Concurrent Separation Logic

Robbert Krebbers[1]

Radboud University Nijmegen, The Netherlands

January 22, 2026 @ CYCLIC winter school, Enschede, NL

---

[1] Based on slides by Robbert Krebbers (for the Type Theory Based Tools workshop, 2017, Paris, France), which were improved by Derek Dreyer (for MFPS, 2017, Ljubljana, Slovenia); Jacques-Henri Jourdan and Robbert Krebbers (for the POPL'18 tutorial, Los Angeles, USA); Joseph Tassarotti, Ralf Jung and Tej Chajed (for the POPL'21 tutorial, Online); and Robbert Krebbers (for the Program Verification course @ RU, 2021–2025)

Outline

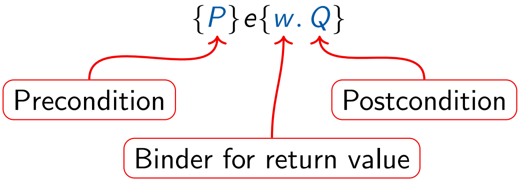
- **Part 1:** Working with invariants and ghost state
- **Part 2:** Modeling ghost state via “PCMs”
- **Hands-on Iris:** Work on the exercises at `https://gitlab.mpi-sws.org/iris/tutorial-popl21`

Part 1:

Working with invariants and ghost state

Hoare triples

Hoare triples for partial program correctness:



$$\{P\}\,e\,\{w.\;Q\}$$

Precondition

Binder for return value

Postcondition

If the initial state satisfies P, then:

- e does not get stuck/crash
- if e terminates with value v, the final state satisfies $Q[v/w]$

Separation logic [O'Hearn, Reynolds, Yang 2001]

The points-to connective $x \mapsto v$

- provides the knowledge that location x has value v, and
- provides exclusive ownership of x

Separating conjunction $P * Q$:
the state consists of *disjoint parts* satisfying P and Q

Separation logic [O'Hearn, Reynolds, Yang 2001]

The points-to connective $x \mapsto v$

- provides the knowledge that location x has value v, and
- provides exclusive ownership of x

Separating conjunction $P * Q$:
the state consists of *disjoint parts* satisfying P and Q

Example:

$$\{x \mapsto v_1 * y \mapsto v_2\}\mathit{swap}(x, y)\{w.\, w = () \land x \mapsto v_2 * y \mapsto v_1\}$$

the $*$ ensures that x and y are different

Concurrent separation logic [O'Hearn, Brookes 2004]

The *par* rule:

$$\frac{\{P_1\}\, e_1\, \{Q_1\} \qquad \{P_2\}\, e_2\, \{Q_2\}}{\{P_1 * P_2\}\, e_1 || e_2\, \{Q_1 * Q_2\}}$$

Concurrent separation logic [O'Hearn, Brookes 2004]

The *par* rule:

$$\frac{\{P_1\}\, e_1 \,\{Q_1\} \qquad \{P_2\}\, e_2 \,\{Q_2\}}{\{P_1 * P_2\}\, e_1 || e_2 \,\{Q_1 * Q_2\}}$$

For example:

$$\{x \mapsto 4 * y \mapsto 6\}$$

$$x := !\,x + 2 \;\Big\|\; y := !\,y + 2$$

$$\{x \mapsto 6 * y \mapsto 8\}$$

Concurrent separation logic [O'Hearn, Brookes 2004]

The *par* rule:

$$\frac{\{P_1\}\, e_1\, \{Q_1\} \qquad \{P_2\}\, e_2\, \{Q_2\}}{\{P_1 * P_2\}\, e_1 \| e_2\, \{Q_1 * Q_2\}}$$

For example:

$$\begin{array}{c}
\{x \mapsto 4 * y \mapsto 6\} \\
\begin{array}{l|l}
\{x \mapsto 4\} & \{y \mapsto 6\} \\
x := !\,x + 2 & y := !\,y + 2 \\
\{x \mapsto 6\} & \{y \mapsto 8\}
\end{array} \\
\{x \mapsto 6 * y \mapsto 8\}
\end{array}$$

Concurrent separation logic [O'Hearn, Brookes 2004]

The *par* rule:

$$\frac{\{P_1\}\, e_1 \,\{Q_1\} \qquad \{P_2\}\, e_2 \,\{Q_2\}}{\{P_1 * P_2\}\, e_1 || e_2 \,\{Q_1 * Q_2\}}$$

For example:

$$\begin{array}{c}
\{x \mapsto 4 * y \mapsto 6\} \\
\begin{array}{l||l}
\{x \mapsto 4\} & \{y \mapsto 6\} \\
x := !\,x + 2 & y := !\,y + 2 \\
\{x \mapsto 6\} & \{y \mapsto 8\}
\end{array} \\
\{x \mapsto 6 * y \mapsto 8\}
\end{array}$$

Concurrent separation logic [O'Hearn, Brookes 2004]

The *par* rule:

$$\frac{\{P_1\}\, e_1\, \{Q_1\} \qquad \{P_2\}\, e_2\, \{Q_2\}}{\{P_1 * P_2\}\, e_1 || e_2\, \{Q_1 * Q_2\}}$$

For example:

$$\begin{array}{c}
\{x \mapsto 4 * y \mapsto 6\} \\
\begin{array}{l||l}
\{x \mapsto 4\} & \{y \mapsto 6\} \\
x := !\,x + 2 & y := !\,y + 2 \\
\{x \mapsto 6\} & \{y \mapsto 8\}
\end{array} \\
\{x \mapsto 6 * y \mapsto 8\}
\end{array}$$

Works great for concurrent programs without shared memory: concurrent quick sort, etc.

What about shared state/racy programs?

A classic problem:

$$\texttt{let}\, x = \texttt{ref}(0)\, \texttt{in}$$

$$\begin{array}{l||l} \texttt{fetchandadd}(x, 2) & \texttt{fetchandadd}(x, 2) \end{array}$$

$$!\, x$$

Where $\texttt{fetchandadd}(x, y)$ is the atomic version of $x := !\, x + y$.

What about shared state/racy programs?

A classic problem:

$$\begin{array}{l}
\{\mathsf{True}\} \\
\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in} \\
\texttt{fetchandadd}(x, 2) \ \left\|\ \texttt{fetchandadd}(x, 2)\right. \\
!\, x \\
\{w.\, w = 4\}
\end{array}$$

Where $\texttt{fetchandadd}(x, y)$ is the atomic version of $x := !\, x + y$.

What about shared state/racy programs?

A classic problem:

$$
\begin{array}{l}
\{\mathsf{True}\} \\
\texttt{let}\; x = \texttt{ref}(0)\; \texttt{in} \\
\{x \mapsto 0\} \\
\texttt{fetchandadd}(x, 2) \;\left\|\; \texttt{fetchandadd}(x, 2)\right. \\
!\, x \\
\{w.\, w = 4\}
\end{array}
$$

Where $\texttt{fetchandadd}(x, y)$ is the atomic version of $x := !\, x + y$.

What about shared state/racy programs?

A classic problem:

$$\begin{array}{l}
\{\mathsf{True}\} \\
\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in} \\
\{x \mapsto 0\} \\
\begin{array}{l||l}
\{??\} & \{??\} \\
\texttt{fetchandadd}(x, 2) & \texttt{fetchandadd}(x, 2) \\
\{??\} & \{??\}
\end{array} \\
!\,x \\
\{w.\, w = 4\}
\end{array}$$

Where $\texttt{fetchandadd}(x, y)$ is the atomic version of $x := !\,x + y$.

Problem: can only give ownership of x to one thread

Invariants

The invariant assertion $\boxed{R}$ expresses that R is maintained as an invariant on the state

Invariants

The invariant assertion $\boxed{R}$ expresses that R is maintained as an invariant on the state

Invariant allocation:

$$\frac{\left\{\boxed{R} * P\right\} e \{Q\}}{\{R * P\} e \{Q\}}$$

Invariants

The invariant assertion $\boxed{R}$ expresses that R is maintained as an invariant on the state

Invariant allocation:

$$\frac{\left\{\boxed{R} * P\right\} e \{Q\}}{\{R * P\} e \{Q\}}$$

Invariant duplication: $\boxed{R} \vdash \boxed{R} * \boxed{R}$

Invariants

The invariant assertion $\boxed{R}$ expresses that R is maintained as an invariant on the state

Invariant allocation:

$$\frac{\left\{\boxed{R} * P\right\} e \{Q\}}{\{R * P\} e \{Q\}}$$

Invariant duplication: $\boxed{R} \vdash \boxed{R} * \boxed{R}$

Invariant opening:

$$\frac{\{R * P\} e \{R * Q\} \qquad e \text{ atomic}}{\left\{\boxed{R} * P\right\} e \left\{\boxed{R} * Q\right\}}$$

Invariants

The invariant assertion $\boxed{R}^{\mathcal{N}}$ expresses that R is maintained as an invariant on the state

Invariant allocation:

$$\frac{\left\{\boxed{R}^{\mathcal{N}} * P\right\} e \{Q\}_{\mathcal{E}}}{\{R * P\} e \{Q\}_{\mathcal{E}}}$$

Invariant duplication: $\boxed{R}^{\mathcal{N}} \vdash \boxed{R}^{\mathcal{N}} * \boxed{R}^{\mathcal{N}}$

Invariant opening:

$$\frac{\{R * P\} e \{R * Q\}_{\mathcal{E}} \qquad e \text{ atomic}}{\left\{\boxed{R}^{\mathcal{N}} * P\right\} e \left\{\boxed{R}^{\mathcal{N}} * Q\right\}_{\mathcal{E} \uplus \mathcal{N}}}$$

Technicalities: names prevent opening the same invariant twice

Invariants

The invariant assertion $\boxed{R}^{\mathcal{N}}$ expresses that R is maintained as an invariant on the state

Invariant allocation:

$$\frac{\left\{\boxed{R}^{\mathcal{N}} * P\right\} e \{Q\}_{\mathcal{E}}}{\{\triangleright R * P\} e \{Q\}_{\mathcal{E}}}$$

Invariant duplication: $\boxed{R}^{\mathcal{N}} \vdash \boxed{R}^{\mathcal{N}} * \boxed{R}^{\mathcal{N}}$

Invariant opening:

$$\frac{\{\triangleright R * P\} e \{\triangleright R * Q\}_{\mathcal{E}} \qquad e \text{ atomic}}{\left\{\boxed{R}^{\mathcal{N}} * P\right\} e \left\{\boxed{R}^{\mathcal{N}} * Q\right\}_{\mathcal{E} \uplus \mathcal{N}}}$$

Technicalities: names prevent opening the same invariant twice and the later $\triangleright$ is needed for impredicativity, i.e., $\boxed{\ldots \boxed{R}^{\mathcal{N}_2} \ldots}^{\mathcal{N}_1}$

Invariants in action

Let us consider a simpler problem first:

$$\{\textsf{True}\}$$

$$\texttt{let}\, x = \texttt{ref}(0)\, \texttt{in}$$

$$\texttt{fetchandadd}(x, 2) \quad \Big\| \quad \texttt{fetchandadd}(x, 2)$$

$$!\, x$$

$$\{n.\, \mathit{even}(n)\}$$

Invariants in action

Let us consider a simpler problem first:

$$\{\mathsf{True}\}$$

$$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$$

$$\{x \mapsto 0\}$$

$$\texttt{fetchandadd}(x, 2) \quad \Big\| \quad \texttt{fetchandadd}(x, 2)$$

$$!\,x$$

$$\{n.\ even(n)\}$$

Invariants in action

Let us consider a simpler problem first:

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

allocate $\boxed{\exists n.\, x \mapsto n * even(n)}$

$$\texttt{fetchandadd}(x, 2) \quad \Big\| \quad \texttt{fetchandadd}(x, 2)$$

$!\,x$

$\{n.\, even(n)\}$

Invariants in action

Let us consider a simpler problem first:

$$
\begin{array}{l}
\{\mathsf{True}\}\\
\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}\\
\{x \mapsto 0\}\\
\text{allocate } \boxed{\exists n.\, x \mapsto n * even(n)}\\
\begin{array}{l||l}
\left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\} & \left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\}\\
\quad \texttt{fetchandadd}(x, 2) & \quad \texttt{fetchandadd}(x, 2)\\
\left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\} & \left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\}
\end{array}\\
\quad !\, x\\
\{n.\, even(n)\}
\end{array}
$$

Invariants in action

Let us consider a simpler problem first:

$$
\begin{array}{l}
\{\mathsf{True}\} \\
\texttt{let}\; x = \texttt{ref}(0)\;\texttt{in} \\
\{x \mapsto 0\} \\
\text{allocate } \boxed{\exists n.\, x \mapsto n * even(n)} \\
\begin{array}{l||l}
\{\boxed{\exists n.\, x \mapsto n * even(n)}\} & \{\boxed{\exists n.\, x \mapsto n * even(n)}\} \\
\quad \{x \mapsto n * even(n)\} & \\
\quad \texttt{fetchandadd}(x, 2) & \quad \texttt{fetchandadd}(x, 2) \\
\quad \{x \mapsto n + 2 * even(n + 2)\} & \\
\{\boxed{\exists n.\, x \mapsto n * even(n)}\} & \{\boxed{\exists n.\, x \mapsto n * even(n)}\}
\end{array} \\
\\
\quad !\, x \\
\\
\{n.\, even(n)\}
\end{array}
$$

Invariants in action

Let us consider a simpler problem first:

$\{\text{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

allocate $\boxed{\exists n.\, x \mapsto n * even(n)}$

$$\begin{array}{l||l}
\left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\} & \left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\} \\
\quad \{x \mapsto n * even(n)\} & \quad \{x \mapsto n * even(n)\} \\
\quad \texttt{fetchandadd}(x, 2) & \quad \texttt{fetchandadd}(x, 2) \\
\quad \{x \mapsto n + 2 * even(n + 2)\} & \quad \{x \mapsto n + 2 * even(n + 2)\} \\
\left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\} & \left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\}
\end{array}$$

$!\, x$

$\{n.\, even(n)\}$

Invariants in action

Let us consider a simpler problem first:

$\{\mathsf{True}\}$

$\texttt{let}\; x = \texttt{ref}(0)\; \texttt{in}$

$\{x \mapsto 0\}$

allocate $\boxed{\exists n.\, x \mapsto n * even(n)}$

$$\begin{array}{l||l}
\{\boxed{\exists n.\, x \mapsto n * even(n)}\} & \{\boxed{\exists n.\, x \mapsto n * even(n)}\} \\
\quad \{x \mapsto n * even(n)\} & \quad \{x \mapsto n * even(n)\} \\
\quad \texttt{fetchandadd}(x, 2) & \quad \texttt{fetchandadd}(x, 2) \\
\quad \{x \mapsto n + 2 * even(n + 2)\} & \quad \{x \mapsto n + 2 * even(n + 2)\} \\
\{\boxed{\exists n.\, x \mapsto n * even(n)}\} & \{\boxed{\exists n.\, x \mapsto n * even(n)}\}
\end{array}$$

$\quad \{x \mapsto n * even(n)\}$

$\quad !\, x$

$\quad \{n.\, x \mapsto n * even(n)\}$

$\{n.\, even(n)\}$

Invariants in action

Let us consider a simpler problem first:

$$\{\mathsf{True}\}$$

$$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$$

$$\{x \mapsto 0\}$$

$$\text{allocate } \boxed{\exists n.\, x \mapsto n * even(n)}$$

$$\begin{array}{l||l}
\left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\} & \left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\} \\
\quad \{x \mapsto n * even(n)\} & \quad \{x \mapsto n * even(n)\} \\
\quad \texttt{fetchandadd}(x, 2) & \quad \texttt{fetchandadd}(x, 2) \\
\quad \{x \mapsto n + 2 * even(n + 2)\} & \quad \{x \mapsto n + 2 * even(n + 2)\} \\
\left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\} & \left\{\boxed{\exists n.\, x \mapsto n * even(n)}\right\}
\end{array}$$

$$\quad \{x \mapsto n * even(n)\}$$

$$\quad !\, x$$

$$\quad \{n.\, x \mapsto n * even(n)\}$$

$$\{n.\, even(n)\}$$

Problem: still cannot prove it returns 4

Ghost variables

Consider the invariant:

$$\boxed{\exists n.\, x \mapsto n * \ldots \qquad\qquad\qquad\qquad\qquad\qquad}$$

How to avoid **information loss** due to existential quantification?

Ghost variables

Consider the invariant:

$$\boxed{\exists n.\, x \mapsto n * \ldots}$$

How to avoid **information loss** due to existential quantification?

Solution: ghost variables

Ghost variables

Consider the invariant:

$$\boxed{\exists n_1, n_2.\, x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2}$$

How to avoid **information loss** due to existential quantification?

Solution: ghost variables

Ghost variables come in "entangled" pairs:

$$\underbrace{\gamma \hookrightarrow_{\bullet} n}_{\substack{\text{in the invariant} \\ (\text{“}\textbf{authoritative}\text{”})}} \quad * \quad \underbrace{\gamma \hookrightarrow_{\circ} n}_{\substack{\text{in the Hoare triple} \\ (\text{“}\textbf{fragment}\text{”})}}$$

Ghost variables

Consider the invariant:

$$\boxed{\exists n_1, n_2.\, x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$$

How to avoid **information loss** due to existential quantification?

Solution: ghost variables

Ghost variables come in “entangled” pairs:

$$\text{True} \quad \Rrightarrow\!\!\ast \quad \exists\gamma. \quad \underbrace{\gamma \hookrightarrow_\bullet n}_{\substack{\text{in the invariant} \\ (\text{“}\textbf{authoritative}\text{”})}} \quad * \quad \underbrace{\gamma \hookrightarrow_\circ n}_{\substack{\text{in the Hoare triple} \\ (\text{“}\textbf{fragment}\text{”})}}$$

Ghost variables

Consider the invariant:

$$\boxed{\exists n_1, n_2.\, x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$$

How to avoid **information loss** due to existential quantification?

Solution: ghost variables

Ghost variables come in “entangled” pairs:

$$\text{True} \quad \Rrightarrow\!\!\!\!\ast \quad \exists \gamma.\ \underbrace{\gamma \hookrightarrow_\bullet n}_{\substack{\text{in the invariant} \\ \text{(``\textbf{authoritative}'')}}} \quad * \quad \underbrace{\gamma \hookrightarrow_\circ n}_{\substack{\text{in the Hoare triple} \\ \text{(``\textbf{fragment}'')}}}$$

When you own both parts you obtain that the values are equal and can update both parts:

$$\gamma \hookrightarrow_\bullet n * \gamma \hookrightarrow_\circ m \quad \Rightarrow \quad n = m$$

$$\gamma \hookrightarrow_\bullet n * \gamma \hookrightarrow_\circ m \quad \Rrightarrow\!\!\!\!\ast \quad \gamma \hookrightarrow_\bullet n' * \gamma \hookrightarrow_\circ n'$$

Aside: Where do these rules come from?

We introduced invariants and now ghost variables, with rules for using them

Where do these rules come from?

Aside: Where do these rules come from?

We introduced invariants and now ghost variables, with rules for using them

Where do these rules come from?

We will get to that in **Part 2** of this lecture

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\, x = \texttt{ref}(0)\, \texttt{in}$

$\texttt{fetchandadd}(x, 2) \quad \| \quad \texttt{fetchandadd}(x, 2)$

$!\, x$

$\{n.\ n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\texttt{fetchandadd}(x, 2) \quad || \quad \texttt{fetchandadd}(x, 2)$

$!\,x$

$\{n.\ n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

$\texttt{fetchandadd}(x, 2) \quad \Vert \quad \texttt{fetchandadd}(x, 2)$

$!\, x$

$\{n.\ n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\texttt{fetchandadd}(x, 2) \quad \Vert \quad \texttt{fetchandadd}(x, 2)$

$!\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$$\texttt{fetchandadd}(x, 2) \quad \Big\| \quad \texttt{fetchandadd}(x, 2)$$

$!\,x$

$\{n.\ n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$
$\mathtt{let}\ x = \mathtt{ref}(0)\ \mathtt{in}$
$\{x \mapsto 0\}$
$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$
allocate $\boxed{\exists n_1, n_2 .\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$
$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0\}$

$\mathtt{fetchandadd}(x, 2)$

$\parallel$

$\{\gamma_2 \hookrightarrow_\circ 0\}$

$\mathtt{fetchandadd}(x, 2)$

$!\,x$

$\{n.\ n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$$\begin{array}{l||l} \{\gamma_1 \hookrightarrow_\circ 0\} & \{\gamma_2 \hookrightarrow_\circ 0\} \\ \quad \texttt{fetchandadd}(x, 2) & \quad \texttt{fetchandadd}(x, 2) \\ \{\gamma_1 \hookrightarrow_\circ 2\} & \{\gamma_2 \hookrightarrow_\circ 2\} \end{array}$$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2\}$

$\quad !\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\mathtt{let}\ x = \mathtt{ref}(0)\ \mathtt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\mathtt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_\circ 2\}$

$\|$

$\{\gamma_2 \hookrightarrow_\circ 0\}$

$\mathtt{fetchandadd}(x, 2)$

$\{\gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2\}$

$!\,x$

$\{n.\ n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_\circ 2\}$

$\|$

$\{\gamma_2 \hookrightarrow_\circ 0\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2\}$

$!\,x$

$\{n.\ n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_\circ 2\}$

$||$

$\{\gamma_2 \hookrightarrow_\circ 0\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2\}$

$!\,x$

$\{n.\ n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2\}$

$\parallel$

$\{\gamma_2 \hookrightarrow_{\circ} 0\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_2 \hookrightarrow_{\circ} 2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2\}$

$!\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2\}$

$\|$

$\{\gamma_2 \hookrightarrow_{\circ} 0\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_2 \hookrightarrow_{\circ} 2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2\}$

$!\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_\bullet 2 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2\}$

$\|$

$\{\gamma_2 \hookrightarrow_\circ 0\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2\}$

$!\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0\}$

- $\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$
- $\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$
- $\texttt{fetchandadd}(x, 2)$
- $\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$
- $\{\gamma_1 \hookrightarrow_{\circ} 2 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 2 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2\}$

$\parallel$

$\{\gamma_2 \hookrightarrow_{\circ} 0\}$

- $\{\ldots\}$
- $\texttt{fetchandadd}(x, 2)$
- $\{\ldots\}$

$\{\gamma_2 \hookrightarrow_{\circ} 2\}$

$!\, x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\mathtt{let}\ x = \mathtt{ref}(0)\ \mathtt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\mathtt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_\bullet 2 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2\}$

$\|$

$\{\gamma_2 \hookrightarrow_\circ 0\}$

$\{\ldots\}$

$\mathtt{fetchandadd}(x, 2)$

$\{\ldots\}$

$\{\gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$!\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\mathtt{let}\ x = \mathtt{ref}(0)\ \mathtt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\mathtt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 2 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2\}$

$\|$

$\{\gamma_2 \hookrightarrow_{\circ} 0\}$

$\{\ldots\}$

$\mathtt{fetchandadd}(x, 2)$

$\{\ldots\}$

$\{\gamma_2 \hookrightarrow_{\circ} 2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$!\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0\} \quad \| \quad \{\gamma_2 \hookrightarrow_{\circ} 0\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 2 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\ldots\}$

$\texttt{fetchandadd}(x, 2)$

$\{\ldots\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2\} \quad \| \quad \{\gamma_2 \hookrightarrow_{\circ} 2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2 * x \mapsto 4 * \gamma_1 \hookrightarrow_{\bullet} 2 * \gamma_2 \hookrightarrow_{\bullet} 2\}$

$!\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\text{allocate}\ \boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_\bullet 2 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2\}$

$\|$

$\{\gamma_2 \hookrightarrow_\circ 0\}$

$\{\ldots\}$

$\texttt{fetchandadd}(x, 2)$

$\{\ldots\}$

$\{\gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2 * x \mapsto 4 * \gamma_1 \hookrightarrow_\bullet 2 * \gamma_2 \hookrightarrow_\bullet 2\}$

$!\,x$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathrm{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2}$

$\{\gamma_1 \hookrightarrow_{\circ} 0 * \gamma_2 \hookrightarrow_{\circ} 0\}$

$$\begin{array}{l||l}
\{\gamma_1 \hookrightarrow_{\circ} 0\} & \{\gamma_2 \hookrightarrow_{\circ} 0\} \\
\quad \{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\} & \\
\quad \{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\} & \quad \{\ldots\} \\
\quad \texttt{fetchandadd}(x, 2) & \quad \texttt{fetchandadd}(x, 2) \\
\quad \{\gamma_1 \hookrightarrow_{\circ} 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 0 * \gamma_2 \hookrightarrow_{\bullet} n_2\} & \quad \{\ldots\} \\
\quad \{\gamma_1 \hookrightarrow_{\circ} 2 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_{\bullet} 2 * \gamma_2 \hookrightarrow_{\bullet} n_2\} & \\
\{\gamma_1 \hookrightarrow_{\circ} 2\} & \{\gamma_2 \hookrightarrow_{\circ} 2\}
\end{array}$$

$\{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2\}$

$\quad \{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_{\bullet} n_1 * \gamma_2 \hookrightarrow_{\bullet} n_2\}$

$\quad \{\gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2 * x \mapsto 4 * \gamma_1 \hookrightarrow_{\bullet} 2 * \gamma_2 \hookrightarrow_{\bullet} 2\}$

$\quad !\, x$

$\quad \{n.\, n = 4 * \gamma_1 \hookrightarrow_{\circ} 2 * \gamma_2 \hookrightarrow_{\circ} 2 * x \mapsto 4 * \gamma_1 \hookrightarrow_{\bullet} 2 * \gamma_2 \hookrightarrow_{\bullet} 2\}$

$\{n.\, n = 4\}$

Ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\{x \mapsto 0 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\circ 0\}$

allocate $\boxed{\exists n_1, n_2.\, x \mapsto n_1 + n_2 * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2}$

$\{\gamma_1 \hookrightarrow_\circ 0 * \gamma_2 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto n_2 * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\texttt{fetchandadd}(x, 2)$

$\{\gamma_1 \hookrightarrow_\circ 0 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_\bullet 0 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * x \mapsto (2 + n_2) * \gamma_1 \hookrightarrow_\bullet 2 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2\}$

$\|$

$\{\gamma_2 \hookrightarrow_\circ 0\}$

$\{\ldots\}$

$\texttt{fetchandadd}(x, 2)$

$\{\ldots\}$

$\{\gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2 * x \mapsto (n_1 + n_2) * \gamma_1 \hookrightarrow_\bullet n_1 * \gamma_2 \hookrightarrow_\bullet n_2\}$

$\{\gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2 * x \mapsto 4 * \gamma_1 \hookrightarrow_\bullet 2 * \gamma_2 \hookrightarrow_\bullet 2\}$

$!\,x$

$\{n.\, n = 4 * \gamma_1 \hookrightarrow_\circ 2 * \gamma_2 \hookrightarrow_\circ 2 * x \mapsto 4 * \gamma_1 \hookrightarrow_\bullet 2 * \gamma_2 \hookrightarrow_\bullet 2\}$

$\{n.\, n = 4\}$

Ghost variables with fractional permissions [Boyland 2003]

What if we have n threads? Using n different ghost variables, results in different proofs for each thread. *That is not modular.*

Better way: ghost variables with a *fractional permission* $(0, 1]_{\mathbb{Q}}$:

$$\gamma \xhookrightarrow{\pi_1 + \pi_2}_{\circ} (n_1 + n_2) \quad \Leftrightarrow \quad \gamma \xhookrightarrow{\pi_1}_{\circ} n_1 * \gamma \xhookrightarrow{\pi_2}_{\circ} n_2$$

When allocating you get full ownership $(\pi = 1)$:

$$\text{True} \Rrightarrow\!\!\ast \quad \exists \gamma.\, \gamma \hookrightarrow_{\bullet} n * \gamma \xhookrightarrow{1}_{\circ} n$$

Ghost variables with fractional permissions [Boyland 2003]

What if we have n threads? Using n different ghost variables, results in different proofs for each thread. *That is not modular.*

Better way: ghost variables with a *fractional permission* $(0,1]_{\mathbb{Q}}$:

$$\gamma \xhookrightarrow{\pi_1+\pi_2}_{\circ} (n_1+n_2) \quad\Leftrightarrow\quad \gamma \xhookrightarrow{\pi_1}_{\circ} n_1 * \gamma \xhookrightarrow{\pi_2}_{\circ} n_2$$

When allocating you get full ownership ($\pi = 1$):

$$\text{True} \Rrightarrow\!\!\!\ast \quad \exists\gamma.\, \gamma \hookrightarrow_{\bullet} n * \gamma \xhookrightarrow{1}_{\circ} n$$

You only get the equality when you have *full ownership* ($\pi = 1$):

$$\gamma \hookrightarrow_{\bullet} n * \gamma \xhookrightarrow{1}_{\circ} m \quad\Rightarrow\quad n = m$$

Ghost variables with fractional permissions [Boyland 2003]

What if we have n threads? Using n different ghost variables, results in different proofs for each thread. *That is not modular.*

Better way: ghost variables with a *fractional permission* $(0,1]_{\mathbb{Q}}$:

$$\gamma \xhookrightarrow{\pi_1+\pi_2}_{\circ} (n_1 + n_2) \quad \Leftrightarrow \quad \gamma \overset{\pi_1}{\hookrightarrow}_{\circ} n_1 * \gamma \overset{\pi_2}{\hookrightarrow}_{\circ} n_2$$

When allocating you get full ownership ($\pi = 1$):

$$\text{True} \Rrightarrow\!\!\!\!* \quad \exists \gamma.\, \gamma \hookrightarrow_{\bullet} n * \gamma \overset{1}{\hookrightarrow}_{\circ} n$$

You only get the equality when you have *full ownership* ($\pi = 1$):

$$\gamma \hookrightarrow_{\bullet} n * \gamma \overset{1}{\hookrightarrow}_{\circ} m \quad \Rightarrow \quad n = m$$

Updating is possible with *partial ownership* ($0 < \pi \leq 1$):

$$\gamma \hookrightarrow_{\bullet} n * \gamma \overset{\pi}{\hookrightarrow}_{\circ} m \quad \Rrightarrow\!\!\!\!* \quad \gamma \hookrightarrow_{\bullet} (n + i) * \gamma \overset{\pi}{\hookrightarrow}_{\circ} (m + i)$$

Ghost variables with fractional permissions [Boyland 2003]

What if we have n threads? Using n different ghost variables, results in different proofs for each thread. *That is not modular.*

Better way: ghost variables with a *fractional permission* $(0, 1]_{\mathbb{Q}}$:

$$\gamma \xhookrightarrow{\pi_1+\pi_2}_{\circ} (n_1 + n_2) \quad \Leftrightarrow \quad \gamma \xhookrightarrow{\pi_1}_{\circ} n_1 * \gamma \xhookrightarrow{\pi_2}_{\circ} n_2$$

When allocating you get full ownership ($\pi = 1$):

$$\text{True} \Rrightarrow\!\!\!\!* \quad \exists \gamma.\, \gamma \hookrightarrow_{\bullet} n * \gamma \xhookrightarrow{1}_{\circ} n$$

You only get the equality when you have *full ownership* ($\pi = 1$):

$$\gamma \hookrightarrow_{\bullet} n * \gamma \xhookrightarrow{1}_{\circ} m \quad \Rightarrow \quad n = m$$

Updating is possible with *partial ownership* ($0 < \pi \leq 1$):

$$\gamma \hookrightarrow_{\bullet} n * \gamma \xhookrightarrow{\pi}_{\circ} m \quad \Rrightarrow\!\!\!\!* \quad \gamma \hookrightarrow_{\bullet} (n + i) * \gamma \xhookrightarrow{\pi}_{\circ} (m + i)$$

Keeps the invariant that all $\gamma \xhookrightarrow{\pi_i}_{\circ} n_i$ sum up to $\gamma \hookrightarrow_{\bullet} \sum n_i$

Fractional ghost variables in action

$\{\mathsf{True}\}$
$\texttt{let}\, x = \texttt{ref}(0)\, \texttt{in}$

$\texttt{fetchandadd}(x, 2) \;\Big\|\; \texttt{fetchandadd}(x, 2) \;\Big\|\; \ldots$

$!\, x$

$\{n.\ n = 2k\}$

Fractional ghost variables in action

$\{\mathsf{True}\}$
$\texttt{let}\, x = \texttt{ref}(0)\, \texttt{in}$
$\{x \mapsto 0\}$

$\texttt{fetchandadd}(x, 2) \quad \Big\|\Big| \quad \texttt{fetchandadd}(x, 2) \quad \Big\|\Big| \quad \ldots$

$!\, x$

$\{n.\ n = 2k\}$

Fractional ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\, x = \texttt{ref}(0)\, \texttt{in}$

$\{x \mapsto 0\}$

$\left\{x \mapsto 0 * \gamma \hookrightarrow_{\bullet} 0 * \gamma \overset{1}{\hookrightarrow}_{\circ} 0\right\}$

$\texttt{fetchandadd}(x, 2) \quad \Big\| \quad \texttt{fetchandadd}(x, 2) \quad \Big\| \quad \ldots$

$!\, x$

$\{n.\, n = 2k\}$

Fractional ghost variables in action

$\{\mathsf{True}\}$

$\mathtt{let}\ x = \mathtt{ref}(0)\ \mathtt{in}$

$\{x \mapsto 0\}$

$\left\{x \mapsto 0 * \gamma \hookrightarrow_{\bullet} 0 * \gamma \overset{1}{\hookrightarrow}_{\circ} 0\right\}$

allocate $\boxed{\exists n.\, x \mapsto n * \gamma \hookrightarrow_{\bullet} n}$

$$\mathtt{fetchandadd}(x, 2) \quad \Big\| \quad \mathtt{fetchandadd}(x, 2) \quad \Big\| \quad \ldots$$

$!\, x$

$\{n.\, n = 2k\}$

Fractional ghost variables in action

$\{\text{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\left\{x \mapsto 0 * \gamma \hookrightarrow_{\bullet} 0 * \gamma \overset{1}{\hookrightarrow}_{\circ} 0\right\}$

allocate $\boxed{\exists n.\, x \mapsto n * \gamma \hookrightarrow_{\bullet} n}$

$$\begin{array}{l||l||l}
\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\} & \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\} & \\
\quad \texttt{fetchandadd}(x, 2) & \quad \texttt{fetchandadd}(x, 2) & \ldots \\
\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\} & \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\} &
\end{array}$$

$\quad !\, x$

$\{n.\ n = 2k\}$

Fractional ghost variables in action

$\{\mathsf{True}\}$

$\mathtt{let}\ x = \mathtt{ref}(0)\ \mathtt{in}$

$\{x \mapsto 0\}$

$\left\{x \mapsto 0 * \gamma \hookrightarrow_{\bullet} 0 * \gamma \overset{1}{\hookrightarrow}_{\circ} 0\right\}$

allocate $\boxed{\exists n.\, x \mapsto n * \gamma \hookrightarrow_{\bullet} n}$

$$\begin{array}{l||l||l}
\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\} & \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\} & \\
\quad \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0 * x \mapsto n * \gamma \hookrightarrow_{\bullet} n\right\} & & \\
\quad \mathtt{fetchandadd}(x, 2) & \mathtt{fetchandadd}(x, 2) & \ldots \\
\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\} & \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\} &
\end{array}$$

$!\,x$

$\{n.\ n = 2k\}$

Fractional ghost variables in action

$\{\text{True}\}$

$\mathtt{let}\ x = \mathtt{ref}(0)\ \mathtt{in}$

$\{x \mapsto 0\}$

$\left\{x \mapsto 0 * \gamma \hookrightarrow_{\bullet} 0 * \gamma \overset{1}{\hookrightarrow}_{\circ} 0\right\}$

allocate $\boxed{\exists n.\, x \mapsto n * \gamma \hookrightarrow_{\bullet} n}$

$$\begin{array}{l||l||l}
\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\} & \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\} & \\
\mid\ \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0 * x \mapsto n * \gamma \hookrightarrow_{\bullet} n\right\} & & \\
\mid\ \mathtt{fetchandadd}(x, 2) & \quad \mathtt{fetchandadd}(x, 2) & \ \ldots \\
\mid\ \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2 * x \mapsto (2+n) * \gamma \hookrightarrow_{\bullet} (2+n)\right\} & & \\
\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\} & \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\} &
\end{array}$$

$!\,x$

$\{n.\ n = 2k\}$

Fractional ghost variables in action

$\{\text{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\left\{x \mapsto 0 * \gamma \hookrightarrow_{\bullet} 0 * \gamma \overset{1}{\hookrightarrow}_{\circ} 0\right\}$

allocate $\boxed{\exists n.\, x \mapsto n * \gamma \hookrightarrow_{\bullet} n}$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\}$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0 * x \mapsto n * \gamma \hookrightarrow_{\bullet} n\right\}$

$\texttt{fetchandadd}(x, 2)$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2 * x \mapsto (2 + n) * \gamma \hookrightarrow_{\bullet} (2 + n)\right\}$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\}$

$\|$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\}$

$\{\ldots\}$

$\texttt{fetchandadd}(x, 2)$

$\{\ldots\}$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\}$

$\|$

$\ldots$

$!\,x$

$\{n.\ n = 2k\}$

Fractional ghost variables in action

$\{\mathsf{True}\}$

$\texttt{let}\ x = \texttt{ref}(0)\ \texttt{in}$

$\{x \mapsto 0\}$

$\left\{x \mapsto 0 * \gamma \hookrightarrow_\bullet 0 * \gamma \overset{1}{\hookrightarrow}_\circ 0\right\}$

allocate $\boxed{\exists n.\, x \mapsto n * \gamma \hookrightarrow_\bullet n}$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_\circ 0\right\}$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_\circ 0 * x \mapsto n * \gamma \hookrightarrow_\bullet n\right\}$

$\texttt{fetchandadd}(x, 2)$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_\circ 2 * x \mapsto (2+n) * \gamma \hookrightarrow_\bullet (2+n)\right\}$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_\circ 2\right\}$

$\left\{\gamma \overset{1}{\hookrightarrow}_\circ 2k * x \mapsto n * \gamma \hookrightarrow_\bullet n\right\}$

$!\,x$

$\{n.\ n = 2k\}$

$\|$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_\circ 0\right\}$

$\{\ldots\}$

$\texttt{fetchandadd}(x, 2)$

$\{\ldots\}$

$\left\{\gamma \overset{1/k}{\hookrightarrow}_\circ 2\right\}$

$\|$

$\ldots$

Fractional ghost variables in action

$\{\mathsf{True}\}$

$\mathtt{let}\ x = \mathtt{ref}(0)\ \mathtt{in}$

$\{x \mapsto 0\}$

$\left\{x \mapsto 0 * \gamma \hookrightarrow_{\bullet} 0 * \gamma \overset{1}{\hookrightarrow}_{\circ} 0\right\}$

allocate $\boxed{\exists n.\, x \mapsto n * \gamma \hookrightarrow_{\bullet} n}$

$$\begin{array}{l|l|l}
\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\} & \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0\right\} & \\
\quad \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 0 * x \mapsto n * \gamma \hookrightarrow_{\bullet} n\right\} & \quad \{\dots\} & \\
\quad \mathtt{fetchandadd}(x, 2) & \quad \mathtt{fetchandadd}(x, 2) & \dots \\
\quad \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2 * x \mapsto (2 + n) * \gamma \hookrightarrow_{\bullet} (2 + n)\right\} & \quad \{\dots\} & \\
\left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\} & \left\{\gamma \overset{1/k}{\hookrightarrow}_{\circ} 2\right\} &
\end{array}$$

$\quad \left\{\gamma \overset{1}{\hookrightarrow}_{\circ} 2k * x \mapsto n * \gamma \hookrightarrow_{\bullet} n\right\}$

$\quad !\,x$

$\quad \left\{n.\ n = 2k \wedge \gamma \overset{1}{\hookrightarrow}_{\circ} 2k * x \mapsto 2k * \gamma \hookrightarrow_{\bullet} 2k\right\}$

$\{n.\ n = 2k\}$

Part 2:

Modeling ghost state via “PCMs”

Mechanisms for concurrent reasoning

We have seen so far:

- Invariants $\boxed{R}^{\mathcal{N}}$
- Ghost variables $\gamma \hookrightarrow_{\bullet} n$ and $\gamma \hookrightarrow_{\circ} n$
- Fractional ghost variables $\gamma \hookrightarrow_{\bullet} n$ and $\gamma \overset{\pi}{\hookrightarrow}_{\circ} n$

How can we make sure we have all the mechanisms we will need?

Mechanisms for concurrent reasoning

We have seen so far:

- Invariants $\boxed{R}^{\mathcal{N}}$
- Ghost variables $\gamma \hookrightarrow_\bullet n$ and $\gamma \hookrightarrow_\circ n$
- Fractional ghost variables $\gamma \hookrightarrow_\bullet n$ and $\gamma \overset{\pi}{\hookrightarrow}_\circ n$

How can we make sure we have all the mechanisms we will need?

The Iris approach: these mechanisms can be **encoded** using a simple mechanism of *ghost resource ownership*

Resource algebras (RAs): A generalization of PCMs

Resource algebra (RA) with carrier M:

- Composition $(\cdot) : M \to M \to M$
- Validity predicate $\mathcal{V} \subseteq M$

Satisfying:

$$a \cdot b = b \cdot a \qquad a \cdot (b \cdot c) = (a \cdot b) \cdot c \qquad (a \cdot b) \in \mathcal{V} \Rightarrow a \in \mathcal{V}$$

Resource algebras (RAs): A generalization of PCMs

Resource algebra (RA) with carrier M:

- Composition $(\cdot) : M \rightarrow M \rightarrow M$
- Validity predicate $\mathcal{V} \subseteq M$

Satisfying:

$$a \cdot b = b \cdot a \qquad a \cdot (b \cdot c) = (a \cdot b) \cdot c \qquad (a \cdot b) \in \mathcal{V} \Rightarrow a \in \mathcal{V}$$

Iris provides $\boxed{a : M}^{\gamma}$ expressing ownership of an element a of resource algebra M (with name γ)

Ghost variables revisited

Resource algebra for ghost variables:

$$M \triangleq \bullet\, n \mid \circ\, n \mid \bullet\!\!\circ\, n \mid \bot$$

$$\mathcal{V} \triangleq \{a \neq \bot \mid a \in M\}$$

$$\bullet\, n \cdot \circ\, n' = \circ\, n' \cdot \bullet\, n \triangleq \begin{cases} \bullet\!\!\circ\, n & \text{if } n = n' \\ \bot & \text{otherwise} \end{cases}$$

$$\text{other combinations} \triangleq \bot$$

Ghost variables revisited

Resource algebra for ghost variables:

$$\begin{aligned} M &\triangleq \bullet\, n \mid \circ\, n \mid \bullet\!\circ\, n \mid \bot \\ \mathcal{V} &\triangleq \{a \neq \bot \mid a \in M\} \\ \bullet\, n \cdot \circ\, n' = \circ\, n' \cdot \bullet\, n &\triangleq \begin{cases} \bullet\!\circ\, n & \text{if } n = n' \\ \bot & \text{otherwise} \end{cases} \\ \text{other combinations} &\triangleq \bot \end{aligned}$$

And define:

$$\gamma \hookrightarrow_{\bullet} n \triangleq \left[\bullet\, n\right]^{\gamma} \qquad\qquad \gamma \hookrightarrow_{\circ} n \triangleq \left[\circ\, n\right]^{\gamma}$$

Ghost resource laws

Iris provides general laws for ghost resources:

$$a \in \mathcal{V} \Rrightarrow\!\!\!\ast\ \exists\gamma.\, \boxed{a}^{\gamma} \qquad \boxed{a \cdot b}^{\gamma} \Leftrightarrow \boxed{a}^{\gamma} * \boxed{b}^{\gamma} \qquad \boxed{a}^{\gamma} \Rightarrow a \in \mathcal{V}$$

Ghost resource laws

Iris provides general laws for ghost resources:

$$a \in \mathcal{V} \Rrightarrow\!\!\!\!\ast\ \exists \gamma.\, \boxed{a}^{\gamma} \qquad \boxed{a \cdot b}^{\gamma} \Leftrightarrow \boxed{a}^{\gamma} * \boxed{b}^{\gamma} \qquad \boxed{a}^{\gamma} \Rightarrow a \in \mathcal{V}$$

The ghost variable laws follow from these:

$$\text{True} \Rrightarrow\!\!\!\!\ast\ \exists \gamma.\, \boxed{\bullet\, n}^{\gamma} * \boxed{\circ\, n}^{\gamma}$$

Ghost resource laws

Iris provides general laws for ghost resources:

$$a \in \mathcal{V} \mathrel{\Rrightarrow\!\!\!\!\ast} \exists\gamma.\, \boxed{a}^{\gamma} \qquad\qquad \boxed{a \cdot b}^{\gamma} \Leftrightarrow \boxed{a}^{\gamma} \ast \boxed{b}^{\gamma} \qquad\qquad \boxed{a}^{\gamma} \Rightarrow a \in \mathcal{V}$$

The ghost variable laws follow from these:

$$\text{True} \mathrel{\Rrightarrow\!\!\!\!\ast} \exists\gamma.\, \boxed{\bullet\!\circ\, n}^{\gamma} \mathrel{\Rrightarrow\!\!\!\!\ast} \exists\gamma.\, \boxed{\bullet\, n}^{\gamma} \ast \boxed{\circ\, n}^{\gamma}$$

Ghost resource laws

Iris provides general laws for ghost resources:

$$a \in \mathcal{V} \Rrightarrow\!\!\ast\ \exists \gamma.\, \boxed{a}^{\gamma} \qquad \boxed{a \cdot b}^{\gamma} \Leftrightarrow \boxed{a}^{\gamma} * \boxed{b}^{\gamma} \qquad \boxed{a}^{\gamma} \Rightarrow a \in \mathcal{V}$$

The ghost variable laws follow from these:

$$\text{True} \Rrightarrow\!\!\ast\ \exists \gamma.\, \boxed{\bullet\!\circ\, n}^{\gamma} \Rrightarrow\!\!\ast\ \exists \gamma.\, \boxed{\bullet\, n}^{\gamma} * \boxed{\circ\, n}^{\gamma}$$

Remember:

$$\bullet\, n \cdot \circ\, n' = \circ\, n' \cdot \bullet\, n \triangleq \begin{cases} \bullet\!\circ\, n & \text{if } n = n' \\ \bot & \text{otherwise} \end{cases}$$

Ghost resource laws

Iris provides general laws for ghost resources:

$$a \in \mathcal{V} \Rrightarrow\!\!\ast\; \exists \gamma.\, \boxed{a}^{\gamma} \qquad \boxed{a \cdot b}^{\gamma} \Leftrightarrow \boxed{a}^{\gamma} * \boxed{b}^{\gamma} \qquad \boxed{a}^{\gamma} \Rightarrow a \in \mathcal{V}$$

The ghost variable laws follow from these:

$$\text{True} \Rrightarrow\!\!\ast\; \exists \gamma.\, \boxed{\bullet\!\circ n}^{\gamma} \Rrightarrow\!\!\ast\; \exists \gamma.\, \boxed{\bullet\, n}^{\gamma} * \boxed{\circ\, n}^{\gamma}$$

$$\boxed{\bullet\, n}^{\gamma} * \boxed{\circ\, m}^{\gamma} \Rightarrow n = m$$

Remember:

$$\bullet\, n \cdot \circ\, n' = \circ\, n' \cdot \bullet\, n \triangleq \begin{cases} \bullet\!\circ n & \text{if } n = n' \\ \bot & \text{otherwise} \end{cases}$$

Ghost resource laws

Iris provides general laws for ghost resources:

$$a \in \mathcal{V} \mathrel{\Rrightarrow\!\!\ast} \exists\gamma.\, \boxed{a}^{\gamma} \qquad \boxed{a \cdot b}^{\gamma} \Leftrightarrow \boxed{a}^{\gamma} * \boxed{b}^{\gamma} \qquad \boxed{a}^{\gamma} \Rightarrow a \in \mathcal{V}$$

The ghost variable laws follow from these:

$$\text{True} \mathrel{\Rrightarrow\!\!\ast} \exists\gamma.\, \boxed{\bullet\!\circ\, n}^{\gamma} \mathrel{\Rrightarrow\!\!\ast} \exists\gamma.\, \boxed{\bullet\, n}^{\gamma} * \boxed{\circ\, n}^{\gamma}$$

$$\boxed{\bullet\, n}^{\gamma} * \boxed{\circ\, m}^{\gamma} \Rightarrow (\bullet\, n \cdot \circ\, m) \in \mathcal{V} \Rightarrow n = m$$

Remember:

$$\bullet\, n \cdot \circ\, n' = \circ\, n' \cdot \bullet\, n \triangleq \begin{cases} \bullet\!\circ\, n & \text{if } n = n' \\ \bot & \text{otherwise} \end{cases}$$

$$\mathcal{V} \triangleq \{a \neq \bot \mid a \in M\}$$

Ghost resource laws

Iris provides general laws for ghost resources:

$$a \in \mathcal{V} \Rrightarrow \exists \gamma.\, \boxed{a}^{\gamma} \qquad \boxed{a \cdot b}^{\gamma} \Leftrightarrow \boxed{a}^{\gamma} * \boxed{b}^{\gamma} \qquad \boxed{a}^{\gamma} \Rightarrow a \in \mathcal{V}$$

The ghost variable laws follow from these:

$$\text{True} \Rrightarrow \exists \gamma.\, \boxed{\bullet\!\circ\, n}^{\gamma} \Rrightarrow \exists \gamma.\, \boxed{\bullet\, n}^{\gamma} * \boxed{\circ\, n}^{\gamma}$$

$$\boxed{\bullet\, n}^{\gamma} * \boxed{\circ\, m}^{\gamma} \Rightarrow (\bullet\, n \cdot \circ\, m) \in \mathcal{V} \Rightarrow n = m$$

Remember:

$$\bullet\, n \cdot \circ\, n' = \circ\, n' \cdot \bullet\, n \triangleq \begin{cases} \bullet\!\circ\, n & \text{if } n = n' \\ \bot & \text{otherwise} \end{cases}$$

$$\mathcal{V} \triangleq \{a \neq \bot \mid a \in M\}$$

Updating resources

Resources can be *updated* using frame-preserving updates:

$$\frac{a \rightsquigarrow b}{\boxed{a}^{\gamma} \Rrightarrow\!\!\ast \boxed{b}^{\gamma}} \qquad a \rightsquigarrow b \triangleq \forall a_{\mathrm{f}}.\, a \cdot a_{\mathrm{f}} \in \mathcal{V} \Rightarrow b \cdot a_{\mathrm{f}} \in \mathcal{V}$$

Updating resources

Resources can be *updated* using frame-preserving updates:

$$\frac{a \rightsquigarrow b}{\boxed{a}^{\gamma} \Rrightarrow\!\!\ast \boxed{b}^{\gamma}} \qquad a \rightsquigarrow b \triangleq \forall a_{\mathrm{f}}.\, a \cdot a_{\mathrm{f}} \in \mathcal{V} \Rightarrow b \cdot a_{\mathrm{f}} \in \mathcal{V}$$

Key idea: a resource can be updated if the update does not invalidate the resources of concurrently-running threads

$$\begin{array}{ccccccccc}
\text{Thread 1} & & \text{Thread 2} & & \ldots & & \text{Thread n} & \\
a & \cdot & a_2 & \cdot & \ldots & \cdot & a_n & \in \mathcal{V} \\
\rotatebox{-90}{$\rightsquigarrow$} & & & & & & & \\
b & \cdot & a_2 & \cdot & \ldots & \cdot & a_n & \in \mathcal{V}
\end{array}$$

Updating resources

Resources can be *updated* using frame-preserving updates:

$$\frac{a \rightsquigarrow b}{\boxed{a}^{\gamma} \Rrightarrow\!\!\ast \boxed{b}^{\gamma}} \qquad a \rightsquigarrow b \triangleq \forall a_{\mathrm{f}}.\, a \cdot a_{\mathrm{f}} \in \mathcal{V} \Rightarrow b \cdot a_{\mathrm{f}} \in \mathcal{V}$$

For ghost variables:

$$\frac{\bullet\!\circ\, n \rightsquigarrow \bullet\!\circ\, n'}{\gamma \hookrightarrow_{\bullet} n * \gamma \hookrightarrow_{\circ} n \Rrightarrow\!\!\ast \gamma \hookrightarrow_{\bullet} n' * \gamma \hookrightarrow_{\circ} n'}$$

Updating resources

Resources can be *updated* using frame-preserving updates:

$$\frac{a \rightsquigarrow b}{\boxed{a}^{\gamma} \Rrightarrow\!\!\ast\; \boxed{b}^{\gamma}} \qquad a \rightsquigarrow b \triangleq \forall a_{\mathrm{f}}.\, a \cdot a_{\mathrm{f}} \in \mathcal{V} \Rightarrow b \cdot a_{\mathrm{f}} \in \mathcal{V}$$

For ghost variables:

$$\frac{\bullet\!\circ\, n \rightsquigarrow \bullet\!\circ\, n'}{\gamma \hookrightarrow_{\bullet} n * \gamma \hookrightarrow_{\circ} n \Rrightarrow\!\!\ast\; \gamma \hookrightarrow_{\bullet} n' * \gamma \hookrightarrow_{\circ} n'}$$

$$\bullet\!\circ\, n \rightsquigarrow \bullet\!\circ\, n' = \forall a_{\mathrm{f}}.\, \bullet\!\circ\, n \cdot a_{\mathrm{f}} \in \mathcal{V} \Rightarrow \bullet\!\circ\, n' \cdot a_{\mathrm{f}} \in \mathcal{V}$$

Updating resources

Resources can be *updated* using frame-preserving updates:

$$\frac{a \rightsquigarrow b}{\boxed{a}^{\gamma} \Rrightarrow\!\!\ast\ \boxed{b}^{\gamma}} \qquad a \rightsquigarrow b \triangleq \forall a_{\mathrm{f}}.\, a \cdot a_{\mathrm{f}} \in \mathcal{V} \Rightarrow b \cdot a_{\mathrm{f}} \in \mathcal{V}$$

For ghost variables:

$$\frac{\bullet\!\circ\, n \rightsquigarrow \bullet\!\circ\, n'}{\gamma \hookrightarrow_{\bullet} n * \gamma \hookrightarrow_{\circ} n \Rrightarrow\!\!\ast\ \gamma \hookrightarrow_{\bullet} n' * \gamma \hookrightarrow_{\circ} n'}$$

$$\bullet\!\circ\, n \rightsquigarrow \bullet\!\circ\, n' = \forall a_{\mathrm{f}}.\, \bullet\!\circ\, n \cdot a_{\mathrm{f}} \in \mathcal{V} \Rightarrow \bullet\!\circ\, n' \cdot a_{\mathrm{f}} \in \mathcal{V}$$

$\bullet\!\circ\, n \cdot a_{\mathrm{f}} = \bot$, so the premise holds vacuously.

Generalizing to a library of RA combinators

Iris comes with a library of useful RA combinators

- $\textsc{Auth}(M)$: Generalizes the $\bullet$, $\circ$, $\bullet\!\circ$ construction over an arbitrary RA M – we call it the "authoritative" RA.
- $\textsc{Excl}(X)$: The "exclusive" RA, whose valid elements are the elements of X, and where composition is always undefined.
- $\textsc{Frac}$: The RA for fractions in $(0, 1]$ with addition.
- The expected RA liftings of products, sums, etc.

Using these combinators, we can easily construct the necessary models of many desired forms of ghost state:

- Ghost variables: $\textsc{Auth}\ (\textsc{Excl}\ \textsc{Nat})$
- Fractional ghost variables: $\textsc{Auth}\ (\textsc{Frac} \times \textsc{Nat}_{+})$

Many things I have not covered

Modal basis of Iris: $\square$, $\triangleright$, $\Rrightarrow$

- Persistent modality $\square\, P$: Says P holds forever, i.e., only relying on duplicable resources, such as invariants
- Later modality $\triangleright P$: Says P holds one step-index later (lower); needed to model impredicative invariants
- Update modality $\Rrightarrow P$: Says P holds after some frame-preserving update to ghost state

Higher-order ghost state, e.g., named propositions $\gamma \mapsto P$

- $\gamma \mapsto P$ means γ is a name for proposition P
- $\gamma \mapsto P * \gamma \mapsto Q \Rightarrow \triangleright(P = Q)$
- Sounds arcane, but turns out to be surprisingly useful!
- Achieved by equipping RAs with a step-indexing structure

Encoding of Iris program logic (including invariants)
within the modal base logic (with higher-order ghost state)

Homework

- Clone the tutorial lecture material `https://gitlab.mpi-sws.org/iris/tutorial-popl21`
- Read the `README.md` and install `coq-iris-heap-lang`
- Exercise 1 and 2 are about sequential programs (so you can get used to the `wp_` tactics and style of Hoare triples in Iris)
- Exercise 3, 4 and 5 are about concurrency