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 $\lambda_{\text{ref,conc}}$
  - $e, (h, e) \leadsto (h, e')$, and $(h, E) \rightarrow (h', E')$

Today:

- Basic Logic of Resources
  - $l \leftrightarrow v$, $P \ast Q$, $P \ast\ast Q$, $\Gamma \mid P \vdash Q$
Iris

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

\[ \tau ::= T \mid \mathbb{Z} \mid Val \mid Exp \mid \text{Prop} \mid 1 \mid \tau + \tau \mid \tau \times \tau \mid \tau \to \tau \]

where

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

\[ t, P := x \mid n \mid v \mid e \mid F(t_1, \ldots, t_n) \mid (t, t) \mid \pi_i t \mid \lambda x : \tau. t \mid t(t) \mid \text{inl} \ t \mid \text{inr} \ t \mid \text{case} (t, x.t, y.t) \mid \text{False} \mid \text{True} \mid t =_\tau t \mid P \Rightarrow P \mid P \land P \mid P \lor P \mid P \ast P \mid P \rightarrow P \mid \exists x : \tau. P \mid \forall x : \tau. P \mid \Box P \mid \triangleright P \mid \{P\} t \{P\} \mid t \hookrightarrow t \]

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 \( S. \)
Well-typed Terms (\( \Gamma \vdash_S t : \tau \))

- Typing relation

\[ \Gamma \vdash_S t : \tau \]

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:

\[
\begin{align*}
  \Gamma, x : \tau & \vdash t : \tau' \\
  \Gamma & \vdash \lambda x. t : \tau \rightarrow \tau' \\
  \Gamma, x : \tau & \vdash u : \tau \\
  \Gamma & \vdash t(u) : \tau' \\
  \Gamma & \vdash \text{True} : \text{Prop}
\end{align*}
\]

\[
\begin{align*}
  \Gamma & \vdash t : \tau \\
  \Gamma & \vdash u : \tau \\
  \Gamma & \vdash t =_\tau u : \text{Prop}
\end{align*}
\]

\[
\begin{align*}
  \Gamma & \vdash P : \text{Prop} \\
  \Gamma & \vdash Q : \text{Prop} \\
  \Gamma & \vdash P \Rightarrow Q : \text{Prop}
\end{align*}
\]

\[
\begin{align*}
  \Gamma & \vdash \forall x : \tau. P : \text{Prop}
\end{align*}
\]
Entailment ($\Gamma | P \vdash Q$)

- Entailment relation

\[ \Gamma | P \vdash Q \]

for $\Gamma \vdash P : \text{Prop}$ and $\Gamma \vdash Q : \text{Prop}$.

- 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 $\land$
  - This choice makes it easy to extend the context also with $\ast$.

- 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 $\mathcal{R} = \text{Heap}$. 
Intuition for Iris Propositions

- Canonical example: $R = \text{Heap}$, the set of heaps from $\lambda_{\text{ref,conc}}$.
- Recall: $\text{Heap} = \text{Loc} \xrightarrow{\text{fin}} \text{Val}$, 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., $\text{dom}(h_1) \cap \text{dom}(h_2) = \emptyset$).
- The composition of two disjoint heaps $h_1$ and $h_2$ is the heap $h = h_1 \cdot h_2$ defined by

$$h(x) = \begin{cases} h_1(x) & \text{if } x \in \text{dom}(h_1) \\ h_2(x) & \text{if } x \in \text{dom}(h_2) \end{cases}$$
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.”
- $P \in P(R)$.
- 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., $\forall r \in R. r \in P \Rightarrow r \in Q$).
Describing Resources in the Logic

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

\[
\Gamma \vdash \ell : \text{Val} \quad \Gamma \vdash v : \text{Val} \\
\Gamma \vdash \ell \leftrightarrow v : \text{Prop}
\]

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

\[
x \leftrightarrow v = \{h \mid x \in \text{dom}(h) \land h(x) = v\}
\]

- **Ownership reading:** if I assert $\ell \leftrightarrow v$, then I express that I have the ownership of $\ell$ and hence I may modify what $\ell$ points to, without invalidating invariants of other parts of the program.
Intuition for $\ast$ and $\rightarrow$

$P \ast Q = \{ r \mid \exists r_1, r_2. r = r_1 \cdot r_2 \land r_1 \in P \land r_2 \in Q \}$

For example, $x \hookrightarrow u \ast y \hookrightarrow v$ describes the set of heaps with two disjoint locations $x$ and $y$, the first stores $u$ and the second $v$.

Note: $x \hookrightarrow v \ast x \hookrightarrow u \vdash \text{False}$.

$P \rightarrow Q = \{ r \mid \forall r_1. r \# r \land r_1 \in P \Rightarrow r \cdot r_1 \in Q \}$

For example, the proposition

$$x \hookrightarrow u \rightarrow (x \hookrightarrow u \ast y \hookrightarrow v)$$

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:

\[ \text{*-WEAK} \]

\[ P_1 \ast P_2 \vdash P_1 \]

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

\[ x \leftrightarrow u \ast y \leftrightarrow v \vdash x \leftrightarrow u \]

- Suppose \( h \in (x \leftrightarrow u \ast y \leftrightarrow v) \).
- Then \( h(x) = u \) and \( h(y) = v \).
- Therefore \( h \in (x \leftrightarrow u) \).
- 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 \( r_1 \cdot r_2 \in P_1 \cdot P_2 \).
  - 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 $\ast$

Basic structural rules:

\[
\begin{align*}
&\text{*-ASSOC} \\
&P_1 \ast (P_2 \ast P_3) \iff (P_1 \ast P_2) \ast P_3
\end{align*}
\]

\[
\begin{align*}
&\text{*-COMM} \\
&P_1 \ast P_2 \iff P_2 \ast P_1
\end{align*}
\]

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

\[ *I \]
\[ P_1 \vdash Q_1 \quad P_2 \vdash Q_2 \]
\[ P_1 \ast P_2 \vdash Q_1 \ast Q_2 \]

- To show a separating conjunction \( Q_1 \ast Q_2 \), 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: \( P \vdash P \ast P \) is **not** provable in general
Introduction rule intuitively sound because
- Suppose $r \in R$. TS $r \in P \notstar Q$.
- Thus let $r_1 \in P$ and suppose $r_1 \not\# r$. TS $r \cdot r_1 \in Q$.
- We have $r \cdot r_1 \in R \star P$.
- Hence, by antecedent, $r \cdot r_1 \in Q$, as required.

Elimination rule intuitively sound because
- ...