

State of Iris

Robbert Krebbers, Ralf Jung

Iris Workshop, May 24th 2023

Since the last Iris workshop...

- Iris 4.0: Later credits!

Since the last Iris workshop...

- Iris 4.0: Later credits!
- A bunch of small quality of life improvements
- 516 Iris commits by 22 authors
- 246 std++ commits by 8 authors

Since the last Iris workshop...

- Adam, Armaël Guéneau, Arthur Azevedo de Amorim, Dorian Lesbre, Gregory Malecha,
- Herman Bergwerf, Ike Mulder, Jonas Kastberg, Lennard Gäher, Marijn van Wezel, Michael Sammler, Niklas Mück, Paolo G. Giarrusso,
- Pierre Roux, Ralf Jung, Robbert Krebbers,
- Rodolphe Lepigre, Silvus, Simcha van Collem, Simon Friis Vindum, Simon Spies, Tej Chajed, Yixuan Chen, Yusuke Matsushita.

Thank you!

Since the last Iris workshop...

- Iris 4.0: Later credits!
- A bunch of small quality of life improvements
- 516 Iris commits by 22 authors
- 246 std++ commits by 8 authors

We had no release since August 2022.
Does anyone actually care about releases?

Recent change: entailment notation

These used to be equivalent:

Lemma wand : $P \multimap Q$.

Lemma entails : $P \vdash Q$.

Recent change: entailment notation

These used to be equivalent:

```
Lemma wand : P  $\multimap$  Q.
```

```
Lemma entails : P  $\vdash$  Q.
```

This recently changed. Now these are equivalent:

```
Lemma wand : P  $\multimap$  Q.
```

```
Lemma emp_valid :  $\vdash$  P  $\multimap$  Q.
```

Lemmas used for rewriting, and lemmas involving more than one entailment, should be written with \vdash .

Everything else should use \multimap .

Breaking change announcements

We have a GitLab group sending notifications about breaking changes.

Leave Ralf a message if you want to be added!

Coq version support

“Coq deprecation policy is to enable easy support of 2 consecutive versions”

Iris aims to support 2 consecutive Coq versions

std++ aims to support 3 consecutive Coq versions (as long as that is possible)

Coq version support

“Coq deprecation policy is to enable easy support of 2 consecutive versions”

Iris aims to support 2 consecutive Coq versions

std++ aims to support 3 consecutive Coq versions (as long as that is possible)

Does anyone care about support for more than 2 Coq versions?

Iris LaTeX files

The Iris repo contains LaTeX macro collections
for Iris notation.

Please check it out and submit MRs if you have
improvements!

Upcoming new
features

Improved iCombine

Problem: Rules for combining ghost state are difficult to find and tedious to apply

Solution: Automatically combine ghost state and derive validity information

Now: Better `iCombine` with `gives` clause (by Ike)

Soon: Generic mechanism for `own`

Transfinite step-indexing

Do you need

\triangleright^ω ? Liveness? Angelic non-determinism?

Transfinite step-indexing

Do you need

▷^ω? Liveness? Angelic non-determinism?

Then you want **transfinite step-indexing!**

We are slowly working on landing the required infrastructure in Iris.

Pure step-indexed propositions

Problem: Results about internal equality \equiv and internal camera validity \checkmark are tied to `iProp`

Solution: New `Sbi` interface with:

```
<si_pure> : siProp → PROP  
<si_emp_valid> : PROP → siProp
```

Soon: Uniform definition of \equiv , \checkmark , \blacksquare for any `Sbi`

Future: More use of `siProp`, e.g., for view camera

A world of integers

Problem: All program arithmetic is in `Z`, but `list` operations need `nat`

The `Z.of_nat : nat → Z` coercion is confusing

A world of integers

Problem: All program arithmetic is in `Z`, but `list` operations need `nat`

The `Z.of_nat : nat → Z` coercion is confusing

Possible solution: `Z`-based `list` APIs
⇒ lots of API duplication

Only duplicate APIs when the need comes up?

Your lemma here?

Quick poll: who in the room has lemmas that should be upstream, but are not?

Your lemma here?

Quick poll: who in the room has lemmas that should be upstream, but are not?

How can we improve the upstreaming process?

Q & A