Concurrent Separation Logic with Probabilities

Ira Fesefeldt Joost-Pieter Katoen Thomas Noll

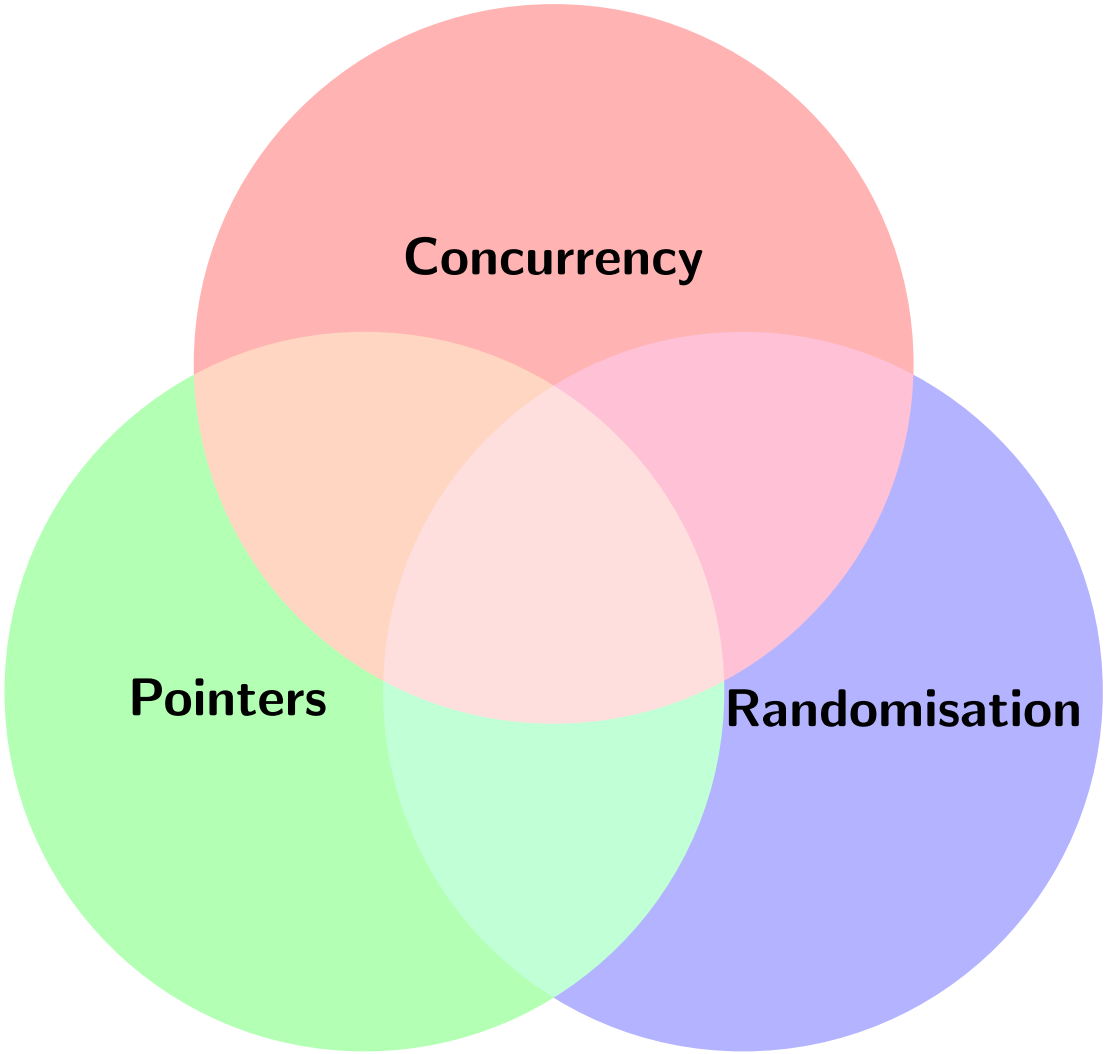
RWTH Aachen University, Germany

OPCT 2023, Bertinoro, Italy; June 26–30, 2023

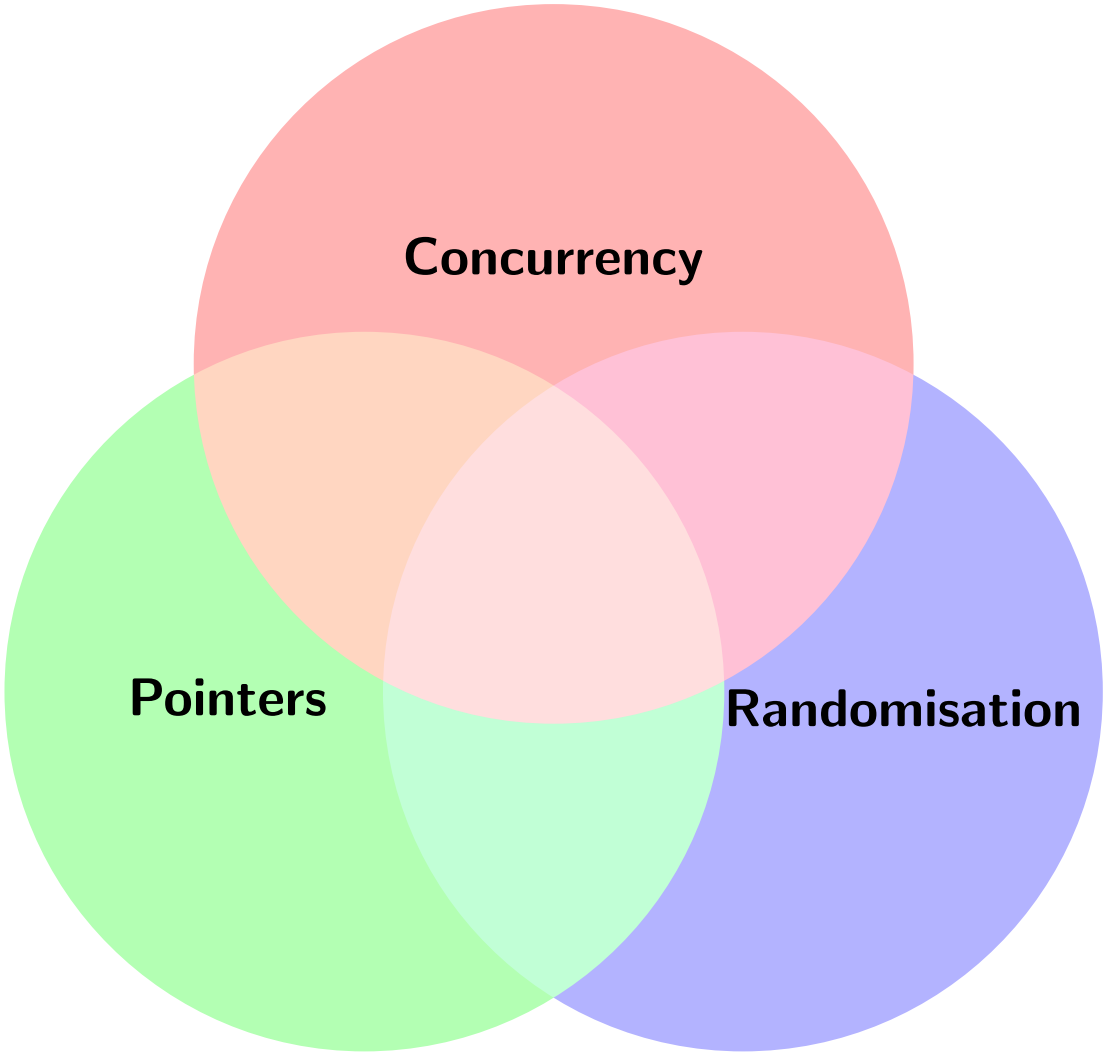
Software Modeling and Verification Chair

RWTH AACHEN UNIVERSITY

Motivation

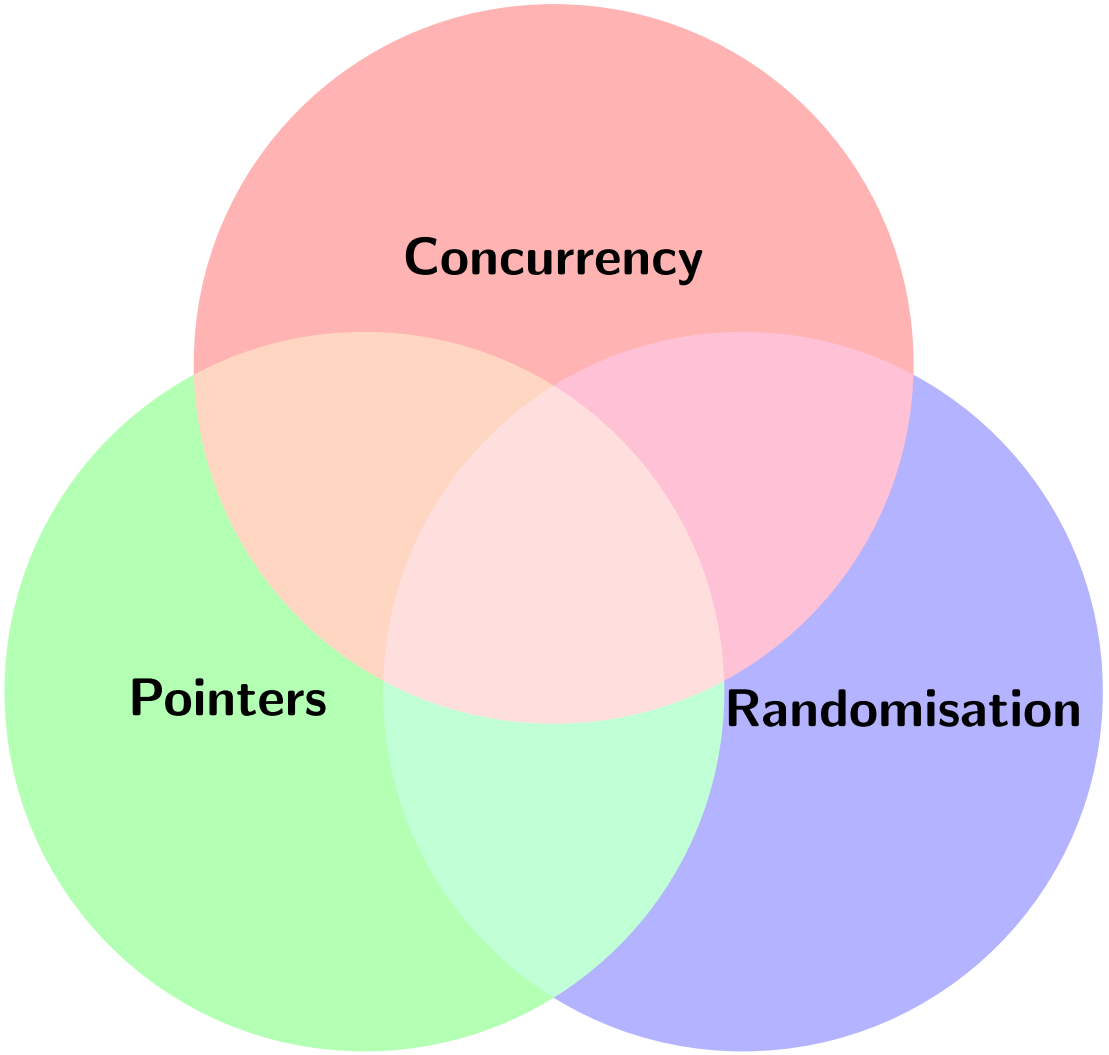


Motivation



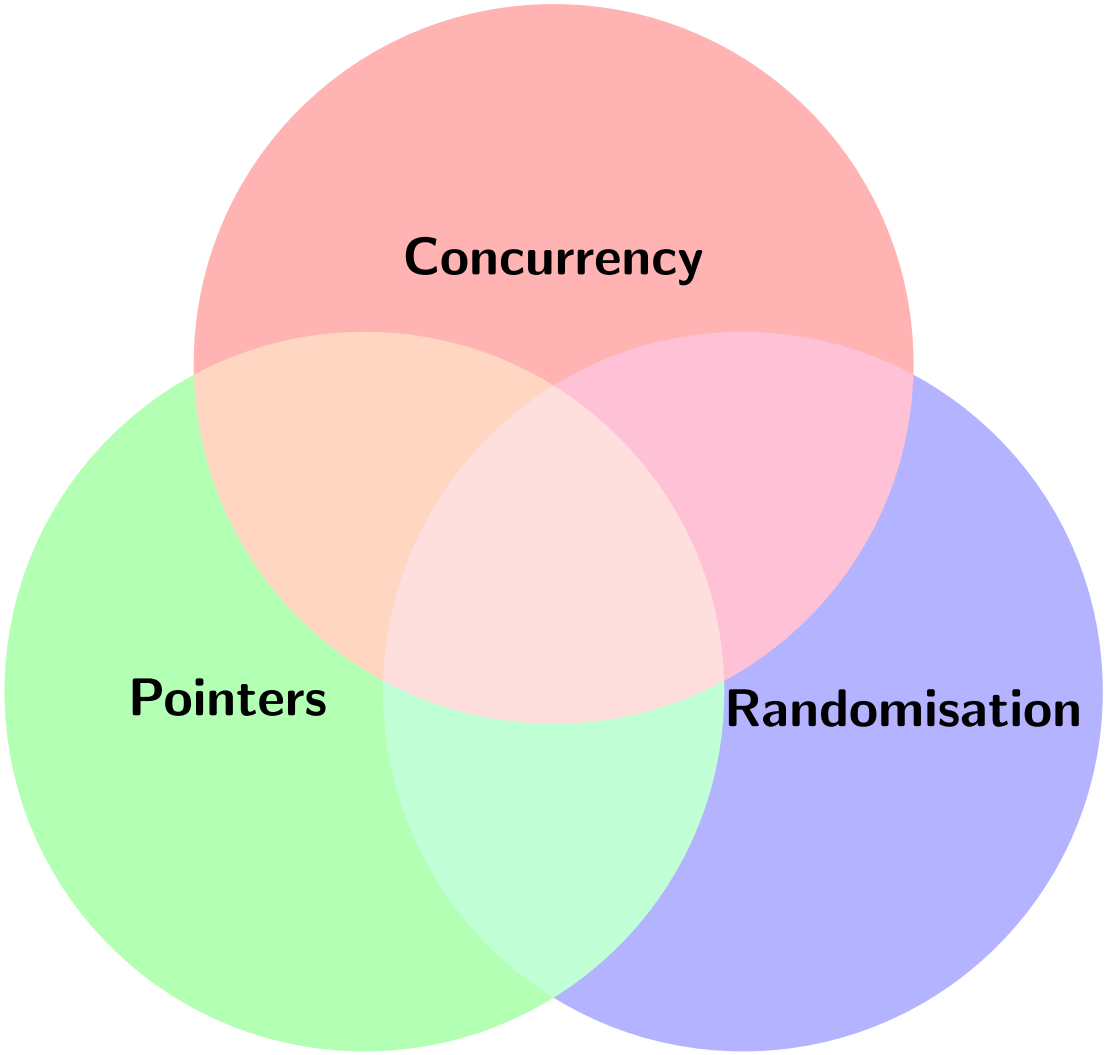
- Concurrency
  - distributed computing
  - parallel algorithms
  - ...

Motivation



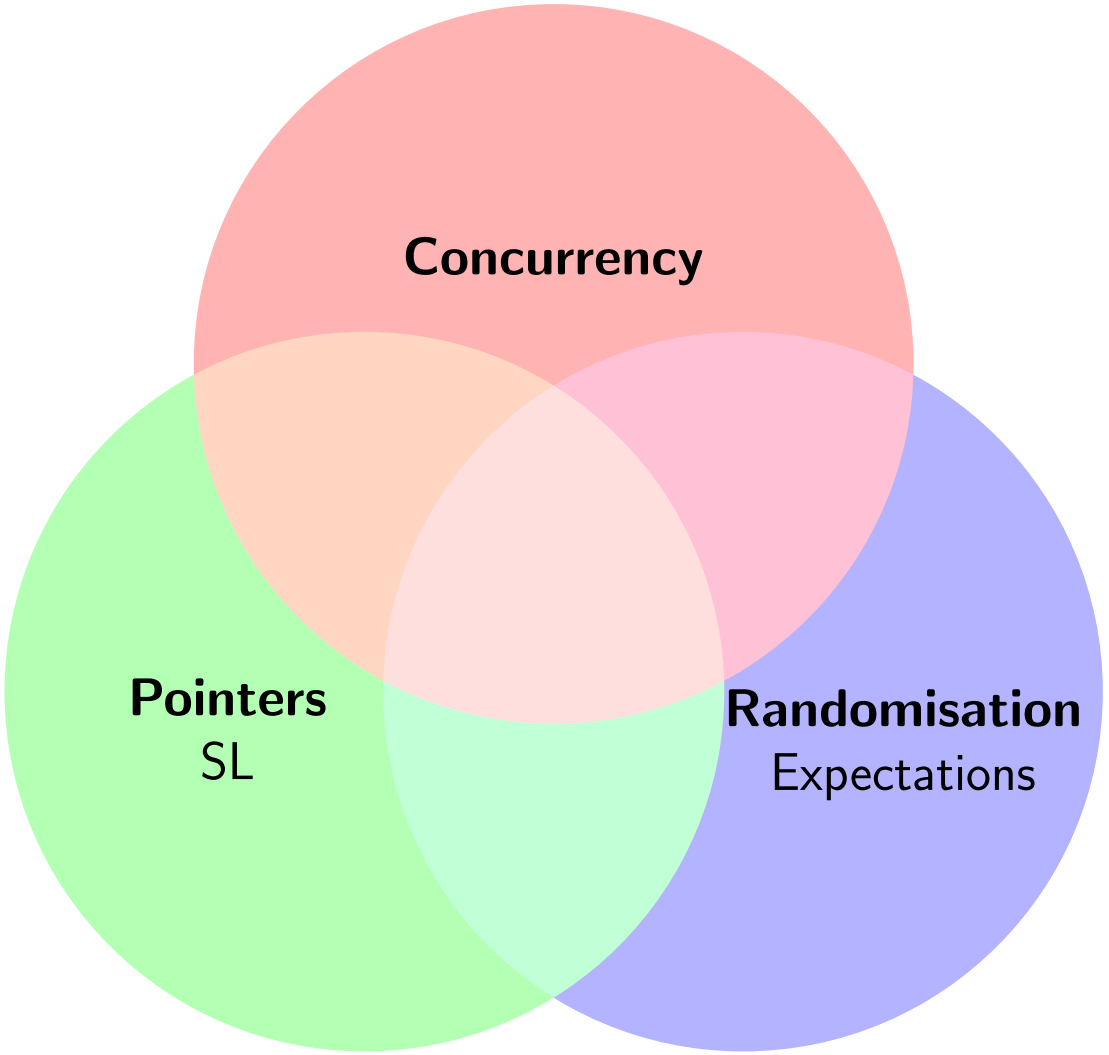
- Concurrency
  - distributed computing
  - parallel algorithms
  - ...
- Pointers
  - dynamic data structures
  - shared memory
  - ...

Motivation



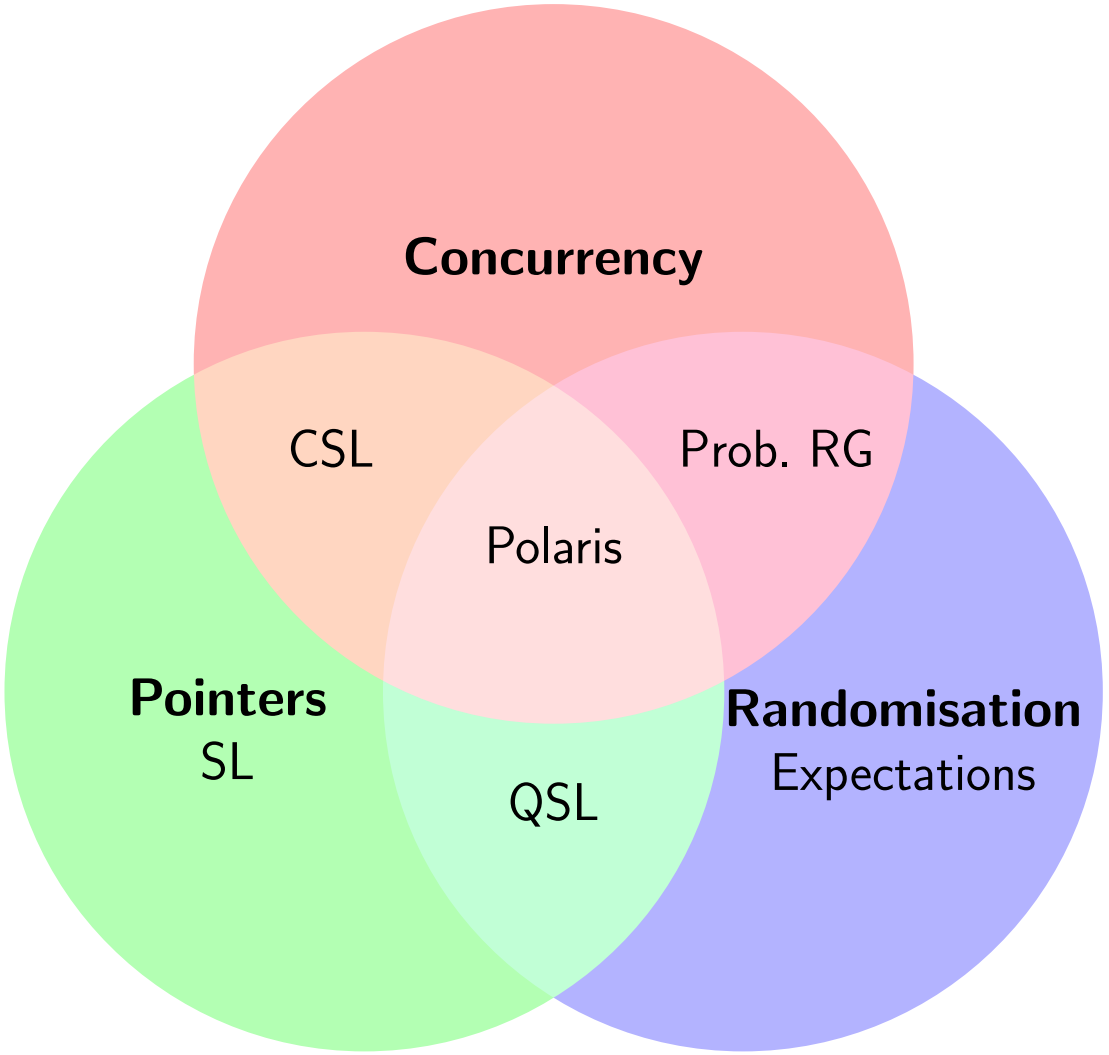
- Concurrency
  - distributed computing
  - parallel algorithms
  - ...
- Pointers
  - dynamic data structures
  - shared memory
  - ...
- Randomisation
  - randomised algorithms
  - unreliable hardware
  - ...

Motivation



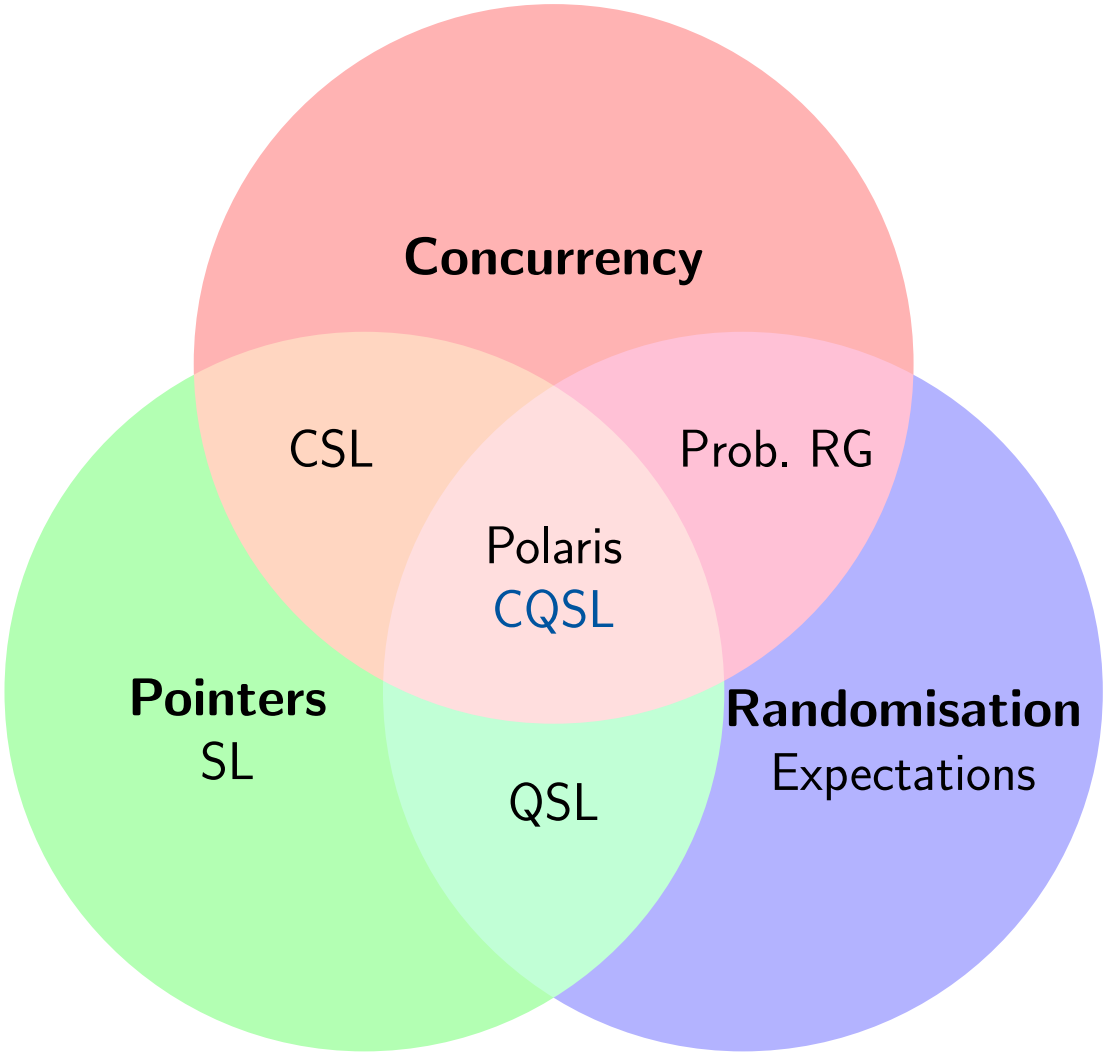
- Concurrency
  - distributed computing
  - parallel algorithms
  - ...
- Pointers
  - dynamic data structures
  - shared memory
  - ...
- Randomisation
  - randomised algorithms
  - unreliable hardware
  - ...

Motivation



- Concurrency
  - distributed computing
  - parallel algorithms
  - ...
- Pointers
  - dynamic data structures
  - shared memory
  - ...
- Randomisation
  - randomised algorithms
  - unreliable hardware
  - ...

Motivation



- Concurrency
  - distributed computing
  - parallel algorithms
  - ...
- Pointers
  - dynamic data structures
  - shared memory
  - ...
- Randomisation
  - randomised algorithms
  - unreliable hardware
  - ...

An Example

$$
\begin{array}{l}
\mathbin{/\!/} \\
\langle r \rangle := -1; \\
\mathbin{/\!/} \\
\left(\begin{array}{l}
\mathbin{/\!/} \\
\langle r \rangle := 0 \\
\\
\\
\mathbin{/\!/} \\
\mathbin{/\!/}
\end{array}\right)
\;\Big\|\;
\left(\begin{array}{l}
\mathbin{/\!/} \\
y := \langle r \rangle; \\
\mathbin{/\!/} \\
\texttt{while}\,(y = -1)\,\{\, y := \langle r \rangle \,\} \\
\mathbin{/\!/}
\end{array}\right)
\end{array}
$$

An Example

$$
\begin{array}{l}
/\!/ \\
\langle r \rangle := -1; \\
/\!/ \\
\left(\begin{array}{l} /\!/ \\ \langle r \rangle := 0 \\ \\ /\!/ \end{array} \right) \;\Big\|\; \left(\begin{array}{l} /\!/ \\ y := \langle r \rangle; \\ /\!/ \\ \texttt{while}(y = -1)\{\, y := \langle r \rangle \} \\ /\!/ \end{array} \right) \\
/\!/ \;\; y = 0
\end{array}
$$

An Example

Given resource invariant $I_R \coloneqq r \mapsto -1 \lor r \mapsto 0$, we have:

$$
\begin{array}{l}
/\!/ \ \mathit{true} \\
\texttt{<}r\texttt{>} := -1\,; \\
/\!/ \ \mathit{true} \\
\left.\begin{array}{l}
/\!/ \ \mathit{true} \\
\texttt{<}r\texttt{>} := 0 \\
\\
\\
/\!/ \ \mathit{true}
\end{array}\right\| \left.\begin{array}{l}
/\!/ \ \mathit{true} \\
y := \texttt{<}r\texttt{>}\,; \\
/\!/ \ y = -1 \lor y = 0 \\
\texttt{while}\,(y = -1)\,\{\, y := \texttt{<}r\texttt{>}\,\} \\
/\!/ \ y = 0
\end{array}\right. \\
/\!/ \ y = 0
\end{array}
$$

An Example

Given resource invariant $I_R \coloneqq r \mapsto -1 \lor r \mapsto 0$, we have:

$$
\begin{array}{l||l}
\text{// ???} & \\
\langle r \rangle := -1\,; & \\
\text{// ???} & \\
\quad \text{// ???} & \quad \text{// } \mathit{true} \\
\{\langle r \rangle := 0\} & y := \langle r \rangle\,; \\
\quad [0.5] & \quad \text{// } y = -1 \lor y = 0 \\
\{\langle r \rangle := 1\} & \texttt{while}\,(y = -1)\,\{\, y := \langle r \rangle\} \\
\quad \text{// } \mathit{true} & \quad \text{// } y = 0 \\
\text{// } y = 0 &
\end{array}
$$

Expectations and Weakest Liberal Preexpectations [McIver & Morgan, 2001]

Definition

For program C, expectations[1] $X, Y : \mathit{States} \to [0, 1]$ (with $\mathit{States} := \mathit{Stacks} \times \mathit{Heaps}$):

$$X \leq \mathsf{wlp}[\![C]\!](Y) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$

[1] Actually random variables and not expected values.

Definition

For program C, expectations[1] $X, Y : \mathit{States} \to [0,1]$ (with $\mathit{States} := \mathit{Stacks} \times \mathit{Heaps}$):

$$X \leq \mathsf{wlp}[\![C]\!](Y) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$

probability of Y when $Y : \mathit{States} \to \{0,1\}$

[1] Actually random variables and not expected values.

Expectations and Weakest Liberal Preexpectations [McIver & Morgan, 2001]

Definition

For program C, expectations[1] $X, Y : \mathit{States} \to [0,1]$ (with $\mathit{States} := \mathit{Stacks} \times \mathit{Heaps}$):

$$X \leq \mathsf{wlp}[\![C]\!](Y) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$

Example

- $0.5 \leq \mathsf{wlp}[\![C_{asgn}]\!]([r \mapsto 0])$
- $0.5 \leq \mathsf{wlp}[\![C_{asgn}]\!]([r \mapsto 1])$

$$C_{asgn} : \{ < r > := 0 \} [0.5] \{ < r > := 1 \}$$

[1] Actually random variables and not expected values.

Expectations and Weakest Liberal Preexpectations [McIver & Morgan, 2001]

Definition

For program C, expectations[1] $X, Y : \mathit{States} \to [0,1]$ (with $\mathit{States} := \mathit{Stacks} \times \mathit{Heaps}$):

$$X \leq \mathsf{wlp}[\![C]\!](Y) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$

Example

- $0.5 \leq \mathsf{wlp}[\![C_{asgn}]\!]([r \mapsto 0])$
- $0.5 \leq \mathsf{wlp}[\![C_{asgn}]\!]([r \mapsto 1])$
- $0.5 \leq \mathsf{wlp}[\![C_{run}]\!]([y = 0])$

$$C_{asgn} : \{ <r> := 0 \} \, [0.5] \, \{ <r> := 1 \}$$

$$C_{run} : \; <r> := -1 \,; \begin{array}{l|l} \{<r> := 0\} & y := <r>\,; \\ \quad [0.5] & \texttt{while}\,(y = -1)\,\{ \\ \{<r> := 1\} & \quad y := <r>\}\,; \end{array}$$

[1] Actually random variables and not expected values.

Expectations and Weakest Liberal Preexpectations [McIver & Morgan, 2001]

Definition

For program C, expectations[1] $X, Y : \mathit{States} \to [0,1]$ (with $\mathit{States} := \mathit{Stacks} \times \mathit{Heaps}$):

$$X \leq \mathsf{wlp}[\![C]\!](Y) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$

Example

- $0.5 \leq \mathsf{wlp}[\![C_{asgn}]\!]([r \mapsto 0])$
- $0.5 \leq \mathsf{wlp}[\![C_{asgn}]\!]([r \mapsto 1])$

$$C_{asgn} : \{ <r> := 0 \} [0.5] \{ <r> := 1 \}$$

- $0.5 \leq \mathsf{wlp}[\![C_{run}]\!]([y = 0])$
- $1 \leq \mathsf{wlp}[\![C_{run}]\!]([y = 0])$
  (probability of 0.5 to not terminate)

$$C_{run} : \; <r> := -1 \,; \begin{array}{l|l} \{<r> := 0\} & y := <r> \,; \\ \quad [0.5] & \texttt{while}\,(y = -1 \lor y = 1)\,\{ \\ \{<r> := 1\} & \quad y := <r>\} \,; \end{array}$$

[1] Actually random variables and not expected values.

Separating Multiplication [POPL 2019]

Separating conjunction: $\sigma \models \psi \star \varphi \quad \text{iff} \quad \exists \sigma_1, \sigma_2 : \sigma = \sigma_1 \uplus \sigma_2, \sigma_1 \models \psi \land \sigma_2 \models \varphi$

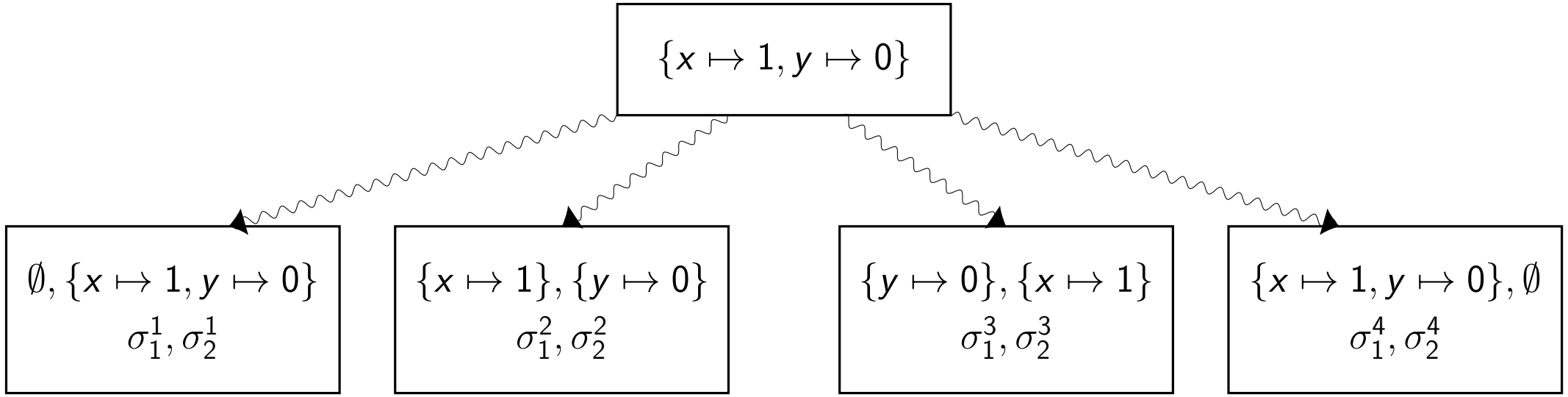
Separating Multiplication [POPL 2019]

Separating conjunction: $\sigma \models \psi \star \varphi \quad \text{iff} \quad \exists \sigma_1, \sigma_2 : \sigma = \sigma_1 \uplus \sigma_2, \sigma_1 \models \psi \land \sigma_2 \models \varphi$

Definition

For state $\sigma \in \mathit{States}$ and expectations $X, Y : \mathit{States} \to [0, 1]$, we define

$$(X \star Y)(\sigma) = \sup \{ X(\sigma_1) \cdot Y(\sigma_2) \mid \sigma = \sigma_1 \uplus \sigma_2 \}$$



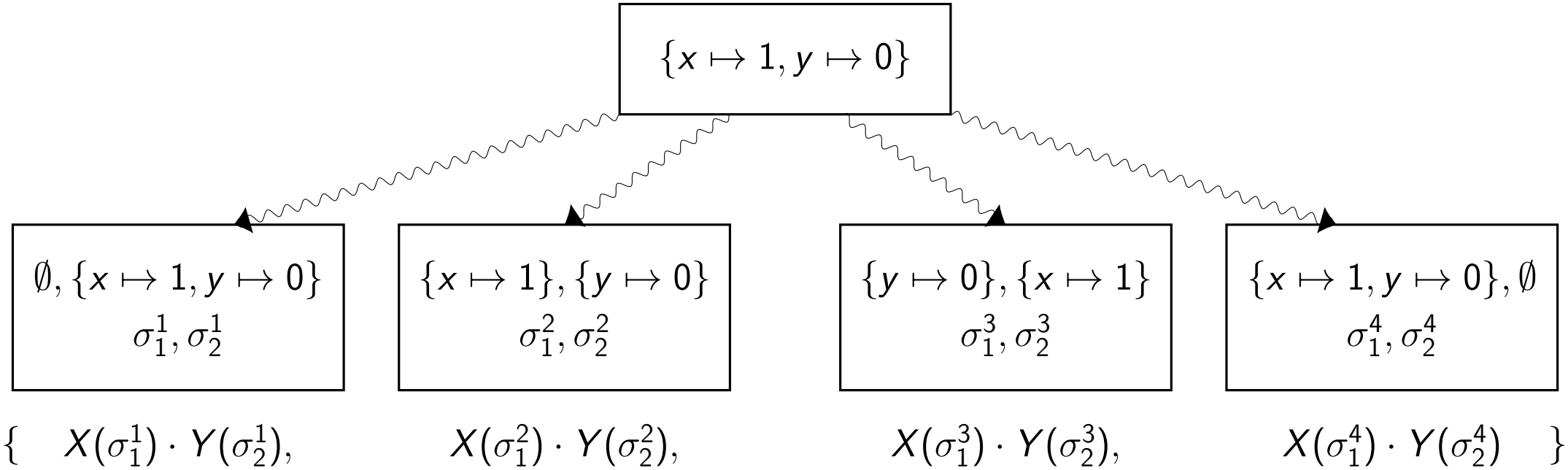
Separating Multiplication

Separating conjunction: $\sigma \models \psi \star \varphi \quad \text{iff} \quad \exists \sigma_1, \sigma_2 : \sigma = \sigma_1 \uplus \sigma_2, \sigma_1 \models \psi \land \sigma_2 \models \varphi$

Definition

For state $\sigma \in \mathit{States}$ and expectations $X, Y : \mathit{States} \to [0, 1]$, we define

$$(X \star Y)(\sigma) = \sup \{\, X(\sigma_1) \cdot Y(\sigma_2) \mid \sigma = \sigma_1 \uplus \sigma_2 \,\}$$



Quantitative Magic Wand [POPL 2019]

Qualitative magic wand (separating implication): $\sigma \models \psi \mathrel{-\!\!*} \varphi$ iff $\forall \sigma' : \sigma' \models \psi \implies \sigma \uplus \sigma' \models \varphi$

Quantitative Magic Wand [POPL 2019]

Qualitative magic wand (separating implication): $\sigma \models \psi \mathrel{-\!\!*} \varphi \quad \text{iff} \quad \forall \sigma' : \sigma' \models \psi \implies \sigma \uplus \sigma' \models \varphi$

Definition

For state $\sigma \in \mathit{States}$, predicate $\psi : \mathit{States} \to \{0,1\}$ and expectation $Y : \mathit{States} \to [0,1]$, we define

$$(\psi \mathrel{-\!\!*} Y)(\sigma) = \inf \{\, Y(\sigma \uplus \sigma') \mid \sigma' \models \psi \,\}$$

$\{x \mapsto 1\}$

$\uplus$ $\{x \mapsto 1\}$ not disjoint $\lightning$

$\uplus$ $\{y \mapsto 1\}$ $\{y \mapsto 1\} \not\models \psi$ $\lightning$

$\uplus$ $\{y \mapsto 2\}$ $\{y \mapsto 2\} \models \psi$ ✓

⋮

Quantitative Magic Wand [POPL 2019]

Qualitative magic wand (separating implication): $\sigma \models \psi \mathrel{-\!\!*} \varphi$ iff $\forall \sigma' : \sigma' \models \psi \implies \sigma \uplus \sigma' \models \varphi$

Definition

For state $\sigma \in \mathit{States}$, predicate $\psi : \mathit{States} \to \{0,1\}$ and expectation $Y : \mathit{States} \to [0,1]$, we define

$$(\psi \mathrel{-\!\!*} Y)(\sigma) = \inf \{\, Y(\sigma \uplus \sigma') \mid \sigma' \models \psi \,\}$$

$\{x \mapsto 1\}$

$\uplus$ $\{x \mapsto 1\}$ not disjoint $\lightning$

$\uplus$ $\{y \mapsto 1\}$ $\{y \mapsto 1\} \not\models \psi$ $\lightning$

$\uplus$ $\{y \mapsto 2\}$ $\{y \mapsto 2\} \models \psi$ ✓

$\vdots$

$\inf\{$

$Y(\{x \mapsto 1, y \mapsto 2\}),$

$\vdots$ $\}$

Definition

For program C, expectations $X, Y : \mathit{States} \to [0,1]$ and resource invariant $\xi : \mathit{States} \to \{0,1\}$

$$X \leq \mathsf{wrlp}[\![C]\!](Y \mid \xi) \quad \text{iff} \quad \begin{array}{l} X \text{ lower bounds the expected value of } Y \text{ after executing } C \\ \quad + \text{ probability to diverge} \\ \text{and } \xi \text{ always preserved by some sub-heap during execution} \end{array}$$

Definition

For program C, expectations $X, Y : \mathit{States} \to [0,1]$ and resource invariant $\xi : \mathit{States} \to \{0,1\}$

$$X \leq \mathsf{wrlp}[\![C]\!](Y \mid \xi) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$

and ξ always preserved by some sub-heap during execution

$$\frac{X \leq \xi \mathrel{-\!\!\!\star} \mathsf{wlp}[\![x := e]\!](Y \star \xi)}{X \leq \mathsf{wrlp}[\![x := e]\!](Y \mid \xi)}\ \text{assign-adv}$$

Definition

For program C, expectations $X, Y : \mathit{States} \to [0,1]$ and resource invariant $\xi : \mathit{States} \to \{0,1\}$

$$X \leq \mathsf{wrlp}[\![C]\!](Y \mid \xi) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$

and ξ always preserved by some sub-heap during execution

$$\frac{X \leq \xi \mathrel{-\!\!\!\circledast} \mathsf{wlp}[\![x := e]\!](Y \star \xi)}{X \leq \mathsf{wrlp}[\![x := e]\!](Y \mid \xi)}\ \text{assign-adv} \qquad \frac{X \leq \mathsf{wlp}[\![x := e]\!](Y)}{X \leq \mathsf{wrlp}[\![x := e]\!](Y \mid \xi)}\ \text{assign-easy}$$

Definition

For program C, expectations $X, Y : \mathit{States} \to [0,1]$ and resource invariant $\xi : \mathit{States} \to \{0,1\}$

$$X \leq \mathsf{wrlp}[\![C]\!](Y \mid \xi) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$
and ξ always preserved by some sub-heap during execution

$$\frac{X \leq \xi \mathrel{-\!\!\!\circledast} \mathsf{wlp}[\![x := e]\!](Y \star \xi)}{X \leq \mathsf{wrlp}[\![x := e]\!](Y \mid \xi)}\ \text{assign-adv} \qquad \frac{X \leq \mathsf{wlp}[\![x := e]\!](Y)}{X \leq \mathsf{wrlp}[\![x := e]\!](Y \mid \xi)}\ \text{assign-easy}$$

$$\frac{X_1 \leq \mathsf{wrlp}[\![C_1]\!](Y_1 \mid \xi) \quad X_2 \leq \mathsf{wrlp}[\![C_2]\!](Y_2 \mid \xi) \quad \forall i \in \{1,2\} : \mathsf{Write}(C_i) \cap \mathsf{Vars}(C_{3-i}, Y_{3-i}, \xi) = \emptyset}{X_1 \star X_2 \leq \mathsf{wrlp}[\![C_1 \parallel C_2]\!](Y_1 \star Y_2 \mid \xi)}\ \text{concur}$$

Definition

For program C, expectations $X, Y : \mathit{States} \to [0,1]$ and resource invariant $\xi : \mathit{States} \to \{0,1\}$

$$X \leq \mathsf{wrlp}[\![C]\!](Y \mid \xi) \quad \text{iff} \quad X \text{ lower bounds the expected value of } Y \text{ after executing } C + \text{probability to diverge}$$
and ξ always preserved by some sub-heap during execution

$$\frac{X \leq \xi \mathrel{-\!\!\star} \mathsf{wlp}[\![x := e]\!](Y \star \xi)}{X \leq \mathsf{wrlp}[\![x := e]\!](Y \mid \xi)}\ \text{assign-adv} \qquad \frac{X \leq \mathsf{wlp}[\![x := e]\!](Y)}{X \leq \mathsf{wrlp}[\![x := e]\!](Y \mid \xi)}\ \text{assign-easy}$$

$$\frac{X_1 \leq \mathsf{wrlp}[\![C_1]\!](Y_1 \mid \xi) \quad X_2 \leq \mathsf{wrlp}[\![C_2]\!](Y_2 \mid \xi) \quad \forall i \in \{1,2\} : \mathsf{Write}(C_i) \cap \mathsf{Vars}(C_{3-i}, Y_{3-i}, \xi) = \emptyset}{X_1 \star X_2 \leq \mathsf{wrlp}[\![C_1 \parallel C_2]\!](Y_1 \star Y_2 \mid \xi)}\ \text{concur}$$

$$\frac{X \leq \mathsf{wrlp}[\![C]\!](Y \mid \xi \star \pi)}{X \star \pi \leq \mathsf{wrlp}[\![C]\!](Y \star \pi \mid \xi)}\ \text{share}$$

Proof for Running Example C_{run}

Example

Pick $\xi = \max\{\,[r \mapsto -1],\ [r \mapsto 0]\,\}$:

$$
\begin{array}{l}
\quad <r> := -1 \\
\\
\{ <r> := 0 \}\,[0.5]\,\{ <r> := 1 \} \quad \Big\| \quad y := <r>; \\
\\
\quad /\!/ \ \ 1 \star [y = 0] \ \ \mid \ \ \xi \qquad\qquad \texttt{while}\,(y = -1)\,\{ y := <r> \};
\end{array}
$$

Proof for Running Example C_{run}

Example

Pick $\xi = \max\{\,[r \mapsto -1],\ [r \mapsto 0]\,\}$:

$$
\begin{array}{l}
\texttt{<}r\texttt{>} := -1 \\
\\
\{\,\texttt{<}r\texttt{>} := 0\,\}\,[0.5]\,\{\,\texttt{<}r\texttt{>} := 1\,\} \\
\quad /\!/ \ 1 \ \mid \ \xi \\
\\
\quad /\!/ \ 1 \star [y = 0] \ \mid \ \xi
\end{array}
\quad \Big\| \quad
\begin{array}{l}
y := \texttt{<}r\texttt{>}\,; \\
\\
\texttt{while}\,(y = -1)\,\{\,y := \texttt{<}r\texttt{>}\,\}\,; \\
\quad /\!/ \ [y = 0] \ \mid \ \xi
\end{array}
$$

Proof for Running Example C_{run}

Example

Pick $\xi = \max\{\,[r \mapsto -1],\ [r \mapsto 0]\,\}$:

$$
\begin{array}{l}
\quad <r> := -1 \\
\\
\quad /\!/\ 0.5 \ \mid\ \xi \\
\{<r> := 0\}\,[0.5]\,\{<r> := 1\} \\
\quad /\!/\ 1 \ \mid\ \xi \\
\\
\quad /\!/\ 1 \star [y = 0] \ \mid\ \xi
\end{array}
\quad \Big\| \quad
\begin{array}{l}
y := <r>; \\
\\
\texttt{while}\,(y = -1)\,\{\,y := <r>\,\}\,; \\
\quad /\!/\ [y = 0] \ \mid\ \xi
\end{array}
$$

Proof for Running Example C_{run}

Example

Pick $\xi = \max\{\,[r \mapsto -1],\ [r \mapsto 0]\,\}$:

$$
\begin{array}{l}
\texttt{<}r\texttt{>} := -1 \\
\\
\quad /\!/\ 0.5 \ \mid\ \xi \\
\{\,\texttt{<}r\texttt{>} := 0\,\}\,[0.5]\,\{\,\texttt{<}r\texttt{>} := 1\,\} \\
\quad /\!/\ 1 \ \mid\ \xi \\
\\
/\!/\ \ 1 \star [y = 0] \ \ \mid\ \ \xi
\end{array}
\quad \Big\| \quad
\begin{array}{l}
/\!/\ \ 1 \ \mid\ \xi \\
y := \texttt{<}r\texttt{>}\,; \\
\quad /\!/\ \max\{\,[y = -1],\ [y = 0]\,\} \ \mid\ \xi \\
\texttt{while}\,(y = -1)\,\{\,y := \texttt{<}r\texttt{>}\,\}\,; \\
\quad /\!/\ \ [y = 0] \ \mid\ \xi
\end{array}
$$

Proof for Running Example C_{run}

Example

Pick $\xi = \max\{\,[r \mapsto -1],\ [r \mapsto 0]\,\}$:

$$
\begin{array}{l}
\quad /\!/\ \ 0.5 \ \mid\ \xi \\
<r> := -1 \\
\quad /\!/\ \ 0.5 \star 1 \ \mid\ \xi \\
\\
\quad /\!/\ \ 0.5 \ \mid\ \xi \\
\{\,<r> := 0\,\}\,[0.5]\,\{\,<r> := 1\,\} \\
\quad /\!/\ \ 1 \ \mid\ \xi \\
\\
\quad /\!/\ \ 1 \star [y = 0] \ \mid\ \xi
\end{array}
\quad \Big\| \quad
\begin{array}{l}
\quad /\!/\ \ 1 \ \mid\ \xi \\
y := <r>; \\
\quad /\!/\ \ \max\{\,[y = -1],\ [y = 0]\,\} \ \mid\ \xi \\
\texttt{while}\,(y = -1)\,\{\,y := <r>\,\}\,; \\
\quad /\!/\ \ [y = 0] \ \mid\ \xi
\end{array}
$$

Proof for Running Example C_{run}

Example

Pick $\xi = \max\{\,[r \mapsto -1],\ [r \mapsto 0]\,\}$:

$$
\begin{array}{l}
\quad /\!/\ 0.5 \mid \xi \\
< r > := -1 \\
\quad /\!/\ 0.5 \star 1 \mid \xi \\
\\
\quad /\!/\ 0.5 \mid \xi \\
\{ < r > := 0 \}\,[0.5]\,\{ < r > := 1 \} \\
\quad /\!/\ 1 \mid \xi \\
\\
\quad /\!/\ 1 \star [y = 0] \mid \xi
\end{array}
\quad \Big\| \quad
\begin{array}{l}
\quad /\!/\ 1 \mid \xi \\
y := < r >; \\
\quad /\!/\ \max\{\,[y = -1],\ [y = 0]\,\} \mid \xi \\
\texttt{while}\,(y = -1)\,\{\, y := < r > \,\}\,; \\
\quad /\!/\ [y = 0] \mid \xi
\end{array}
$$

$$\implies 0.5 \leq \mathsf{wrlp}[\![C_{run}]\!]\,([y = 0] \mid \xi)$$

For the empty resource invariant, wrlp and wlp coincide:

Lemma

For the empty-state predicate $[\mathbf{emp}]$*, a program* C *and an expectation* $Y : \mathit{States} \to [0,1]$*, we have:*

$$\mathrm{wrlp}[\![C]\!]\,(Y \mid [\mathbf{emp}]) = \mathrm{wlp}[\![C]\!]\,(Y)$$

For the empty resource invariant, wrlp and wlp coincide:

Lemma

For the empty-state predicate $[\mathbf{emp}]$*, a program* C *and an expectation* $Y : \mathit{States} \to [0,1]$*, we have:*

$$\mathrm{wrlp}[\![C]\!]\,(Y \mid [\mathbf{emp}]) = \mathrm{wlp}[\![C]\!]\,(Y)$$

Can be used to compute lower bounds on wlp using wrlp:

Example

$$\begin{aligned}
& 0.5 \leq \mathrm{wrlp}[\![C_{run}]\!]\,([y=0] \mid \xi) && \text{(previous example)} \\
\Longrightarrow\quad & 0.5 \star \xi \leq \mathrm{wrlp}[\![C_{run}]\!]\,([y=0] \star \xi \mid [\mathbf{emp}]) && \text{(share rule)} \\
\Longrightarrow\quad & 0.5 \star \xi \leq \mathrm{wrlp}[\![C_{run}]\!]\,([y=0] \mid [\mathbf{emp}]) && \text{(monotonicity)} \\
\Longrightarrow\quad & 0.5 \star \xi \leq \mathrm{wlp}[\![C_{run}]\!]\,([y=0]) && \text{(Lemma)}
\end{aligned}$$

Further Properties of w(r)lp

Conservativity w.r.t. CSL [O'Hearn/Vafeiadis]:

Theorem

If C is non-probabilistic and every trace in C enables certain framing criteria (left out here):

$$\xi \models_{\mathit{CSL}} \{\psi\}\ C\ \{\varphi\} \qquad \textit{iff} \qquad \psi \leq \mathsf{wrlp}[\![C]\!]\,(\varphi \mid \xi)$$

Conservativity w.r.t. CSL [O'Hearn/Vafeiadis]:

Theorem

If C is non-probabilistic and every trace in C enables certain framing criteria (left out here):

$$\xi \models_{CSL} \{\psi\}\ C\ \{\varphi\} \qquad \textit{iff} \qquad \psi \leq \mathsf{wrlp}[\![C]\!](\varphi \mid \xi)$$

Superlinearity to eliminate addition:

Lemma

When C is almost surely terminating:

$$\mathsf{wlp}[\![C]\!](X) + \mathsf{wlp}[\![C]\!](Y) \leq \mathsf{wlp}[\![C]\!](X + Y)$$

Conservativity w.r.t. CSL [O'Hearn/Vafeiadis]:

Theorem

If C is non-probabilistic and every trace in C enables certain framing criteria (left out here):

$$\xi \models_{CSL} \{\psi\}\ C\ \{\varphi\} \qquad \textit{iff} \qquad \psi \leq \mathsf{wrlp}[\![C]\!](\varphi \mid \xi)$$

Superlinearity to eliminate addition:

Lemma

When C is almost surely terminating:

$$\mathsf{wlp}[\![C]\!](X) + \mathsf{wlp}[\![C]\!](Y) \leq \mathsf{wlp}[\![C]\!](X + Y)$$

Many more rules requiring preciseness of ξ:

Definition

ξ is precise iff for $\forall (s,h) \in \mathit{States}$ with $(s,h) \models \xi$ there exists no $h' \in \mathit{Heaps}$ such that $h \subsetneq h'$ and $(s,h') \models \xi$.

The End

Conclusion

- CQSL combines CSL and QSL to support
  - concurrency
  - pointers
  - probabilities
- Conservative w.r.t. to both CSL and QSL

Example

$$
\begin{array}{l}
C : \langle r \rangle := -1 ; \\
\quad \begin{array}{l|l}
\{\langle r \rangle := 0\} & y := \langle r \rangle ; \\
\quad [0.5] & \texttt{while}\,(y = -1)\,\{ \\
\{\langle r \rangle := 1\} & \quad y := \langle r \rangle\} ;
\end{array}
\end{array}
$$

$$0.5 \star \xi \leq \mathsf{wlp}[\![C]\!]([y = 0])$$

The End

Conclusion

- CQSL combines CSL and QSL to support
  - concurrency
  - pointers
  - probabilities
- Conservative w.r.t. to both CSL and QSL

Open Problems

- Certain rules still require preciseness of resource invariant
- Quantitative resource invariants
  (require fully quantitative version of magic wand)
- Upper bounds for weakest (resource-safe) liberal preexpectations
- Extension to weakest non-liberal preexpectations

Example

$$
\begin{array}{l}
C : \langle r \rangle := -1 \,; \\
\quad \begin{array}{l|l}
\{\langle r \rangle := 0\} & y := \langle r \rangle \,; \\
\quad [0.5] & \texttt{while}\,(y = -1)\,\{ \\
\{\langle r \rangle := 1\} & \quad y := \langle r \rangle\} \,;
\end{array}
\end{array}
$$

$$0.5 \star \xi \leq \mathsf{wlp}[\![C]\!]\,([y = 0])$$

The End

Conclusion

- CQSL combines CSL and QSL to support
  - concurrency
  - pointers
  - probabilities
- Conservative w.r.t. to both CSL and QSL

Open Problems

- Certain rules still require preciseness of resource invariant
- Quantitative resource invariants
  (require fully quantitative version of magic wand)
- Upper bounds for weakest (resource-safe) liberal preexpectations
- Extension to weakest non-liberal preexpectations

Proof Automation

- Expectations are generalised propositions
- Quantitative extensions for dealing with probabilities required

Example

$$C : \langle r \rangle := -1 ;$$
$$\begin{array}{l|l} \{\langle r \rangle := 0\} & y := \langle r \rangle ; \\ [0.5] & \texttt{while}\,(y = -1)\,\{ \\ \{\langle r \rangle := 1\} & \quad y := \langle r \rangle\} ; \end{array}$$

$$0.5 \star \xi \leq \mathsf{wlp}[\![C]\!]([y = 0])$$