BI Hyperdoctrines and Higher-Order Separation Logic

Lars Birkedal

The IT University of Copenhagen

IT University of Copenhagen

Goals

- Intro to Higher-order Separation Logic
  - examples of why it is useful to use higher-order logic
- Intro to (BI) hyperdoctrines
- Observe some benefits of “abstract approach”
- Main reference: [B. Biering and L. Birkedal and N. Torp-Smith: BI-Hyperdoctrines, Higher-order Separation Logic, and Abstraction. *ACM Transactions on Programming Languages and Systems*, 29(5): 2007.
  (Journal version of ESOP’05 paper.)]

HOSL Example

From [Petersen et. al.: A Realizability Model of HTT, ESOP'08]: Imperative ADT:

$$
\begin{array}{ll}
\multicolumn{2}{l}{\mathsf{stacktype} =} \\
\multicolumn{2}{l}{\quad \prod \alpha : \mathsf{Type}. \sum \beta : \mathsf{Type}. \sum inv : \beta \times \alpha\, \mathsf{list} \to \mathsf{Prop}.} \\
/ * \, \mathtt{new} * / & (-).\{\mathbf{emp}\}s : \beta\{inv(s, [])\} \times \\
/ * \, \mathtt{push} * / & \prod s : \beta. \prod x : \alpha. \\
& \quad (l : \alpha\, \mathsf{list}).\{inv(s, l)\}u : 1\{inv(s, x :: l)\} \times \\
/ * \, \mathtt{pop} * / & \prod s : \beta. \\
& \quad (x : \alpha, l : \alpha\, \mathsf{list}). \\
& \qquad \{inv(s, x :: l)\}y : \alpha\{inv(s, l) \wedge y =_\alpha x\} \times \\
/ * \, \mathtt{del} * / & \prod s : \beta. \\
& \quad (l : \alpha\, \mathsf{list}).\{inv(s, l)\}u : 1\{\mathsf{emp}\}
\end{array}
$$

IT University
of Copenhagen

Overview

- Earlier work [Pym, O'Hearn, et. al.] has established correspondence between a part of separation logic and propositional BI
- We extend the correspondence to full separation logic and a simple version of *predicate* BI, and, moreover, to *higher-order*
  - define a class of sound and complete models: BI Hyperdoctrines
  - show that one cannot simply use toposes as models
  - argue that higher-order separation logic is useful for formalizations of separation logic and for data abstraction

IT University of Copenhagen

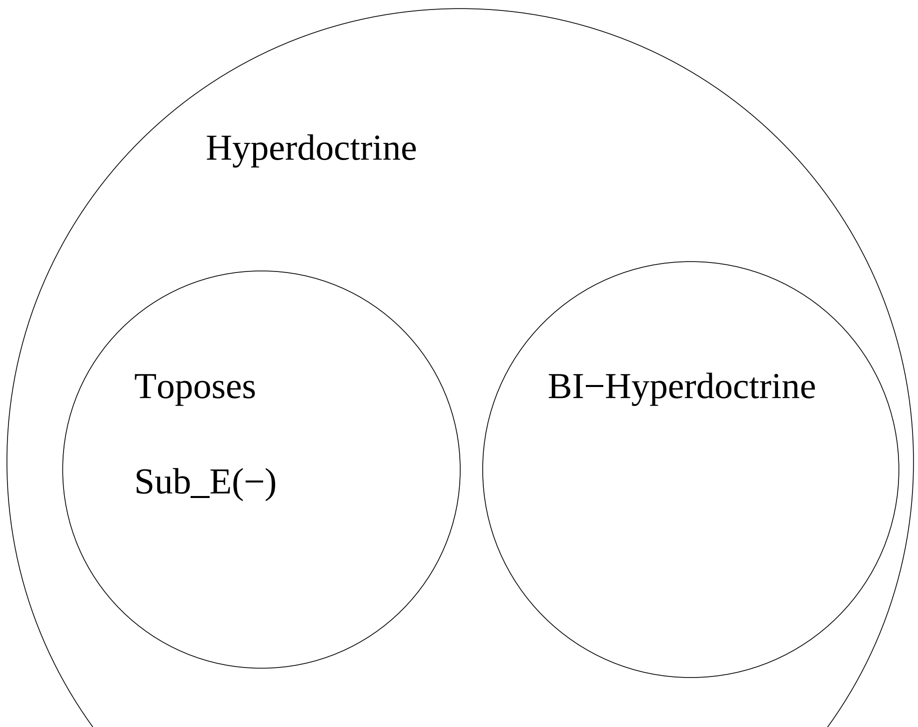
Why abstract approach?

Results applicable in many different situations, e.g.:

- Relational Parametricity and Separation Logic [LB-Yang: FOSSACS'07]
- Higher-order store [LB et. al.: ICALP'08]
- Hoare Type Theory [Petersen et. al.: ESOP'08]
- Idealized ML [Krishnaswami: thesis proposal, Krishnaswami et. al.: submitted]
- HOSL for Java [LB-Parkinson, ongoing]

BI Hyperdoctrines — Overview

- A hyperdoctrine is a categorical formalization of a model of predicate logic [Lawvere 1969]. Sound and complete for IHOL.
- Toposes also sound and complete for IHOL.
- BI Hyperdoctrines sound and complete for IHOL + BI



First-order Hyperdoctrines, I

Let $\mathcal{C}$ be a category with finite products. A *first-order hyperdoctrine* $\mathcal{P}$ over $\mathcal{C}$ is a contravariant functor $\mathcal{P} : \mathcal{C}^{op} \to \mathrm{Poset}$ s.t.:

- Each $\mathcal{P}(X)$ is a Heyting algebra.
- Each $\mathcal{P}(f) : \mathcal{P}(Y) \to \mathcal{P}(X)$ is a Heyting algebra homomorphism.
- There is an element $=_X$ of $\mathcal{P}(X \times X)$ satisfying that for all $A \in \mathcal{P}(X \times X)$,

$$\top \leq \mathcal{P}(\Delta_X)(A) \quad \text{iff} \quad =_X \leq A.$$

IT University of Copenhagen

First-order Hyperdoctrines, II

- For each product projection $\pi : \Gamma \times X \to \Gamma$ in $\mathcal{C}$, $\mathcal{P}(\pi) : \mathcal{P}(\Gamma) \to \mathcal{P}(\Gamma \times X)$ has both a left adjoint $(\exists X)_\Gamma$ and a right adjoint $(\forall X)_\Gamma$:

  $$A \leq \mathcal{P}(\pi)(A') \quad \text{if and only if} \quad (\exists X)_\Gamma(A) \leq A'$$

  $$\mathcal{P}(\pi)(A') \leq A \quad \text{if and only if} \quad A' \leq (\forall X)_\Gamma(A).$$

  Natural in Γ.

Interpretation in Hyperdoctrines

- Types and terms interpreted by objects and morphisms of $\mathcal{C}$
- Each formula ϕ with free variables in Γ is interpreted as a $\mathcal{P}$-predicate $[\![\phi]\!] \in \mathcal{P}([\![\Gamma]\!])$ by induction on the structure of ϕ using definining properties of hyperdoctrine.
- A formula ϕ with free variables in Γ is *satisfied* if $[\![\phi]\!]$ is $\top$ in $\mathcal{P}([\![\Gamma]\!])$.
- Sound and complete for intuitionistic predicate logic.
- A first-order hyperdoctrine is sound for *classical* predicate logic in case all the fibres $\mathcal{P}(X)$ are Boolean algebras and all the reindexing functions $\mathcal{P}(f)$ are Boolean algebra homomorphisms.

Hyperdoctrines

A (general) *hyperdoctrine* is a first-order hyperdoctrine with the following additional properties:

- $\mathcal{C}$ is cartesian closed; and
- there is $H \in \mathcal{C}$ and a natural bijection $\Theta_X : Obj(\mathcal{P}(X)) \simeq \mathcal{C}(X, H)$.

Cartesian closure interprets higher types.

Type of propositions is interpreted by H.

IT University of Copenhagen

BI Hyperdoctrines

- Recall: A *BI algebra* is a Heyting algebra, which has an additional symmetric monoidal closed structure $(\mathrm{I}, *, -\!*)$
- Define: A first-order hyperdoctrine $\mathcal{P}$ over $\mathcal{C}$ is a *first-order BI hyperdoctrine* in case
  - all the fibres $\mathcal{P}(X)$ are BI algebras, and
  - all the reindexing functions $\mathcal{P}(f)$ are BI algebra homomorphisms
- Likewise for general BI hyperdoctrines.

First-order Predicate BI, I

- Predicate logic with equality extended with I, $\phi * \psi$, $\phi \mathrel{-\!\!*} \psi$ satisfying the usual rules for BI (in any context Γ):

$$(\phi * \psi) * \theta \vdash_\Gamma \phi * (\psi * \theta) \qquad \phi * (\psi * \theta) \vdash_\Gamma (\phi * \psi) * \theta$$

$$\vdash_\Gamma \phi \leftrightarrow \phi * \mathrm{I} \qquad \phi * \psi \vdash_\Gamma \psi * \phi$$

$$\frac{\phi \vdash_\Gamma \psi \qquad \theta \vdash_\Gamma \omega}{\phi * \theta \vdash_\Gamma \psi * \omega} \qquad \frac{\phi * \psi \vdash_\Gamma \theta}{\phi \vdash_\Gamma \psi \mathrel{-\!\!*} \theta}$$

IT University of Copenhagen

First-Order Predicate BI, II

Notice

- No BI structure on contexts (in [Pym:2002] there is)
- In particular, weakening on the level of variables is always allowed

$$\frac{\phi \vdash_{\Gamma} \psi}{\phi \vdash_{\Gamma \cup \{x:X\}} \psi}$$

- Fine because simple and what we need for separation logic
- Can be interpreted in first-order BI hyperdoctrines
- **Theorem** The interpretation of first-order predicate BI is sound and complete.
- Also for classical predicate BI, of course

IT University of Copenhagen

Higher-order Predicate BI

- Higher-order predicate logic extended with BI as above.
- BI hyperdoctrines sound and complete class of models.

Example of BI hyperdoctrine

Let B be a complete BI algebra. Define $\mathbf{Set}$-indexed BI hyperdoctrine:

- $\mathcal{P}(X) = B^X$, functions from X to B, ordered pointwise
- For $f : X \to Y$, $\mathcal{P}(f) : B^Y \to B^X$ is comp. with f.
- $=_X (x, x')$ is $\top$ if $x = x'$, otherwise $\bot$.
- Quantification: for $A \in B^{\Gamma \times X}$

$$(\exists X)_\Gamma(A) \stackrel{def}{=} \lambda i \in \Gamma. \bigvee_{x \in X} A(i, x)$$
$$(\forall X)_\Gamma(A) \stackrel{def}{=} \lambda i \in \Gamma. \bigwedge_{x \in X} A(i, x)$$

in B^Γ.

IT University of Copenhagen

Toposes and BI Hyperdoctrines

- Earlier work showed how to use some toposes to model propostional BI ($\mathrm{Sub}_{\mathcal{E}}(1)$ is a BI-algebra, for certain $\mathcal{E}$)
- Toposes model (higher-order) predicate logic, since $\mathrm{Sub}_{\mathcal{E}}$ is a hyperdoctrine.
- But, surprise, we cannot interpret predicate BI in toposes:

Theorem Let $\mathcal{E}$ be a topos and suppose $\mathrm{Sub}_{\mathcal{E}} : \mathcal{E}^{op} \rightarrow \mathrm{Poset}$ is a BI hyperdoctrine. Then the BI structure on each lattice $\mathrm{Sub}_{\mathcal{E}}(X)$ is trivial, i.e., for all $\varphi, \psi \in \mathrm{Sub}_{\mathcal{E}}(X)$, $\varphi * \psi \leftrightarrow \varphi \wedge \psi$.

Higher-order Separation Logic

Next:

- Recall pointer model and interpretation of separation logic in pointer model
- Show how to view pointer model as a BI hyperdoctrine and that the standard interpretation therein coincides with standard interpretation of separation logic.
- Leads to obvious extension of separation logic to higher-order.
- Some implications thereof.

Pointer Model of Sep. Logic

- set $[\![\mathrm{Val}]\!]$ interpreting the type Val
- set $[\![\mathrm{Loc}]\!]$ of locations with $[\![\mathrm{Loc}]\!] \subseteq [\![\mathrm{Val}]\!]$
- set of heaps $H = [\![\mathrm{Loc}]\!] \rightharpoonup_{fin} [\![\mathrm{Val}]\!]$, ordered discretely, with partial binary operation $*$ defined by

$$h_1 * h_2 = \begin{cases} h_1 \cup h_2 & \text{if } h_1 \# h_2 \\ \text{undefined} & \text{otherwise,} \end{cases}$$

- set $\mathrm{Var} \rightharpoonup_{fin} [\![\mathrm{Val}]\!]$ of stacks

IT University of Copenhagen

Standard Int. of Formulas

Given by a forcing relation $s, h \Vdash \phi$, where $\mathrm{FV}(\phi) \subseteq \mathrm{dom}(s)$:

$s, h \Vdash t_1 = t_2$	iff	$[\![t_1]\!]s = [\![t_2]\!]s$
$s, h \Vdash t_1 \mapsto t_2$	iff	$\mathrm{dom}(h) = \{[\![t_1]\!]s\}$ and $h([\![t_1]\!]s) = [\![t_2]\!]s$
$s, h \Vdash \mathbf{emp}$	iff	$h = \emptyset$
$s, h \Vdash \phi * \psi$	iff	there exists $h_1, h_2 \in H.\, h_1 * h_2 = h$ and $s, h_1 \Vdash \phi$ and $s, h_2 \Vdash \psi$
$s, h \Vdash \phi \mathrel{-\!\!*} \psi$	iff	for all $h', h' \# h$ and $s, h' \Vdash \phi$ implies $s, h * h' \Vdash \psi$
$\ldots$		
$s, h \Vdash \forall x.\phi$	iff	for all $v \in [\![\mathrm{Val}]\!].s[x \mapsto v], h \Vdash \phi$

IT University of Copenhagen

Separation Logic as a BI Hyp.

- $\mathcal{P}(H)$ is a complete Boolean BI algebra, ordered by inclusion.
- Let S be the BI hyperdoctrine induced by the complete Boolean BI algebra
- **Theorem** $h \in [\![\phi]\!](v_1, \ldots, v_n)$ iff $[x_1 \mapsto v_1, \ldots, x_n \mapsto v_n], h \Vdash \phi$.
- (also works for other models of separation logic, e.g., intuitionistic and permissions models)

IT University of Copenhagen

Higher-order Sep. Logic

- The BI hyperdoctrine S also gives a model of *higher-order* separation logic, with $\mathcal{P}(H)$ the set of truth values.
- Now consider some applications of higher-order.

IT University of Copenhagen

Formalization of Sep. Logic, I

- Applications of sep. logic have used various extensions, with sets of lists, trees, relations, etc.
- Our point here is that they can be seen as trivial definitional extensions, since they are all definable in higher-order logic.
- Let $2 = \{\bot, \top\}$. There is a canonical map $\iota : 2 \to \mathcal{P}(H)$. Say $\phi : X \to \mathcal{P}(H)$ is *pure* if there is a map $\chi_\phi : X \to 2$ s.t.

$$\begin{tikzcd} X \arrow[rr, "\phi"] \arrow[dr, "\chi_\phi"'] & & \mathcal{P}(H) \\ & 2 \arrow[ur, "\iota"'] & \end{tikzcd}$$

commutes.

IT University of Copenhagen

Formalization of Sep. Logic, I

- The sub-logic of pure predicates is simply the standard classical higher-order logic of $\mathbf{Set}$.
- Allows to use classical higher-order logic for defining lists, trees, etc.
- In particular, recursive definitions of predicates, earlier done at the meta-level, can now be done inside the higher-order logic itself.

IT University of Copenhagen

Logical Characterizations...

of classes of formulas:

- Traditional definition of a *precise*: q is precise iff, for s, h, there is at most one subheap h_0 of h such that $s, h_0 \Vdash q$.
- **Prop.** q is precise iff

$$\forall p_1, p_2 : \mathsf{prop} \,.\, (p_1 * q) \wedge (p_2 * q) \rightarrow (p_1 \wedge p_2) * q$$

  is valid in the BI hyperdoctrine S.
- Thus: can make *logical* proofs about precise formulas.

Characterizations, II

- Traditional: q is *monotone* iff whenever $h \in [\![q]\!]$ then also $h' \in [\![q]\!]$, for all extensions $h' \supseteq h$.
- **Prop.** q is *monotone* iff

$$\forall p : \mathsf{prop} \,.\, p * q \longrightarrow q$$

  is valid in the BI hyperdoctrine S.
- **Prop.** q is *pure* iff

$$\forall p_1, p_2 : \mathsf{prop} \,.\, (q \wedge p_1) * p_2 \leftrightarrow q \wedge (p_1 * p_2)$$

  is valid in the BI hyperdoctrine S.

IT University of Copenhagen

Applications in Program Proving

- one can use existential quantification over hidden (abstract) resource invariants to reason about programs using abstract data types, c.f. the stack example from the beginning.
- see also examples in Ynot paper [Nanevski et. al.] and in design patterns paper [Krishnaswami et. al.]
- polymorphic types using universal quantification (generic reasoning)

Ongoing / Future Work

- Systematic investigation of relation between assertion and specification logic.
- HOSL for Java / C#.
- Formalizations / Automation
  - finding loop / data structure invariants
  - theorem proving for higher-order logic
  - experiments so far:
    - HOSL in Isabelle/HOLCF [Varming-LB: MFPS'08] (Cheney's g.c. verified)
    - Ynot in Coq [Nanevski et. al.: ICFP'08] (finite map data structures + design patterns verified)

IT University of Copenhagen

Strengthening

Theorem Let $\mathcal{P}$ be an indexed preorder, fibres all BI algebras, preserved under reindexing, with *full* subset types. Then the BI structure on each lattice $\mathcal{P}(X)$ is trivial, i.e., for all $\varphi, \psi \in \mathcal{P}(X)$, $\varphi * \psi \leftrightarrow \varphi \wedge \psi$.

- The BI hyperdoctrine for separation logic has subset types, but not *full* subset types.
- Full subset types:

$$\frac{y : \{x : X \mid \varphi\} \mid \theta \vdash \psi}{x : X \mid \theta, \varphi \vdash \psi}$$

IT University of Copenhagen