Nola: Later-free ghost state for verifying termination in Iris

Yusuke Matsushita Kyoto University
Takeshi Tsukada Chiba University

June 20, 2025 – PLDI 2025 @ Seoul

Shared Mutable State × Termination is Hard

Shared Mutable State × Termination is Hard

Landin's knot

```
let r = ref (λ_.()) in
*r = λ_.(*r)(); (*r)()
```

Shared Mutable State × Termination is Hard

Landin's knot

```
let r = ref (λ_.()) in
*r = λ_.(*r)(); (*r)()
```

Shared mutable ref to a **function** `r : ref (() → ())`

Shared Mutable State × Termination is Hard

Landin's knot

```
let r = ref (λ_. ()) in
*r = λ_.(*r)(); (*r)()
```

Shared mutable ref to a **function** `r : ref (() → ())`

No loop appears in the program

Shared Mutable State × Termination is Hard

Landin's knot

```
let r = ref (λ_.()) in
*r = λ_.(*r)(); (*r)()
```

Shared mutable ref to a **function** `r : ref (() → ())`

No loop appears in the program

But it **loops infinitely**… 🤯

Shared Mutable State × Termination is Hard

Landin's knot

```
let r = ref (λ_.()) in
*r = λ_.(*r)(); (*r)()
```

Shared mutable ref to a **function** `r : ref (() → ())`

No loop appears in the program

But it **loops infinitely**… 🤯

Shared Mutable State × Termination is Hard

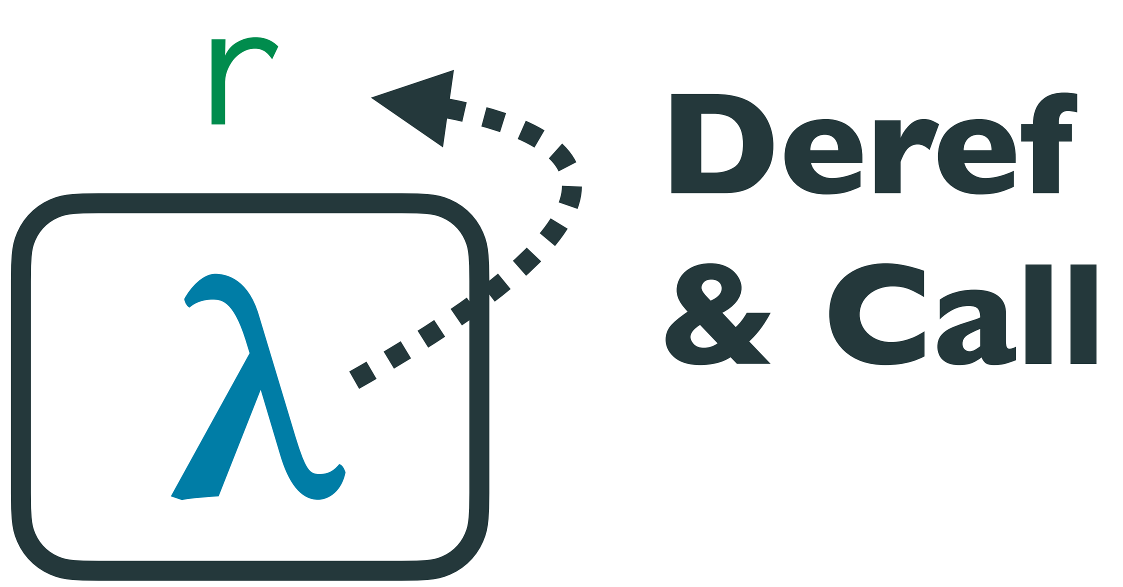
Landin's knot

```
let r = ref (λ_.()) in
*r = λ_.(*r)();  (*r)()
```

Shared mutable ref to a **function** `r : ref (() → ())`

No loop appears in the program

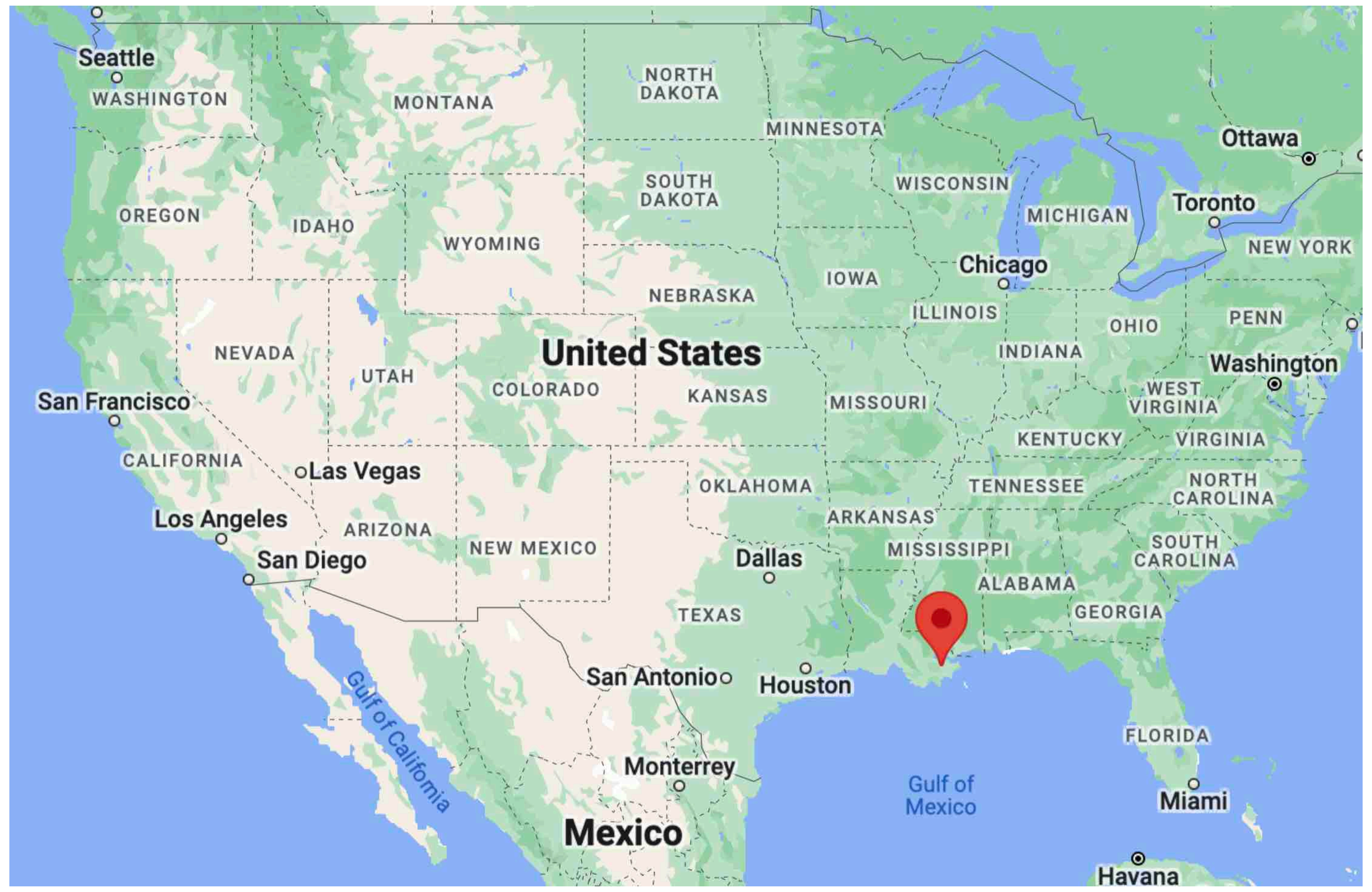
But it **loops infinitely**… 🤯



Self-reference

Our work, Nola

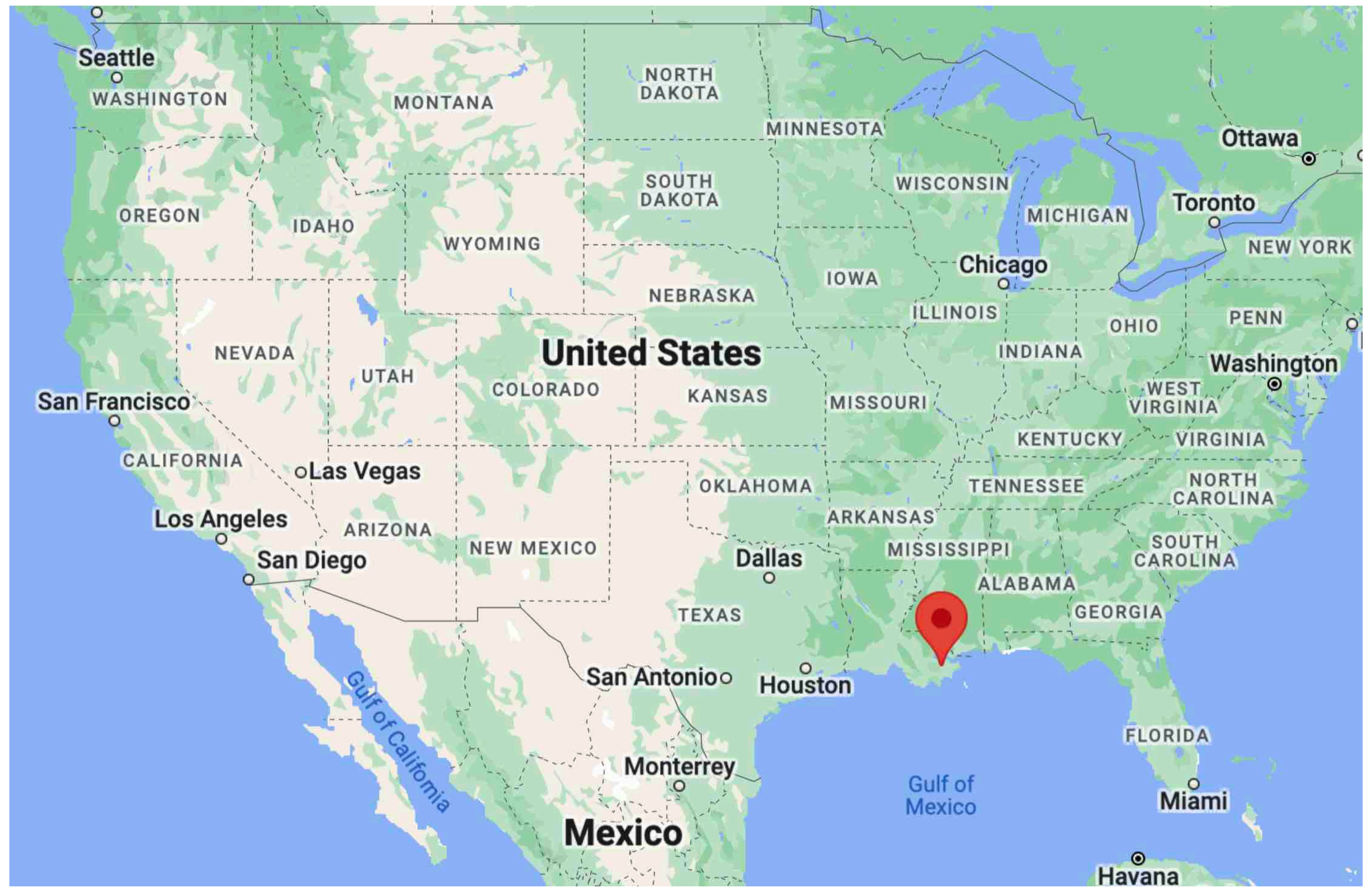
Our work, Nola

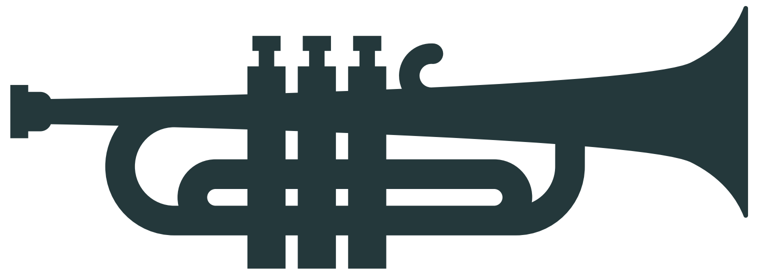


New Orleans, Louisiana

= NOLA

Our work, Nola



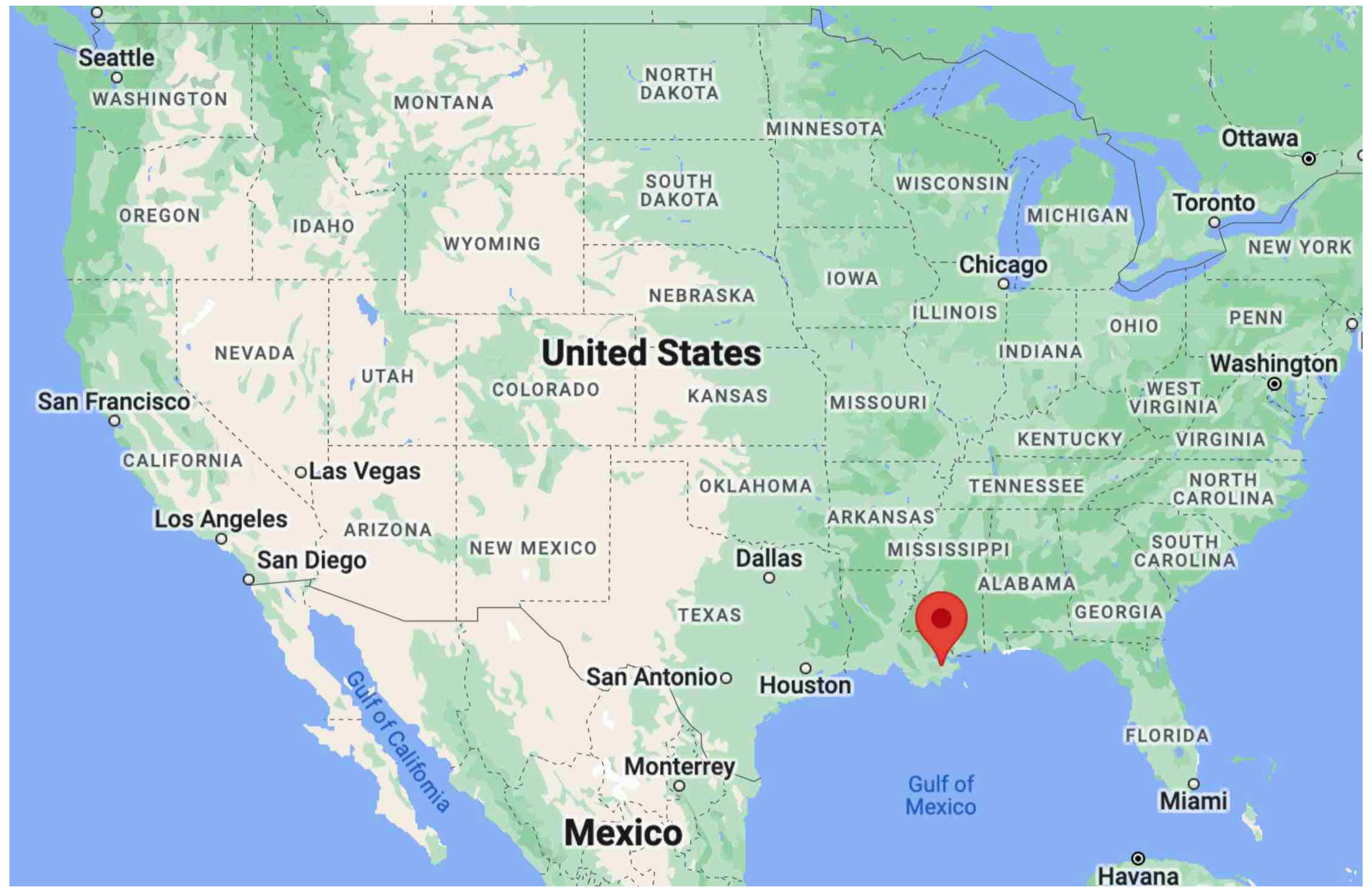


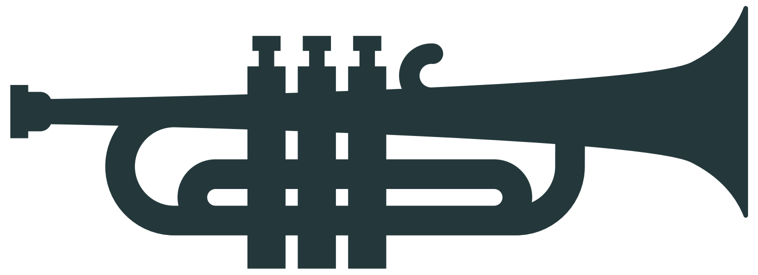
New Orleans, Louisiana

= NOLA

Our work, Nola

Matsushita & Tsukada PLDI ’25

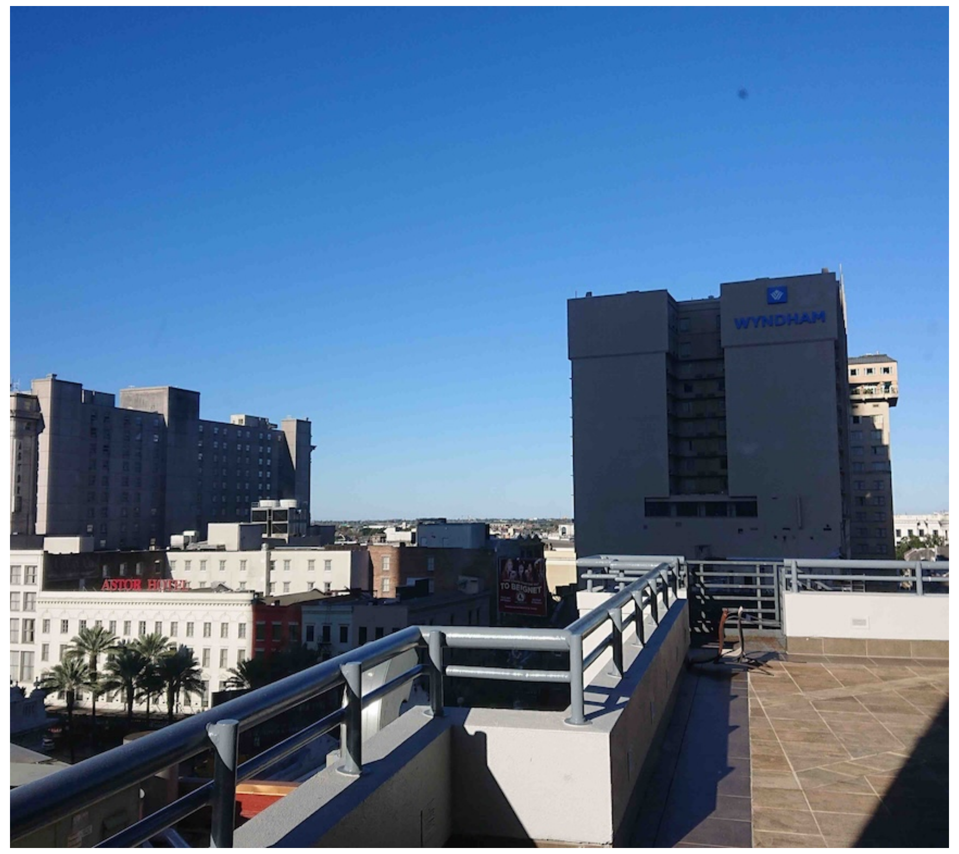




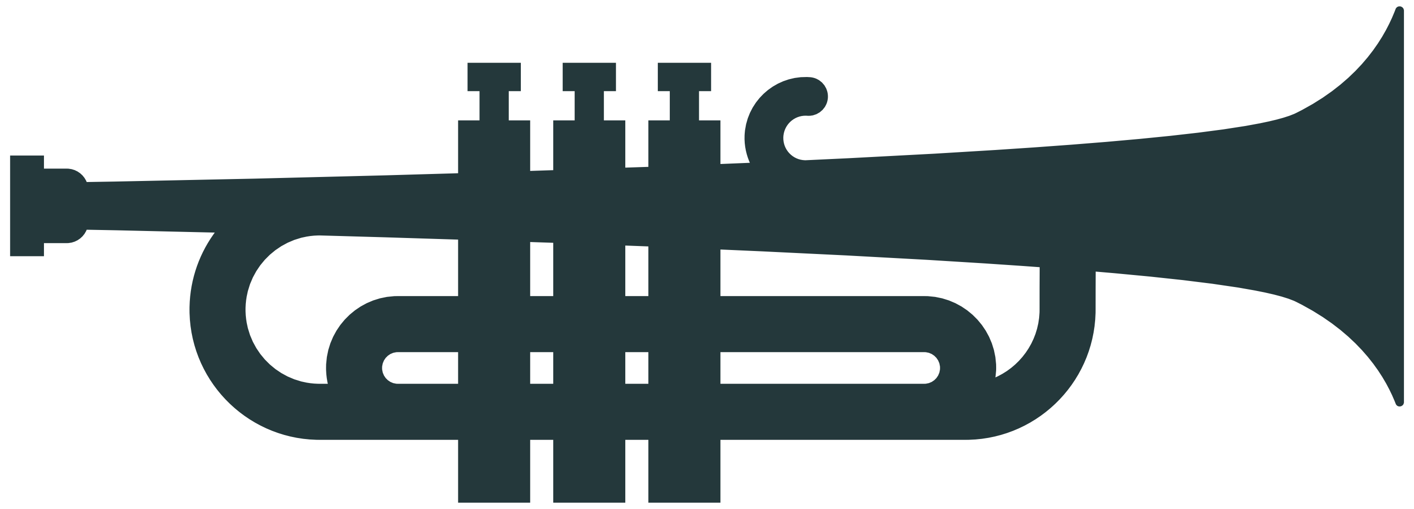
POPL 2020 New Orleans

New Orleans, Louisiana

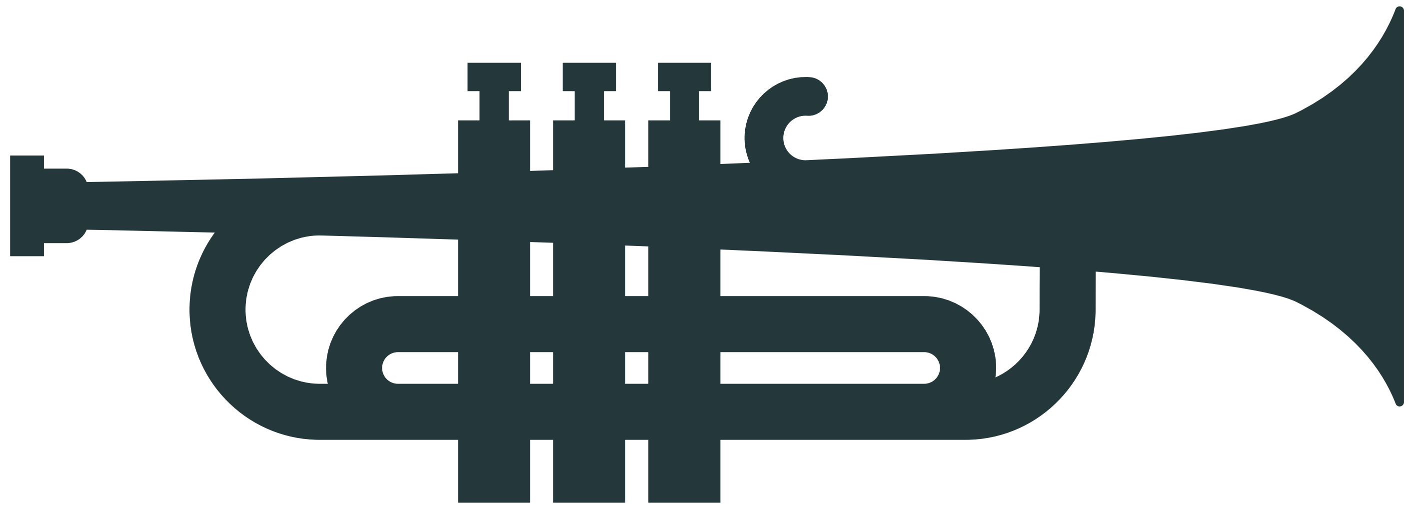
= NOLA



Our work, Nola



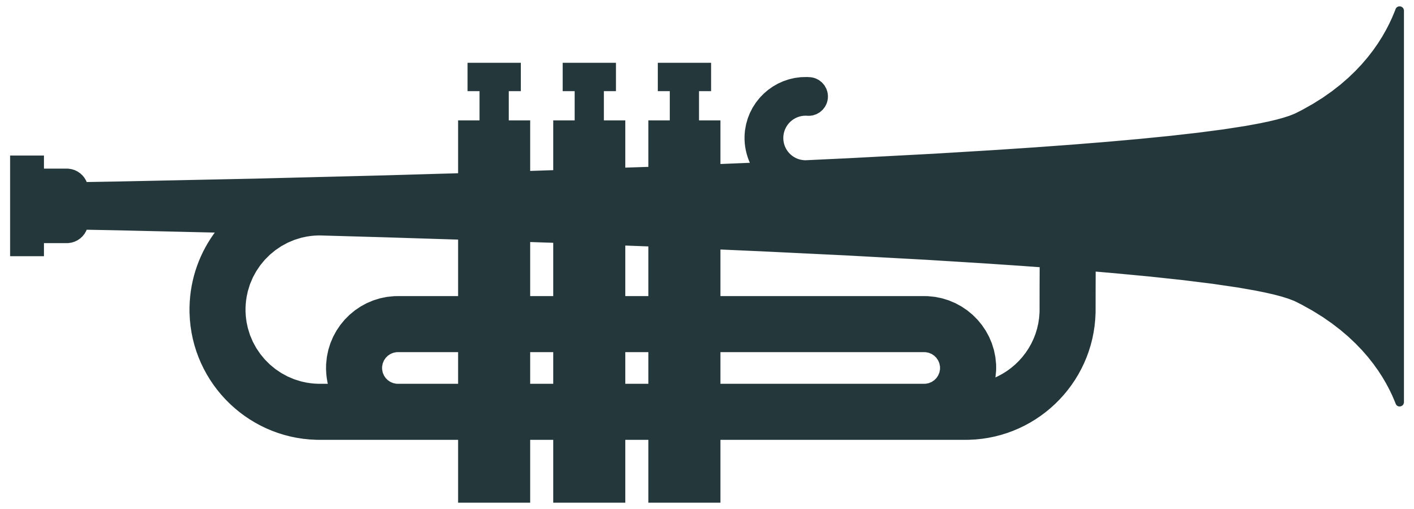
Our work, Nola



Nola

No later

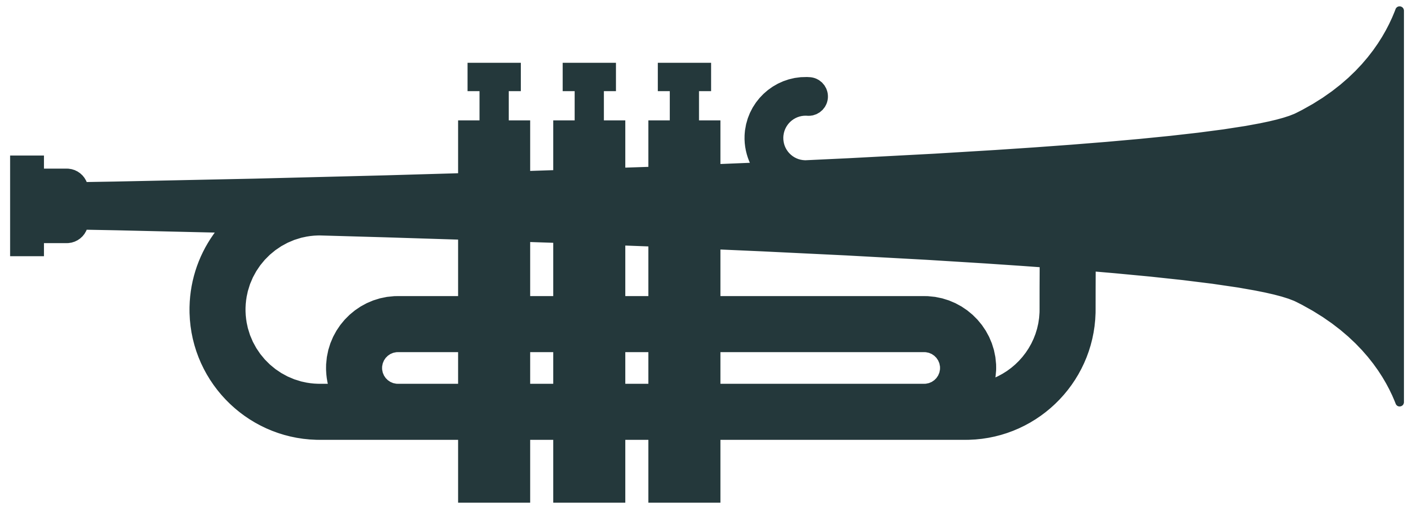
Our work, Nola



No later

Helps you finish your work…

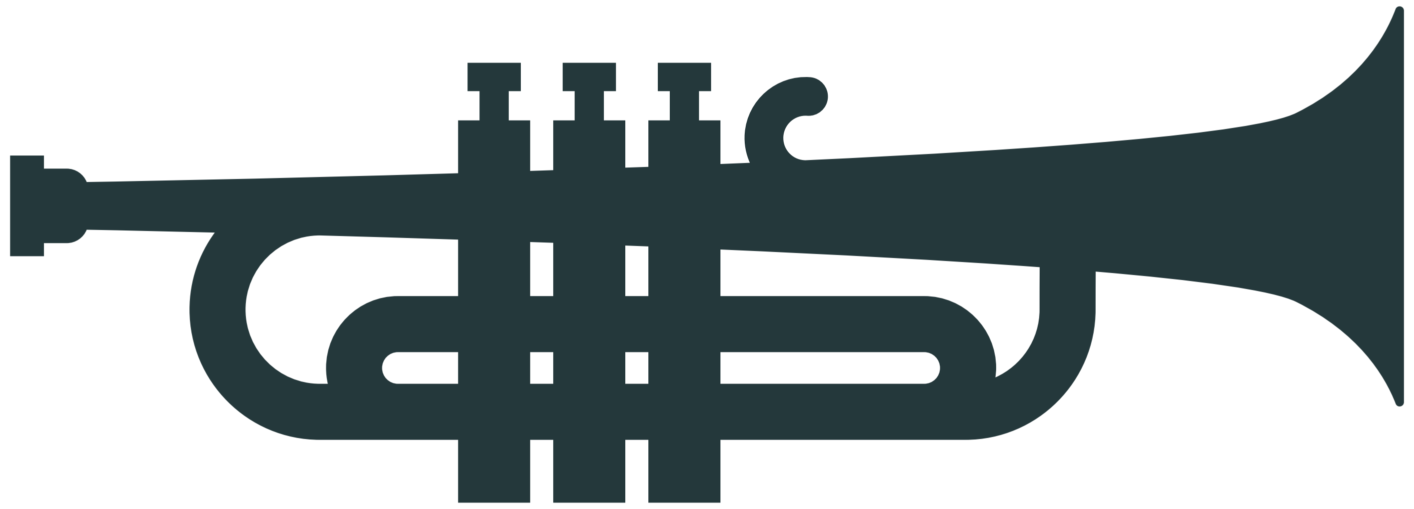
Our work, Nola



Later modality ▷

No later

Our work, Nola

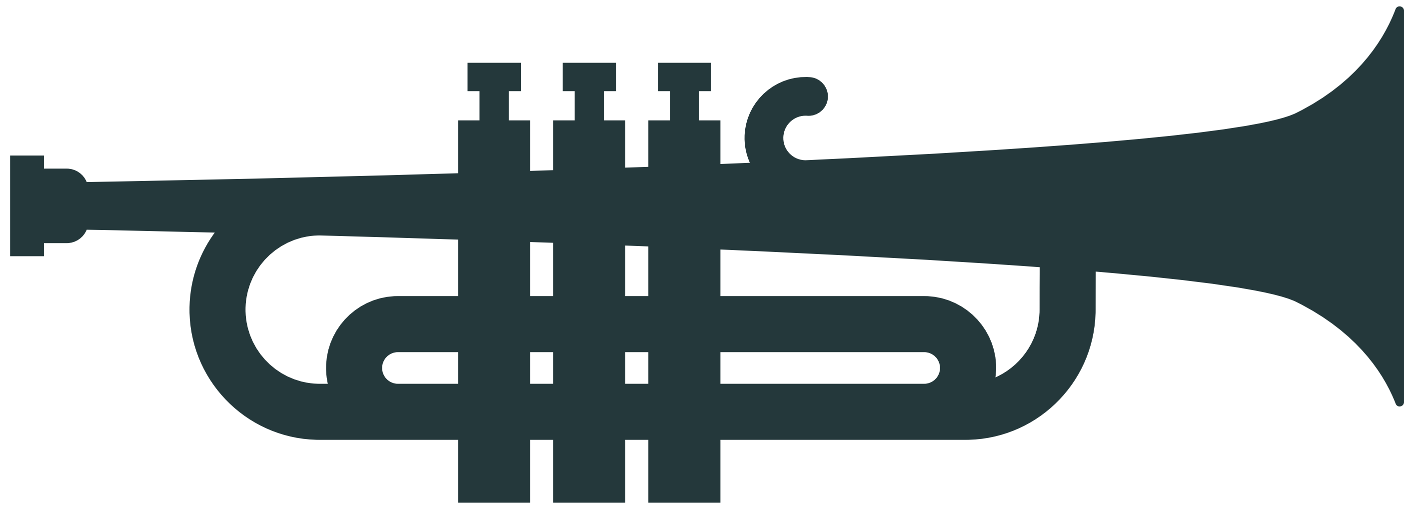


No later

Later modality ▷

Iris Separation Logic

Our work, Nola

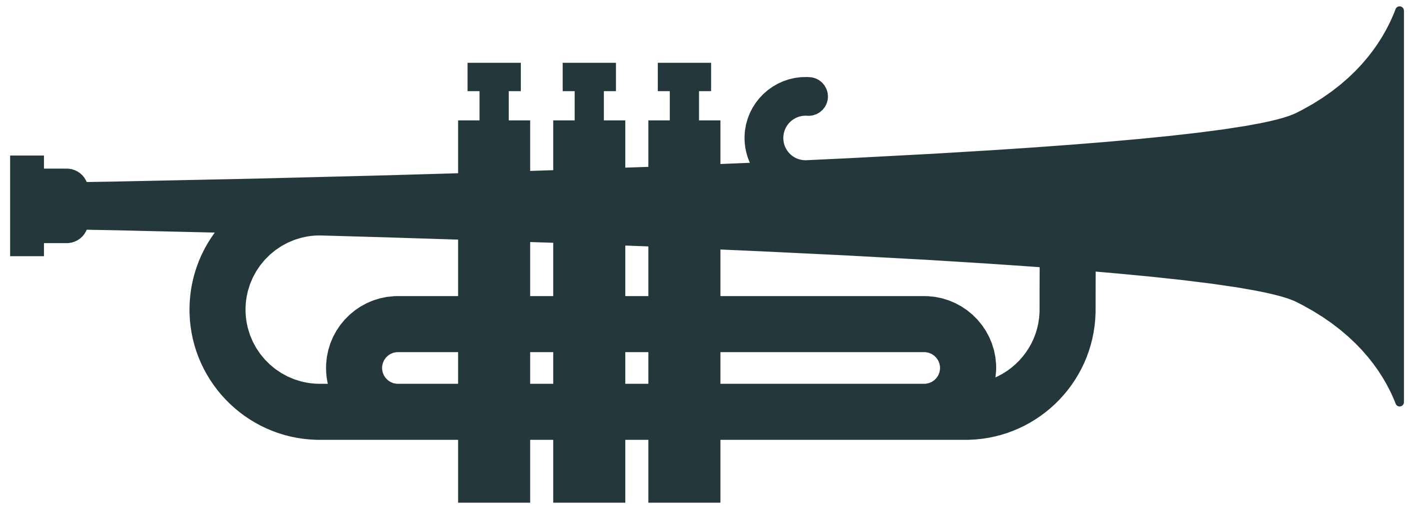


No later

~~Later modality $\triangleright$~~

Iris **Separation Logic**

Our work, Nola

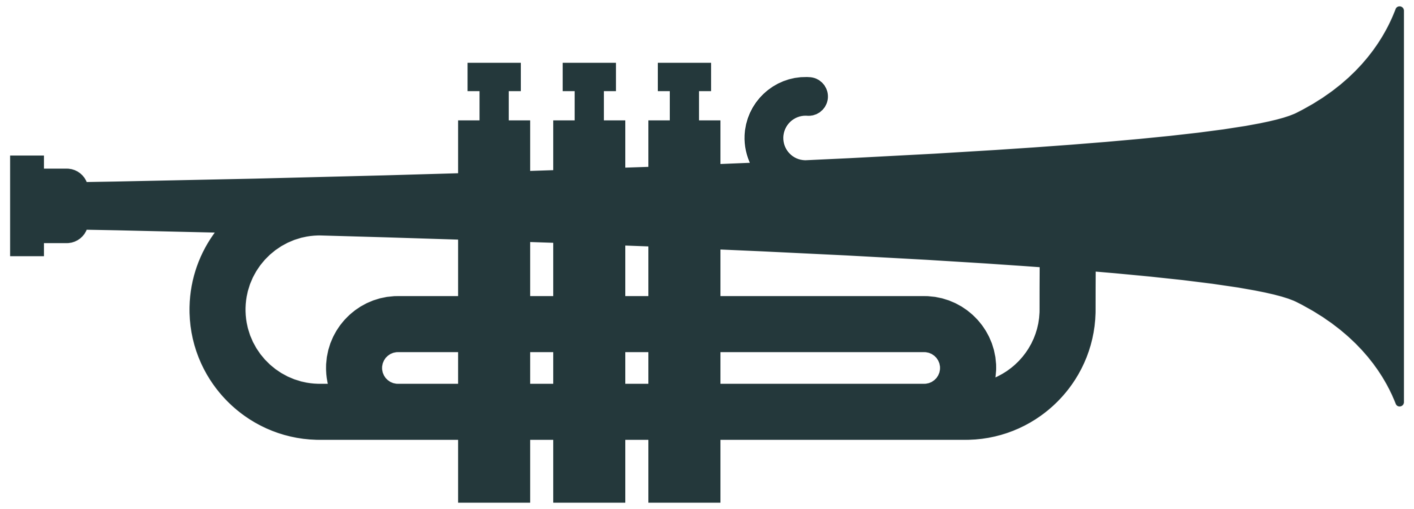


No later

Later modality $\triangleright$

Iris **Separation Logic**

Our work, Nola



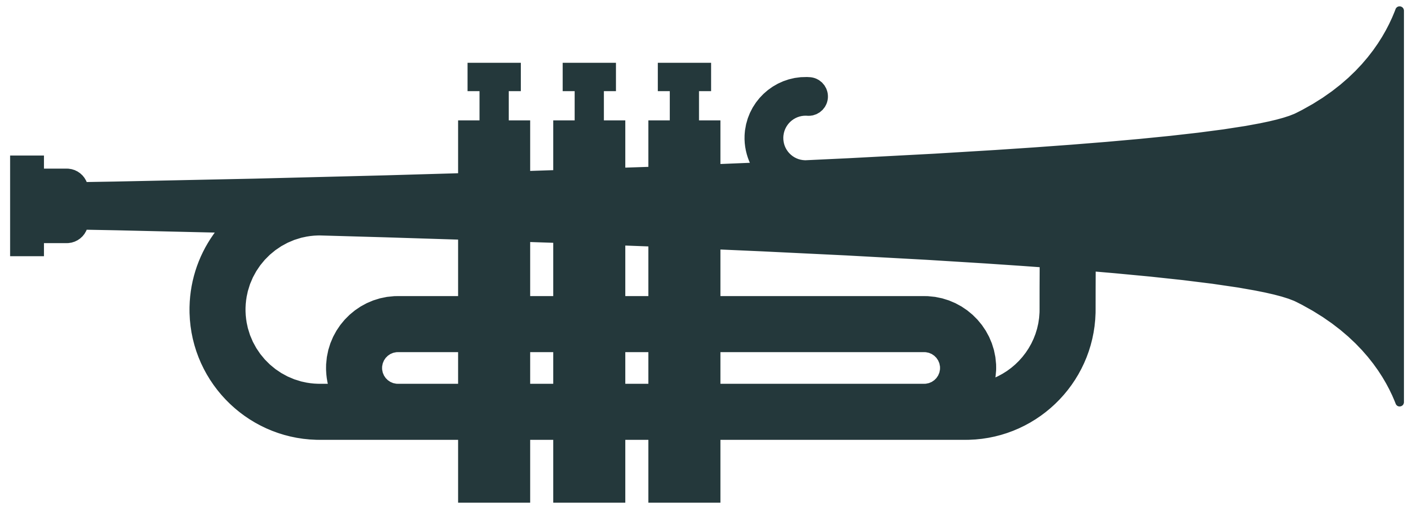
No later

Later modality $\triangleright$

Iris Separation Logic

Shared mutable state

Our work, Nola



No later

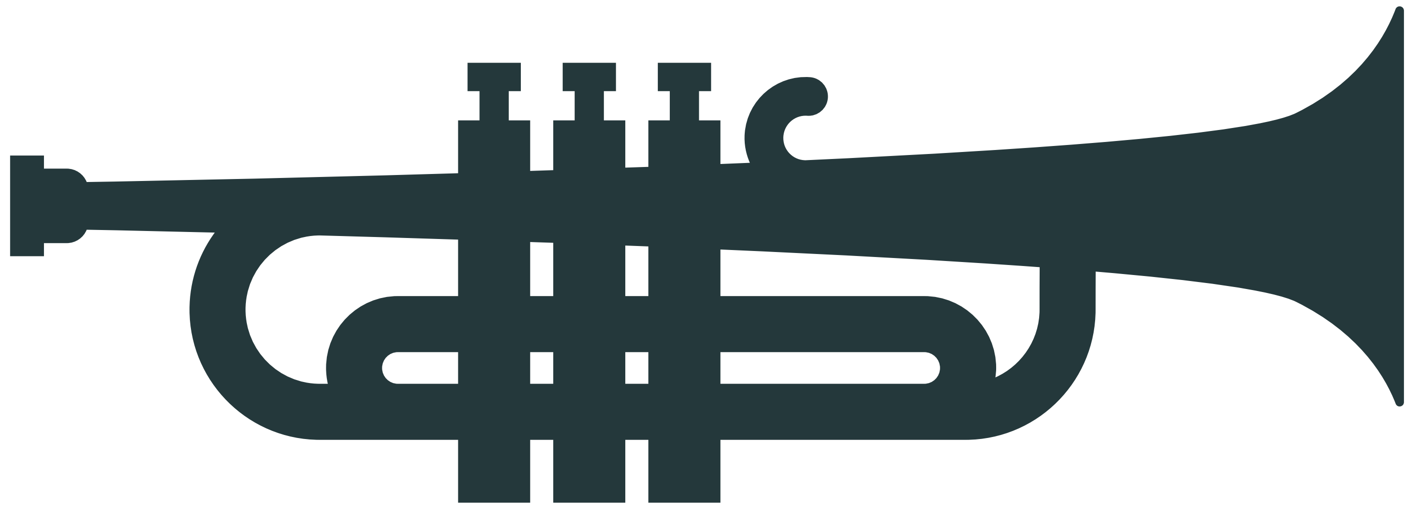
Later modality ▷

Iris Separation Logic

Shared mutable state

Invariants & Borrows

Our work, Nola



No later

Later modality ▷

Separation Logic

Shared mutable state

Invariants & Borrows

Sound

Our work, Nola

No later

Later modality $\triangleright$

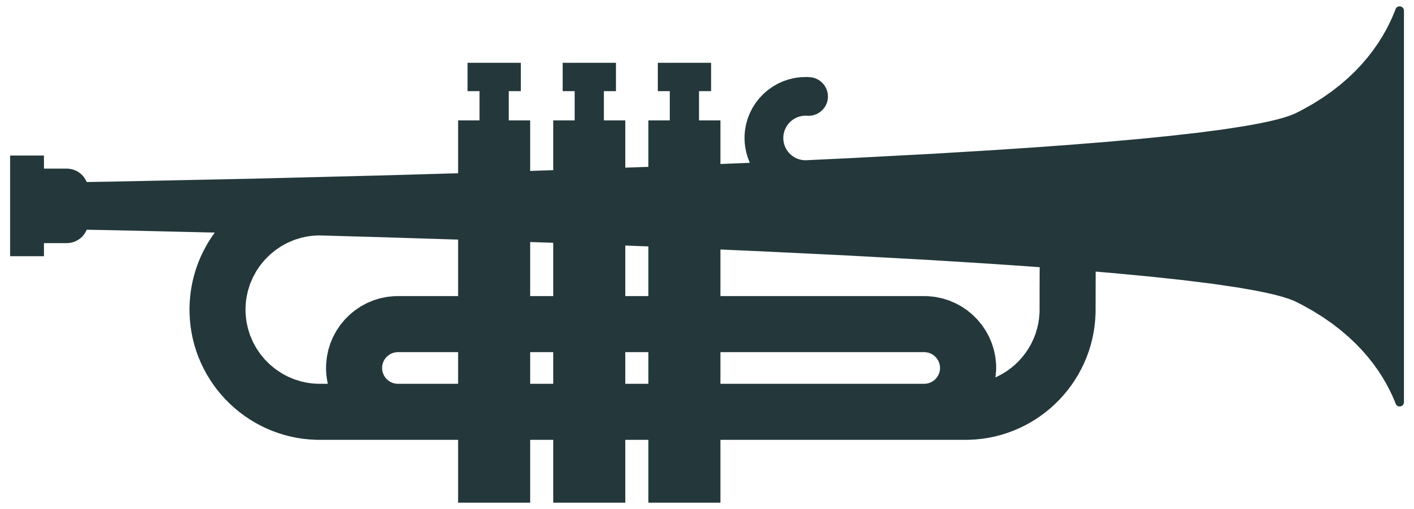
Iris **Separation Logic**

Shared mutable state

Invariants & Borrows

Sound Termination?

Our work, Nola



No later

~~Later modality ▷~~

Iris **Separation Logic**

Shared mutable state

Invariants & Borrows

Sound

Termination ✓

Our work, Nola

No later

~~Later modality~~ $\cancel{\triangleright}$

Iris Separation Logic

Shared mutable state

Invariants & Borrows

Sound

Termination ✓

Invariant $\boxed{P}$ Modern usage established by Iris [Jung+ '15]

Invariant $\boxed{P}$

Modern usage established by Iris [Jung+ '15]

- **Invariant** $\boxed{P}$: Roughly, the situation P always holds

Invariant $\boxed{P}$ Modern usage established by Iris [Jung+ '15]

- **Invariant** $\boxed{P}$: Roughly, the situation P always holds
  - **Shareable!** $\boxed{P} \Rightarrow \boxed{P} * \boxed{P}$ → **Shared mutable state**

Invariant $\boxed{P}$

Modern usage established by Iris [Jung+ '15]

- **Invariant** $\boxed{P}$: Roughly, the situation P always holds
  - **Shareable!** $\boxed{P} \Rightarrow \boxed{P} * \boxed{P}$ → **Shared mutable state**

Shared mutable ref $\quad \mathtt{r} : \mathtt{ref}\ \mathtt{T} \triangleq \boxed{\exists \mathtt{v}.\ \mathtt{r} \mapsto \mathtt{v} * \mathtt{v} : \mathtt{T}}$

Invariant $\boxed{P}$

Modern usage established by Iris [Jung+ '15]

- **Invariant** $\boxed{P}$: Roughly, the situation P always holds
  - **Shareable!** $\boxed{P} \Rightarrow \boxed{P} * \boxed{P}$ → **Shared mutable state**

Shared mutable ref $\mathtt{r} : \mathtt{ref}\ \mathtt{T} \triangleq \boxed{\exists \mathtt{v}.\ \mathtt{r} \mapsto \mathtt{v} * \mathtt{v} : \mathtt{T}}$

Can be nested! $\mathtt{r} : \mathtt{ref}\ (\mathtt{ref}\ \mathtt{T})$

Invariant $\boxed{P}$

Modern usage established by Iris [Jung+ '15]

- **Invariant** $\boxed{P}$: Roughly, the situation P always holds
  - **Shareable!** $\boxed{P} \Rightarrow \boxed{P} * \boxed{P}$ → **Shared mutable state**

Shared mutable ref $\mathtt{r} : \mathtt{ref}\ \mathtt{T} \triangleq \boxed{\exists \mathtt{v}.\ \mathtt{r} \mapsto \mathtt{v} * \mathtt{v} : \mathtt{T}}$

Can be nested! $\mathtt{r} : \mathtt{ref}\ (\mathtt{ref}\ \mathtt{T})$ $\quad P \in iProp$ ➜ $\boxed{P} \in iProp$

SL Prop

Invariant $\boxed{P}$

Modern usage established by Iris [Jung+ '15]

- **Invariant** $\boxed{P}$: Roughly, the situation P always holds
  - **Shareable!** $\boxed{P} \Rightarrow \boxed{P} * \boxed{P}$ → **Shared mutable state**

    Shared mutable ref $\mathtt{r} : \mathtt{ref}\ \mathtt{T} \triangleq \boxed{\exists \mathtt{v}.\ \mathtt{r} \mapsto \mathtt{v} * \mathtt{v} : \mathtt{T}}$

    Can be nested! $\mathtt{r} : \mathtt{ref}\ (\mathtt{ref}\ \mathtt{T})$ $P \in iProp$ ➜ $\boxed{P} \in iProp$

    SL Prop
  - Need it, as we are in **Separation Logic** [Ishtaq & O'Hearn '01]

Invariant $\boxed{P}$

Modern usage established by Iris [Jung+ '15]

- **Invariant** $\boxed{P}$: Roughly, the situation P always holds
  - **Shareable!** $\boxed{P} \Rightarrow \boxed{P} * \boxed{P}$ → **Shared mutable state**

    Shared mutable ref $\mathtt{r : ref\ T} \triangleq \boxed{\exists \mathtt{v}.\ \mathtt{r} \mapsto \mathtt{v} * \mathtt{v : T}}$

    Can be nested! $\mathtt{r : ref\ (ref\ T)}$ $\quad P \in iProp$ ➜ $\boxed{P} \in iProp$

    SL Prop
  - Need it, as we are in **Separation Logic** [Ishtaq & O'Hearn '01]

    Not shareable $\mathtt{r} \mapsto \mathtt{v} \not\Rightarrow \mathtt{r} \mapsto \mathtt{v} * \mathtt{r} \mapsto \mathtt{v}$

Invariant $\boxed{P}$

Modern usage established by Iris [Jung+ '15]

- **Invariant** $\boxed{P}$: Roughly, the situation P always holds
  - **Shareable!** $\boxed{P} \Rightarrow \boxed{P} * \boxed{P}$ → **Shared mutable state**

    Shared mutable ref $\mathtt{r} : \mathtt{ref}\ \mathtt{T} \triangleq \boxed{\exists \mathtt{v}.\ \mathtt{r} \mapsto \mathtt{v} * \mathtt{v} : \mathtt{T}}$

    Can be nested! $\mathtt{r} : \mathtt{ref}\ (\mathtt{ref}\ \mathtt{T})$ $\quad P \in iProp \Rightarrow \boxed{P} \in iProp$

    SL Prop
  - Need it, as we are in **Separation Logic** [Ishtaq & O'Hearn '01]

    Not shareable $\mathtt{r} \mapsto \mathtt{v} \not\Rightarrow \mathtt{r} \mapsto \mathtt{v} * \mathtt{r} \mapsto \mathtt{v}$

    $\because$ **State mutation** $[\mathtt{r} \mapsto \mathtt{v} * P]\ {*\mathtt{r}} = \mathtt{w}\ [\mathtt{r} \mapsto \mathtt{w} * P]$

Soundness & Later modality

Soundness & Later modality

- **Naive** invariant is **unsound**!

Soundness & Later modality

- **Naive** invariant is **unsound**!

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$

Soundness & Later modality

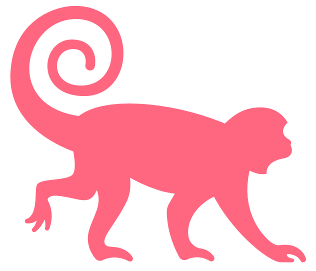
- **Naive** invariant is **unsound**!
  - Can't just store **any** $P \in iProp$

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$

Soundness & Later modality

- **Naive** invariant is **unsound**!
  - Can't just store **any** $P \in iProp$
    - **Paradoxes** [Krebbers+ '17] [M&T '25] $\approx$ **Landin's knot** 🤯

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$



Soundness & Later modality

- **Naive** invariant is **unsound**!

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$

  - Can't just store **any** $P \in iProp$
    - **Paradoxes** [Krebbers+ '17] [M&T '25] $\approx$ **Landin's knot** 🤯

- Known **fix**: Weaken by the **later** modality $\triangleright$ [Nakano '00]

Soundness & Later modality

- **Naive** invariant is **unsound**!

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$

  - Can't just store **any** $P \in iProp$
    - **Paradoxes** [Krebbers+ '17] [M&T '25] ≈ **Landin's knot** 🤯

- Known **fix:** Weaken by the **later** modality $\triangleright$ [Nakano '00]

$$\frac{[\triangleright P * Q]\ \mathsf{e}\ [\triangleright P * R]}{[\boxed{\triangleright P} * Q]\ \mathsf{e}\ [R]}$$

Iris

Soundness & Later modality

- **Naive invariant is unsound!**

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$

  - Can't just store **any** $P \in \mathit{iProp}$
    - **Paradoxes** [Krebbers+ '17] [M&T '25] ≈ **Landin's knot**

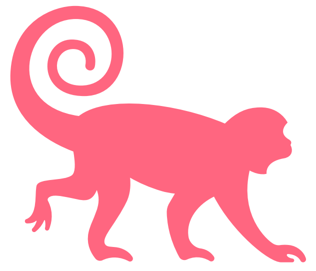
- Known **fix:** **Weaken by the later modality** $\triangleright$ [Nakano '00]
  - $\boxed{\triangleright P}$: Situation $\triangleright P$ always holds

$$\frac{[\triangleright P * Q]\ \mathsf{e}\ [\triangleright P * R]}{[\boxed{\triangleright P} * Q]\ \mathsf{e}\ [R]}$$

Soundness & Later modality

- **Naive invariant is unsound!**

$$\frac{[P * Q]\ e\ [P * R]}{[\boxed{P} * Q]\ e\ [R]}$$



  - Can't just store **any** $P \in iProp$
    - **Paradoxes** [Krebbers+ '17] [M&T '25] $\approx$ **Landin's knot**

- **Known fix: Weaken by the later modality** $\triangleright$ [Nakano '00]
  - $\boxed{\triangleright P}$: Situation $\triangleright P$ always holds

$$\frac{[\triangleright P * Q]\ e\ [\triangleright P * R]}{[\boxed{\triangleright P} * Q]\ e\ [R]}$$

Iris

Later weakens: $P \Rightarrow \triangleright P$

Soundness & Later modality

- **Naive invariant is unsound!**

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$

  - Can't just store **any** $P \in iProp$
    - **Paradoxes** [Krebbers+ '17] [M&T '25] $\approx$ **Landin's knot**

- Known **fix:** **Weaken by the later modality** $\triangleright$ [Nakano '00]

  - $\boxed{\triangleright P}$: Situation $\triangleright P$ always holds

$$\frac{[\triangleright P * Q]\ \mathsf{e}\ [\triangleright P * R]}{[\boxed{\triangleright P} * Q]\ \mathsf{e}\ [R]}$$

Iris

Later weakens: $P \Rightarrow \triangleright P$ **but** $\triangleright P \not\Rightarrow P$

Soundness & Later modality

- **Naive invariant is unsound!**

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$

  - Can't just store **any** $P \in iProp$
    - **Paradoxes** [Krebbers+ '17] [M&T '25] $\approx$ **Landin's knot** 🤯

- **Known fix: Weaken by the later modality** $\triangleright$ [Nakano '00]

  - $\boxed{\triangleright P}$: Situation $\triangleright P$ always holds

$$\frac{[\triangleright P * Q]\ \mathsf{e}\ [\triangleright P * R]}{[\boxed{\triangleright P} * Q]\ \mathsf{e}\ [R]}$$

Iris

Later weakens: $P \Rightarrow \triangleright P$ **but** $\triangleright P \not\Rightarrow P$ **even** $\triangleright\triangleright P \not\Rightarrow \triangleright P$

Soundness & Later modality

- **Naive invariant is unsound!**

$$\frac{[P * Q]\ \mathsf{e}\ [P * R]}{[\boxed{P} * Q]\ \mathsf{e}\ [R]}$$

  - Can't just store **any** $P \in iProp$
    - **Paradoxes** [Krebbers+ '17] [M&T '25] $\approx$ **Landin's knot**

- Known **fix:** **Weaken by the later modality** $\triangleright$ [Nakano '00]
  - $\boxed{\triangleright P}$: Situation $\triangleright P$ always holds
  - **Sound!** But **weaker…**

$$\frac{[\triangleright P * Q]\ \mathsf{e}\ [\triangleright P * R]}{[\boxed{\triangleright P} * Q]\ \mathsf{e}\ [R]}$$

Later weakens: $P \Rightarrow \triangleright P$ **but** $\triangleright P \not\Rightarrow P$ **even** $\triangleright \triangleright P \not\Rightarrow \triangleright P$

Challenge: Termination & liveness verification

Challenge: Termination & liveness verification

- **Later ▷ blocks** verifying **termination** & **liveness**!

Challenge: Termination & liveness verification

- **Later ▷ blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

Challenge: Termination & liveness verification

- **Later $\triangleright$ blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

$$r : \mathbf{ref}\ T \ \triangleq\ \boxed{\exists v.\ r \mapsto v \ast v : T}$$

Challenge: Termination & liveness verification

- **Later ▷ blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

$$\mathtt{r} : \mathbf{ref}\ \mathtt{T} \triangleq \boxed{\triangleright \exists \mathtt{v}.\ \mathtt{r} \mapsto \mathtt{v} * \mathtt{v} : \mathtt{T}}$$

Challenge: Termination & liveness verification

- **Later** $\triangleright$ **blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

$$r : \mathtt{ref}\ T \ \triangleq\ \boxed{\triangleright\, \exists v.\, r \mapsto v \ast v : T}$$

$$\textbf{Total}\qquad [\, r : \mathtt{ref}\ T \,]\ {*r}\ [\, v.\ \triangleright\, v : T \,]$$

Challenge: Termination & liveness verification

- **Later** $\triangleright$ **blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

$$r : \mathtt{ref}\ T \triangleq \boxed{\triangleright\, \exists v.\, r \mapsto v \ast v : T}$$

Total $\left[r : \mathtt{ref}\ T \right] \ast r \left[v.\ \triangleright\, v : T \right]$

Blocks access

Challenge: Termination & liveness verification

- **Later ▷ blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

$$r : \mathtt{ref}\ T \ \triangleq\ \boxed{\triangleright \exists v.\ r \mapsto v \ast v : T}$$

Total $[\, r : \mathtt{ref}\ T \,]\ {*r}\ [\, v.\ \triangleright v : T \,]$ **Blocks access**

Partial $\{\, r : \mathtt{ref}\ T \,\}\ {*r}\ \{\, v.\ v : T \,\}$

Challenge: Termination & liveness verification

- **Later** $\triangleright$ **blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

$$r : \mathtt{ref}\ T \ \triangleq\ \boxed{\triangleright\, \exists v.\, r \mapsto v \ast v : T}$$

Total $\left[\, r : \mathtt{ref}\ T \,\right] \ast r \left[\, v.\ \triangleright\, v : T \,\right]$ Blocks access

Partial $\left\{\, r : \mathtt{ref}\ T \,\right\} \ast r \left\{\, v.\ v : T \,\right\}$ Step-indexing

Challenge: Termination & liveness verification

- **Later** $\triangleright$ **blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

$$r : \mathbf{ref}\ T \triangleq \boxed{\triangleright \exists v.\ r \mapsto v * v : T}$$

Total $\quad [\, r : \mathbf{ref}\ T \,] *r\ [\, v.\ \triangleright v : T \,]$ **Blocks access**

Partial $\quad \{\, r : \mathbf{ref}\ T \,\} *r\ \{\, v.\ v : T \,\}$ **Step-indexing**

Simuliris
Gäher+ '22

"However, Iris's use of **step-indexing** means that **Iris-**based approaches … do **not** support reasoning about **liveness** properties such as **termination** preservation."

Challenge: Termination & liveness verification

- **Later ▹ blocks** verifying **termination** & **liveness**!
  - **Shared mutable refs** cannot be rightly **dereferenced**

?

$$r : \mathbf{ref}\ T \triangleq \boxed{\triangleright \exists v.\ r \mapsto v * v : T}$$

Total $[\,r : \mathbf{ref}\ T\,]\ {*}r\ [\,v.\ \triangleright v : T\,]$ **Blocks access**

Partial $\{\,r : \mathbf{ref}\ T\,\}\ {*}r\ \{\,v.\ v : T\,\}$ **Step-indexing**

Simuliris
Gäher+ ’22

“However, Iris’s use of **step-indexing** means that **Iris-**based approaches … do **not** support reasoning about **liveness** properties such as **termination** preservation.”

Our work, Nola

Our work, Nola

Use **syntax**!

Our work, Nola

Use syntax! For **SL assertions** to share

Our work, Nola

Use syntax! For **SL assertions** to share

Syntax gives you **better control** of what can be **shared**, than the **later** modality $\triangleright$, to achieve **soundness**

Our work, Nola

Use syntax! For **SL assertions** to share

Syntax gives you **better control** of what can be **shared**, than the **later** modality $\triangleright$, to achieve **soundness**

$$\frac{[\llbracket P \rrbracket * Q]\ e\ [\llbracket P \rrbracket * R]^{\mathrm{Winv}\,\llbracket\ \rrbracket}}{[\boxed{P} * Q]\ e\ [R]^{\mathrm{Winv}\,\llbracket\ \rrbracket}}$$

Our work, Nola

Matsushita & Tsukada PLDI '25

Use syntax! For SL assertions to share

Syntax gives you **better control** of what can be **shared**, than the **later** modality $\triangleright$, to achieve **soundness**

$$\frac{[\llbracket P \rrbracket * Q]\ \mathsf{e}\ [\llbracket P \rrbracket * R]^{\mathrm{Winv}\,\llbracket\ \rrbracket}}{[\boxed{P} * Q]\ \mathsf{e}\ [R]^{\mathrm{Winv}\,\llbracket\ \rrbracket}}$$

$P \in \mathit{Fml}$ **Syntactic SL formula**

Our work, Nola

Use syntax! For **SL assertions** to share

Syntax gives you **better control** of what can be **shared**, than the **later** modality $\triangleright$, to achieve **soundness**

$$\frac{[\llbracket P \rrbracket * Q]\ \mathsf{e}\ [\llbracket P \rrbracket * R]^{\mathrm{Winv}\,\llbracket\ \rrbracket}}{[\boxed{P} * Q]\ \mathsf{e}\ [R]^{\mathrm{Winv}\,\llbracket\ \rrbracket}}$$

$P \in \mathit{Fml}$ **Syntactic SL formula**

$\llbracket\ \rrbracket : \mathit{Fml} \to \mathit{iProp}$ **Semantics**

Our work, Nola

Use syntax! For **SL assertions** to share

Syntax gives you **better control** of what can be **shared**, than the **later** modality $\triangleright$, to achieve **soundness**

$$\frac{[\,[\![P]\!] * Q\,]\ \mathsf{e}\ [\,[\![P]\!] * R\,]^{\mathrm{Winv}\,[\![\ ]\!]}}{[\,\boxed{P} * Q\,]\ \mathsf{e}\ [\,R\,]^{\mathrm{Winv}\,[\![\ ]\!]}}$$

$P \in \mathit{Fml}$ **Syntactic SL formula**

$[\![\ ]\!] : \mathit{Fml} \longrightarrow \mathit{iProp}$ **Semantics**

Proofs can be written **semantically** with iProp in Iris !

Nola's syntax clears the later modality

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

$$\mathit{Fml} \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid P \mathrel{-\!\!*} Q \mid P \lor Q \mid \forall_A \Phi \mid \exists_A \Phi$$
$$\mid \phi \quad (\phi \in \mathit{Prop}) \mid \mathsf{r} \mapsto \mathsf{v} \mid \boxed{P} \quad (P \in_{\nu} \mathit{Fml})$$

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

$$\mathit{Fml} \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid P \mathrel{-\!\!*} Q \mid P \vee Q \mid \forall_A \Phi \mid \exists_A \Phi$$
$$\mid \phi \quad (\phi \in \mathit{Prop}) \mid \mathsf{r} \mapsto \mathsf{v} \mid \boxed{P} \quad (P \in_\nu \mathit{Fml})$$

Invariant

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

$$Fml \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid P -\!\!* \, Q \mid P \vee Q \mid \forall_A \Phi \mid \exists_A \Phi$$
$$\mid \phi \quad (\phi \in Prop) \mid \mathsf{r} \mapsto \mathsf{v} \mid \boxed{P} \quad (P \in_\nu Fml)$$

Invariant

$$[\![\ ]\!] : Fml \longrightarrow iProp$$

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

$$Fml \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid P \mathrel{-\!\!*} Q \mid P \vee Q \mid \forall_A \Phi \mid \exists_A \Phi$$
$$\mid \phi \quad (\phi \in Prop) \mid \mathsf{r} \mapsto \mathsf{v} \mid \boxed{P} \quad (P \in_{\nu} Fml)$$

Invariant

$$[\![\ ]\!]: Fml \longrightarrow iProp$$

$$[\![P * Q]\!] \triangleq [\![P]\!] * [\![Q]\!]$$

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

$$Fml \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid P \mathrel{-\!\!*} Q \mid P \lor Q \mid \forall_A \Phi \mid \exists_A \Phi$$
$$\mid \phi \quad (\phi \in Prop) \mid \mathsf{r} \mapsto \mathsf{v} \mid \boxed{P} \quad (P \in_\nu Fml)$$

Invariant

$$[\![\ ]\!] : Fml \to iProp$$

$$[\![P * Q]\!] \triangleq [\![P]\!] * [\![Q]\!] \qquad [\![\boxed{P}]\!] \triangleq \boxed{P}$$

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

$$\mathit{Fml} \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid P \mathrel{-\!\!*} Q \mid P \lor Q \mid \forall_A \Phi \mid \exists_A \Phi$$

$$\mid \phi \ \ (\phi \in \mathit{Prop}) \mid \mathsf{r} \mapsto \mathsf{v} \mid \boxed{P} \ \ (P \in_\nu \mathit{Fml})$$

Invariant

$$[\![\ ]\!] : \mathit{Fml} \to \mathit{iProp}$$

$$[\![P * Q]\!] \triangleq [\![P]\!] * [\![Q]\!] \qquad [\![\boxed{P}]\!] \triangleq \boxed{P}$$

- Now **later-free!**

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

$$\mathit{Fml} \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid P -\!\!* Q \mid P \vee Q \mid \forall_A \Phi \mid \exists_A \Phi$$
$$\mid \phi \quad (\phi \in \mathit{Prop}) \mid \mathsf{r} \mapsto \mathsf{v} \mid \boxed{P} \quad (P \in_\nu \mathit{Fml})$$

Invariant

$$[\![\ ]\!] : \mathit{Fml} \to \mathit{iProp}$$

$$[\![P * Q]\!] \triangleq [\![P]\!] * [\![Q]\!] \qquad [\![\boxed{P}]\!] \triangleq \boxed{P}$$

- **Now later-free!**

$$\mathsf{r} : \mathbf{ref}\ \mathsf{T} \triangleq \boxed{\exists \mathsf{v}.\ \mathsf{r} \mapsto \mathsf{v} * \mathsf{v} : \mathsf{T}}$$

Nola's syntax clears the later modality

- Just build the **syntax** for SL formulas & its **semantics**

$$\mathit{Fml} \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid P \mathrel{-\!\!*} Q \mid P \lor Q \mid \forall_A \Phi \mid \exists_A \Phi$$
$$\mid \phi \quad (\phi \in \mathit{Prop}) \mid \mathtt{r} \mapsto \mathtt{v} \mid \boxed{P} \quad (P \in_\nu \mathit{Fml})$$

Invariant

$$[\![\ ]\!] : \mathit{Fml} \to \mathit{iProp}$$

$$[\![P * Q]\!] \triangleq [\![P]\!] * [\![Q]\!] \qquad [\![\boxed{P}]\!] \triangleq \boxed{P}$$

- **Now later-free!**

$$\mathtt{r} : \mathbf{ref}\ \mathtt{T} \triangleq \boxed{\exists\, \mathtt{v}.\ \mathtt{r} \mapsto \mathtt{v} * \mathtt{v} : \mathtt{T}}$$

Later-free access! $\left[[\![\mathtt{r} : \mathbf{ref}\ \mathtt{T}]\!] \right] *\mathtt{r} \left[\mathtt{v}.\ [\![\mathtt{v} : \mathtt{T}]\!] \right]$

Verification example: Infinite shared mutable list

Verification example: Infinite shared mutable list

- **No later** remains if we use **Nola**

$$\mathtt{list}\ \Phi\ r \triangleq \boxed{\Phi\ r} * \boxed{\exists s.\ r{+}1 \mapsto s * \mathtt{list}\ \Phi\ s}$$

$$\left[\ [\![\mathtt{list}\ \Phi\ r]\!]\ \right] *(r{+}1) \left[\ s.\ [\![\mathtt{list}\ \Phi\ s]\!]\ \right]^{\mathrm{Winv}\,[\![\ ]\!]}$$

Iris

$$\mathrm{llist}\ \Phi\ r \triangleq \boxed{\triangleright\,\Phi\ r} * \boxed{\triangleright\,\exists s.\ r{+}1 \mapsto s * \mathrm{llist}\ \Phi\ s}$$

$$\left[\ \mathrm{llist}\ \Phi\ r\ \right] *(r{+}1) \left[\ s.\ \triangleright\,\mathrm{llist}\ \Phi\ s\ \right]$$

Verification example: Infinite shared mutable list

- **No later** remains if we use **Nola**

Iris

$$\mathtt{list}\ \Phi\ \mathtt{r} \triangleq \boxed{\Phi\ \mathtt{r}} * \boxed{\exists \mathtt{s}.\ \mathtt{r}+1 \mapsto \mathtt{s} * \mathtt{list}\ \Phi\ \mathtt{s}}$$

$$\big[\, [\![\mathtt{list}\ \Phi\ \mathtt{r}]\!] \,\big]\ {*}(\mathtt{r}+1)\ \big[\, \mathtt{s}.\ [\![\mathtt{list}\ \Phi\ \mathtt{s}]\!] \,\big]^{\mathrm{Winv}\,[\![\ ]\!]}$$

$$\mathrm{llist}\ \Phi\ \mathtt{r} \triangleq \boxed{\triangleright\, \Phi\ \mathtt{r}} * \boxed{\triangleright\, \exists \mathtt{s}.\ \mathtt{r}+1 \mapsto \mathtt{s} * \mathrm{llist}\ \Phi\ \mathtt{s}}$$

$$\big[\, \mathrm{llist}\ \Phi\ \mathtt{r} \,\big]\ {*}(\mathtt{r}+1)\ \big[\, \mathtt{s}.\ \triangleright\, \mathrm{llist}\ \Phi\ \mathtt{s} \,\big]$$

- **Can verify termination of iteration naturally!**

```
fn iterc(f,c,r) { if *c > 0 {
    f(r); *c = *c − 1;
    iterc(f,c,*(r+1)) } }
```

$$\frac{\forall \mathtt{r}.\ \big[\, \boxed{\Phi\ \mathtt{r}} \,\big]\ \mathtt{f}(\mathtt{r})\ \big[\, \top \,\big]^{\mathrm{Winv}\,[\![\ ]\!]}}{\big[\, [\![\mathtt{list}\ \Phi\ \mathtt{r}]\!] * \mathtt{c} \mapsto n \,\big]\ \mathtt{iterc}(\mathtt{f},\mathtt{c},\mathtt{r})\ \big[\, \mathtt{c} \mapsto 0 \,\big]^{\mathrm{Winv}\,[\![\ ]\!]}}$$

Nola’s soundness & expressivity

Nola's soundness & expressivity

- **Well-definedness** of $[\![\ ]\!]$ is the key to **soundness**

Nola's soundness & expressivity

- **Well-definedness** of $[\![\ ]\!]$ is the key to **soundness**
  - No **Landin's knot paradoxes**!

Nola's soundness & expressivity

- **Well-definedness of** $[\![\ ]\!]$ is the key to **soundness**
  - No **Landin's knot paradoxes**!

$$[\![[\top]\, e\, [\top]]\!] \triangleq_? [\top]\, e\, [\top]^{\mathrm{Winv}\, [\![\ ]\!]}$$

Nola's soundness & expressivity

- **Well-definedness of $[\![\ ]\!]$ is the key to soundness**
  - No **Landin's knot paradoxes**!

$$[\![\,[\top]\ \mathsf{e}\ [\top]\,]\!] \triangleq_{?} [\top]\ \mathsf{e}\ [\top]^{\mathrm{Winv}\,[\![\ ]\!]}$$

Bad self-reference

Nola's soundness & expressivity

- **Well-definedness** of $[\![\ ]\!]$ is the key to **soundness**
  - No **Landin's knot paradoxes**!

$$[\![\, [\top]\ \mathsf{e}\ [\top] \,]\!] \triangleq_? [\top]\ \mathsf{e}\ [\top]^{\mathrm{Winv}\,[\![\ ]\!]}$$

Bad self-reference

- Allows **flexible** construction for **extra expressivity**

Nola's soundness & expressivity

- **Well-definedness** of $[\![\ ]\!]$ is the key to **soundness**
  - No **Landin's knot paradoxes**!

$$[\![\, [\top]\ \mathsf{e}\ [\top] \,]\!] \triangleq_{?} [\top]\ \mathsf{e}\ [\top]^{\mathrm{Winv}\,[\![\ ]\!]}$$

Bad self-reference

- Allows **flexible** construction for **extra expressivity**
  - **Later-weakening** $[\![\, \triangleright [\top]\ \mathsf{e}\ [\top] \,]\!] \triangleq \triangleright [\top]\ \mathsf{e}\ [\top]^{\mathrm{Winv}\,[\![\ ]\!]}$

Nola's soundness & expressivity

- **Well-definedness of** $[\![\ ]\!]$ is the key to **soundness**
  - No **Landin's knot paradoxes**!

$$[\![\, [\top]\ \mathsf{e}\ [\top] \,]\!] \triangleq_{?} \left[\top\right]\ \mathsf{e}\ \left[\top\right]^{\mathrm{Winv}\,[\![\ ]\!]}$$

Bad self-reference

- Allows **flexible** construction for **extra expressivity**
  - **Later-weakening** $[\![\, \triangleright [\top]\ \mathsf{e}\ [\top] \,]\!] \triangleq \triangleright \left[\top\right]\ \mathsf{e}\ \left[\top\right]^{\mathrm{Winv}\,[\![\ ]\!]}$
  - **Stratification** $[\![\ ]\!]_i : \mathit{Fml}_i \to \mathit{iProp}$ $\quad [\![\, [P]\ \mathsf{e}\ [Q] \,]\!]_1 \triangleq \left[[\![P]\!]_1\right]\ \mathsf{e}\ \left[[\![Q]\!]_1\right]^{\mathrm{Winv}\,[\![\ ]\!]_0}$

To experts: Power of SL formulas

To experts: Power of SL formulas

- **Any semantic** SL props can be stored under **later**
  - Precisely **subsuming** the **existing** later-weakened approach

$$\mathit{Fml} \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid \cdots \mid \boxed{P} \quad (P \in_\nu \mathit{Fml})$$
$$\mid \check{\triangleright} \hat{P} \quad (\hat{P} \in \blacktriangleright \mathit{iProp})$$

$$[\![\check{\triangleright} \hat{P}]\!] \triangleq \check{\triangleright} \hat{P} \qquad \triangleright P \triangleq \check{\triangleright}\, \mathrm{next}\, P \qquad \check{\triangleright}\, \mathrm{next}\, P \triangleq \triangleright P$$

To experts: Power of SL formulas

- Any **semantic** SL props can be stored under **later**
  - Precisely **subsuming** the **existing** later-weakened approach

$$Fml \ni P, Q \quad ::=_{\nu,\mu} \quad P * Q \mid \cdots \mid \boxed{P} \quad (P \in_\nu Fml)$$
$$\mid \check{\triangleright} \hat{P} \quad (\hat{P} \in \blacktriangleright iProp)$$

$$[\![\check{\triangleright} \hat{P}]\!] \triangleq \check{\triangleright} \hat{P} \qquad \triangleright P \triangleq \check{\triangleright}\, \mathrm{next}\, P \qquad \check{\triangleright}\, \mathrm{next}\, P \triangleq \triangleright P$$

- The set of SL formulas can even be **extensible**
  - By **parameterizing** over the constructors, just like **iProp's Σ**

To experts: Nola's model just generalizes Iris's

To experts: Nola's model just generalizes Iris's

- **Nola's model for the invariant generalizes Iris's**
  - Fml for $\blacktriangleright \mathit{iProp}$, $[\![\ ]\!] : \mathit{Fml} \to \mathit{iProp}$ for $\check{\triangleright} : \blacktriangleright \mathit{iProp} \to \mathit{iProp}$

$$\textsc{Inv}\ \mathit{Fml} \ \triangleq\ \textsc{Auth}\left(\mathbb{N} \xrightarrow{\text{fin}} \textsc{Ag}\ \mathit{Fml}\right)$$

$$\boxed{P} \ \triangleq\ \exists \iota.\ \left[\circ\, [\iota \leftarrow \mathrm{ag}\ P]\right]^{\gamma_{\textsc{Inv}}}$$

$$\mathrm{Winv}\ [\![\ ]\!] \ \triangleq\ \exists\, I : \mathbb{N} \xrightarrow{\text{fin}} \mathit{Fml}.\ \left[\bullet\, \mathrm{ag}\ I\right]^{\gamma_{\textsc{Inv}}} * \mathop{\huge\ast}_{\iota \in \mathrm{dom}\, I} \left(\left([\![I\ \iota]\!] * [\mathrm{D}]_\iota\right) \vee [\mathrm{E}]_\iota\right)$$

$$\mathrm{l}\textsc{Inv} \ \triangleq\ \textsc{Auth}\left(\mathbb{N} \xrightarrow{\text{fin}} \textsc{Ag}\ (\blacktriangleright \mathit{iProp})\right)$$

$$\boxed{\triangleright P} \ \triangleq\ \exists \iota.\ \left[\circ\, [\iota \leftarrow \mathrm{ag}\,(\mathrm{next}\ P)]\right]^{\gamma_{\mathrm{l}\textsc{Inv}}}$$

$$\mathrm{Wlinv} \ \triangleq\ \exists\, \hat{I} : \mathbb{N} \xrightarrow{\text{fin}} \blacktriangleright \mathit{iProp}.\ \left[\bullet\, \mathrm{ag}\ \hat{I}\right]^{\gamma_{\mathrm{l}\textsc{Inv}}} * \mathop{\huge\ast}_{\iota \in \mathrm{dom}\, \hat{I}} \left(\left(\check{\triangleright}\, \hat{I}\ \iota * [\mathrm{D}]_\iota\right) \vee [\mathrm{E}]_\iota\right)$$

Rust-style borrows

Rust-style borrows

- RustBelt's **lifetime** logic [Jung+ '18], but **later-free**

Rust-style borrows

- RustBelt's **lifetime** logic [Jung+ '18], but **later-free**

Rust's borrow

```
let mut l = 0;
let b = &mut l;
*b += 7;
print(l);
```

Rust-style borrows

- RustBelt's **lifetime** logic [Jung+ '18], but **later-free**

Rust's borrow

```
let mut l = 0;
let b = &mut l;
*b += 7;
print(l);
```

α (lifetime of `let b = &mut l;` and `*b += 7;`)

Reasoning in SL

$\exists n.\ \mathtt{l} \mapsto n \quad \mathtt{l} : \mathtt{int}$

$\triangle^{\alpha}\ (\exists n.\ \mathtt{l} \mapsto n)$

$\mathtt{l} :^{\dagger\alpha} \mathtt{int}$

$\exists n.\ \mathtt{l} \mapsto n \quad \mathtt{l} : \mathtt{int}$

$[\alpha]_1$

$\dagger\alpha$

$[\alpha]_{1-q}$

$\&^{\alpha}\ (\exists n.\ \mathtt{l} \mapsto n)$

$\mathtt{b} : \&\alpha\ \mathtt{mut\ int}$

$\rtimes^{\alpha}_{q}\ (\exists n.\ \mathtt{l} \mapsto n)$

$\exists n.\ \mathtt{l} \mapsto n$

Rust-style borrows

- RustBelt's **lifetime** logic [Jung+ '18], but **later-free**

Rust's borrow

$$\alpha\left\{\begin{array}{l}\texttt{let b = \&mut l;}\\ \texttt{*b += 7;}\end{array}\right.$$

```
let mut l = 0;
let b = &mut l;
*b += 7;
print(l);
```

Reasoning in SL

$\exists n.\ \mathtt{l} \mapsto n \quad \mathtt{l} : \mathtt{int}$

$\triangle^\alpha (\exists n.\ \mathtt{l} \mapsto n)$

$\mathtt{l} :^{\dagger\alpha} \mathtt{int}$

$\exists n.\ \mathtt{l} \mapsto n \quad \mathtt{l} : \mathtt{int}$

$[\alpha]_1$

$\dagger\alpha$

$[\alpha]_{1-q}$

$\&^\alpha (\exists n.\ \mathtt{l} \mapsto n)$

$\mathtt{b} : \&\alpha\ \mathtt{mut}\ \mathtt{int}$

$⅋^\alpha_q (\exists n.\ \mathtt{l} \mapsto n)$

$\exists n.\ \mathtt{l} \mapsto n$

$$[\![P]\!] \Rrightarrow^{\mathrm{Wbor}[\![\ ]\!]} \&^\alpha P * \triangle^\alpha P \qquad \dagger\alpha * \triangle^\alpha P \Rrightarrow^{\mathrm{Wbor}[\![\ ]\!]} [\![P]\!]$$

$$\&^\alpha P * [\alpha]_q \Rrightarrow^{\mathrm{Wbor}[\![\ ]\!]} ⅋^\alpha_q P * [\![P]\!] \qquad ⅋^\alpha_q P * [\![P]\!] \Rrightarrow^{\mathrm{Wbor}[\![\ ]\!]} \&^\alpha P * [\alpha]_q$$

Case study: RustHalt

Case study: RustHalt

- **Semantic** foundation for verifying **Rust termination**

Case study: RustHalt

- **Semantic** foundation for verifying **Rust termination**

Example `fn iter(f,l) { match l { Nil => (), Cons(a,l′) => { f(a); iter(f,*l′) } } }`

$$\frac{\forall \mathtt{a}.\ \ \mathtt{a} : \&\alpha\ \mathbf{mut}\ \mathtt{T} \vdash \mathtt{f(a)} \dashv \_.\ \rightsquigarrow \lambda\psi, [(a, a')].\ a' = f\ a \to \psi\ []}{\mathtt{l} : \&\alpha\ \mathbf{mut}\ \mathtt{List<T>} \vdash \mathtt{iter(f,l)} \dashv \_.\ \rightsquigarrow \lambda\psi, [(l, l')].\ l' = \mathrm{map}\ f\ l \to \psi\ []}$$

Case study: RustHalt

- **Semantic foundation for verifying Rust termination**
  - Refines **RustHornBelt** [M+ '22] with **Nola**

Example `fn iter(f,l) { match l { Nil ⇒ (), Cons(a,l′) ⇒ { f(a); iter(f,*l′) } } }`

$$\frac{\forall \mathtt{a}.\ \ \mathtt{a} : \&\alpha\ \mathbf{mut}\ \mathtt{T} \vdash \mathtt{f(a)} \dashv \_.\ \rightsquigarrow \lambda\psi, [(a, a')].\ a' = f\,a \to \psi\,[]}{\mathtt{l} : \&\alpha\ \mathbf{mut}\ \mathtt{List<T>} \vdash \mathtt{iter(f,l)} \dashv \_.\ \rightsquigarrow \lambda\psi, [(l, l')].\ l' = \mathrm{map}\,f\,l \to \psi\,[]}$$

Case study: RustHalt

- **Semantic** foundation for verifying **Rust termination**
  - Refines **RustHornBelt** [M+ '22] with **Nola**
  - **Semantic** typing / **logical** relation that enjoys **extensibility**

Example `fn iter(f,l) { match l { Nil ⇒ (), Cons(a,l′) ⇒ { f(a); iter(f,*l′) } } }`

$$\frac{\forall \texttt{a}.\ \ \texttt{a} : \&\alpha\,\textbf{mut}\,\texttt{T} \vdash \texttt{f(a)} \dashv \_.\ \rightsquigarrow \lambda\psi, [(a, a')].\ a' = f\,a \to \psi\,[]}{\texttt{l} : \&\alpha\,\textbf{mut}\,\texttt{List<T>} \vdash \texttt{iter(f,l)} \dashv \_.\ \rightsquigarrow \lambda\psi, [(l, l')].\ l' = \mathrm{map}\,f\,l \to \psi\,[]}$$

Case study: RustHalt

- **Semantic** foundation for verifying **Rust termination**
  - Refines **RustHornBelt** [M+ '22] with **Nola**
  - **Semantic** typing / **logical** relation that enjoys **extensibility**

Example `fn iter(f,l) { match l { Nil => (), Cons(a,l′) => { f(a); iter(f,*l′) } } }`

$$\frac{\forall \mathtt{a}.\ \ \mathtt{a} : \&\alpha\ \mathbf{mut}\ \mathtt{T} \vdash \mathtt{f(a)} \dashv \_.\ \rightsquigarrow \lambda\psi, [(a, a')].\ a' = f\ a \to \psi\ []}{\mathtt{l} : \&\alpha\ \mathbf{mut}\ \mathtt{List\langle T\rangle} \vdash \mathtt{iter(f,l)} \dashv \_.\ \rightsquigarrow \lambda\psi, [(l, l')].\ l' = \mathrm{map}\ f\ l \to \psi\ []}$$

Semantics!

$$[\![\Gamma \vdash_\alpha \mathtt{e} \dashv \mathtt{r}.\Gamma' \rightsquigarrow \mathit{pre}]\!] \ \triangleq\ \forall \hat{\psi}, t, q.\ \Big[\ \exists \bar{\hat{a}}.\ \big\langle \lambda\pi.\ \mathit{pre}\,(\hat{\psi}\,\pi)\,(\overline{\hat{a}\,\pi})\big\rangle * [\alpha]_q * [t] * [\![\Gamma]\!](\bar{\hat{a}}, t)\ \Big]$$

$$\mathtt{e}\ \Big[\ \lambda\mathtt{r}.\ \exists \bar{\hat{b}}.\ \big\langle \lambda\pi.\ \hat{\psi}\,\pi\,(\overline{\hat{b}\,\pi})\big\rangle * [\alpha]_q * [t] * [\![\Gamma']\!](\bar{\hat{b}}, t)\ \Big]^{\mathrm{Wrh}\,[\![\ ]\!]}$$

Recent application: Lilo Lee+ OOPSLA ’25

Recent application: Lilo Lee+ OOPSLA '25

- **Fair liveness** verification with **Nola**-style **invariants**

Recent application: Lilo Lee+ OOPSLA '25

- **Fair liveness** verification with **Nola**-style **invariants**
  - **Stratification** for **higher-order** features

Recent application: Lilo Lee+ OOPSLA '25

- **Fair liveness** verification with **Nola**-style **invariants**
  - **Stratification** for **higher-order** features

Example

$$\begin{array}{l}\texttt{while}\ (1)\ \{\ \mathcal{Y};\ X \coloneqq 1; \\ \quad \texttt{do}\ \{\ \mathcal{Y};\ a \coloneqq X;\ \}\ \texttt{while}\ (a = 1);\ \mathcal{Y};\ \mathit{print}(a);\ \}\end{array} \;\Bigg\|\; \begin{array}{l}\texttt{while}\ (1)\ \{\ \mathcal{Y};\ X \coloneqq 2; \\ \quad \texttt{do}\ \{\ \mathcal{Y};\ b \coloneqq X;\ \}\ \texttt{while}\ (b = 2);\ \mathcal{Y};\ \mathit{print}(b);\ \}\end{array}$$

refines

$$\texttt{while}\ (1)\ \{\ \mathcal{Y};\ \mathit{print}(2);\ \} \;\Big\|\; \texttt{while}\ (1)\ \{\ \mathcal{Y};\ \mathit{print}(1);\ \}$$

preserving **termination** under scheduler **fairness**

To experts: Magic derivability, our finding

To experts: Magic derivability, our finding

- **Semantic** alteration of SL formulas for the body
  - **Goal**: Prove **subtyping** on shared mutable refs **semantically**

Goal $$\frac{\mathsf{T} \le \mathsf{U} \qquad \mathsf{U} \le \mathsf{T}}{\texttt{ref}\ \mathsf{T} \le \texttt{ref}\ \mathsf{U}}$$

Need something like $$\frac{[\![P]\!] \Leftrightarrow [\![Q]\!]}{[\![\boxed{P}]\!] \Leftrightarrow [\![\boxed{Q}]\!]}$$

To experts: Magic derivability, our finding

- **Semantic** alteration of SL formulas for the body
  - **Goal**: Prove **subtyping** on shared mutable refs **semantically**

Goal $\dfrac{\mathsf{T} \le \mathsf{U} \qquad \mathsf{U} \le \mathsf{T}}{\texttt{ref}\ \mathsf{T} \le \texttt{ref}\ \mathsf{U}}$ **Need** something like $\dfrac{[\![P]\!] \Leftrightarrow [\![Q]\!]}{[\![\boxed{P}]\!] \Leftrightarrow [\![\boxed{Q}]\!]}$

- **Magic derivability** enables this by a kind of **fixpoint**

$$[\![\boxed{P}]\!]_\delta \triangleq \exists Q \text{ s.t. } \delta(P \Leftrightarrow Q).\ \boxed{Q}$$

$$\mathit{Judg} \ni J ::= P \Leftrightarrow Q$$

$$[\![P \Leftrightarrow Q]\!]^+_\delta \triangleq [\![P]\!]_\delta \Leftrightarrow [\![Q]\!]_\delta$$

$$\dfrac{\forall\, \delta \in \mathit{Deriv}.\ [\![P]\!]_\delta \Leftrightarrow [\![Q]\!]_\delta}{\forall\, \delta \in \mathit{Deriv}.\ [\![\boxed{P}]\!]_\delta \Leftrightarrow [\![\boxed{Q}]\!]_\delta}$$

$$\mathrm{der}\ J \Rightarrow [\![J]\!]^+_{\mathrm{der}} \qquad \mathrm{der} \in \mathit{Deriv}$$

Our work, Nola

Matsushita & Tsukada PLDI ’25

- **Sound later $\triangleright$-free shared mutable state**
  - Refine Iris’s **invariants** $\boxed{\triangleright P}$ & RustBelt’s **borrows** $\&^{\alpha} \triangleright P$
  - Great for **termination** & **liveness** verification
    - Case study: **RustHalt**, RustHornBelt revised for termination
  - **Fully mechanized** in **Rocq** as a **library** of Iris
- **Syntax P for SL formulas to share:** $\boxed{P}$ **&** $\&^{\alpha} P$
  - **Extensible** & **Semantic** SL props under later
  - **Magic derivability** for **semantic** alteration