Iris: Higher-Order Concurrent Separation Logic

Lecture 2: Basic Logic of Resources

Lars Birkedal

Aarhus University, Denmark

November 10, 2017

Overview

Earlier:

- Operational Semantics of $\lambda_{\mathrm{ref,conc}}$
  - e, $(h, e) \rightsquigarrow (h, e')$, and $(h, \mathcal{E}) \rightarrow (h', \mathcal{E}')$

Today:

- Basic Logic of Resources
  - $l \hookrightarrow v$, $P * Q$, $P \mathrel{-\!\!*} Q$, $\Gamma \mid P \vdash Q$

Iris

- A higher-order separation logic over a simple type theory with new base types and base terms defined in signature $\mathcal{S}$.
- Terms and types are as in simply typed lambda calculus, types include a type Prop of propositions.
- Do not confuse the lambda calculus of Iris with the programming language lambda abstractions in $\lambda_{\mathrm{ref,conc}}$
  - The lambda calculus of Iris is an equational theory of functions, no operational semantics (think standard mathematical functions)
  - In $\lambda_{\mathrm{ref,conc}}$ one can define functions whose behaviour is defined by the operational semantics of $\lambda_{\mathrm{ref,conc}}$

Syntax: Types

$$\tau ::= T \mid \mathbb{Z} \mid \mathit{Val} \mid \mathit{Exp} \mid \mathsf{Prop} \mid 1 \mid \tau + \tau \mid \tau \times \tau \mid \tau \to \tau$$

where

- T stands for additional base types which we will add later
- Val and Exp are types of values and expressions in $\lambda_{\mathrm{ref,conc}}$
- Prop is the type of Iris propositions.

Syntax: Terms

$$
\begin{aligned}
t, P ::= {} & x \mid n \mid v \mid e \mid F(t_1, \ldots, t_n) \mid \\
& () \mid (t, t) \mid \pi_i\ t \mid \lambda x : \tau.\, t \mid t(t) \mid \mathsf{inl}\, t \mid \mathsf{inr}\, t \mid \mathsf{case}(t, x.t, y.t) \mid \\
& \mathsf{False} \mid \mathsf{True} \mid t =_\tau t \mid P \Rightarrow P \mid P \wedge P \mid P \vee P \mid P * P \mid P \mathrel{-\!\!*} P \mid \\
& \exists x : \tau.\, P \mid \forall x : \tau.\, P \mid \\
& \square\, P \mid \triangleright P \mid \\
& \{P\}\, t\, \{P\} \mid \\
& t \hookrightarrow t
\end{aligned}
$$

where

- x are variables
- n are integers
- v and e range over values of the language, *i.e.*, they are primitive terms of types *Val* and *Exp*
- F ranges over the function symbols in the signature $\mathcal{S}$.

Well-typed Terms ($\Gamma \vdash_{\mathcal{S}} t : \tau$)

- Typing relation

$$\Gamma \vdash_{\mathcal{S}} t : \tau$$

  defined inductively by inference rules.

- Here $\Gamma = x_1 : \tau_1, x_2 : \tau_2, \ldots, x_n : \tau_n$ is a context, assigning types to variables
- Selected rules:

$$\frac{\Gamma, x : \tau \vdash t : \tau'}{\Gamma \vdash \lambda x.\, t : \tau \to \tau'} \qquad \frac{\Gamma \vdash t : \tau \to \tau' \qquad u : \tau}{\Gamma \vdash t(u) : \tau'} \qquad \frac{}{\Gamma \vdash \mathrm{True} : \mathrm{Prop}}$$

$$\frac{\Gamma \vdash t : \tau \qquad \Gamma \vdash u : \tau}{\Gamma \vdash t =_\tau u : \mathrm{Prop}} \qquad \frac{\Gamma \vdash P : \mathrm{Prop} \qquad \Gamma \vdash Q : \mathrm{Prop}}{\Gamma \vdash P \Rightarrow Q : \mathrm{Prop}} \qquad \frac{\Gamma, x : \tau \vdash P : \mathrm{Prop}}{\Gamma \vdash \forall x : \tau.\, P : \mathrm{Prop}}$$

Entailment ($\Gamma \mid P \vdash Q$)

- Entailment relation
$$\Gamma \mid P \vdash Q$$
for $\Gamma \vdash P : \mathrm{Prop}$ and $\Gamma \vdash Q : \mathrm{Prop}$.
- The relation is defined by induction, using standard rules from intuitionistic higher-order logic extended with new rules for the new connectives.
- We only have one proposition P on the left of the turnstile.
  - You may be used to seeing a list of assumptions separated by commas
  - Instead we extend the context by using $\wedge$
  - This choice makes it easy to extend the context also with $*$.
- To understand the entailment rules for the new connectives, we need to have an intuitive understanding of the semantics of the logical connectives.
- Note: in this course, we do not present a formal semantics of the logic and formally prove the logic sound (for that, see "Iris from the Ground Up: A Modular Foundation for Higher-Order Concurrent Separation Logic" on iris-project.org).

Intuition for Iris Propositions

- Intuition: A proposition P describes a set of resources.
- Write $\mathcal{R}$ for the set of resources, and write r_1, r_2, etc., for elements in $\mathcal{R}$.
- We assume that
  - there is an empty resource
  - there is a way to compose (or combine) resources r_1 and r_2, denoted $r_1 \cdot r_2$
  - the composition is defined for resources that are suitably disjoint, denoted $r_1 \# r_2$.
- Later on we will formalize such notions of resources using certain commutative monoids. For now, it suffices to think about the example of $\mathcal{R} = \mathit{Heap}$.

Intuition for Iris Propositions

- Canonical example: $\mathcal{R} = \mathit{Heap}$, the set of heaps from $\lambda_{\mathrm{ref,conc}}$.
- Recall: $\mathit{Heap} = \mathit{Loc} \xrightharpoonup{\mathrm{fin}} \mathit{Val}$, the set of partial functions from locations to values
- The empty resource is the empty heap, denoted $[]$.
- Two heaps h_1 and h_2 are disjoint, denoted $h_1 \# h_2$, if their domains do not overlap (*i.e.*, $\mathrm{dom}(h_1) \cap \mathrm{dom}(h_2) = \emptyset$).
- The composition of two disjoint heaps h_1 and h_2 is the heap $h = h_1 \cdot h_2$ defined by

$$h(x) = \begin{cases} h_1(x) & \text{if } x \in \mathrm{dom}(h_1) \\ h_2(x) & \text{if } x \in \mathrm{dom}(h_2) \end{cases}$$

Intuition for Iris Propositions

- We said: "A proposition P *describes* a set of resources."
- Also say: "P *is* a set of resources."
- Also say: "P *denotes* a set of resources."
- $P \in P(\mathcal{R})$.
- When r is a resource described by P, we also say that r *satisfies* P, or that r *is in* P.
- The intuition for $P \vdash Q$ is then that all resources in P are also in Q (*i.e.*, $\forall r \in \mathcal{R}.\, r \in P \Rightarrow r \in Q$).

Describing Resources in the Logic

- Primitive: the points-to predicate $x \hookrightarrow v$.
- It is a formula, *i.e.*, a term of type Prop

$$\frac{\Gamma \vdash \ell : \mathit{Val} \qquad \Gamma \vdash v : \mathit{Val}}{\Gamma \vdash \ell \hookrightarrow v : \mathrm{Prop}}$$

- It describes the set of heap fragments that map location x to value v

$$x \hookrightarrow v = \{h \mid x \in \mathrm{dom}(h) \wedge h(x) = v\}$$

- Ownership reading: if I assert $\ell \hookrightarrow v$, then I express that I have the ownership of ℓ and hence I may modify what ℓ pointsto, without invalidating invariants of other parts of the program.

Intuition for $*$ and $-\!\!*$

- $P * Q = \{r \mid \exists r_1, r_2. r = r_1 \cdot r_2 \wedge r_1 \in P \wedge r_2 \in Q\}$
- For example, $x \hookrightarrow u * y \hookrightarrow v$ describes the set of heaps with two *disjoint* locations x and y, the first stores u and the second v.
- Note: $x \hookrightarrow v * x \hookrightarrow u \vdash \text{False}$.
- $P -\!\!* Q = \{r \mid \forall r_1. r_1 \# r \wedge r_1 \in P \Rightarrow r \cdot r_1 \in Q\}$
- For example, the proposition

  $$x \hookrightarrow u -\!\!* (x \hookrightarrow u * y \hookrightarrow v)$$

  describes those heap fragments that map y to v, because when we combine it with a heap fragment mapping x to u, then we get a heap fragment mapping x to u and y to v.

Weakening Rule

Weakening rule:

$$\frac{}{P_1 * P_2 \vdash P_1}\ \ast\text{-WEAK}$$

- Thus Iris is an affine separation logic.
- Example:

$$x \hookrightarrow u * y \hookrightarrow v \vdash x \hookrightarrow u$$

  - Suppose $h \in (x \hookrightarrow u * y \hookrightarrow v)$.
  - Then $h(x) = u$ and $h(y) = v$.
  - Therefore $h \in (x \hookrightarrow u)$.
  - Generally, if $h \in P$ and $h' \geq h$, then also $h' \in P$.

Weakening Rule

In a bit more detail:

- Intuitively, the fact that this rule is sound means that propositions are interpreted by upwards closed sets of resources:
  - We say that $r_1 \geq r_2$ iff $r_1 = r_2 \cdot r_3$, for some r_3.
  - Suppose $r_1 \in P_1$ and that $r \geq r_1$. Then there is r_2 such that $r = r_1 \cdot r_2$.
  - Let P_2 be $\{r_2\}$.
  - Then $r_1 \cdot r_2 \in P_1 * P_2$.
  - By the weakening rule, we then also have that $r = r_1 \cdot r_2 \in P_1$.
  - Hence P_1 is upwards closed.
- The above is not a formal proof, hence the stress on “intuitively”.

Associativity and Commutativity of $*$

Basic structural rules:

$$\frac{}{P_1 * (P_2 * P_3) \dashv\vdash (P_1 * P_2) * P_3}\ {*}\text{-}\textsc{Assoc} \qquad \frac{}{P_1 * P_2 \dashv\vdash P_2 * P_1}\ {*}\text{-}\textsc{Comm}$$

Sound because composition of resources, $\cdot$, is commutative and associative.

Separating Conjunction Introduction

$$\frac{P_1 \vdash Q_1 \qquad P_2 \vdash Q_2}{P_1 * P_2 \vdash Q_1 * Q_2}\ {*}\mathrm{I}$$

- To show a separating conjuction $Q_1 * Q_2$, we need to split the assumption and decide which resources to use to prove Q_1 and which ones to use to prove Q_2.
- Example: $P \vdash P * P$ is not provable in general

Magic wand introduction and elimination

$$\frac{R * P \vdash Q}{R \vdash P \mathrel{-\!\!*} Q}\;{\mathrel{-\!\!*}}\mathrm{I} \qquad\qquad \frac{R_1 \vdash P \mathrel{-\!\!*} Q \qquad R_2 \vdash P}{R_1 * R_2 \vdash Q}\;{\mathrel{-\!\!*}}\mathrm{E}$$

- Introduction rule intuitively sound because
  - Suppose $r \in R$. TS $r \in P \mathrel{-\!\!*} Q$.
  - Thus let $r_1 \in P$ and suppose $r_1 \# r$. TS $r \cdot r_1 \in Q$.
  - We have $r \cdot r_1 \in R * P$.
  - Hence, by antecedent, $r \cdot r_1 \in Q$, as required.
- Elimination rule intuitively sound because
  - ...