# Deadlock-Free Separation Logic

## Linearity Yields Progress for Dependent Higher-Order Message Passing

Jules Jacobs Jonas Kastberg Hinrichsen Robbert Krebbers

**Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

JULES JACOBS, Radboud University Nijmegen, The Netherlands
JONAS KASTBERG HINRICHSEN, Aarhus University, Denmark
ROBBERT KREBBERS, Radboud University Nijmegen, The Netherlands

We introduce a linear concurrent separation logic, called **LinearActris**, designed to guarantee deadlock and leak freedom for message-passing concurrency. LinearActris combines the strengths of session types and concurrent separation logic, allowing for the verification of challenging higher-order programs with mutable state through dependent protocols. The key challenge is to prove the adequacy theorem of LinearActris, which says that the logic indeed gives deadlock and leak freedom "for free" from linearity. We prove this theorem by defining a step-indexed model of separation logic, based on *connectivity graphs*. To demonstrate the expressive power of LinearActris, we prove soundness of a higher-order (GV-style) session type system using the technique of logical relations. All our results and examples have been mechanized in Coq.

CCS Concepts: • **Theory of computation → Separation logic**; *Program verification*; Programming logic.

Additional Key Words and Phrases: Message passing, deadlocks, session types, separation logic, Iris, Coq

**ACM Reference Format:**
Jules Jacobs, Jonas Kastberg Hinrichsen, and Robbert Krebbers. 2024. Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing. *Proc. ACM Program. Lang.* 8, POPL, Article 47 (January 2024), 33 pages. https://doi.org/10.1145/3632889

POPL'24

# Deadlock-Free Separation Logic

## Linearity Yields Progress for Dependent Higher-Order Message Passing

Jules Jacobs Jonas Kastberg Hinrichsen Robbert Krebbers

**Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

JULES JACOBS, Radboud University Nijmegen, The Netherlands
JONAS KASTBERG HINRICHSEN, Aarhus University, Denmark
ROBBERT KREBBERS, Radboud University Nijmegen, The Netherlands

We introduce a linear concurrent separation logic, called **LinearActris**, designed to guarantee deadlock and leak freedom for message-passing concurrency. LinearActris combines the strengths of session types and concurrent separation logic, allowing for the verification of challenging higher-order programs with mutable state through dependent protocols. The key challenge is to prove the adequacy theorem of LinearActris, which says that the logic indeed gives deadlock and leak freedom "for free" from linearity. We prove this theorem by defining a step-indexed model of separation logic, based on *connectivity graphs*. To demonstrate the expressive power of LinearActris, we prove soundness of a higher-order (GV-style) session type system using the technique of logical relations. All our results and examples have been mechanized in Coq.

CCS Concepts: • **Theory of computation** → **Separation logic**; *Program verification*; Programming logic.

Additional Key Words and Phrases: Message passing, deadlocks, session types, separation logic, Iris, Coq

**ACM Reference Format:**
Jules Jacobs, Jonas Kastberg Hinrichsen, and Robbert Krebbers. 2024. Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing. *Proc. ACM Program. Lang.* 8, POPL, Article 47 (January 2024), 33 pages. https://doi.org/10.1145/3632889

**POPL'24**

# The Iris Masterplan

Jules Jacobs, 2023-12-11

# History

# History

$\{P\}\ e\ \{Q\}$

# History

$$\{P\}\ e\ \{Q\}$$

If P holds, and we run e, then Q holds

# History

$$\{P\}\ e\ \{Q\}$$

~~If P holds, and we run e, then Q holds~~

# History

$$\{P\}\ e\ \{Q\}$$

~~If P holds, and we run e, then Q holds~~

If P holds, then e is *safe*, and afterwards Q holds

# History

$\{P\}\ e\ \{Q\}$

~~If P holds, and we run e, then Q holds~~

If P holds, then e is *safe*, and afterwards Q holds

| **Verification of programs with** | **Using** |
| --- | --- |
| *mutable local variables* | *Hoare logic (Hoare)* |
| + *pointers & data structures* | *Separation logic (Reynolds)* |
| + *shared-memory concurrency* | *Concurrent separation logic (O'Hearn)* |

# What is Iris?

## *Iris from the ground up*

### *A modular foundation for higher-order concurrent separation logic*


RALF JUNG
*MPI-SWS, Germany*
(*e-mail:* jung@mpi-sws.org)

ROBBERT KREBBERS
*Delft University of Technology, The Netherlands*
(*e-mail:* mail@robbertkrebbers.nl)

JACQUES-HENRI JOURDAN
*MPI-SWS, Germany*
(*e-mail:* jjourdan@mpi-sws.org)

ALEŠ BIZJAK
*Aarhus University, Denmark*
(*e-mail:* abizjak@cs.au.dk)

LARS BIRKEDAL
*Aarhus University, Denmark*
(*e-mail:* birkedal@cs.au.dk)

DEREK DREYER
*MPI-SWS, Germany*
(*e-mail:* dreyer@mpi-sws.org)



**Abstract**

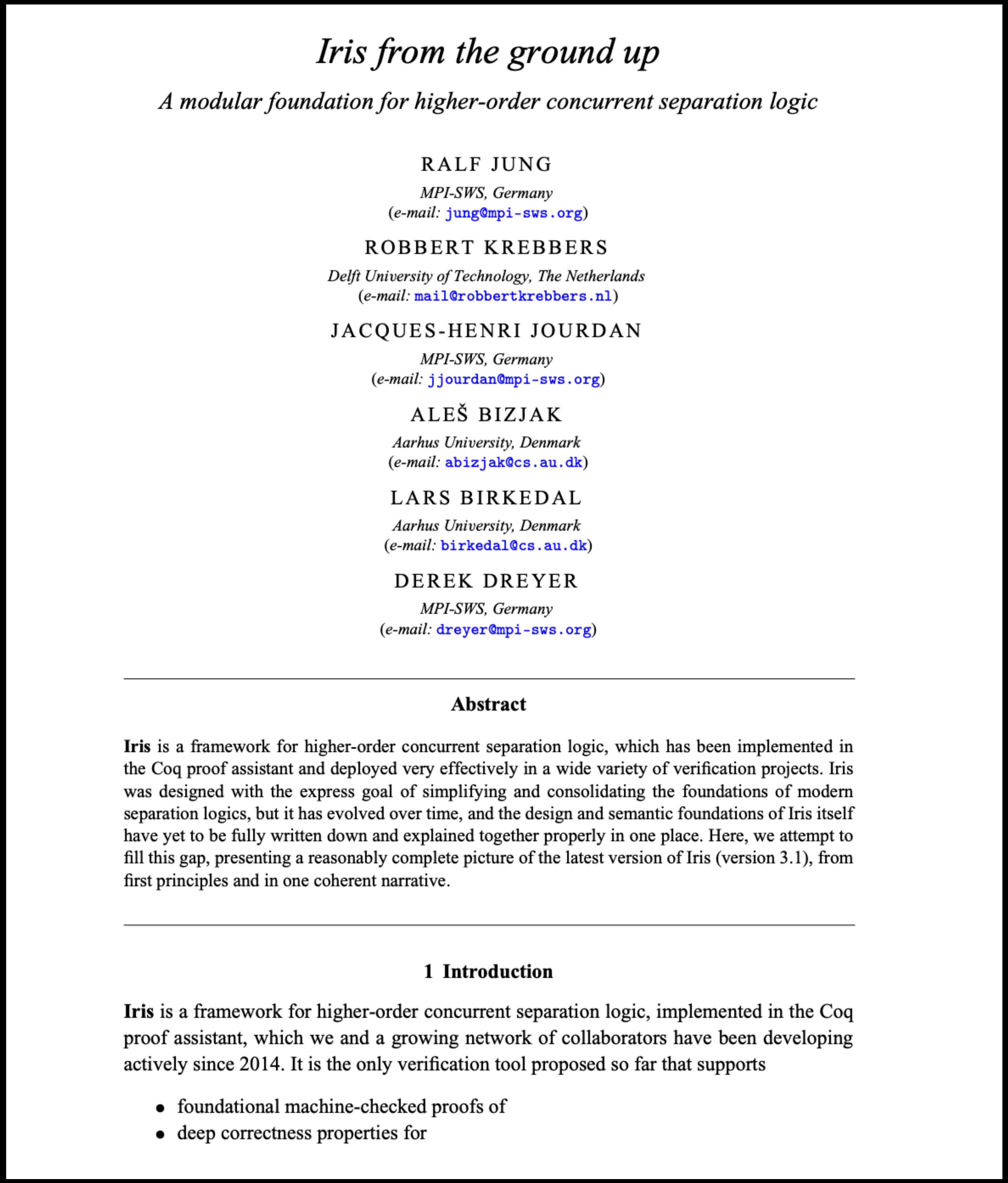
**Iris** is a framework for higher-order concurrent separation logic, which has been implemented in the Coq proof assistant and deployed very effectively in a wide variety of verification projects. Iris was designed with the express goal of simplifying and consolidating the foundations of modern separation logics, but it has evolved over time, and the design and semantic foundations of Iris itself have yet to be fully written down and explained together properly in one place. Here, we attempt to fill this gap, presenting a reasonably complete picture of the latest version of Iris (version 3.1), from first principles and in one coherent narrative.


### 1 Introduction

**Iris** is a framework for higher-order concurrent separation logic, implemented in the Coq proof assistant, which we and a growing network of collaborators have been developing actively since 2014. It is the only verification tool proposed so far that supports

- foundational machine-checked proofs of
- deep correctness properties for

# What is Iris?



*Iris from the ground up*

*A modular foundation for higher-order concurrent separation logic*

RALF JUNG
*MPI-SWS, Germany*
(*e-mail:* jung@mpi-sws.org)

ROBBERT KREBBERS
*Delft University of Technology, The Netherlands*
(*e-mail:* mail@robbertkrebbers.nl)

JACQUES-HENRI JOURDAN
*MPI-SWS, Germany*
(*e-mail:* jjourdan@mpi-sws.org)

ALEŠ BIZJAK
*Aarhus University, Denmark*
(*e-mail:* abizjak@cs.au.dk)

LARS BIRKEDAL
*Aarhus University, Denmark*
(*e-mail:* birkedal@cs.au.dk)

DEREK DREYER
*MPI-SWS, Germany*
(*e-mail:* dreyer@mpi-sws.org)

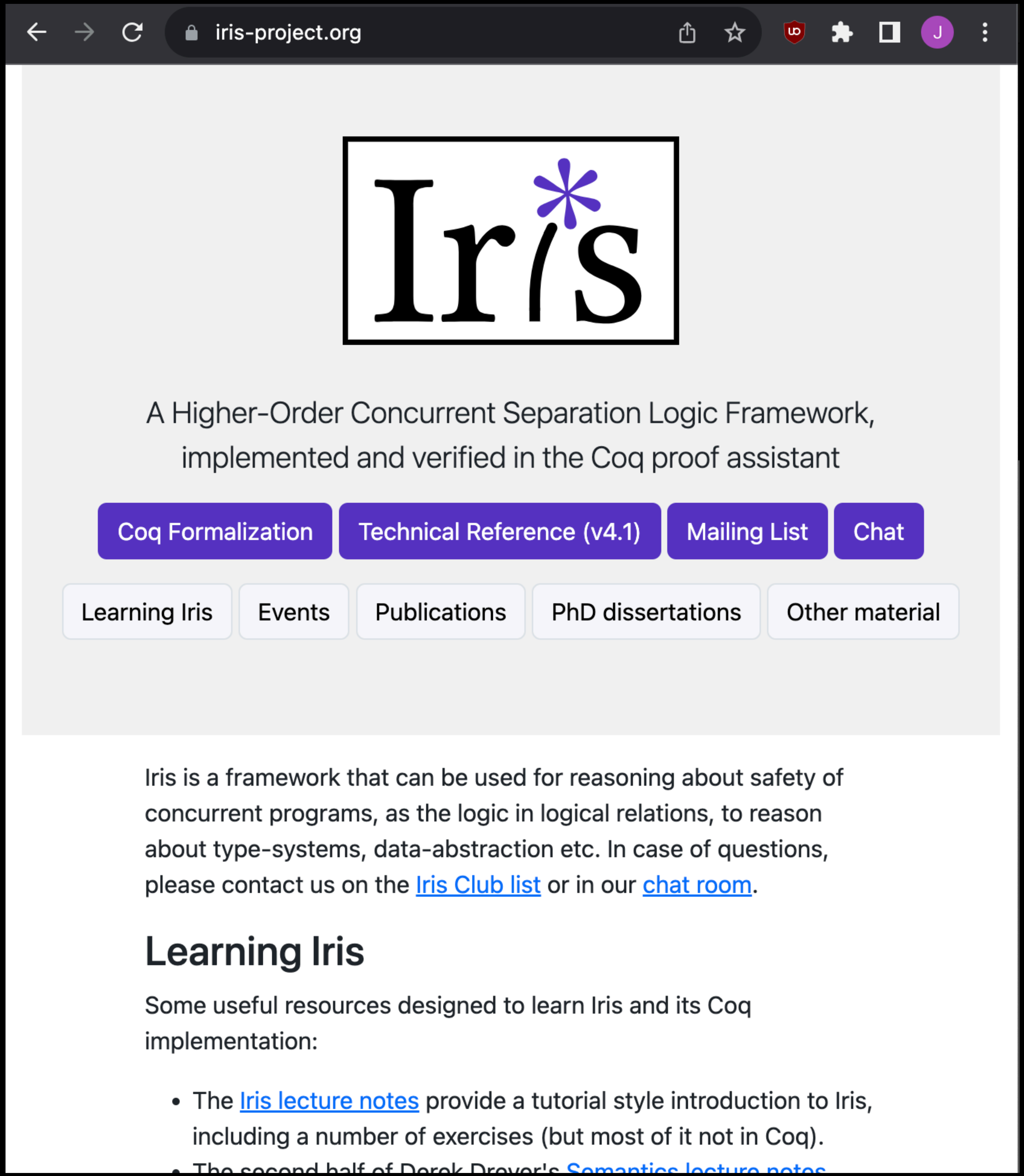
**Abstract**

**Iris** is a framework for higher-order concurrent separation logic, which has been implemented in the Coq proof assistant and deployed very effectively in a wide variety of verification projects. Iris was designed with the express goal of simplifying and consolidating the foundations of modern separation logics, but it has evolved over time, and the design and semantic foundations of Iris itself have yet to be fully written down and explained together properly in one place. Here, we attempt to fill this gap, presenting a reasonably complete picture of the latest version of Iris (version 3.1), from first principles and in one coherent narrative.

**1 Introduction**

**Iris** is a framework for higher-order concurrent separation logic, implemented in the Coq proof assistant, which we and a growing network of collaborators have been developing actively since 2014. It is the only verification tool proposed so far that supports

- foundational machine-checked proofs of
- deep correctness properties for



# What is Iris, really?

# What is Iris, really?

Coq

# What is Iris, really?

Coq

# What is Iris, really?

HeapLang {P} e {Q}

```
Definition newlock : val := λ: <>, ref #false.

Definition acquire : val := rec: "acquire" "l" :=
    if: CAS "l" #false #true then #() else "acquire" "l".

Definition release : val := λ: "l", "l" <- #false.
```

Coq

Iris

# What is Iris, really?

Coq

Iris

## HeapLang {P} e {Q}

```
Definition newlock : val := λ: <>, ref #false.

Definition acquire : val := rec: "acquire" "l" :=
    if: CAS "l" #false #true then #() else "acquire" “l".

Definition release : val := λ: "l", "l" <- #false.
```

## iProp {P} e {Q}

```
Definition lock_inv (γ : gname) (l : loc) (R : iProp Σ) : iProp Σ :=
    ∃ b : bool, l ↦ #b ∗ if b then True else own γ (Excl ()) ∗ R.

Definition is_lock (γ : gname) (lk : val) (R : iProp Σ) : iProp Σ :=
    ∃ l: loc, ⌜lk = #l⌝ ∧ inv N (lock_inv γ l R).

Definition locked (γ : gname) : iProp Σ := own γ (Excl ()).
```

# What is Iris, really?

Coq

Iris

## HeapLang {P} e {Q}

```
Definition newlock : val := λ: <>, ref #false.

Definition acquire : val := rec: "acquire" "l" :=
    if: CAS "l" #false #true then #() else "acquire" "l".

Definition release : val := λ: "l", "l" <- #false.
```

## iProp {P} e {Q}

```
Definition lock_inv (γ : gname) (l : loc) (R : iProp Σ) : iProp Σ :=
    ∃ b : bool, l ↦ #b ∗ if b then True else own γ (Excl ()) ∗ R.

Definition is_lock (γ : gname) (lk : val) (R : iProp Σ) : iProp Σ :=
    ∃ l: loc, ⌜lk = #l⌝ ∧ inv N (lock_inv γ l R).

Definition locked (γ : gname) : iProp Σ := own γ (Excl ()).
```

## Hoare Triples {P} e {Q}

```
Lemma acquire_spec γ lk R :
  {{{ is_lock γ lk R }}} acquire lk {{{ RET #(); locked γ ∗ R }}}.
```

# What is Iris, really?

Coq

Iris

## HeapLang {P} e {Q}

```
Definition newlock : val := λ: <>, ref #false.

Definition acquire : val := rec: "acquire" "l" :=
    if: CAS "l" #false #true then #() else "acquire" "l".

Definition release : val := λ: "l", "l" <- #false.
```

## iProp {P} e {Q}

```
Definition lock_inv (γ : gname) (l : loc) (R : iProp Σ) : iProp Σ :=
    ∃ b : bool, l ↦ #b ∗ if b then True else own γ (Excl ()) ∗ R.

Definition is_lock (γ : gname) (lk : val) (R : iProp Σ) : iProp Σ :=
    ∃ l: loc, ⌜lk = #l⌝ ∧ inv N (lock_inv γ l R).

Definition locked (γ : gname) : iProp Σ := own γ (Excl ()).
```

## Hoare Triples {P} e {Q}

```
Lemma acquire_spec γ lk R :
  {{{ is_lock γ lk R }}} acquire lk {{{ RET #(); locked γ ∗ R }}}.
Proof.
  iIntros (Φ) "#Hl HΦ". iLöb as "IH". wp_rec.
  wp_apply (try_acquire_spec with "Hl"). iIntros ([]).
  - iIntros "[Hlked HR]". wp_if. iApply "HΦ"; auto with iFrame.
  - iIntros "_". wp_if. iApply ("IH" with "[HΦ]"). auto.
Qed.
```

## Proof Mode

# What is Iris, really?

Coq

Iris

## HeapLang {P} e {Q}

```
Definition newlock : val := λ: <>, ref #false.

Definition acquire : val := rec: "acquire" "l" :=
    if: CAS "l" #false #true then #() else "acquire" "l".

Definition release : val := λ: "l", "l" <- #false.
```

## iProp {P} e {Q}

```
Definition lock_inv (γ : gname) (l : loc) (R : iProp Σ) : iProp Σ :=
    ∃ b : bool, l ↦ #b ∗ if b then True else own γ (Excl ()) ∗ R.

Definition is_lock (γ : gname) (lk : val) (R : iProp Σ) : iProp Σ :=
    ∃ l: loc, ⌜lk = #l⌝ ∧ inv N (lock_inv γ l R).

Definition locked (γ : gname) : iProp Σ := own γ (Excl ()).
```

## Hoare Triples {P} e {Q}

```
Lemma acquire_spec γ lk R :
  {{{ is_lock γ lk R }}} acquire lk {{{ RET #(); locked γ ∗ R }}}.
Proof.
    iIntros (Φ) "#Hl HΦ". iLöb as "IH". wp_rec.
    wp_apply (try_acquire_spec with "Hl"). iIntros ([]).
    - iIntros "[Hlked HR]". wp_if. iApply "HΦ"; auto with iFrame.
    - iIntros "_". wp_if. iApply ("IH" with "[HΦ]"). auto.
Qed.
```

## Proof Mode

Iris has many innovations over CSL!

I will not talk about these

Ask me, and I'll tell you what I know

# Iris’ adequacy theorem

# Iris' adequacy theorem

$$\{P\}\ e\ \{Q\}$$

If P holds, then e is *safe*, and afterwards Q holds

# Iris' adequacy theorem

{P} e {Q}

If P holds, then e is *safe*, and afterwards Q holds

safe(e) := "no illegal operations when running e"

(based on operational semantics for HeapLang in Coq, with heap + thread pool)

# Iris' adequacy theorem

$\{P\}\ e\ \{Q\}$

If P holds, then e is *safe*, and afterwards Q holds

safe(e) := "no illegal operations when running e"

(based on operational semantics for HeapLang in Coq, with heap + thread pool)

Partial correctness: e may loop!

# Iris' adequacy theorem

{P} e {Q}

If P holds, then e is *safe*, and afterwards Q holds

safe(e) := "no illegal operations when running e"

(based on operational semantics for HeapLang in Coq, with heap + thread pool)

Partial correctness: e may loop!

Very partial: e may deadlock!

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**
Jules Jacobs, Jonas Kastberg Hinrichsen, Robbert Krebbers
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**
Alexandre Moine, Sam Westrick, Stephanie Balzer
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**
Simon Oddershede Gregersen, Alejandro Aguirre, Philipp G. Haselwarter, Joseph Tassarotti, Lars Birkedal
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**
Amin Timany, Simon Oddershede Gregersen, Léo Stefanesco, Léon Gondelman, Abel Nieto, Lars Birkedal
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**
Angus Hammond, Zongyuan Liu, Thibaut Pérami, Peter Sewell, Lars Birkedal, Jean Pichon-Pharabod
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs
William Mansky, Ke Du

[10] **Proof Automation for Linearizability in Separation Logic**
Ike Mulder, Robbert Krebbers
In OOPSLA 2023: ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**
Michael Sammler, Simon Spies, Youngju Song, Emanuele D'Osualdo, Robbert Krebbers, Deepak Garg, Derek Dreyer
In POPL 2023: ACM SIGPLAN Symposium on Principles of Programming Languages
Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**
Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg
In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation
Recipient of **PLDI 2021 Distinguished Paper Award**
Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**
Paulo Emílio de Vilhena, François Pottier
In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] **Actris: Session-Type Based Reasoning in Separation Logic**
Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers
In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**
Glen Mével, Jacques-Henri Jourdan, François Pottier
In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**
Dan Frumin, Robbert Krebbers, Lars Birkedal
In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**
Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer
In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**
Alexandre Moine, Sam Westrick, Stephanie Balzer
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**
Simon Oddershede Gregersen, Alejandro Aguirre, Philipp G. Haselwarter, Joseph Tassarotti, Lars Birkedal
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**
Amin Timany, Simon Oddershede Gregersen, Léo Stefanesco, Léon Gondelman, Abel Nieto, Lars Birkedal
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**
Angus Hammond, Zongyuan Liu, Thibaut Pérami, Peter Sewell, Lars Birkedal, Jean Pichon-Pharabod
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs
William Mansky, Ke Du

[10] **Proof Automation for Linearizability in Separation Logic**
Ike Mulder, Robbert Krebbers
In OOPSLA 2023: ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**
Michael Sammler, Simon Spies, Youngju Song, Emanuele D'Osualdo, Robbert Krebbers, Deepak Garg, Derek Dreyer
In POPL 2023: ACM SIGPLAN Symposium on Principles of Programming Languages
Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**
Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg
In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation
Recipient of **PLDI 2021 Distinguished Paper Award**
Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**
Paulo Emílio de Vilhena, François Pottier
In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] **Actris: Session-Type Based Reasoning in Separation Logic**
Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers
In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**
Glen Mével, Jacques-Henri Jourdan, François Pottier
In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**
Dan Frumin, Robbert Krebbers, Lars Birkedal
In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**
Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer
In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**
Simon Oddershede Gregersen, Alejandro Aguirre, Philipp G. Haselwarter, Joseph Tassarotti, Lars Birkedal
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**
Amin Timany, Simon Oddershede Gregersen, Léo Stefanesco, Léon Gondelman, Abel Nieto, Lars Birkedal
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**
Angus Hammond, Zongyuan Liu, Thibaut Pérami, Peter Sewell, Lars Birkedal, Jean Pichon-Pharabod
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages
.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs
William Mansky, Ke Du

[10] **Proof Automation for Linearizability in Separation Logic**
Ike Mulder, Robbert Krebbers
In OOPSLA 2023: ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**
Michael Sammler, Simon Spies, Youngju Song, Emanuele D'Osualdo, Robbert Krebbers, Deepak Garg, Derek Dreyer
In POPL 2023: ACM SIGPLAN Symposium on Principles of Programming Languages
Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**
Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg
In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation
Recipient of **PLDI 2021 Distinguished Paper Award**
Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**
Paulo Emílio de Vilhena, François Pottier
In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] **Actris: Session-Type Based Reasoning in Separation Logic**
Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers
In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**
Glen Mével, Jacques-Henri Jourdan, François Pottier
In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**
Dan Frumin, Robbert Krebbers, Lars Birkedal
In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**
Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer
In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**
Amin Timany, Simon Oddershede Gregersen, Léo Stefanesco, Léon Gondelman, Abel Nieto, Lars Birkedal
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**
Angus Hammond, Zongyuan Liu, Thibaut Pérami, Peter Sewell, Lars Birkedal, Jean Pichon-Pharabod
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages

.pdf | Coq development | Artifact

An Iris Instance for Verifying CompCert C Programs
William Mansky, Ke Du

[10] **Proof Automation for Linearizability in Separation Logic**
Ike Mulder, Robbert Krebbers
In OOPSLA 2023: ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**
Michael Sammler, Simon Spies, Youngju Song, Emanuele D'Osualdo, Robbert Krebbers, Deepak Garg, Derek Dreyer
In POPL 2023: ACM SIGPLAN Symposium on Principles of Programming Languages
Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**
Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg
In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation
Recipient of **PLDI 2021 Distinguished Paper Award**
Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**
Paulo Emílio de Vilhena, François Pottier
In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] **Actris: Session-Type Based Reasoning in Separation Logic**
Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers
In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**
Glen Mével, Jacques-Henri Jourdan, François Pottier
In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**
Dan Frumin, Robbert Krebbers, Lars Birkedal
In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**
Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer
In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**
Angus Hammond, Zongyuan Liu, Thibaut Pérami, Peter Sewell, Lars Birkedal, Jean Pichon-Pharabod
In POPL 2024: ACM SIGPLAN Symposium on Principles of Programming Languages

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs
William Mansky, Ke Du

[10] **Proof Automation for Linearizability in Separation Logic**
Ike Mulder, Robbert Krebbers
In OOPSLA 2023: ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**
Michael Sammler, Simon Spies, Youngju Song, Emanuele D'Osualdo, Robbert Krebbers, Deepak Garg, Derek Dreyer
In POPL 2023: ACM SIGPLAN Symposium on Principles of Programming Languages
Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**
Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg
In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation
Recipient of **PLDI 2021 Distinguished Paper Award**
Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**
Paulo Emílio de Vilhena, François Pottier
In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] **Actris: Session-Type Based Reasoning in Separation Logic**
Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers
In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**
Glen Mével, Jacques-Henri Jourdan, François Pottier
In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**
Dan Frumin, Robbert Krebbers, Lars Birkedal
In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**
Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer
In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

**[1] Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

**[2] DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

**[3] Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

**[4] Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

**[5] An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs
William Mansky, Ke Du

**[10] Proof Automation for Linearizability in Separation Logic**
Ike Mulder, Robbert Krebbers
In OOPSLA 2023: ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
Recipient of **OOPSLA 2023 Distinguished Artifact Award**

**[24] DimSum: A Decentralized Approach to Multi-language Semantics and Verification**
Michael Sammler, Simon Spies, Youngju Song, Emanuele D'Osualdo, Robbert Krebbers, Deepak Garg, Derek Dreyer
In POPL 2023: ACM SIGPLAN Symposium on Principles of Programming Languages
Recipient of **POPL 2023 Distinguished Paper Award**

**[50] RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**
Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg
In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation
Recipient of **PLDI 2021 Distinguished Paper Award**
Recipient of **PLDI 2021 Distinguished Artifact Award**

**[58] A Separation Logic for Effect Handlers**
Paulo Emílio de Vilhena, François Pottier
In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

**[68] Actris: Session-Type Based Reasoning in Separation Logic**
Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers
In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

**[74] Time Credits and Time Receipts in Iris**
Glen Mével, Jacques-Henri Jourdan, François Pottier
In ESOP 2019: European Symposium on Programming

**[80] ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**
Dan Frumin, Robbert Krebbers, Lars Birkedal
In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

**[81] RustBelt: Securing the Foundations of the Rust Programming Language**
Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

**[91] Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer
In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] **Proof Automation for Linearizability in Separation Logic**
Ike Mulder, Robbert Krebbers
In OOPSLA 2023: ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications
Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**
Michael Sammler, Simon Spies, Youngju Song, Emanuele D'Osualdo, Robbert Krebbers, Deepak Garg, Derek Dreyer
In POPL 2023: ACM SIGPLAN Symposium on Principles of Programming Languages
Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**
Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg
In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation
Recipient of **PLDI 2021 Distinguished Paper Award**
Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**
Paulo Emílio de Vilhena, François Pottier
In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] **Actris: Session-Type Based Reasoning in Separation Logic**
Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers
In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**
Glen Mével, Jacques-Henri Jourdan, François Pottier
In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**
Dan Frumin, Robbert Krebbers, Lars Birkedal
In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**
Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer
In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] **Proof Automation for Linearizability in Separation Logic**

**Automated proofs (with concurrency)**

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**
Michael Sammler, Simon Spies, Youngju Song, Emanuele D'Osualdo, Robbert Krebbers, Deepak Garg, Derek Dreyer
In POPL 2023: ACM SIGPLAN Symposium on Principles of Programming Languages
Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**
Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg
In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation
Recipient of **PLDI 2021 Distinguished Paper Award**
Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**
Paulo Emílio de Vilhena, François Pottier
In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] **Actris: Session-Type Based Reasoning in Separation Logic**
Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers
In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**
Glen Mével, Jacques-Henri Jourdan, François Pottier
In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**
Dan Frumin, Robbert Krebbers, Lars Birkedal
In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**
Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer
In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] **Proof Automation for Linearizability in Separation Logic**

**Automated proofs (with concurrency)**

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**

Michae … Garg, Derek Dreyer

In POPL

**Multi-language interoperability**

Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**

Michael Sammler, Rodolphe Lepigre, Robbert Krebbers, Kayvan Memarian, Derek Dreyer, Deepak Garg

In PLDI 2021: ACM SIGPLAN Conference on Programming Language Design and Implementation

Recipient of **PLDI 2021 Distinguished Paper Award**

Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**

Paulo Emílio de Vilhena, François Pottier

In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] **Actris: Session-Type Based Reasoning in Separation Logic**

Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers

In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**

Glen Mével, Jacques-Henri Jourdan, François Pottier

In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**

Dan Frumin, Robbert Krebbers, Lars Birkedal

In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**

Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer

In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**

Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] DisLog: A Separation Logic for Disentanglement

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] Asynchronous Probabilistic Couplings in Higher-Order Separation Logic

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic

**Arm semantics (weak memory model)**

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] Proof Automation for Linearizability in Separation Logic

**Automated proofs (with concurrency)**

ns

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] DimSum: A Decentralized Approach to Multi-language Semantics and Verification

Michae Garg, Derek Dreyer

In POPL

**Multi-language interoperability**

Recipient of **POPL 2023 Distinguished Paper Award**

[50] RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types

Michael Samm Deepak Garg

In PLDI 2021: A

**C verification (automated)**

Recipient of PL

Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] A Separation Logic for Effect Handlers

Paulo Emílio de Vilhena, François Pottier

In POPL 2021: ACM SIGPLAN Symposium on Principles of Programming Languages

[68] Actris: Session-Type Based Reasoning in Separation Logic

Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers

In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] Time Credits and Time Receipts in Iris

Glen Mével, Jacques-Henri Jourdan, François Pottier

In ESOP 2019: European Symposium on Programming

[80] ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency

Dan Frumin, Robbert Krebbers, Lars Birkedal

In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] RustBelt: Securing the Foundations of the Rust Programming Language

Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer

In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")

Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] **Proof Automation for Linearizability in Separation Logic**

**Automated proofs (with concurrency)**

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**

Michae … Garg, Derek Dreyer

In POPL

**Multi-language interoperability**

Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**

Michael Samm … Deepak Garg

In PLDI 2021: A

**C verification (automated)**

Recipient of **PL**

Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**

Paulo Emílio de Vilhena, Fran

In POPL 2021: ACM SIGPLAN Sy

**Effect handlers**

[68] **Actris: Session-Type Based Reasoning in Separation Logic**

Jonas Kastberg Hinrichsen, Jesper Bengtson, Robbert Krebbers

In POPL 2020: ACM SIGPLAN Symposium on Principles of Programming Languages

[74] **Time Credits and Time Receipts in Iris**

Glen Mével, Jacques-Henri Jourdan, François Pottier

In ESOP 2019: European Symposium on Programming

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**

Dan Frumin, Robbert Krebbers, Lars Birkedal

In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**

Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer

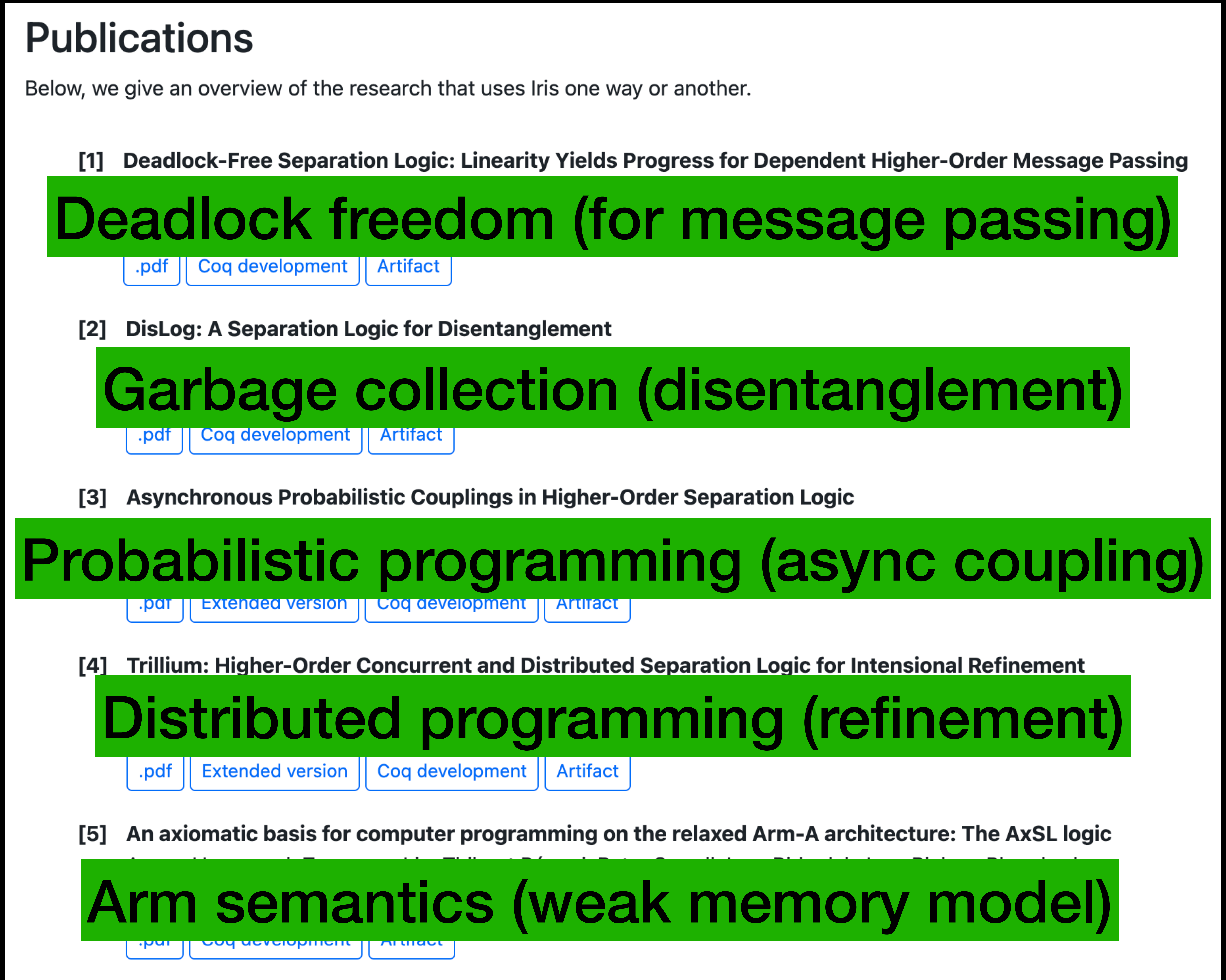
In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**

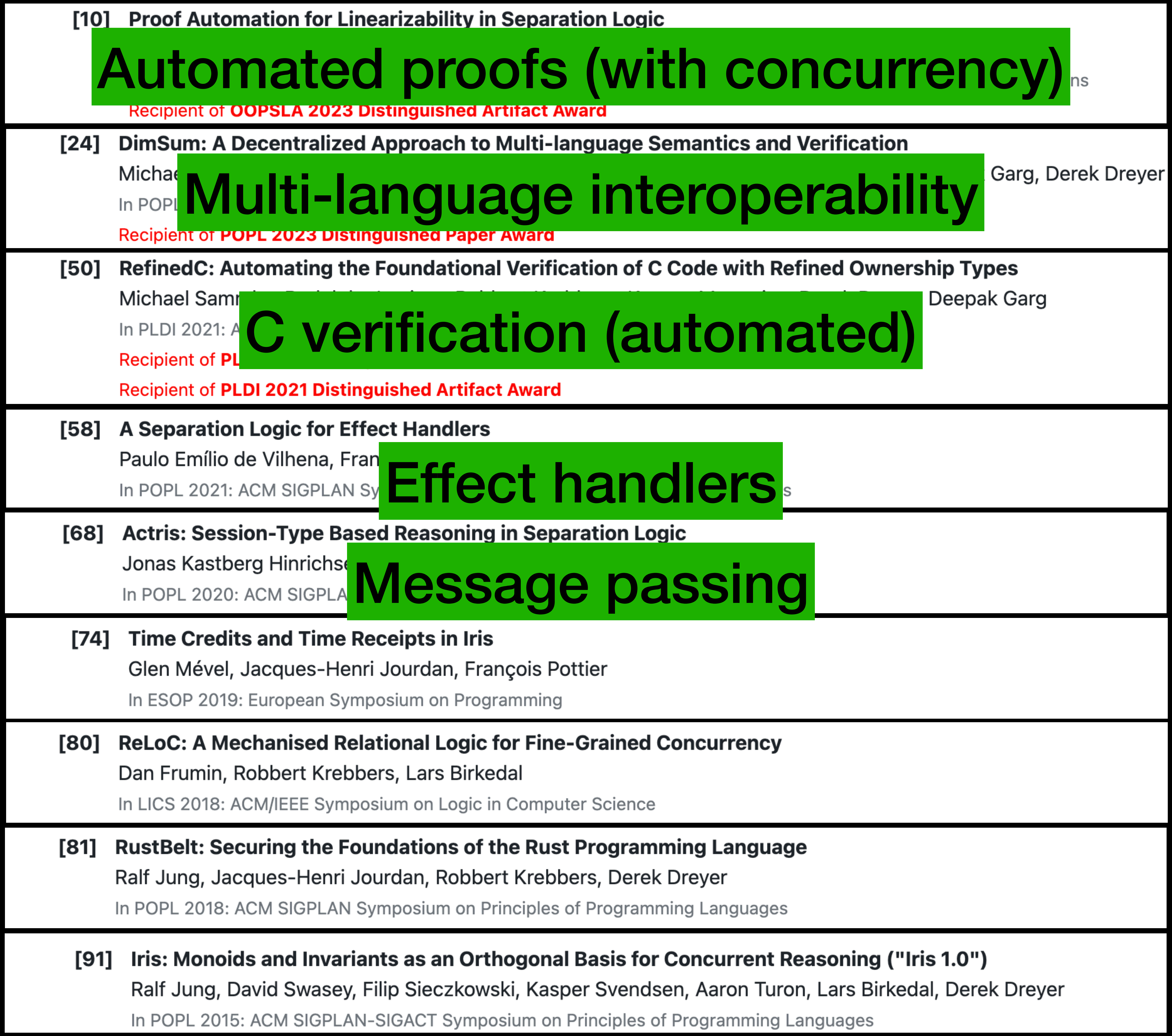
Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?







# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

Deadlock freedom (for message passing)

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

Garbage collection (disentanglement)

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

Probabilistic programming (async coupling)

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

Distributed programming (refinement)

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

Arm semantics (weak memory model)

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk…

Iris + CompCert

[10] **Proof Automation for Linearizability in Separation Logic**

Automated proofs (with concurrency)

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**

Michae… Garg, Derek Dreyer

In POPL…

Multi-language interoperability

Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**

Michael Samm… Deepak Garg

In PLDI 2021: A…

C verification (automated)

Recipient of **PL…**

Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**

Paulo Emílio de Vilhena, Fran…

In POPL 2021: ACM SIGPLAN Sy…

Effect handlers

[68] **Actris: Session-Type Based Reasoning in Separation Logic**

Jonas Kastberg Hinrichs…

In POPL 2020: ACM SIGPLA…

Message passing

[74] **Time Credits and Time Receipts in Iris**

Glen Mével, Jacques-Henr…

In ESOP 2019: European Sym…

Time complexity

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**

Dan Frumin, Robbert Krebbers, Lars Birkedal

In LICS 2018: ACM/IEEE Symposium on Logic in Computer Science

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**

Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer

In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**

Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] **Proof Automation for Linearizability in Separation Logic**

**Automated proofs (with concurrency)**

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**

Michae … Garg, Derek Dreyer

In POPL

**Multi-language interoperability**

Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**

Michael Samm … Deepak Garg

In PLDI 2021: A

**C verification (automated)**

Recipient of PL

Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**

Paulo Emílio de Vilhena, Fran

In POPL 2021: ACM SIGPLAN Sy

**Effect handlers**

[68] **Actris: Session-Type Based Reasoning in Separation Logic**

Jonas Kastberg Hinrichs

In POPL 2020: ACM SIGPLA

**Message passing**

[74] **Time Credits and Time Receipts in Iris**

Glen Mével, Jacques-Henr

In ESOP 2019: European Sym

**Time complexity**

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**

Dan Frumin, Robbert Krebb

In LICS 2018: ACM/IEEE Sympo

**Hyperproperties**

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**

Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer

In POPL 2018: ACM SIGPLAN Symposium on Principles of Programming Languages

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**

Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] **Proof Automation for Linearizability in Separation Logic**

**Automated proofs (with concurrency)**

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**

Michae … Garg, Derek Dreyer

In POPL

**Multi-language interoperability**

Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**

Michael Samm … Deepak Garg

In PLDI 2021: A

**C verification (automated)**

Recipient of PL

Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**

Paulo Emílio de Vilhena, Fran

In POPL 2021: ACM SIGPLAN Sy

**Effect handlers**

[68] **Actris: Session-Type Based Reasoning in Separation Logic**

Jonas Kastberg Hinrichs

In POPL 2020: ACM SIGPLA

**Message passing**

[74] **Time Credits and Time Receipts in Iris**

Glen Mével, Jacques-Henr

In ESOP 2019: European Sym

**Time complexity**

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**

Dan Frumin, Robbert Krebb

In LICS 2018: ACM/IEEE Sympo

**Hyperproperties**

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**

**Rust type system (soundness proof)**

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**

Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] **Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing**

**Deadlock freedom (for message passing)**

.pdf | Coq development | Artifact

[2] **DisLog: A Separation Logic for Disentanglement**

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] **Asynchronous Probabilistic Couplings in Higher-Order Separation Logic**

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] **Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement**

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] **An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic**

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] **Proof Automation for Linearizability in Separation Logic**

**Automated proofs (with concurrency)**

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] **DimSum: A Decentralized Approach to Multi-language Semantics and Verification**

Michae... Garg, Derek Dreyer

In POPL...

**Multi-language interoperability**

Recipient of **POPL 2023 Distinguished Paper Award**

[50] **RefinedC: Automating the Foundational Verification of C Code with Refined Ownership Types**

Michael Samm... Deepak Garg

In PLDI 2021: A...

Recipient of PL...

**C verification (automated)**

Recipient of **PLDI 2021 Distinguished Artifact Award**

[58] **A Separation Logic for Effect Handlers**

Paulo Emílio de Vilhena, Fran...

In POPL 2021: ACM SIGPLAN Sy...

**Effect handlers**

[68] **Actris: Session-Type Based Reasoning in Separation Logic**

Jonas Kastberg Hinrichs...

In POPL 2020: ACM SIGPLA...

**Message passing**

[74] **Time Credits and Time Receipts in Iris**

Glen Mével, Jacques-Henr...

In ESOP 2019: European Sym...

**Time complexity**

[80] **ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency**

Dan Frumin, Robbert Krebb...

In LICS 2018: ACM/IEEE Sympo...

**Hyperproperties**

[81] **RustBelt: Securing the Foundations of the Rust Programming Language**

**Rust type system (soundness proof)**

[91] **Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")**

Ralf Jung, David Swasey, ... ars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLA... uages

**Original Iris paper**

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing

**Dea**

[2] DisLog: A Separation Logic for Disentanglement

**Garbage collection (disentanglement)**

.pdf | Coq development | Artifact

[3] Asynchronous Probabilistic Couplings in Higher-Order Separation Logic

**Probabilistic programming (async coupling)**

.pdf | Extended version | Coq development | Artifact

[4] Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement

**Distributed programming (refinement)**

.pdf | Extended version | Coq development | Artifact

[5] An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic

**Arm semantics (weak memory model)**

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

**Iris + CompCert**

[10] Proof Automation for Linearizability in Separation Logic

**Automated proofs (with concurrency)**

Recipient of **OOPSLA 2023 Distinguished Artifact Award**

[24] DimSum: A Decentralized Approach to Multi-language Semantics and Verification

Michae ... Garg, Derek Dreyer

**Every paper re-uses a different subset of Iris**

In PLDI 2021: A

Recipient of PL

Recipient of **PLDI 2021 Distinguished Artifact Award**

**C verification (automated)**

[58] A Separation Logic for Effect Handlers

Paulo Emílio de Vilhena, Fran

In POPL 2021: ACM SIGPLAN Sy

**Effect handlers**

[68] Actris: Session-Type Based Reasoning in Separation Logic

Jonas Kastberg Hinrichs

In POPL 2020: ACM SIGPLA

**Message passing**

[74] Time Credits and Time Receipts in Iris

Glen Mével, Jacques-Henr

In ESOP 2019: European Sym

**Time complexity**

[80] ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency

Dan Frumin, Robbert Krebb

In LICS 2018: ACM/IEEE Sympo

**Hyperproperties**

[81] RustBelt: Securing the Foundations of the Rust Programming Language

**Rust type system (soundness proof)**

[91] Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")

Ralf Jung, David Swasey, ... ars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLA ... uages

**Original Iris paper**

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing

**Every paper re-uses a different subset of Iris**

[2] DisLog: A Separation Logic for Disentanglement

Garbage collection (disentanglement)

**[1] re-uses step-indexing and the Iris proof mode**

Probabilistic programming (async coupling)

.pdf | Extended version | Coq development | Artifact

[4] Trillium: Higher-Order Concurrent and Distributed Separation Logic for Intensional Refinement

Distributed programming (refinement)

.pdf | Extended version | Coq development | Artifact

[5] An axiomatic basis for computer programming on the relaxed Arm-A architecture: The AxSL logic

Arm semantics (weak memory model)

An Iris Instance for Verifying CompCert C Programs

William Mansk…

Iris + CompCert

[10] Proof Automation for Linearizability in Separation Logic

Automated proofs (with concurrency)

Recipient of OOPSLA 2023 Distinguished Artifact Award

[24] DimSum: A Decentralized Approach to Multi-language Semantics and Verification

Michae… Garg, Derek Dreyer

…Garg

In PLDI 2021: …

Recipient of PL…

C verification (automated)

[68] Actris: Session-Type Based Reasoning in Separation Logic

Jonas Kastberg Hinrichs…

In POPL 2020: ACM SIGPLA…

Message passing

[74] Time Credits and Time Receipts in Iris

Glen Mével, Jacques-Henri…

In ESOP 2019: European Sym…

Time complexity

[80] ReLoC: A Mechanised Relational Logic for Fine-Grained Concurrency

Dan Frumin, Robbert Krebb…

In LICS 2018: ACM/IEEE Sympo…

Hyperproperties

[81] RustBelt: Securing the Foundations of the Rust Programming Language

Rust type system (soundness proof)

[91] Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")

Ralf Jung, David Swasey, … Lars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLA… …uages

Original Iris paper

# Why are there 91 Iris papers?

## Publications

Below, we give an overview of the research that uses Iris one way or another.

[1] Deadlock-Free Separation Logic: Linearity Yields Progress for Dependent Higher-Order Message Passing

[2] DisLog: A Separation Logic for Disentanglement

Garbage collection (disentanglement)

Probabilistic programming (async coupling)

[4]

[5]

Arm semantics (weak memory model)

.pdf | Coq development | Artifact

☆ An Iris Instance for Verifying CompCert C Programs

William Mansk

Iris + CompCert

[10] Proof Automation for Linearizability in Separation Logic

Automated proofs (with concurrency)

Recipient of OOPSLA 2023 Distinguished Artifact Award

[24] DimSum: A Decentralized Approach to Multi-language Semantics and Verification

Michae… Garg, Derek Dreyer

…Garg

In PLDI 2021: A…

Recipient of PL…

C verification (automated)

[68] Actris: Session-Type Based Reasoning in Separation Logic

Hyperproperties

[81] RustBelt: Securing the Foundations of the Rust Programming Language

Rust type system (soundness proof)

[91] Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning ("Iris 1.0")

Ralf Jung, David Swasey, … Lars Birkedal, Derek Dreyer

In POPL 2015: ACM SIGPLA… …uages

Original Iris paper

**Every paper re-uses a different subset of Iris**

**[1] re-uses step-indexing and the Iris proof mode**

**Comparison: 200+ papers on effect handlers & way more on probabilistic programming**

# The Masterplan

Iris

# The Masterplan

Iris

Build a fully verified software stack

Hypervisor, OS, compiler, type system, database,

web server, …, and your code!

# The Masterplan

All programming paradigms

Strong adequacy theorems

Fully mechanised proofs

End-to-end theorems

Free theorems from types

Strong reasoning principles

Proof automation

Modular proofs

Iris

Build a fully verified software stack

Hypervisor, OS, compiler, type system, database,

web server, …, and your code!