Completeness of Iris-based Program Logics

Zichen Zhang[*]

with

Johannes Hostert[†], Puming Liu[‡], Simon Gregersen[§], Ralf Jung[†], and Joseph Tassarotti[*]

[*] New York University, [†] ETH Zurich, [‡] NYU Shanghai, [§] CISPA Helmholtz Center for Information Security

June 10, 2026

The Sixth Edition of the Iris Workshop Vienna, Austria

A Missing Piece in Iris

Logic

Truth

A Missing Piece in Iris

Logic

Truth

$\mathsf{safe}_{\varphi}(e, \sigma)$

Never gets stuck

Result satisfies φ

A Missing Piece in Iris

$\vdash \mathsf{wp}\ e\ \{v.\ \ulcorner\varphi(v)\urcorner\}$

Logic

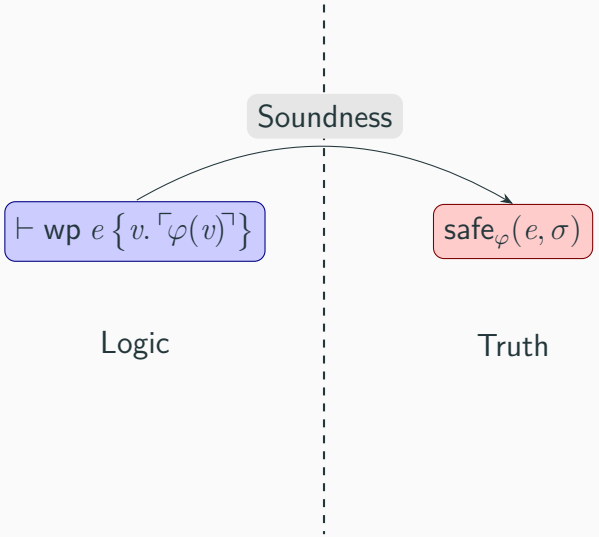
$\mathsf{safe}_{\varphi}(e, \sigma)$

Truth

Never gets stuck

Result satisfies φ

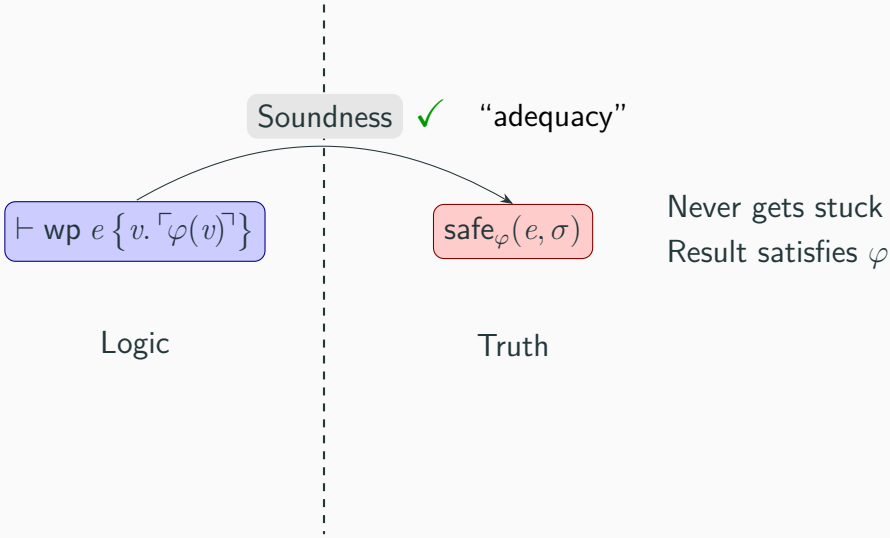
A Missing Piece in Iris



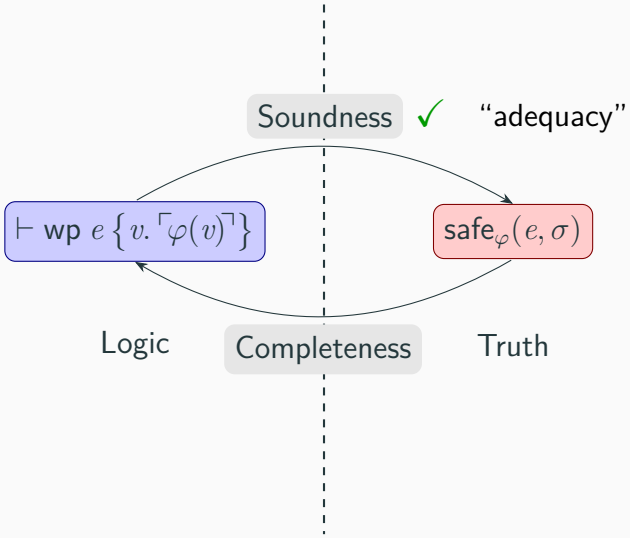
Never gets stuck

Result satisfies φ

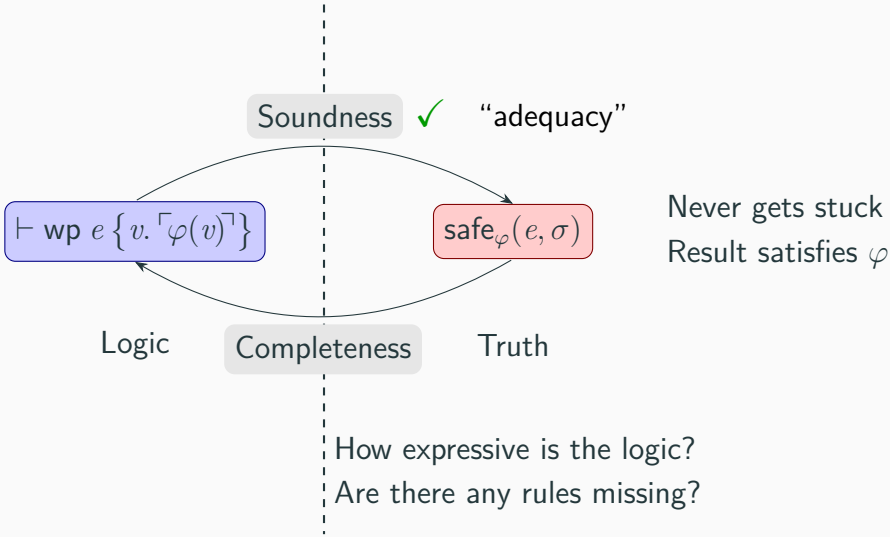
A Missing Piece in Iris



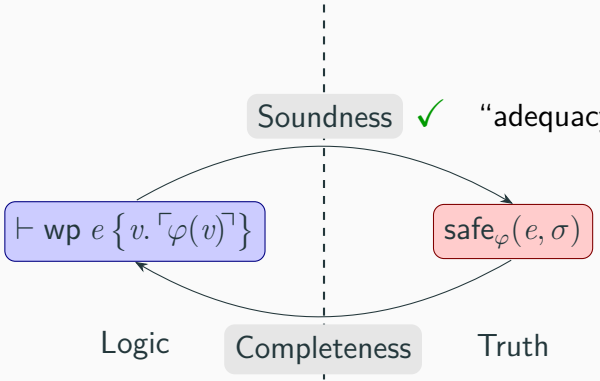
A Missing Piece in Iris



A Missing Piece in Iris



A Missing Piece in Iris

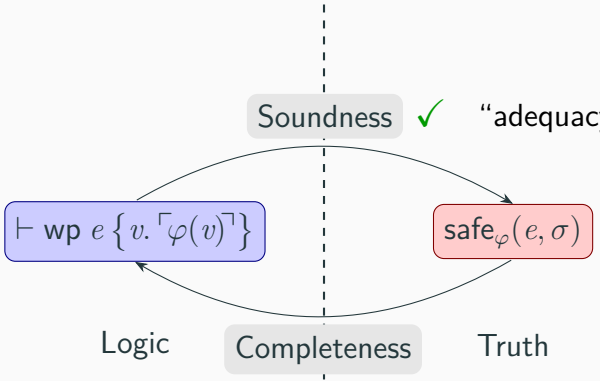


Prior work:

Cook (1978), Owicki (1975),
Jones (1983), Stirling (1988), etc.

How expressive is the logic?
Are there any rules missing?

A Missing Piece in Iris



Prior work:

Cook (1978), Owicki (1975),
Jones (1983), Stirling (1988), etc.

Iris: ?

How expressive is the logic?

Are there any rules missing?

Our Contributions

Part 1: completeness of whole program specifications

- Generic completeness theorems for any Iris-based program logics for
  - Partial correctness
  - Total correctness
  - Refinement
- Concrete languages:
  - HeapLang, λ_{Rust}, Time credits, Eris
  - **Probably your favorite program logics!**

Our Contributions

Part 1: completeness of whole program specifications

- Generic completeness theorems for any Iris-based program logics for
  - Partial correctness
  - Total correctness
  - Refinement
- Concrete languages:
  - HeapLang, λ_{Rust}, Time credits, Eris
  - **Probably your favorite program logics!**

Part 2 (WIP): completeness of library specifications

- Completeness of logically atomic triples in Iris

Whole Program Completeness

Johannes Hostert **Zichen Zhang** Puming Liu

Simon Gregersen Ralf Jung Joseph Tassarotti

Warm-up: Sequential Completeness

Language: $\rightarrow_{\mathsf{pure}} \mid \texttt{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without `fork`, `free`, and prophecy variables

Warm-up: Sequential Completeness

Language: $\to_{\mathsf{pure}} \mid \mathbf{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without **`fork`**, **`free`**, and prophecy variables

$$\forall e \quad .\ \mathsf{safe}_{\varphi}(e, \varnothing) \implies \qquad \vdash \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$$

Warm-up: Sequential Completeness

Language: $\rightarrow_{\mathsf{pure}} \mid \mathbf{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without **`fork`**, **`free`**, and prophecy variables

$$\forall e \quad . \ \mathsf{safe}_{\varphi}(e, \varnothing) \implies \qquad \vdash \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$$

Löb induction

Warm-up: Sequential Completeness

Language: $\rightarrow_{\mathsf{pure}} \mid \texttt{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without **`fork`**, **`free`**, and prophecy variables

$$\forall e, \sigma.\, \mathsf{safe}_{\varphi}(e, \sigma) \implies \qquad \vdash \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$$

Löb induction

Warm-up: Sequential Completeness

Language: $\rightarrow_{\mathsf{pure}} \mid \texttt{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without `fork`, `free`, and prophecy variables

$$\forall e, \sigma.\, \mathsf{safe}_{\varphi}(e, \sigma) \implies \mathsf{S}_{\circ}(\sigma) \vdash \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$$

$$\mathsf{S}_{\circ}(\sigma) \triangleq \mathop{\Large *}_{(\ell \leftarrow v) \in \sigma} (\ell \mapsto v) \qquad \text{(the ownership of } \sigma\text{)}$$

Löb induction

Warm-up: Sequential Completeness

Language: $\to_{\mathsf{pure}} \mid \texttt{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without `fork`, `free`, and prophecy variables

$$\forall e, \sigma.\, \mathsf{safe}_\varphi(e, \sigma) \implies \mathsf{S}_\circ(\sigma) \vdash \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$$

$$\mathsf{S}_\circ(\sigma) \triangleq \mathop{\Large *}_{(\ell \leftarrow v) \in \sigma} (\ell \mapsto v) \qquad \text{(the ownership of } \sigma\text{)}$$

Löb induction $\quad \triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\} \right)$

Warm-up: Sequential Completeness

Language: $\rightarrow_{\mathsf{pure}} \mid \mathbf{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without **fork**, **free**, and prophecy variables

$$\forall e, \sigma.\, \mathsf{safe}_\varphi(e, \sigma) \implies \mathsf{S}_\circ(\sigma) \vdash \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$$

$$\mathsf{S}_\circ(\sigma) \triangleq \mathop{\Large *}_{(\ell \leftarrow v) \in \sigma} (\ell \mapsto v) \qquad \text{(the ownership of } \sigma\text{)}$$

Löb induction $\quad \triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\} \right)$

- If $\mathsf{safe}_\varphi(v, \sigma)$ ($v \in \mathit{Val}$)
- If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$),

Warm-up: Sequential Completeness

Language: $\to_{\mathsf{pure}} \mid \mathtt{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without `fork`, `free`, and prophecy variables

$$\forall e, \sigma.\, \mathsf{safe}_\varphi(e, \sigma) \implies \mathsf{S}_\circ(\sigma) \vdash \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$$

$$\mathsf{S}_\circ(\sigma) \triangleq \mathop{\Large *}_{(\ell \leftarrow v) \in \sigma} (\ell \mapsto v) \qquad \text{(the ownership of } \sigma\text{)}$$

Löb induction $\quad \triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\} \right)$

- If $\mathsf{safe}_\varphi(v, \sigma)$ ($v \in \mathit{Val}$) $\implies \ulcorner \varphi(v) \urcorner \vdash \mathsf{wp}\ v \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$ (by WpValue)
- If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$),

Warm-up: Sequential Completeness

Language: $\to_{\mathsf{pure}} \mid \mathbf{ref}\ e \mid !\, e \mid e_1 \leftarrow e_2$

HeapLang, but without **`fork`**, **`free`**, and prophecy variables

$$\forall e, \sigma.\, \mathsf{safe}_\varphi(e, \sigma) \implies \mathsf{S}_\circ(\sigma) \vdash \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$$

$$\mathsf{S}_\circ(\sigma) \triangleq \mathop{\Large *}_{(\ell \leftarrow v) \in \sigma} (\ell \mapsto v) \qquad \text{(the ownership of } \sigma\text{)}$$

Löb induction $\quad \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right)$

- If $\mathsf{safe}_\varphi(v, \sigma)$ ($v \in \mathit{Val}$) $\implies \ulcorner \varphi(v) \urcorner \vdash \mathsf{wp}\ v \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$ (by WpValue)
- If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$), then e_1 is reducible

case analysis on the first step

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp $\quad$ `Hsafe1`: $\mathsf{safe}_\varphi(e_1, \sigma_1)$

Persistent hyp $\quad$ `IH`: $\triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$ □

Spatial hyp $\quad$ `H`$\circ$: $\mathsf{S}_\circ(\sigma_1)$

Goal $\quad$ $\mathsf{wp}\quad e_1\ \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$ ∗

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp — $\texttt{Hsafe1}: \mathsf{safe}_\varphi(e_1, \sigma_1)$

Persistent hyp — $\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp — $\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal — $\mathsf{wp}\ \ e_1\ \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp — $\texttt{Hsafe1}: \mathsf{safe}_\varphi(e_1, \sigma_1)$

$\texttt{Hprim}: (e_1, \sigma_1) \rightarrow_{\mathsf{prim}} (e_2, \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(e_2, \sigma_2)$

Persistent hyp — $\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp — $\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal — $\mathsf{wp}\quad e_1\ \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp — $\texttt{Hsafe1}: \mathsf{safe}_\varphi(e_1, \sigma_1)$

$\texttt{Hprim}: (e_1, \sigma_1) \rightarrow_{\mathsf{prim}} (e_2, \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(e_2, \sigma_2)$

Persistent hyp — $\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$ □

Spatial hyp — $\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal — $\mathsf{wp}\quad e_1\ \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$ ∗

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(e_1, \sigma_1)$

unfold prim-step $e_1 = K[e_1'] \land e_2 = K[e_2']$

$\texttt{Hprim}: (e_1, \sigma_1) \to_{\mathsf{prim}} (e_2, \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(e_2, \sigma_2)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal

$\mathsf{wp}\quad e_1 \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$ unfold prim-step $e_1 = K[e_1'] \land e_2 = K[e_2']$

$\texttt{Hbase}: (e_1', \sigma_1) \to_{\mathsf{base}} (e_2', \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_2)$

Persistent hyp $\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal $\mathsf{wp}\ K[e_1'] \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hbase}: (e_1', \sigma_1) \rightarrow_{\mathsf{base}} (e_2', \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_2)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal

$\mathsf{wp}\ K[e_1'] \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hbase}: (e_1', \sigma_1) \rightarrow_{\mathsf{base}} (e_2', \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_2)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ K[e_1'] \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$ by WpBind

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hbase}: (e_1', \sigma_1) \rightarrow_{\mathsf{base}} (e_2', \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_2)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ e_1' \left\{ w. \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$ by WpBind

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hbase}: (e_1', \sigma_1) \rightarrow_{\mathsf{base}} (e_2', \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_2)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ e_1' \left\{ w. \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$ case analysis

$\texttt{Hbase}: (e_1', \sigma_1) \rightarrow_{\mathsf{base}} (e_2', \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_2)$

Persistent hyp $\quad \texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal $\quad \mathsf{wp}\ e_1' \left\{ w. \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$)

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$ case analysis

$\texttt{Hbase}: (e_1', \sigma_1) \rightarrow_{\mathsf{base}} (e_2', \sigma_2)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_2)$

Persiste

Spatial

- $\square\ e_1' \rightarrow_{\mathsf{pure}} e_2'$
- $\square\ \sigma_1(\ell) = v \land (!\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$
- $\square\ \sigma_1(\ell) = v \land (\ell \leftarrow w, \sigma_1) \rightarrow_{\mathsf{base}} ((), \sigma_1[\ell \leftarrow w])$
- $\square\ \ell_0 \notin \sigma_1 \land (\texttt{ref}\ v, \sigma_1) \rightarrow_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

Goal

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal

$\mathsf{wp}\ e_1' \left\{ w. \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ e_1' \left\{ w. \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

by WpPure

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal

$\triangleright\, \mathsf{wp}\ e_2' \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

by WpPure

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal

$\triangleright\, \mathsf{wp}\ e_2' \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

strip later

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\triangleright\, \mathsf{wp}\ e_2' \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \quad (\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \{ v. \ulcorner \varphi(v) \urcorner \})$

strip later □

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ e_2' \{ w.\ \mathsf{wp}\ K[w] \{ v. \ulcorner \varphi(v) \urcorner \} \}$ ∗

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \quad \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal

$\mathsf{wp}\ e_2' \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \quad \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ e_2' \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

by WpBindInv

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \quad \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---✱

Goal

$\mathsf{wp}\ K[e_2'] \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

by WpBindInv

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \quad \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal

$\mathsf{wp}\ K[e_2'] \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp

$\texttt{IH}: \quad \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$ apply

---

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ K[e_2'] \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

If e_1' takes a pure step

Pure hyp `Hsafe1`: $\mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

`Hpure`: $e_1' \to_{\mathsf{pure}} e_2'$

`Hsafe2`: $\mathsf{safe}_\varphi(K[e_2'], \sigma_1)$

Persistent hyp `IH`: $\left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$ apply

---

Spatial hyp `Ho`: $\mathsf{S}_\circ(\sigma_1)$

Goals $\ulcorner \mathsf{safe}_\varphi(K[e_2'], \sigma_1) \urcorner$ $\quad$ $\mathsf{S}_\circ(\sigma_1)$

If e_1' takes a pure step

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

$\texttt{Hpure}: e_1' \to_{\mathsf{pure}} e_2'$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[e_2'], \sigma_1)$ frame

---

Persistent hyp

$\texttt{IH}: \quad \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$ frame

Goals

$\ulcorner \mathsf{safe}_\varphi(K[e_2'], \sigma_1) \urcorner$ $\mathsf{S}_\circ(\sigma_1)$

If e_1' takes a pure step

Pure hyp `Hsafe1`: $\mathsf{safe}_\varphi(K[e_1'], \sigma_1)$

- ☑ $e_1' \to_{\mathsf{pure}} e_2'$
- ☐ $\sigma_1(\ell) = v \land (!\ell, \sigma_1) \to_{\mathsf{base}} (v, \sigma_1)$
- ☐ $\sigma_1(\ell) = v \land (\ell \leftarrow w, \sigma_1) \to_{\mathsf{base}} ((), \sigma_1[\ell \leftarrow w])$
- ☐ $\ell_0 \notin \sigma_1 \land (\mathbf{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

Persiste

Spatial

Goal ✓

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ !\,\ell \left\{ w.\, \mathsf{wp}\ K[w] \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\} \right\}$

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}\colon \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}\colon \sigma_1(\ell) = v$

$\texttt{Hbase}\colon (!\,\ell, \sigma_1) \to_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}\colon \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp $\quad \texttt{IH}\colon \triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ\colon \mathsf{S}_\circ(\sigma_1)$ $\qquad$ destruct $\qquad \mathsf{S}_\circ(\sigma_1) \triangleq \ast_{(\ell \leftarrow v) \in \sigma_1} (\ell \mapsto v)$

Goal $\quad \mathsf{wp}\ !\,\ell \left\{ w.\, \mathsf{wp}\ K[w] \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\} \right\}$

If $e_1' = !\,\ell$

Pure hyp

Hsafe1: $\mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$

Hlookup: $\sigma_1(\ell) = v$

Hbase: $(!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

Hsafe2: $\mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp

IH: $\triangleright \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

Ho: $\mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

Hl: $\ell \mapsto v$

destruct $\mathsf{S}_\circ(\sigma_1) \triangleq \ast_{(\ell \leftarrow v) \in \sigma_1} (\ell \mapsto v)$

Goal

$\mathsf{wp}\ !\,\ell \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\} \right\}$

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}\colon \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}\colon \sigma_1(\ell) = v$

$\texttt{Hbase}\colon (!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}\colon \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp

$\texttt{IH}\colon \triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ\colon \mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

$\texttt{Hl}\colon \ell \mapsto v$

Goal

$\mathsf{wp}\ !\,\ell \left\{ w.\, \mathsf{wp}\ K[w] \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\} \right\}$

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp $\quad \texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

$\texttt{Hl}: \ell \mapsto v$

Goal $\quad \mathsf{wp}\ !\,\ell \left\{ w. \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

by WpLoad

If $e_1' = !\,\ell$

Pure hyp

`Hsafe1`: $\mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ `Hlookup`: $\sigma_1(\ell) = v$

`Hbase`: $(!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

`Hsafe2`: $\mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp `IH`: $\triangleright\, \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e\, \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp `H`$\circ$: $\mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

Goal $\triangleright\, (\ell \mapsto v \mathrel{-\!\!*} \mathsf{wp}\ K[v]\, \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\})$

by WpLoad

If $e_1' = \,!\,\ell$

Pure hyp

$\texttt{Hsafe1}\colon \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}\colon \sigma_1(\ell) = v$

$\texttt{Hbase}\colon (!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}\colon \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp $\quad \texttt{IH}\colon \triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ\colon \mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

Goal $\quad \triangleright (\ell \mapsto v \mathrel{-\!\!*} \mathsf{wp}\ K[v] \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\})$

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\quad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp $\quad \texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ: \mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

Goal $\quad \triangleright (\ell \mapsto v \mathrel{-\!\!*} \mathsf{wp}\ K[v] \left\{ v. \ulcorner \varphi(v) \urcorner \right\})$

strip later and intro

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp

$\texttt{IH}: \quad \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

$\texttt{Hl}: \ell \mapsto v$

Goal

$\mathsf{wp}\ K[v] \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$

strip later and intro

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp

$\texttt{IH}: \quad \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$ □

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

$\texttt{Hl}: \ell \mapsto v$

Goal

$\mathsf{wp}\ K[v] \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$ ∗

If $e_1' = !\,\ell$

Pure hyp

Hsafe1: $\mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ Hlookup: $\sigma_1(\ell) = v$

Hbase: $(!\,\ell, \sigma_1) \to_{\mathsf{base}} (v, \sigma_1)$

Hsafe2: $\mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp

IH: $\left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

Ho: $\mathsf{S}_\circ(\sigma_1 \setminus \{\ell\})$

Hl: $\ell \mapsto v$

combine $\mathsf{S}_\circ(\sigma_1) \triangleq \ast_{(\ell \leftarrow v) \in \sigma_1} (\ell \mapsto v)$

Goal

$\mathsf{wp}\ K[v] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}$

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (!\,\ell, \sigma_1) \to_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp $\quad \texttt{IH}: \quad \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

combine $\quad \mathsf{S}_\circ(\sigma_1) \triangleq \mathop{\Large *}_{(\ell \leftarrow v) \in \sigma_1} (\ell \mapsto v)$

Goal $\quad \mathsf{wp}\ K[v] \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$

If $e_1' = !\,\ell$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\qquad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp $\quad \texttt{IH}: \quad \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal $\quad \mathsf{wp}\ K[v] \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$

If $e_1' = !\,\ell$

Pure hyp

`Hsafe1`: $\mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ `Hlookup`: $\sigma_1(\ell) = v$

`Hbase`: $(!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

`Hsafe2`: $\mathsf{safe}_\varphi(K[v], \sigma_1)$ frame

Persistent hyp `IH`: $\left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e\ \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$ apply

Spatial hyp `H`$\circ$: $\mathsf{S}_\circ(\sigma_1)$ frame

Goal $\mathsf{wp}\ K[v]\ \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}$

If $e_1' = !\,\ell$

Pure hyp

`Hsafe1`: $\mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ `Hlookup`: $\sigma_1(\ell) = v$

`Hbase`: $(!\,\ell, \sigma_1) \rightarrow_{\mathsf{base}} (v, \sigma_1)$

`Hsafe2`: $\mathsf{safe}_\varphi(K[v], \sigma_1)$

Persistent hyp `IH`: $\left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$ apply

---

Spatial hyp `H`$\circ$: $\mathsf{S}_\circ(\sigma_1)$ framed

Goal ✓

If $e_1' = !\,\ell$

Pure hyp $\quad$ `Hsafe1`: $\mathsf{safe}_\varphi(K[!\,\ell], \sigma_1)$ $\quad$ `Hlookup`: $\sigma_1(\ell) = v$

Persiste

Spatial

- ☑ $e_1' \to_{\mathsf{pure}} e_2'$
- ☑ $\sigma_1(\ell) = v \land (!\ell, \sigma_1) \to_{\mathsf{base}} (v, \sigma_1)$
- ☐ $\sigma_1(\ell) = v \land (\ell \leftarrow w, \sigma_1) \to_{\mathsf{base}} ((), \sigma_1[\ell \leftarrow w])$
- ☐ $\ell_0 \notin \sigma_1 \land (\mathbf{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

Goal $\quad$ ✓

If $e_1' = \ell \leftarrow w$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\ell \leftarrow w], \sigma_1)$ $\quad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (\ell \leftarrow w, \sigma_1) \rightarrow_{\mathsf{base}} ((), \sigma_1[\ell \leftarrow w])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[()], \sigma_1[\ell \leftarrow w])$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

---∗

Goal

$\mathsf{wp}\ \ell \leftarrow w \left\{ w. \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

If $e_1' = \ell \leftarrow w$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\ell \leftarrow w], \sigma_1)$ $\quad$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

$\texttt{Hbase}: (\ell \leftarrow w, \sigma_1) \rightarrow_{\mathsf{base}} ((), \sigma_1[\ell \leftarrow w])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[()], \sigma_1[\ell \leftarrow w])$

Persistent hyp $\quad \texttt{IH}: \triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal $\quad \mathsf{wp}\ \ell \leftarrow w \left\{ w. \mathsf{wp}\ K[w] \left\{ v. \ulcorner \varphi(v) \urcorner \right\} \right\}$

by WpStore

If $e_1' = \ell \leftarrow w$

Pure hyp $\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\ell \leftarrow w], \sigma_1)$ $\texttt{Hlookup}: \sigma_1(\ell) = v$

☑ $e_1' \to_{\mathsf{pure}} e_2'$

☑ $\sigma_1(\ell) = v \land (!\ell, \sigma_1) \to_{\mathsf{base}} (v, \sigma_1)$

Persiste ☑ $\sigma_1(\ell) = v \land (\ell \leftarrow w, \sigma_1) \to_{\mathsf{base}} ((), \sigma_1[\ell \leftarrow w])$

Spatial ☐ $\ell_0 \notin \sigma_1 \land (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

Goal $\mathsf{wp}\ \ell \leftarrow w \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\} \right\}$ $*$

by WpStore

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \rightarrow_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ \texttt{ref}\ v \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\} \right\}$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \rightarrow_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

Persistent hyp

$\texttt{IH}: \triangleright \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

$\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1)$

Goal

$\mathsf{wp}\ \texttt{ref}\ v \left\{ w.\ \mathsf{wp}\ K[w] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\} \right\}$

by WpAlloc

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\qquad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

Persistent hyp $\quad \texttt{IH}: \quad \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ: \mathsf{S}_\circ(\sigma_1[\ell \leftarrow v])$

Goal $\quad \mathsf{wp}\ K[\ell] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}$

by WpAlloc

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\qquad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

Persistent hyp $\texttt{IH}: \quad \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\texttt{H}\circ: \mathsf{S}_\circ(\sigma_1[\ell \leftarrow v])$

Goal $\mathsf{wp}\ K[\ell] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\quad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

ℓ_0 and ℓ are unrelated

Persistent hyp $\quad \texttt{IH}: \quad \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp $\quad \texttt{H}\circ: \mathsf{S}_\circ(\sigma_1[\ell \leftarrow v])$

Goal $\quad \mathsf{wp}\ K[\ell] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\quad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

ℓ_0 and ℓ are unrelated

Persistent hyp $\quad \texttt{IH}: \quad \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

---□

Spatial hyp $\quad \texttt{H}\circ: \mathsf{S}_\circ(\sigma_1[\ell \leftarrow v])$

Goal $\quad \mathsf{wp}\ K[\ell] \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\quad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

Persistent hyp $\quad \texttt{IH}: \quad \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

Goal $\quad \ulcorner \mathsf{safe}_\varphi(K[\ell], \sigma_1[\ell \leftarrow v]) \urcorner$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\qquad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

pure intro

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

Persistent hyp $\quad \texttt{IH}: \quad \left(\forall e, \sigma.\ \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\circ(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\ \ulcorner \varphi(v) \urcorner \right\}\right)$

Spatial hyp

Goal $\quad \ulcorner \mathsf{safe}_\varphi(K[\ell], \sigma_1[\ell \leftarrow v]) \urcorner$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\qquad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

pure intro

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

$\mathsf{safe}_\varphi(K[\ell], \sigma_1[\ell \leftarrow v])$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\quad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

$\mathsf{safe}_\varphi(K[\ell], \sigma_1[\ell \leftarrow v])$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\qquad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

$\mathsf{safe}_\varphi(K[\ell], \sigma_1[\ell \leftarrow v])$

$\Uparrow$

$(\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell, \sigma_1[\ell \leftarrow v])$

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

$\mathsf{safe}_\varphi(K[\ell], \sigma_1[\ell \leftarrow v])$

$\Uparrow$

$(\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell, \sigma_1[\ell \leftarrow v])$ ✓

(because **ref** chooses ℓ non-deterministically)

If $e_1' = \texttt{ref}\ v$

Pure hyp

$\texttt{Hsafe1}: \mathsf{safe}_\varphi(K[\texttt{ref}\ v], \sigma_1)$ $\quad$ $\texttt{Hfresh}: \ell \notin \sigma_1$

$\texttt{Hbase}: (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

$\texttt{Hsafe2}: \mathsf{safe}_\varphi(K[\ell_0], \sigma_1[\ell_0 \leftarrow v])$

$\mathsf{safe}_\varphi(K[\ell], \sigma_1[\ell \leftarrow v])$

$\Uparrow$

- ☑ $e_1' \to_{\mathsf{pure}} e_2'$
- ☑ $\sigma_1(\ell) = v \land (!\ell, \sigma_1) \to_{\mathsf{base}} (v, \sigma_1)$
- ☑ $\sigma_1(\ell) = v \land (\ell \leftarrow w, \sigma_1) \to_{\mathsf{base}} ((), \sigma_1[\ell \leftarrow w])$
- ☑ $\ell_0 \notin \sigma_1 \land (\texttt{ref}\ v, \sigma_1) \to_{\mathsf{base}} (\ell_0, \sigma_1[\ell_0 \leftarrow v])$

Recap

- Löb induction $\quad \triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\diamond(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$
- If $\mathsf{safe}_\varphi(v, \sigma)$ ($v \in \mathit{Val}$) $\implies \ulcorner \varphi(v) \urcorner \vdash \mathsf{wp}\ v \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$ (by $\textsc{WpValue}$)
- If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$), then e_1 is reducible

  case analysis on the first step

Recap

- Löb induction $\triangleright \left(\forall e, \sigma. \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\diamond(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}\right)$
- If $\mathsf{safe}_\varphi(v, \sigma)$ ($v \in \mathit{Val}$) $\Longrightarrow$ $\ulcorner \varphi(v) \urcorner \vdash \mathsf{wp}\ v \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$ (by $\textsc{WpValue}$)
- If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$), then e_1 is reducible

case analysis on the first step

Language-independent, expect for the blue part

Recap

- Löb induction $\triangleright \left(\forall e, \sigma.\, \ulcorner \mathsf{safe}_\varphi(e, \sigma) \urcorner \mathrel{-\!\!*} \mathsf{S}_\diamond(\sigma) \mathrel{-\!\!*} \mathsf{wp}\ e \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}\right)$
- If $\mathsf{safe}_\varphi(v, \sigma)$ ($v \in \mathit{Val}$) $\Longrightarrow$ $\ulcorner \varphi(v) \urcorner \vdash \mathsf{wp}\ v \left\{ v.\, \ulcorner \varphi(v) \urcorner \right\}$ (by WpValue)
- If $\mathsf{safe}_\varphi(e_1, \sigma_1)$ ($e_1 \notin \mathit{Val}$), then e_1 is reducible

case analysis on the first step

Language-independent, expect for the blue part

For language-dependent part: update $\mathsf{S}_\diamond(\sigma)$ using primitive laws.

Concurrent Completeness

Sequential:

- Proof environment: $\mathsf{safe}_{\varphi}(e, \sigma)$ and $\mathsf{S}_{\circ}(\sigma)$
- One wp $e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

Concurrent Completeness

Sequential:

- Proof environment: $\mathsf{safe}_\varphi(e, \sigma)$ and $\mathsf{S}_\circ(\sigma)$
- One wp $e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

But in concurrent setting, we need to prove...

$$\frac{\text{Spatial hyp} \qquad \ulcorner \mathsf{safe\text{-}tp}_\varphi(\vec{e}, \sigma) \urcorner \qquad \mathsf{S}_\circ(\sigma)}{\text{Goal} \qquad \mathsf{wp}\ \texttt{fork}\ \{e_1\}\,;\, e_2 \left\{ v. \ulcorner \varphi(v) \urcorner \right\}} *$$

Concurrent Completeness

Sequential:

- Proof environment: $\mathsf{safe}_\varphi(e, \sigma)$ and $\mathsf{S}_\circ(\sigma)$
- One wp $e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

But in concurrent setting, we need to prove...

$$\frac{\text{Spatial hyp} \qquad \ulcorner \mathsf{safe\text{-}tp}_\varphi(\vec{e}, \sigma) \urcorner \qquad \mathsf{S}_\circ(\sigma)}{\text{Goal} \qquad \mathsf{wp}\ e_1 \left\{ v.\ \mathsf{True} \right\} * \mathsf{wp}\ e_2 \left\{ v. \ulcorner \varphi(v) \urcorner \right\}} *$$

Concurrent Completeness

Sequential:

- Proof environment: $\mathsf{safe}_\varphi(e, \sigma)$ and $\mathsf{S}_\circ(\sigma)$
- One wp $e\left\{v.\ulcorner\varphi(v)\urcorner\right\}$

But in concurrent setting, we need to prove...

$$\frac{\text{Spatial hyp} \qquad \ulcorner\mathsf{safe\text{-}tp}_\varphi(\vec{e}, \sigma)\urcorner \qquad \mathsf{S}_\circ(\sigma)}{\text{Goal} \qquad \mathsf{wp}\ e_1\left\{v.\ \mathsf{True}\right\} * \mathsf{wp}\ e_2\left\{v.\ulcorner\varphi(v)\urcorner\right\}}\ *$$

$$I_{compl} \triangleq \boxed{\exists \vec{e}, \sigma.\ \ulcorner\mathsf{safe\text{-}tp}_\varphi(\vec{e}, \sigma)\urcorner * \mathsf{S}_\circ(\sigma)}^{\mathcal{N}}$$

$$\forall i, e.\ I_{compl} \qquad \vdash \mathsf{wp}\ e\left\{v.\ulcorner i = 0 \Rightarrow \varphi(v)\urcorner\right\}$$

Concurrent Completeness

Sequential:

- Proof environment: $\mathsf{safe}_\varphi(e, \sigma)$ and $\mathsf{S}_\circ(\sigma)$
- One wp $e\left\{v.\ulcorner\varphi(v)\urcorner\right\}$

But in concurrent setting, we need to prove...

$$\frac{\text{Spatial hyp} \qquad \ulcorner\mathsf{safe\text{-}tp}_\varphi(\vec{e}, \sigma)\urcorner \qquad \mathsf{S}_\circ(\sigma)}{\text{Goal} \qquad \mathsf{wp}\ e_1\left\{v.\ \mathsf{True}\right\} * \mathsf{wp}\ e_2\left\{v.\ulcorner\varphi(v)\urcorner\right\}}\ *$$

$$I_{compl} \triangleq \boxed{\exists \vec{e}, \sigma.\ \ulcorner\mathsf{safe\text{-}tp}_\varphi(\vec{e}, \sigma)\urcorner * \mathsf{S}_\circ(\sigma) * \bullet \vec{e}^{\,\gamma}}^{\mathcal{N}}$$

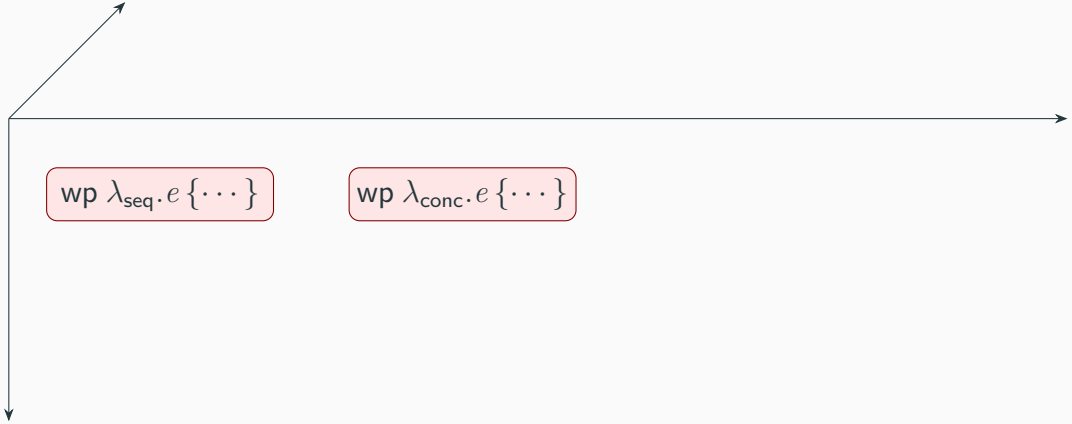
$$\forall i, e.\ I_{compl} * i \hookrightarrow^\gamma e \vdash \mathsf{wp}\ e\left\{v.\ulcorner i = 0 \Rightarrow \varphi(v)\urcorner\right\}$$

Generalization

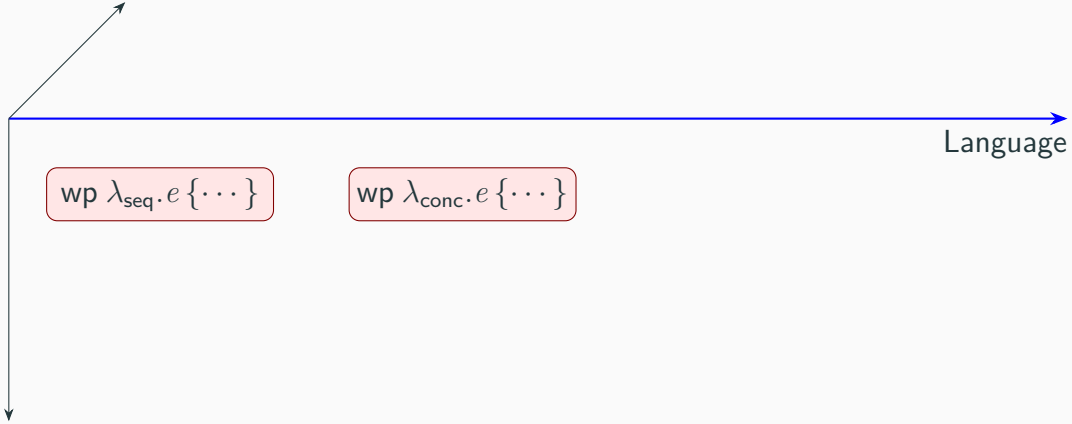
$\mathsf{wp}\ \lambda_{\mathsf{seq}}.e\,\{\cdots\}$

$\mathsf{wp}\ \lambda_{\mathsf{conc}}.e\,\{\cdots\}$

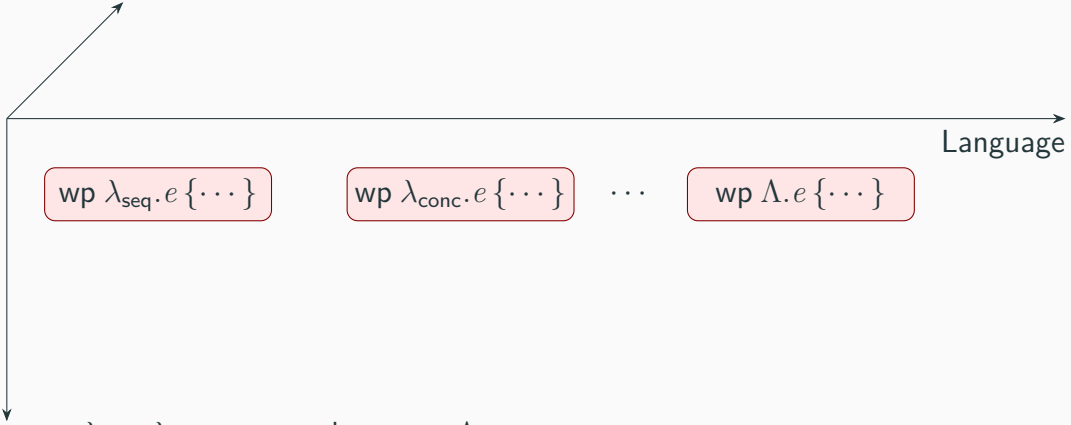
Generalization



Generalization

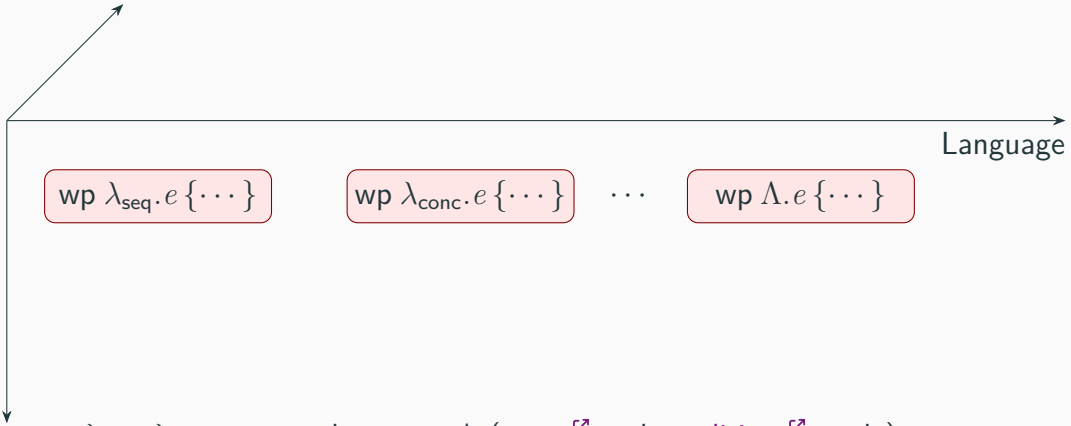


Generalization



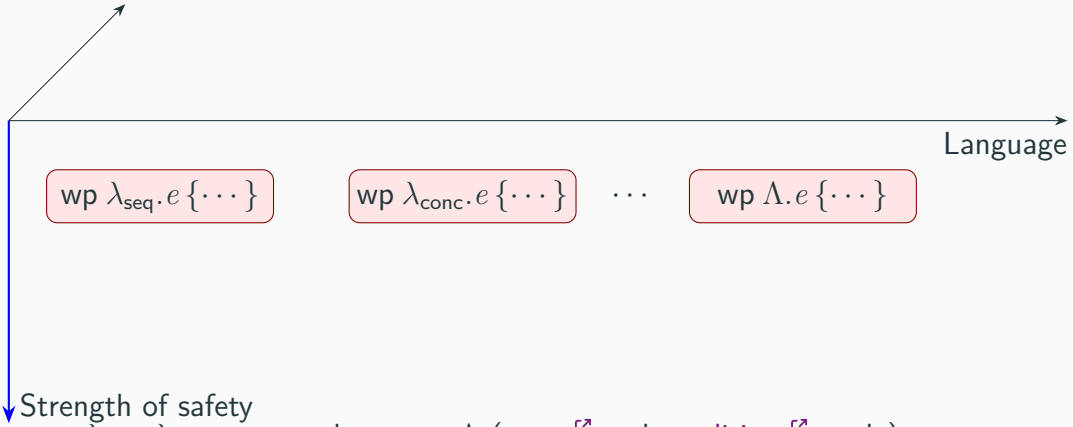
- λ_{seq}, $\lambda_{\mathsf{con}} \longrightarrow$ any language Λ

Generalization



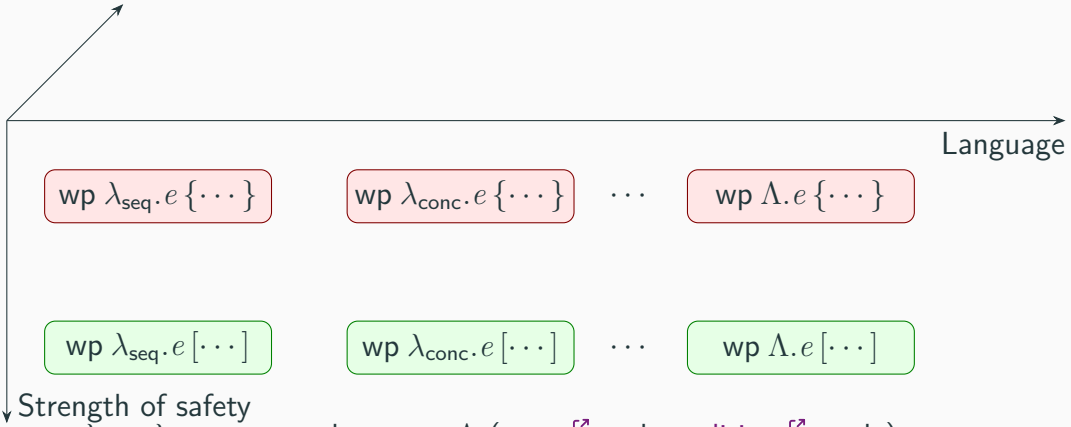
- λ_{seq}, $\lambda_{\mathsf{con}} \longrightarrow$ any language Λ (terms and conditions apply)

Generalization



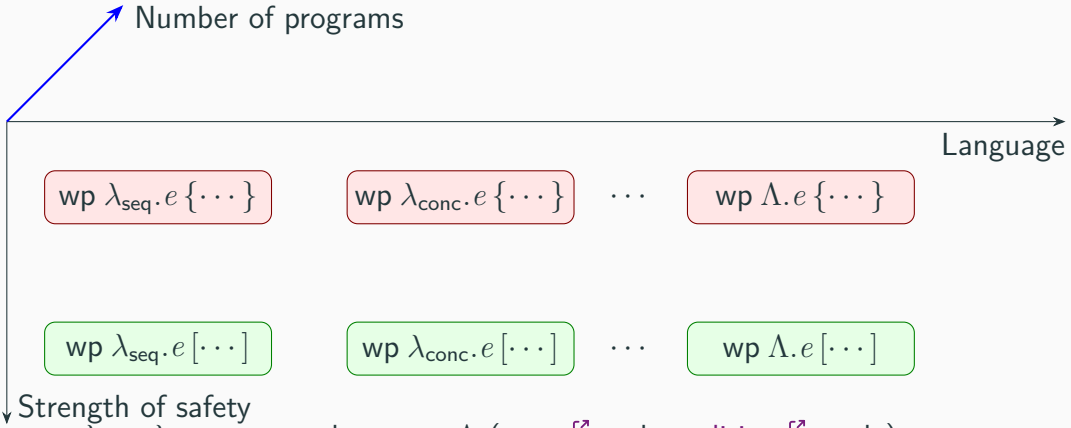
- λ_{seq}, $\lambda_{\mathsf{con}} \longrightarrow$ any language Λ (terms and conditions apply)

Generalization



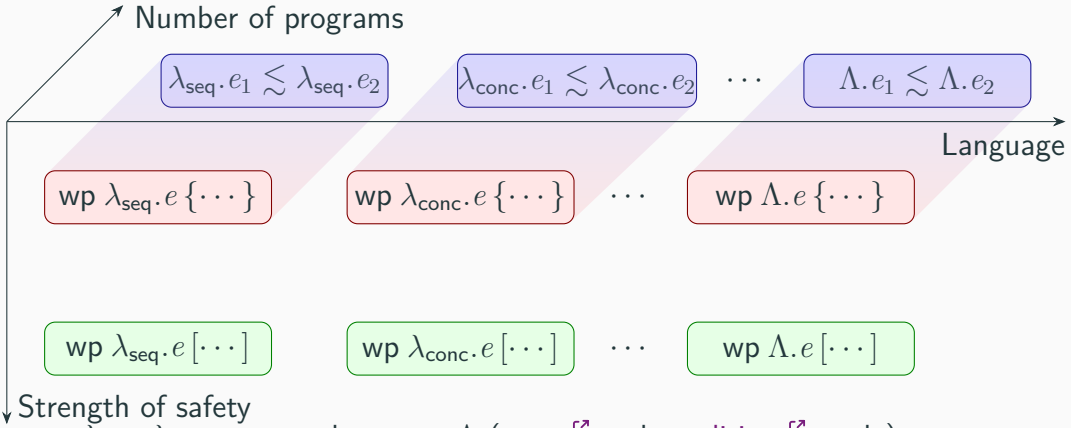
- λ_{seq}, $\lambda_{\text{con}} \longrightarrow$ any language Λ (terms and conditions apply)
- wp $e\,\{\cdots\} \longrightarrow$ wp $e\,[\cdots]$ (Löb induction $\longrightarrow$ well-founded induction)

Generalization



- λ_{seq}, $\lambda_{\mathsf{con}} \longrightarrow$ any language Λ (terms and conditions apply)
- wp $e\,\{\cdots\} \longrightarrow$ wp $e\,[\cdots]$ (Löb induction $\longrightarrow$ well-founded induction)

Generalization



- λ_{seq}, $\lambda_{\text{con}} \longrightarrow$ any language Λ (terms and conditions apply)
- wp $e\,\{\cdots\} \longrightarrow$ wp $e\,[\cdots]$ (Löb induction $\longrightarrow$ well-founded induction)
- Unary $\longrightarrow$ binary (built on top of the unary completeness)

Condition 32 (Completeness).

$$\ulcorner \text{base-red}(e,\sigma)\urcorner * \mathrm{S}_\flat(\sigma) * \bullet^\gamma \vec{e} * \ulcorner \mathit{safe\text{-}tp}_\varphi(\vec{e},\sigma)\urcorner * n \hookrightarrow^\gamma K[e] \vdash \Rrightarrow_{\mathcal{E}}$$
$$\left(\ulcorner \text{Atomic } e\urcorner * \forall \Phi.\ \triangleright\ \mathit{WithStatePre}(\mathcal{E},e,n,K,\sigma,\vec{e},\Phi) \twoheadrightarrow \mathrm{wp}_{\mathcal{E}}\ e\ \{\Phi\}\right) \vee$$
$$\left(\mathrm{S}_\flat(\sigma) * \bullet^\gamma \vec{e} * \forall \Phi.\ \triangleright\ \mathit{WithoutStatePre}(\mathcal{E},e,K,\Phi) \twoheadrightarrow \mathrm{wp}_\top\ e\ \{\Phi\}\right)$$

where

$$\mathit{WithStatePre}(\mathcal{E},e,n,K,\sigma,\vec{e},\Phi) \triangleq \forall v',\sigma',\vec{e}_f.\ \ulcorner(e,\sigma) \to (v',\sigma',\vec{e}_f)\urcorner * \mathrm{S}_\flat(\sigma') * \bullet^\gamma \vec{e} * n \hookrightarrow^\gamma K[e] \twoheadrightarrow \Rrightarrow_{\mathcal{E}}$$
$$\Phi(v') * \mathop{\ast}_{e_f \in \vec{e}_f} \mathrm{wp}_\top\ e_f\ \{v.\ \mathrm{True}\}$$

$$\mathit{WithoutStatePre}(\mathcal{E},e,K,\Phi) \triangleq \forall e',\vec{e},\vec{e}_f.\ \mathit{Reduces}(K,e',\vec{e}) \twoheadrightarrow \Rrightarrow_\top$$
$$\mathrm{wp}_\top\ e'\ \{\Phi\} * \mathop{\ast}_{e_f \in \vec{e}_f} \mathrm{wp}_\top\ e_f\ \{v.\ \mathrm{True}\}$$

$$\mathit{Reduces}(e,K,e',\vec{e}) \triangleq \forall \sigma.\ \mathrm{S}_\flat(\sigma) * \bullet^\gamma \vec{e} * \ulcorner \mathit{safe\text{-}tp}_\varphi(\vec{e},\sigma)\urcorner \twoheadrightarrow \Rrightarrow_{\mathcal{E}}$$
$$\exists \sigma'.\ \ulcorner(e,\sigma) \to^+ (e',\sigma',\vec{e}_f)\urcorner * \mathrm{S}_\flat(\sigma') * \bullet^\gamma \vec{e} * n \hookrightarrow^\gamma K[e]$$

Completeness. If Condition 32 holds for a language Λ, then

$$\mathsf{safe}_\varphi(e,\varnothing) \implies \vdash \mathsf{wp}\ e\left\{v.\ \ulcorner\varphi(v)\urcorner\right\}$$

Strength of safety

- λ_{seq}, $\lambda_{\mathsf{con}} \longrightarrow$ any language Λ (terms and conditions apply)
- $\mathsf{wp}\ e\{\cdots\} \longrightarrow \mathsf{wp}\ e[\cdots]$ (Löb induction $\longrightarrow$ well-founded induction)
- Unary $\longrightarrow$ binary (built on top of the unary completeness)

Generalization

Completeness. If Condition 32 holds for a language Λ, then

$$\mathsf{safe}_\varphi(e, \varnothing) \implies \vdash \mathsf{wp}\ e\left\{v.\ulcorner\varphi(v)\urcorner\right\}$$

Strength of safety

- λ_{seq}, λ_{con} $\longrightarrow$ any language Λ (terms and conditions apply)
- wp $e\{\cdots\}$ $\longrightarrow$ wp $e[\cdots]$ (Löb induction $\longrightarrow$ well-founded induction)
- Unary $\longrightarrow$ binary (built on top of the unary completeness)

Case Studies

Completeness. If Condition 32 holds for a language Λ, then

$$\mathsf{safe}_{\varphi}(e, \varnothing) \implies \vdash \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$$

Case studies: HeapLang, λ_{Rust}, time credits, and Eris (error credits)

Case Studies

Completeness. If Condition 32 holds for a language Λ, then

$$\mathsf{safe}_{\varphi}(e, \varnothing) \implies \vdash \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$$

Case studies: HeapLang, λ_{Rust}, time credits, and Eris (error credits)

- λ_{Rust}: handle non-atomic memory access

Case Studies

Completeness. If Condition 32 holds for a language Λ, then

$$\mathsf{safe}_{\varphi}(e, \varnothing) \implies \vdash \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$$

Case studies: HeapLang, λ_{Rust}, time credits, and Eris (error credits)

- λ_{Rust}: handle non-atomic memory access
- Eris: redo the completeness theorem following the same pattern

The missing rules in HeapLang

$$
\begin{aligned}
\mathit{unique} \triangleq\ & \texttt{let}\ \ell_1 = \texttt{ref}()\ \texttt{in} \\
& \texttt{free}\ \ell_1; \\
& \texttt{assert}\ \ell_1 \neq \texttt{ref}()
\end{aligned}
$$

The missing rules in HeapLang

$$unique \triangleq \texttt{let}\ \ell_1 = \texttt{ref}()\ \texttt{in}\ \texttt{free}\ \ell_1;\ \texttt{assert}\ \ell_1 \neq \texttt{ref}()$$

Safe because locations are never reused!

$$\frac{\ell \notin \mathsf{dom}(\sigma)}{(\texttt{ref}\ v, \sigma) \rightarrow_{\mathsf{base}} (\ell, \sigma[\ell \leftarrow v], \epsilon)}$$

$$\frac{\sigma(\ell) = v}{(\texttt{free}\ \ell, \sigma) \rightarrow_{\mathsf{base}} ((), \sigma[\ell \leftarrow \bot]\,\sigma, \epsilon)}$$

The missing rules in HeapLang

$$unique \triangleq \texttt{let}\ \ell_1 = \texttt{ref}()\ \texttt{in}\ \texttt{free}\ \ell_1;\ \texttt{assert}\ \ell_1 \neq \texttt{ref}()$$

Safe because locations are never reused!

Prove safety using meta tokens.

$$\frac{\ell \notin \mathsf{dom}(\sigma)}{(\texttt{ref}\ v, \sigma) \to_{\mathsf{base}} (\ell, \sigma[\ell \leftarrow v], \epsilon)}$$

$$\frac{\sigma(\ell) = v}{(\texttt{free}\ \ell, \sigma) \to_{\mathsf{base}} ((), \sigma[\ell \leftarrow \bot]\, \sigma, \epsilon)}$$

$$\{\mathsf{True}\}\ \texttt{ref}\ v\,\{\ell.\ \ell \mapsto v * \mathsf{metaTok}(\ell, \top)\}$$

$$\{\ell \mapsto v\}\ \texttt{free}\ \ell\,\{v.\ \ulcorner v = () \urcorner\}$$

The missing rules in HeapLang

$$unique \triangleq \texttt{let}\ \ell_1 = \texttt{ref}()\ \texttt{in}\ \texttt{free}\ \ell_1;\ \texttt{assert}\ \ell_1 \neq \texttt{ref}()$$

Safe because locations are never reused!

Prove safety using meta tokens.

$$\frac{\ell \notin \mathsf{dom}(\sigma)}{(\texttt{ref}\ v, \sigma) \to_{\mathsf{base}} (\ell, \sigma[\ell \leftarrow v], \epsilon)}$$

$$\frac{\sigma(\ell) = v}{(\texttt{free}\ \ell, \sigma) \to_{\mathsf{base}} ((), \sigma[\ell \leftarrow \bot]\,\sigma, \epsilon)}$$

$\{\mathsf{True}\}\ \texttt{ref}\ v\,\{\ell.\ \ell \mapsto v * \mathsf{metaTok}(\ell, \top)\}$

$\{\ell \mapsto v\}\ \texttt{free}\ \ell\,\{v.\ \ulcorner v = () \urcorner\}$

$$\frac{\mathsf{metaTok}(\ell_1, \top) * \mathsf{metaTok}(\ell_2, \top)}{\ulcorner \ell_1 \neq \ell_2 \urcorner}$$

The missing rules in HeapLang

$$unique \triangleq \texttt{let}\ \ell_1 = \texttt{ref}()\ \texttt{in}\ \texttt{free}\ \ell_1;\ \texttt{assert}\ \ell_1 \neq \texttt{ref}()$$

Safe because locations are never reused!

Prove safety using meta tokens.

$$\frac{\ell \notin \mathsf{dom}(\sigma)}{(\texttt{ref}\ v, \sigma) \to_{\mathsf{base}} (\ell, \sigma[\ell \leftarrow v], \epsilon)}$$

$$\frac{\sigma(\ell) = v}{(\texttt{free}\ \ell, \sigma) \to_{\mathsf{base}} ((), \sigma[\ell \leftarrow \bot]\,\sigma, \epsilon)}$$

$$\{\mathsf{True}\}\ \texttt{ref}\ v\,\{\ell.\ \ell \mapsto v * \mathsf{metaTok}(\ell, \top)\}$$

$$\{\ell \mapsto v\}\ \texttt{free}\ \ell\,\{v.\ \ulcorner v = () \urcorner\}$$

$$\frac{\mathsf{metaTok}(\ell_1, \top) * \mathsf{metaTok}(\ell_2, \top)}{\ulcorner \ell_1 \neq \ell_2 \urcorner}$$

(was missing)

Another Example

$$resolve \triangleq \texttt{resolve}\ (\texttt{LitProph}\ 1)\ \texttt{to}\ ()$$

Another Example

$resolve \triangleq \texttt{resolve}\ (\texttt{LitProph}\ 1)\ \texttt{to}\ ()$

Analogy: $(\texttt{LitLoc}\ 1) \leftarrow ()$

Another Example

$resolve \triangleq$ **resolve** (LitProph 1) **to** ()

Safe because

resolve does not check existence!

Analogy: (LitLoc 1) $\leftarrow$ ()

$$\frac{}{(\textbf{resolve}\ p\ \textbf{to}\ v, \sigma) \xrightarrow{[(p,((),v))]}_{\mathsf{base}} ((), \sigma, \varepsilon)}$$

Another Example

$resolve \triangleq$ **resolve** (LitProph 1) **to** ()

Safe because

resolve does not check existence!

Analogy: (LitLoc 1) $\leftarrow$ ()

$$\frac{p \in \sigma.pid}{(\textbf{resolve}\ p\ \textbf{to}\ v, \sigma) \xrightarrow{[(p,((),v))]}_{\mathsf{base}} ((), \sigma, \varepsilon)}$$

Another Example

$resolve \triangleq$ **resolve** (LitProph 1) **to** ()

Safe because

resolve does not check existence!

Completeness.

For a HeapLang expression e, $\mathsf{safe}_{\varphi}(e, \varnothing) \implies \vdash \mathsf{wp}\ e \left\{ v. \ulcorner \varphi(v) \urcorner \right\}$

$$\frac{p \in \sigma.pid}{(\textbf{resolve}\ p\ \textbf{to}\ v, \sigma) \xrightarrow{[(p,((),v))]}_{\mathsf{base}} ((), \sigma, \varepsilon)}$$

Insight: What Iris features are needed?

- Löb induction
- First-order ghost state, ghost maps
- Invariants of timeless assertions
- WpValue, WpBind, WpPure, FupdWp, and WpAtomic

Insight: What Iris features are needed?

- Löb induction
- First-order ghost state, ghost maps
- Invariants of timeless assertions
- WpValue, WpBind, WpPure, FupdWp, and WpAtomic
- WpValueInv and WpBindInv

Insight: What Iris features are needed?

- Löb induction
- First-order ghost state, ghost maps
- Invariants of timeless assertions
- WpValue, WpBind, WpPure, FupdWp, and WpAtomic
- WpValueInv and WpBindInv
- Locality of the semantics
  - Allocation must non-deterministically pick a location
  - Resolve must ensure the prophecy variable exists

Insight: What Iris features are needed?

- Löb induction
- First-order ghost state, ghost maps
- Invariants of timeless assertions
- WpValue, WpBind, WpPure, FupdWp, and WpAtomic
- WpValueInv and WpBindInv
- Locality of the semantics
  - Allocation must non-deterministically pick a location
  - Resolve must ensure the prophecy variable exists

We don't need

- Higher-order ghost state
- Impredicative invariants
- Later credits

Concurrent Library Completeness

Zichen Zhang Simon Gregersen Joseph Tassarotti

Logic & Truth for Concurrent Data Structures

Logic

Truth

Logic & Truth for Concurrent Data Structures

Logic

Truth

$\mathsf{lin}_\mu(\mathsf{op})$

Linearizable *w.r.t.* sequential behavior μ

Logic & Truth for Concurrent Data Structures

$I_\mu \vdash \langle P_\mu \rangle\ \mathsf{op}(obj, x)\ \langle y.\ Q_\mu(y) \rangle$

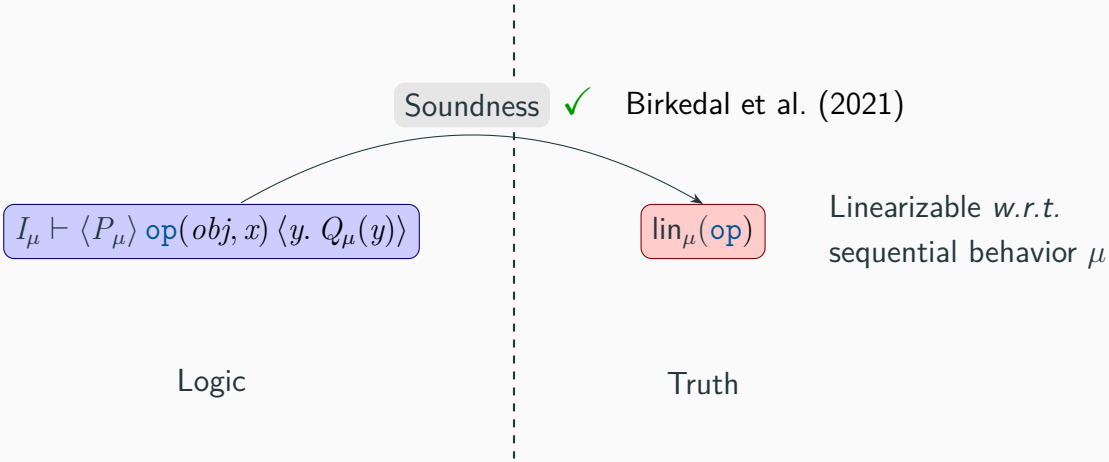
Logic

$\mathsf{lin}_\mu(\mathsf{op})$

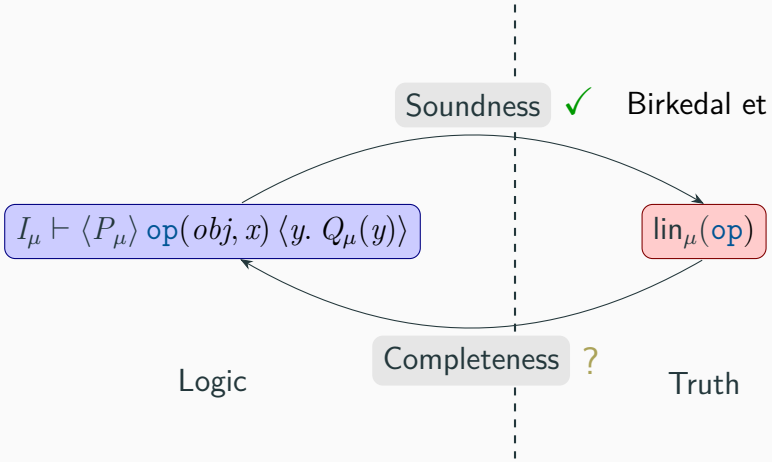
Truth

Linearizable *w.r.t.* sequential behavior μ

Logic & Truth for Concurrent Data Structures

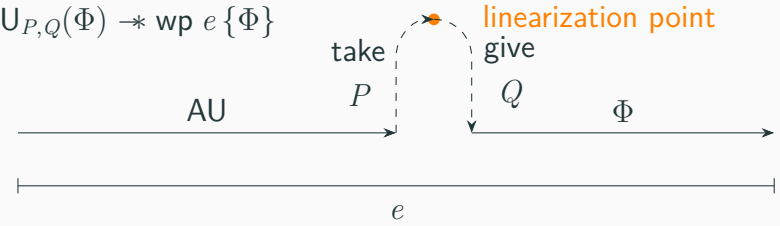


Logic & Truth for Concurrent Data Structures



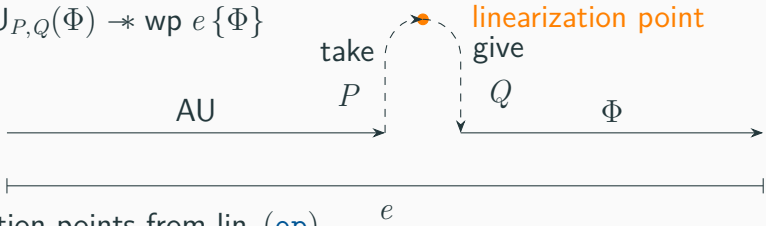
Key Ideas

$\langle P \rangle\ e\ \langle Q \rangle \triangleq \forall \Phi.\ \mathsf{AU}_{P,Q}(\Phi) \twoheadrightarrow \mathsf{wp}\ e\ \{\Phi\}$



Key Ideas

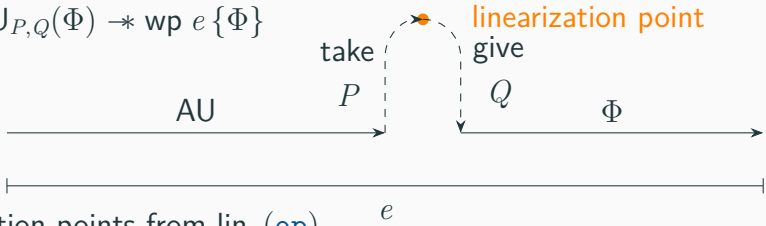
$\langle P \rangle\ e\ \langle Q \rangle \triangleq \forall \Phi.\ \mathsf{AU}_{P,Q}(\Phi) \twoheadrightarrow \mathsf{wp}\ e\ \{\Phi\}$



- Extract linearization points from $\mathsf{lin}_{\mu}(\mathtt{op})$
- "Fire" AU at the linearization points

Key Ideas

$\langle P \rangle\ e\ \langle Q \rangle \triangleq \forall \Phi.\ \mathsf{AU}_{P,Q}(\Phi) \twoheadrightarrow \mathsf{wp}\ e\ \{\Phi\}$



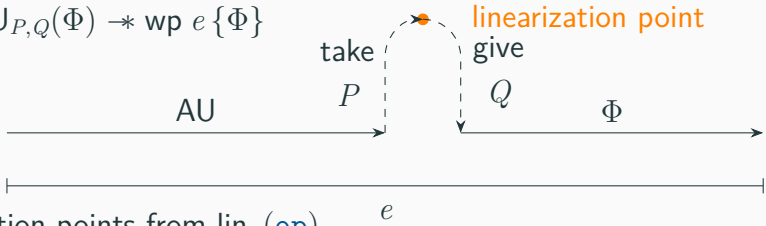
- Extract linearization points from $\mathsf{lin}_{\mu}(\mathtt{op})$
- "Fire" AU at the linearization points

Challenges & Strategies

- Future dependency:

Key Ideas

$\langle P \rangle\ e\ \langle Q \rangle \triangleq \forall \Phi.\ \mathsf{AU}_{P,Q}(\Phi) \twoheadrightarrow \mathsf{wp}\ e\ \{\Phi\}$



- Extract linearization points from $\mathsf{lin}_{\mu}(\mathtt{op})$
- "Fire" AU at the linearization points

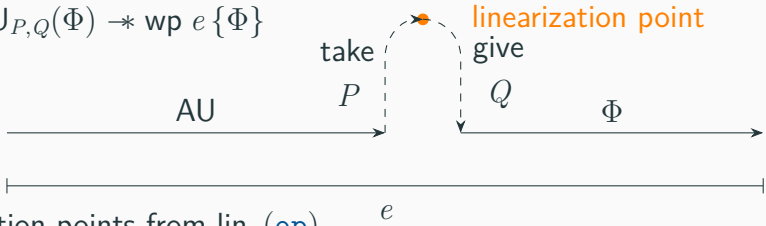
Challenges & Strategies

- Future dependency:

- Non structural linearization points:

Key Ideas

$\langle P \rangle\ e\ \langle Q \rangle \triangleq \forall \Phi.\ \mathsf{AU}_{P,Q}(\Phi) \twoheadrightarrow \mathsf{wp}\ e\ \{\Phi\}$



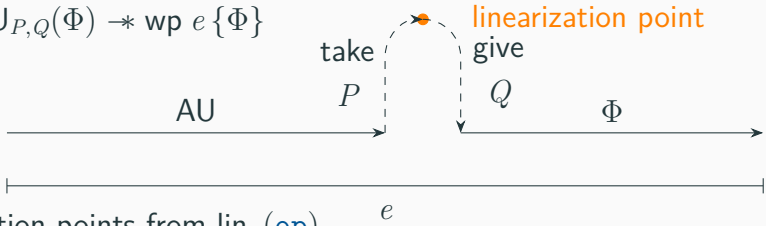
- Extract linearization points from $\mathsf{lin}_{\mu}(\mathtt{op})$
- "Fire" AU at the linearization points

Challenges & Strategies

- Future dependency: prophecy variables
- Non structural linearization points:

Key Ideas

$\langle P \rangle\ e\ \langle Q \rangle \triangleq \forall \Phi.\ \mathsf{AU}_{P,Q}(\Phi) \twoheadrightarrow \mathsf{wp}\ e\ \{\Phi\}$



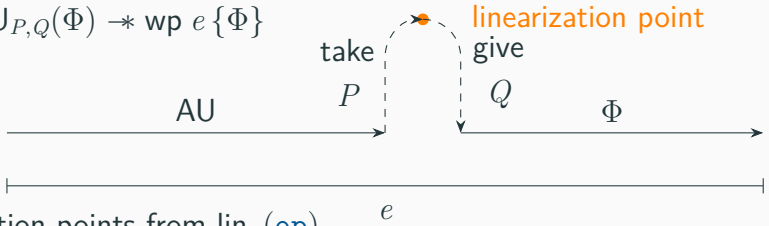
- Extract linearization points from $\mathsf{lin}_{\mu}(\mathtt{op})$
- "Fire" AU at the linearization points

Challenges & Strategies

- Future dependency: typed prophecy variables on undecidable types
  needs excluded middle law
- Non structural linearization points:

Key Ideas

$\langle P \rangle\ e\ \langle Q \rangle \triangleq \forall \Phi.\ \mathsf{AU}_{P,Q}(\Phi) \twoheadrightarrow \mathsf{wp}\ e\ \{\Phi\}$



- Extract linearization points from $\mathsf{lin}_{\mu}(\mathtt{op})$
- "Fire" AU at the linearization points

Challenges & Strategies

- Future dependency: typed prophecy variables on undecidable types
  needs excluded middle law
- Non structural linearization points: helping
  needs impredicative invariants and later credits

Key Ideas

$\langle P \rangle\ e\ \langle Q \rangle \triangleq \forall \Phi.\ \mathsf{AU}_{P,Q}(\Phi) \twoheadrightarrow \mathsf{wp}\ e\ \{\Phi\}$

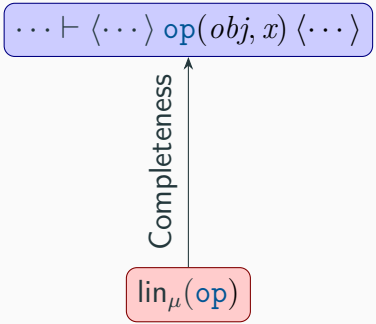
Completeness.
For a library op, $\mathsf{lin}_\mu(\mathtt{op}) \implies \forall obj, x.\ I_\mu \vdash \langle P_\mu \rangle\ \mathtt{op}(obj, x)\ \langle y.\ Q_\mu(y) \rangle$

Challenges & Strategies

- Future dependency: typed prophecy variables on undecidable types
  needs excluded middle law
- Non structural linearization points: helping
  needs impredicative invariants and later credits

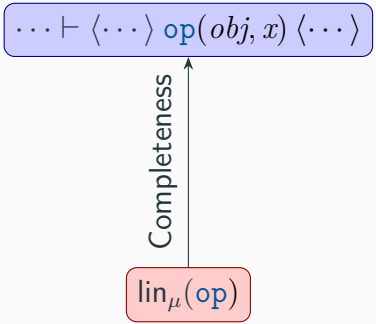
Why completeness?

- Theoretically interesting



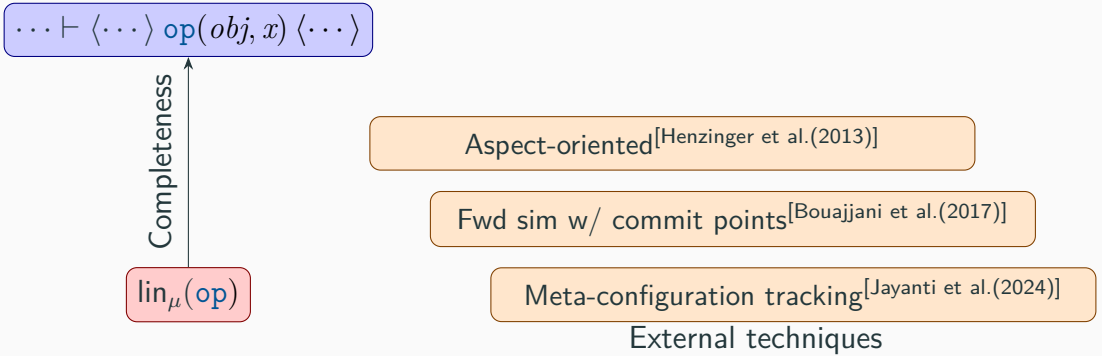
Why completeness?

- Theoretically interesting
- Embedding external techniques



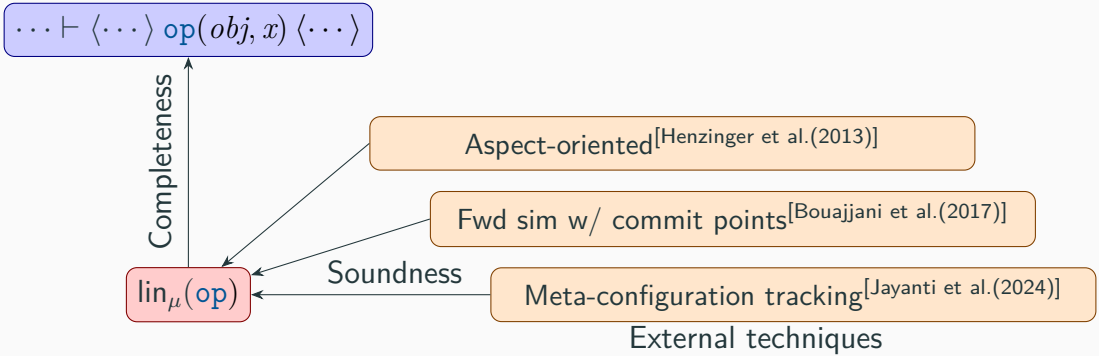
Why completeness?

- Theoretically interesting
- Embedding external techniques



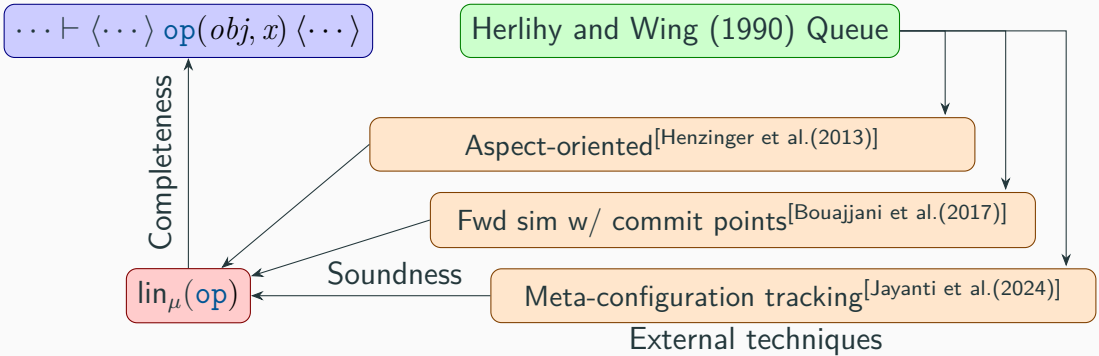
Why completeness?

- Theoretically interesting
- Embedding external techniques



Why completeness?

- Theoretically interesting
- Embedding external techniques



Conclusion

Part 1.

- Language-independent whole program completeness theorems
- Case studies: HeapLang, λ_{Rust}, time credits, and Eris

Conclusion

Part 1.

- Language-independent whole program completeness theorems
- Case studies: HeapLang, λ_{Rust}, time credits, and Eris

Part 2.

- Completeness of logically atomic triples
- Ported external techniques: aspect-oriented proofs, forward simulation with commit points, and meta-configuration tracking

Conclusion

Part 1. (To appear at ICFP 2026)

- Language-independent whole program completeness theorems
- Case studies: HeapLang, λ_{Rust}, time credits, and Eris

Please try it out!

(and/or join our discussion session)

`https://plf.inf.ethz.ch/research/icfp26-completeness.html`

Part 2.

- Completeness of logically atomic triples
- Ported external techniques: aspect-oriented proofs, forward simulation with commit points, and meta-configuration tracking

References i

Lars Birkedal, Thomas Dinsdale-Young, Armaël Guéneau, Guilhem Jaber, Kasper Svendsen, and Nikos Tzevelekos. 2021.
Theorems for free from separation logic specifications.
Proc. ACM Program. Lang. 5, ICFP, Article 81 (Aug. 2021), 29 pages.
doi:10.1145/3473586

Ahmed Bouajjani, Michael Emmi, Constantin Enea, and Suha Orhun Mutluergil. 2017.
Proving Linearizability Using Forward Simulations. In *Computer Aided Verification*, Rupak Majumdar and Viktor Kunčak (Eds.). Springer International Publishing, Cham, 542–563.

References ii


Stephen A. Cook. 1978.
Soundness and Completeness of an Axiom System for Program Verification.
SIAM J. Comput. 7, 1 (1978), 70–90.
doi:10.1137/0207005


Thomas A. Henzinger, Ali Sezgin, and Viktor Vafeiadis. 2013.
Aspect-Oriented Linearizability Proofs. In *CONCUR 2013 – Concurrency Theory*, Pedro R. D'Argenio and Hernán Melgratti (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 242–256.

References iii

- Maurice P. Herlihy and Jeannette M. Wing. 1990.
  Linearizability: a correctness condition for concurrent objects.
  ACM Trans. Program. Lang. Syst. 12, 3 (July 1990), 463–492.
  doi:10.1145/78969.78972
- Prasad Jayanti, Siddhartha Jayanti, Ugur Yavuz, and Lizzie Hernandez. 2024.
  A Universal, Sound, and Complete Forward Reasoning Technique for Machine-Verified Proofs of Linearizability.
  Proc. ACM Program. Lang. 8, POPL, Article 82 (jan 2024), 29 pages.
  doi:10.1145/3632924

References iv


Cliff B. Jones. 1983.
Tentative Steps Toward a Development Method for Interfering Programs.
ACM Trans. Program. Lang. Syst. 5, 4 (1983), 596–619.
doi:10.1145/69575.69577


Susan S. Owicki. 1975.
Axiomatic Proof Techniques for Parallel Programs.
Technical Report. USA.

References v


Colin Stirling. 1988.
A Generalization of Owicki-Gries's Hoare Logic for a Concurrent while Language.
Theor. Comput. Sci. 58 (1988), 347–359.
doi:10.1016/0304-3975(88)90033-3