphi &= ([\![\tau_1]\!]_\Xi\,\varphi) \to (comp([\![\tau_2]\!]_\Xi\,\varphi))
\end{aligned}
$$

Semantic Type Constructors

$$
\begin{aligned}
(\nu_1 \times \nu_2)(\Delta) = \{\, & (in_\times(v_1, v_2), in_\times(v_1', v_2')) \mid \\
& (v_1, v_1') \in \nu_1(\Delta) \wedge (v_2, v_2') \in \nu_2(\Delta) \,\}
\end{aligned}
$$

$$
\begin{aligned}
ref(\nu)(\Delta) = \{\, & (\lambda_l, \lambda_l) \mid l \in \mathrm{dom}(\Delta) \wedge \\
& \forall \Delta_1 \geq \Delta.\ App\,(\Delta(l))\,\Delta_1 = \nu(\Delta_1) \,\} \\
\cup \{\, & (\lambda_l^{n+1}, \lambda_l^{n+1}) \mid l \in \mathrm{dom}(\Delta) \wedge \\
& \forall \Delta_1 \geq \Delta.\ App\,(\Delta(l))\,\Delta_1 \stackrel{n}{=} \nu(\Delta_1) \,\}
\end{aligned}
$$

- Note the use of semantic locations to ensure non-expansiveness in *ref* case.
- Necessary: see Kristian's talk.
- Because of relational parametricity, we need to model *open* types; hence need to compare semantic types above, cannot simply use syntactic worlds and compare types syntactically.

Semantic Type Constructors, II

$$(\nu \to \xi)(\Delta) = \{\, (in_{\to}\, f, in_{\to}\, f') \mid \forall \Delta_1 \geq \Delta. \\ \forall (v, v') \in \nu(\Delta_1)\,.(f\, v, f'\, v') \in \xi(\Delta_1)\,\}$$

$$cont(\nu)(\Delta) = \{\, (k, k') \mid \forall \Delta_1 \geq \Delta. \forall (v, v') \in \nu(\Delta_1). \\ \forall (s, s') \in states(\Delta_1).\, (k\, v\, s, k'\, v'\, s') \in R_{Ans}\,\}$$

$$comp(\nu)(\Delta) = \{\, (c, c') \mid \forall \Delta_1 \geq \Delta. \forall (k, k') \in cont(\nu)(\Delta_1). \\ \forall (s, s') \in states(\Delta_1).\, (c\, k\, s, c'\, k'\, s') \in R_{Ans}\,\}$$

$$states(\Delta) = \{\, (s, s') \mid \mathrm{dom}(s) = \mathrm{dom}(s') = \mathrm{dom}(\Delta) \\ \wedge\ \forall l \in \mathrm{dom}(\Delta).\, (s(l), s'(l)) \in App\,(\Delta(l))\,(\Delta)\,\}$$

$$R_{Ans} = \{\, (\bot, \bot)\,\} \cup \{\, (\lfloor \iota_1\, k \rfloor, \lfloor \iota_1\, k \rfloor) \mid k \in \mathbb{Z}\,\}$$

Typed Semantics of Terms

- For $\Xi \vdash \Gamma$ and $\varphi \in \mathcal{T}^{\Xi}$, let $[\![\Gamma]\!]_{\Xi}\,\varphi$ be the binary relation on $V^{\mathrm{dom}(\Gamma)}$ defined by

$$[\![\Gamma]\!]_{\Xi}\,\varphi = \{\, (\rho, \rho') \mid \forall x \in \mathrm{dom}(\Gamma).\, (\rho(x), \rho'(x)) \in [\![\Gamma(x)]\!]_{\Xi}\,\varphi \,\}\,.$$

- Two typed terms $\Xi \mid \Gamma \vdash t : \tau$ and $\Xi \mid \Gamma \vdash t' : \tau$ of the same type are *semantically related*, written $\Xi \mid \Gamma \models t \sim t' : \tau$, if for all $\varphi \in \mathcal{T}^{\Xi}$, all $(\rho, \rho') \in [\![\Gamma]\!]_{\Xi}\,\varphi$, and all $\Delta \in \mathcal{W}$,

$$\left([\![t]\!]_{\mathrm{dom}(\Gamma)}\,\rho,\, [\![t']\!]_{\mathrm{dom}(\Gamma)}\,\rho' \right) \in \mathit{comp}([\![\tau]\!]_{\Xi}\,\varphi)(\Delta)\,.$$

Typed Semantics of Terms, II

- **Theorem.** Semantic relatedness is a congruence.
- **Corollary.** (FTLR) If $\Xi \mid \Gamma \vdash t : \tau$, then $\Xi \mid \Gamma \models t \sim t : \tau$.
- **Corollary.** If $\emptyset \mid \emptyset \vdash t : \tau$ is a closed term of type τ, then $[\![t]\!]_\emptyset \emptyset \neq \mathit{error}$.
- **Corollary.** If $\Xi \mid \Gamma \models t \sim t' : \tau$ then $t =_{\text{ctx}} t'$.

Overview of Applications and Extensions

- Four strands of work plus mention couple of other applications

Strand I: Nested Triples and (Anti)Frame Rules

Separation Logic with Nested Hoare Triples for reasoning about stored code (higher-order store) with higher-order frame rules [SBRY-CSL'09]

- Interpretation indexed over Kripke world describing "hidden invariants".
- But invariants are simply predicates (think of frame rule where any predicate can be used as an invariant), so get equation in CBUlt:

$$\begin{aligned} Pred &= \tfrac{1}{2}(W \to UAdm(H)) \\ W &\cong Pred \end{aligned}$$

Iso $\iota : Pred \to W$.

Strand I, ctd

- For higher-order frame rules: define non-expansive map $\circ : W \times W \to W$, s.t., for all $p, r, w \in W$,

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

- 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}.
- Semantics allowed to investigate soundness of various higher-order frame rules (tricky, some formulations are not sound, others are, see paper for examples)

Take-home: Use worlds form metric space to ensure well-definedness (via Banach) of recursive operation on worlds.

Strand I, ctd

Models of Pottier's Anti-frame rule [SYBRS - FOSSACS'10]

- Separation Logic for higher-order store with nested triples and formulation of Pottier's anti-frame rule for hiding invariants in direct style.
- Standard existence theorems for the world equation used did not apply directly, constructed solution by hand.
- (Jan's talk)

Strand II: Step-Indexed Models

(cf. Derek's talk)

- Step-indexed model of $F_{\mu,\mathrm{ref}}$ for reasoning about ctx. equiv., with more refined worlds, allows to prove more example programs equivalent. [ADR-POPL'09]
- Logics (LSLR, LADR) for step-indexed models, to avoid reasoning about steps when reasoning about examples (and a bit of the meta-theory) [DAB-LICS'09, DNRB-POPL'10]
- Worlds as transition systems describing how local state can evolve, studying the influence of different langauge features (first vs. higher-order state, with or without call/cc). Handles all known examples. [DNB-ICFP'10]

Take-home: (1) logics for steps for more high-level reasoning, (2) expressive worlds for more useful models.

Strand III: expressive ultrametric worlds

- Scaling up the denotational approach to recursively defined worlds a la those in LADR. [BST-TR] [Thamsborg dissertation]
  - involves using new form of relations that we call Bohr relations (chain-complete and downwards-closed in left-hand side), capturing ctx. *approximation* (instead of equiv.) [as in step-indexed models]
  - involves solving world equation in category of *preordered* ultrametric spaces
    - The Category-Theoretic Solution to Recursive Metric-Space Equations [BST-TCS'10]. Supporting theory. M-categories. (Jacob's talk.)

Take-home (1) Techniques scale well, (2) Resulting model allows for proofs of examples in the model at same level of abstraction as the LADR logic for step-indexed model.

Strand IV: Step-Indexed Kripke Models over Rec. Worlds

Recent work on showing how the approach applies to operational semantics via step-indexing [BRSSTY]

- arguably simpler than denotational approach, 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

To explain idea, let's consider a simple unary model $F_{\mu,\mathrm{ref}}$.

Uniform Predicates

Idea: replace domain V by the set of Val of operational values

- Uniform predicates:

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

- For $p \in UPred(Val)$ 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 $(UPred(Val), d)$ is a well-defined object in CBUlt.

Space of Types

Theorem

There exists $\widehat{\mathcal{T}} \in \mathrm{CBUlt}$ *such that*

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

is an iso in CBUlt.

Interpretation of Types, I

Define non-expansive map

$$[\![\Xi \vdash \tau]\!] \; : \; \mathcal{T}^{|\Xi|} \to \mathcal{T}$$

by induction on τ (only some cases):

$$[\![\Xi \vdash \tau]\!]_\eta : \mathcal{W} \to_{mon} UPred(Val)$$

$$[\![\Xi \vdash 1]\!]_\eta w = \{(k, ()) \mid k \in \mathbb{N}\}$$

$$[\![\Xi \vdash \mathrm{ref}\,\tau]\!]_\eta w = \{(k, l) \mid l \in \mathrm{dom}(w) \wedge$$
$$\forall w' \sqsupseteq w. i(w(l))(w') \stackrel{k}{=} [\![\Xi \vdash \tau]\!]_\eta w'\}$$

Interpretation of Types, II

$$\llbracket \Xi \vdash \tau \to \tau' \rrbracket_\eta w = \{(k, v) \mid \forall v' \in \mathit{Val}.\ \forall w' \sqsupseteq w.\ \forall i \leq k.$$
$$(i, v') \in \llbracket \Xi \vdash \tau \rrbracket_\eta w' \implies (i, v\, v') \in \mathcal{E}\llbracket \Xi \vdash \tau' \rrbracket_\eta w'\}$$

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

$$\mathcal{E}\llbracket \tau \rrbracket_\eta w = \{(k, t) \mid \forall i \leq k.\ \forall h, h'.\ \forall t'.$$
$$(h :_k w \ \wedge\ (t \mid h) \longmapsto^i (t' \mid h') \ \wedge\ (t \mid h') \text{ irreducible})$$
$$\implies (\exists w' \sqsupseteq w.\ \ h' :_{k-i} w' \ \wedge\ (k - i, v) \in \llbracket \tau \rrbracket_\eta w')\}$$

$$h :_k w \iff \forall i < k.\ \mathrm{dom}(h) = \mathrm{dom}(w) \ \wedge$$
$$\forall l \in \mathrm{dom}(w).\ \ (i, h(l)) \in w(l)(w)$$

Interpretation of Types, III

- Recursive Types:

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

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

Well-definedness

- Metric setup tells you what you have to show:
  - non-expansiveness of $[\![\Xi \vdash \tau]\!]$
  - non-expansiveness of $[\![\Xi \vdash \tau]\!]_\eta$
  - contractiveness of map for recursive types.
- Simple calculations.

Example lemma

Lemma

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

Proof.

TS: $\forall j < k.\ \mathrm{dom}(s) = \mathrm{dom}(w') \wedge \forall l \in \mathrm{dom}(w').\ (j, s(l)) \in w'(l)(w')$.
Sps. $k > 0$; then $n > 0$. Let $j < k$. By $w \stackrel{n}{=}_{\mathcal{W}} w'$, we get $\mathrm{dom}(w) = \mathrm{dom}(w') \wedge \forall l \in \mathrm{dom}(w).\ \forall w_0.\ w(l)(w_0) \stackrel{n-1}{=} w'(l)(w_0)$. Since $\mathrm{dom}(s) = \mathrm{dom}(w)$ by the assumption that $s :_k w$ (using $k > 0$), we get $\mathrm{dom}(s) = \mathrm{dom}(w')$. Moreover,

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

since $w(l)$ is non-expansive, and since $w \stackrel{n}{=}_{\mathcal{W}} w'$. 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. $\square$

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 [BBKV-TR]

Applications of Operational Approach

- We have given an alternative model of nested Hoare triples based directly on operational semantics with higher-order frame rules.
- Defined a model of Pottier's Capability Calculus, shown soundness of extension with higher-order frame and anti-frame rules.
- Capability Calculus setup: $\mathcal{W} \cong \frac{1}{2}\mathcal{W} \to \mathit{UPred}^{\uparrow}(\mathit{Heap})$.
- Model expresses that capabilities can be understood as separation logic assertions and is used to show soundness of the type system (new result).

Other recent applications

- A Metric Model of Nakano's calculus of Guarded Recursion [BSS - FICS'10]
  - kripke logical relation for adequacy proof defined using family of natural-number indexed relations.
- Separation logic for storable locks [BBS - ongoing].
- Model of type-and-effect system for higher-order store, extending work of Benton, Beringer, Hofmann, Kennedy [TB - ongoing] (again seems to involve a recursively defined operation on the set of worlds, though a quite different one).
- Krishnaswami-Benton: model of reactive programming using ultrametric spaces, see Neel's talk.

Thank You!