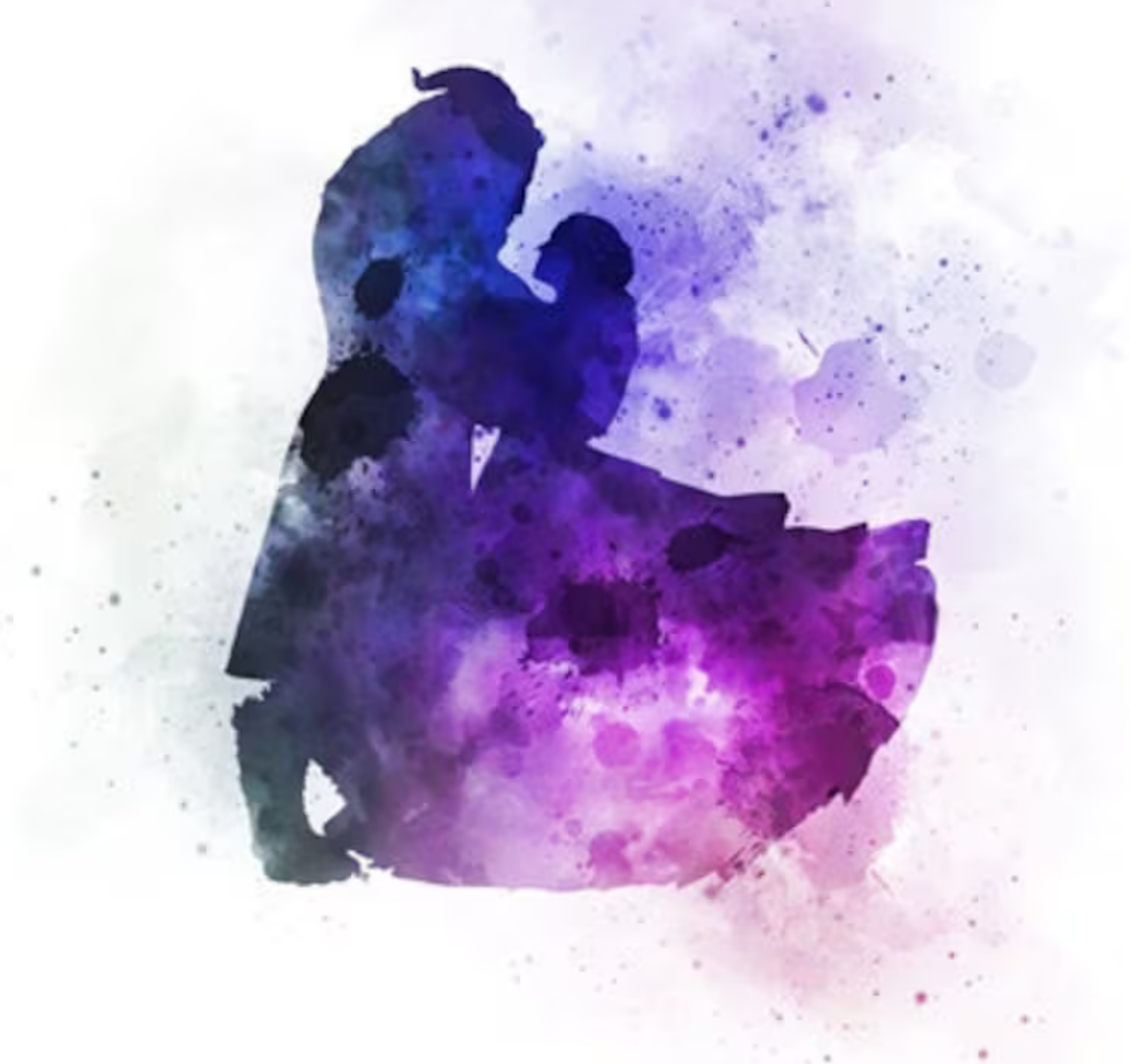
Outcome Logic

A foundational framework for concurrent and probabilistic program analysis

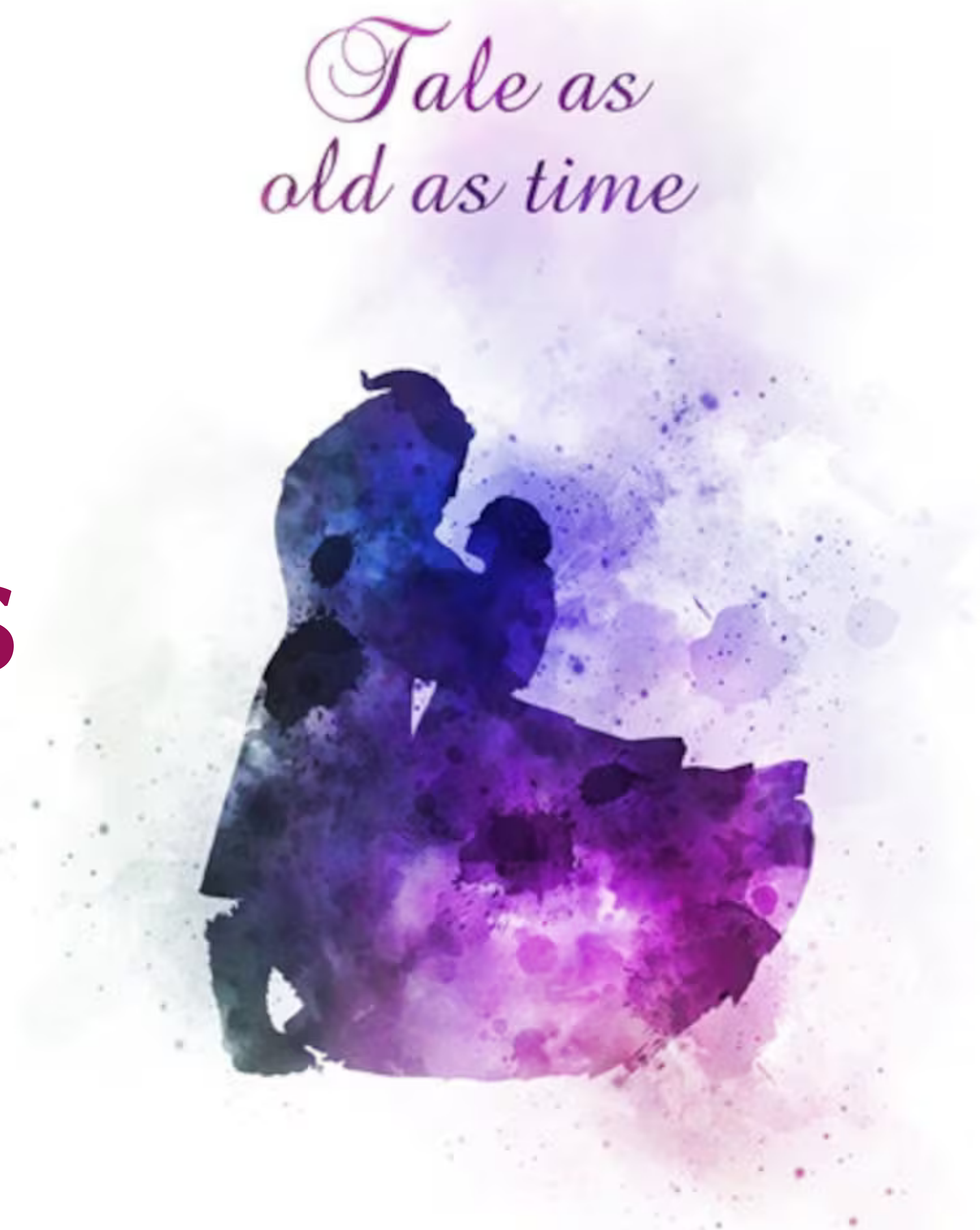
Alexandra Silva

Jan 21, 2025 — VMCAI 2025 — Denver, CO

Tale as old as time



Programs



Programs

Formal
Methods

Programs

Logics

Program Logics

If P holds before running C...

$$\{P\}\ C\ \{Q\}$$

...then Q holds after running C

Basis of analysis and verification tools

Infer

Gillian

Iris

Isabelle

Dafny

LEAN

VIPER

Probabilistic Concurrent Programs

Probabilistic Concurrent Programs

What?

Probabilistic Concurrent Programs

What?

$$y := 0 \;;\; \big(\; x_2 := \mathrm{flip}(1/2) \;\;;\; x_1 := y \;\;;\; z := \mathrm{xor}(x_1, x_2) \;\big|\; y := 1 \;\big)$$

Probabilistic Concurrent Programs

What?

$$y := 0 \; ; \; \left(\; x_2 := \mathrm{flip}({}^1/_2) \; ; \; x_1 := y \; ; \; z := \mathrm{xor}(x_1, x_2) \;\; | \;\; y := 1 \; \right)$$

Is z a good source of randomness?

Probabilistic Concurrent Programs

What?

$y := 0 \; ; \; (\; x_2 := \mathrm{flip}(1/2) \; ; \; x_1 := y \; ; \; z := \mathrm{xor}(x_1, x_2) \; | \; y := 1 \;)$

Is z a good source of randomness?

$$\{ \top \} \; C \; \{ z \sim \mathrm{Ber}(1/2) \}$$

Probabilistic Concurrent Programs

Probabilistic Concurrent Programs

$$\{\ \top\ \}\ C\ \{z \sim \mathsf{Ber}(1/2)\}$$

How do we design a program logic to derive this triple?

Where do these programs appear?

Long Tradition: **Randomised Distributed Algorithms**

Where do these programs appear?

Long Tradition: **Randomised Distributed Algorithms**

Where do these programs appear?

Less Long but Hot: **AI and ML applications**

Good Target for Formal Methods

Program behaviour very subtle

Good Target for Formal Methods

Program behaviour very subtle

Good Target for Formal Methods

Automated Reasoning Essential

Good Target for Program Logics

Complexity demands Scalability

Good Target for Program Logics

Complexity demands Scalability

Good Target for Program Logics

Compositional Reasoning

Challenges

$$y := 0 \;;\; \left(\; x_2 := \mathrm{flip}(1/2) \;;\; x_1 := y \;;\; z := \mathrm{xor}(x_1, x_2) \;\;|\;\; y := 1 \;\right)$$

Challenges

$$y := 0 \;;\; \left(\; \boxed{x_2 := \mathrm{flip}(1/2)} \;;\; x_1 := y \;;\; z := \mathrm{xor}(x_1, x_2) \;\mid\; y := 1 \;\right)$$

Semantics is a probability distribution

Challenges

$$y := 0 \; ; \; \left(\boxed{x_2 := \mathrm{flip}(^1/_2)} \; ; \; x_1 := y \; ; \; z := \mathrm{xor}(x_1, x_2) \; \boxed{\|} \; y := 1 \right)$$

Semantics is a probability distribution

Breaks compositionally of semantics

Challenges

$$y := 0 \;;\; (\; \boxed{x_2 := \mathrm{flip}(1/2)} \;;\; x_1 := y \;;\; z := \mathrm{xor}(x_1, x_2) \; \boxed{\;|\;} \; y := 1 \;)$$

Semantics is a probability distribution

Breaks compositionally of semantics

Minor changes cause strange effects!

Challenges

$$y := 0 \;;\; \big(\; x_1 := y \;;\; x_2 := \mathrm{flip}(1/2) \;;\; z := \mathrm{xor}(x_1, x_2) \;\big|\; y := 1 \;\big)$$

Challenges

High-quality randomness

$$y := 0 \;;\; \big(\; x_1 := y \;;\; x_2 := \mathrm{flip}(^1/_2) \;;\; z := \mathrm{xor}(x_1, x_2) \;\big|\; y := 1 \;\big)$$

Challenges

Low-quality randomness

High-quality randomness

$$y := 0 \;;\; \big(\; x_1 := y \;;\; x_2 := \mathrm{flip}(1/2) \;;\; z := \mathrm{xor}(x_1, x_2) \;\big|\; y := 1 \;\big)$$

Challenges

Low-quality randomness

High-quality randomness

$$y := 0 \;;\; (\; x_1 := y \;;\; x_2 := \mathrm{flip}(^1/_2) \;;\; z := \mathrm{xor}(x_1, x_2) \;\mid\; y := 1 \;)$$

High-quality randomness

Challenges

Low-quality randomness

High-quality randomness

$$y := 0 \;;\; \left(\; x_1 := y \;;\; x_2 := \mathrm{flip}(1/2) \;;\; z := \mathrm{xor}(x_1, x_2) \;\middle|\; y := 1 \;\right)$$

High-quality randomness

$$\{\, \top \,\}\; C\; \{z \sim \mathrm{Ber}(1/2)\}$$ **valid!**

Challenges

$$y := 0 \;;\; \Big(\;\; x_2 := \mathrm{flip}(1/2) \;\;;\; x_1 := y \;;\; z := \mathrm{xor}(x_1, x_2) \;\;\Big|\;\; y := 1 \Big)$$

Challenges

$$y := 0 \;;\; \left(\; x_2 := \mathrm{flip}(^1/_2) \;;\; x_1 := y \;;\; z := \mathrm{xor}(x_1, x_2) \;\middle|\; y := 1 \right)$$

$$\{\ \top\ \}\ C\ \{z \sim \mathsf{Ber}(1/2)\}$$

Challenges

Low-quality randomness!!

$$y := 0 \ ; \ (\ x_2 := \mathrm{flip}(^1/_2) \ ; \ x_1 := y \ ; \ z := \mathrm{xor}(x_1, x_2) \ \mid \ y := 1 \ )$$

$$\{\ \top\ \}\ C\ \{z \sim \mathrm{Ber}(1/2)\}$$

Challenges

Low-quality randomness!!

$$y := 0 \;;\; \left(\; x_2 := \mathrm{flip}(1/2) \;;\; x_1 := y \;;\; z := \mathrm{xor}(x_1, x_2) \;\middle|\; y := 1 \right)$$

$$\{\, \top \,\}\; C \;\{z \sim \mathrm{Ber}(1/2)\}$$

Not valid!

Challenges

$$
\begin{tikzcd}
 & & x_1 := y \arrow[r] & y := 1 \arrow[r] & z := \mathrm{xor}(x_1, x_2) \\
y := 0 \arrow[r] & x_2 := \mathrm{flip}(1/2) \arrow[ur, bend left, "(x_2 = 0)"] \arrow[dr, bend right, "(x_2 = 1)"'] & & & \\
 & & y := 1 \arrow[r] & x_1 := y \arrow[r] & z := \mathrm{xor}(x_1, x_2)
\end{tikzcd}
$$

Challenges

$$\begin{tikzcd}
 & & x_1 := y \arrow[r] & y := 1 \arrow[r] & z := \mathrm{xor}(x_1, x_2) \\
y := 0 \arrow[r] & x_2 := \mathrm{flip}({}^1/_2) \arrow[ur, "(x_2 = 0)"] \arrow[dr, "(x_2 = 1)"'] & & & \\
 & & y := 1 \arrow[r] & x_1 := y \arrow[r] & z := \mathrm{xor}(x_1, x_2)
\end{tikzcd}$$

Scheduler can force z=0

Desiderata

- Compositional Program Semantics
- Intuitive Proof Rules (handling mix of effects!)
- Unbounded Loops

Our work

A first step: Demonic Outcome Logic (POPL'25, on Thursday!)

restricted

✓ Compositional Program Semantics — Convex powerset

✓ Intuitive Proof Rules (handling mix of effects!)

✓ Unbounded Loops — Non-determinism

Our work

A second step: Probabilistic Concurrent Outcome Logic (submitted)

✔ Compositional Program Semantics — New structure!

✔ Intuitive Proof Rules (handling mix of effects!)

✘ Unbounded Loops

Our work

A third step: Denotational Semantics for Probabilistic and Concurrent Programs (submitted)

- ✓ Compositional Program Semantics — New structure!
- ✗ Intuitive Proof Rules (handling mix of effects!)
- ✓ Unbounded Loops — Domain theory developed

Soon.... :-)

✓Compositional Program Semantics
✓Intuitive Proof Rules (handling mix of effects!)
✓Unbounded Loops

Many threads to bring together

Concurrent Separation Logics

Iris

Probabilistic Separation Logics

PSL (POPL'20)
Bluebell (POPL'25)
Lilac (PLDI'23)

Many threads to bring together

Many threads to bring together

CSL: Separation makes compositional reasoning possible!

$$\frac{\langle P_1 \rangle \ C_1 \ \langle Q_1 \rangle \qquad\qquad \langle P_2 \rangle \ C_2 \ \langle Q_2 \rangle}{\langle P_1 * P_2 \rangle \ C_1 \parallel C_2 \ \langle Q_1 * Q_2 \rangle}$$

Many threads to bring together

PSL: Separation = probabilistic independence!

$$\dfrac{\langle\lceil x \mapsto -\rceil\rangle\ x :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right)\ \langle x \sim \mathbf{Ber}\left(\tfrac{1}{2}\right)\rangle \qquad \langle\lceil y \mapsto -\rceil\rangle\ y :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right)\ \langle y \sim \mathbf{Ber}\left(\tfrac{1}{2}\right)\rangle}{\langle\lceil x \mapsto -\rceil * \lceil y \mapsto -\rceil\rangle\ x :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right) \parallel y :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right)\ \langle x \sim \mathbf{Ber}\left(\tfrac{1}{2}\right) * y \sim \mathbf{Ber}\left(\tfrac{1}{2}\right)\rangle}\ \text{pPAR}$$

Many threads to bring together

CSL: Shared State requires invariants

$$\frac{I \vdash \langle \varphi_1 \rangle\ C_1\ \langle \psi_1 \rangle \qquad I \vdash \langle \varphi_2 \rangle\ C_2\ \langle \psi_2 \rangle}{I \vdash \langle \varphi_1 * \varphi_2 \rangle\ C_1 \parallel C_2\ \langle \psi_1 * \psi_2 \rangle}\ \textsc{Par}$$

$$z := 1 \,⨾\, \left(x :\approx \mathbf{Ber}\left(\tfrac{z}{2}\right) \parallel y :\approx \mathbf{Ber}\left(\tfrac{z}{2}\right)\right)$$

Many threads to bring together

Many threads to bring together

Randomised Shared state breaks PAR invariants

Many threads to bring together

Randomised Shared state breaks PAR invariants

$$z :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right) \fatsemi (x := z \parallel y := 1 - z)$$

Many threads to bring together

Randomised Shared state breaks PAR invariants

$$z :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right) \fatsemi (x := z \parallel y := 1 - z)$$

x and y are conditionally independent on z.

Many threads to bring together

Randomised Shared state breaks PAR invariants

$$z :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right) \fatsemi (x := z \parallel y := 1 - z)$$

x and y are conditionally independent on z.

Conditioning = breaking down **outcomes** of sampling operation

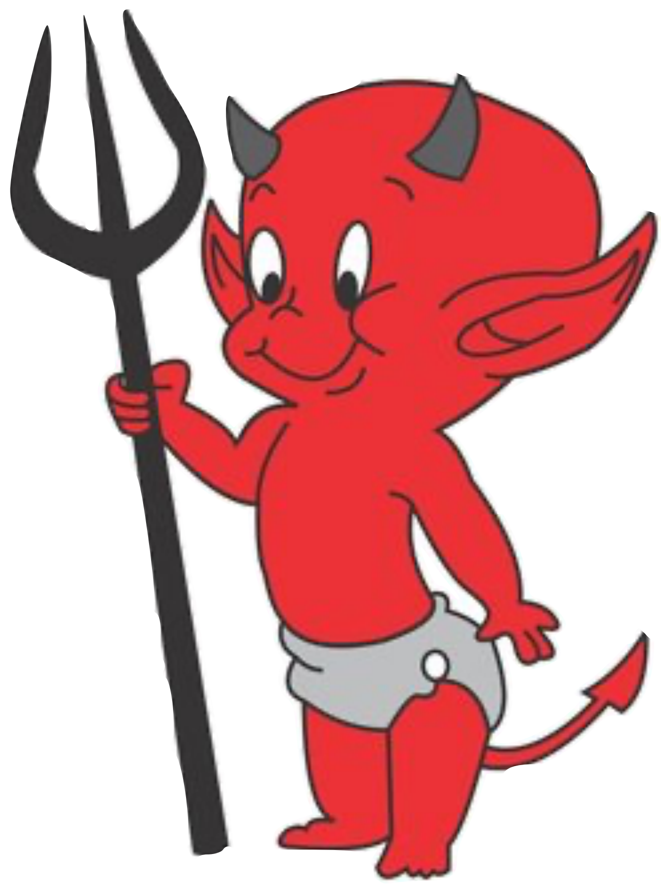
$z \sim$ Ber(1/2) expands to $\displaystyle\bigoplus_{Z \sim \mathrm{Ber}(1/2)} \quad z \mapsto Z$

PCOL: probabilistic concurrent outcome logic

Key Ingredients in the Logic

$\bigoplus_{Z \sim \mathrm{Ber}(1/2)}$ $\quad z \mapsto Z \quad$ and $\quad \varphi \star \psi$

The devil is in the details aka semantics!



Outcomes

$$\oplus$$

Originally developed in Outcome Logic to unify correctness and incorrectness

Outcomes

$$\oplus$$

Originally developed in Outcome Logic to unify correctness and incorrectness

Probabilistic setting: enables reasoning on **entire distribution** of program outcomes

Outcomes

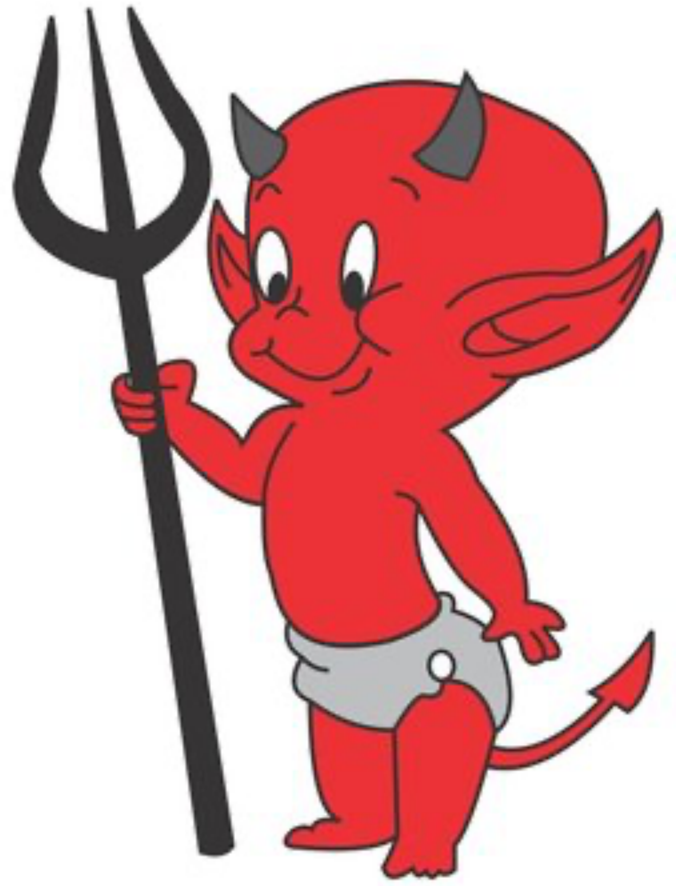
$$\oplus$$

Originally developed in Outcome Logic to unify correctness and incorrectness

Probabilistic setting: enables reasoning on **entire distribution** of program outcomes

Key for probabilistic concurrency: program outcomes can be effectful and composed

Back to the devil

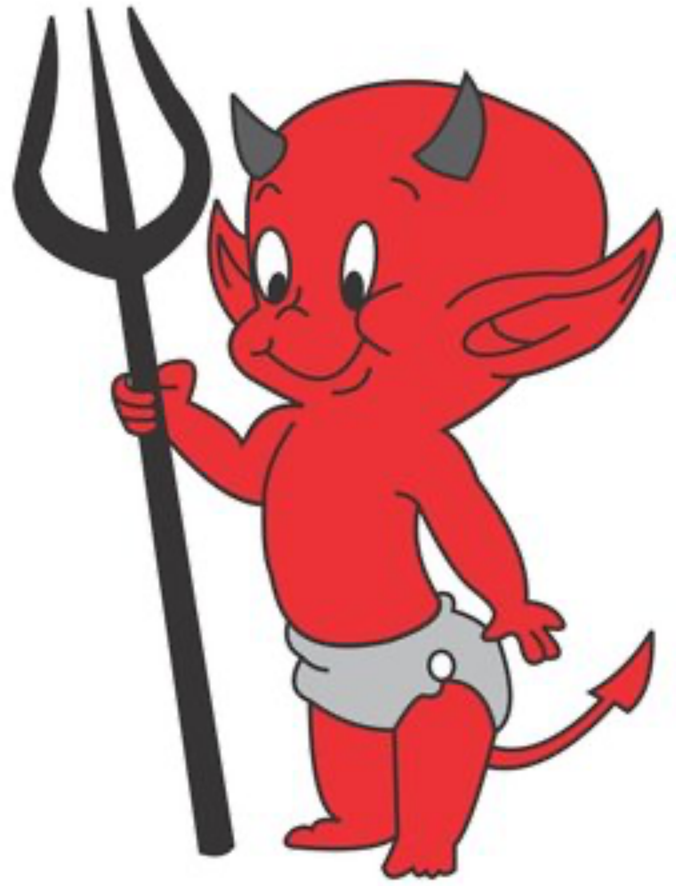


Probability

Markov Kernels

Convex spaces

Back to the devil



Probability

Markov Kernels

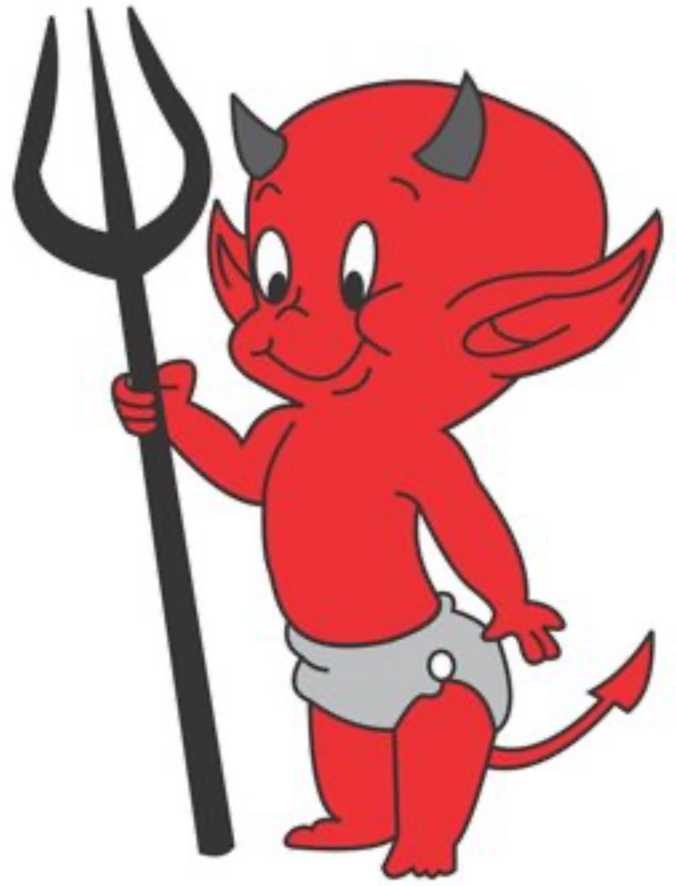
Convex spaces

Concurrency

Pomsets

Event Structures

Back to the devil



Probability

Markov Kernels

Convex spaces

Concurrency

Pomsets

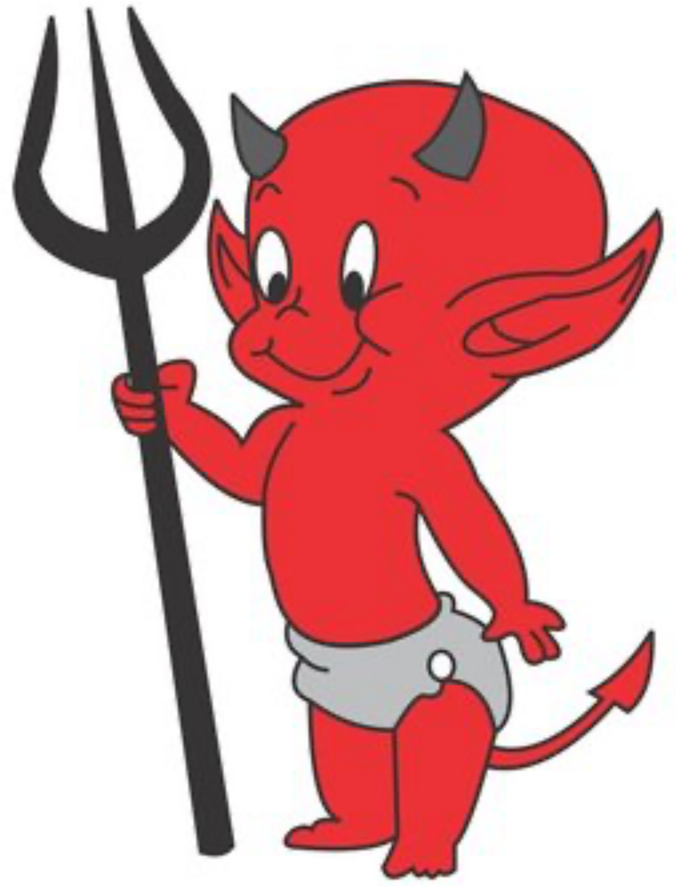
Event Structures

Control Flow

Boolean Algebra

Control-Flow Graphs

Back to the devil



Probability

+

Concurrency

+

Control Flow

Markov Kernels

Convex spaces

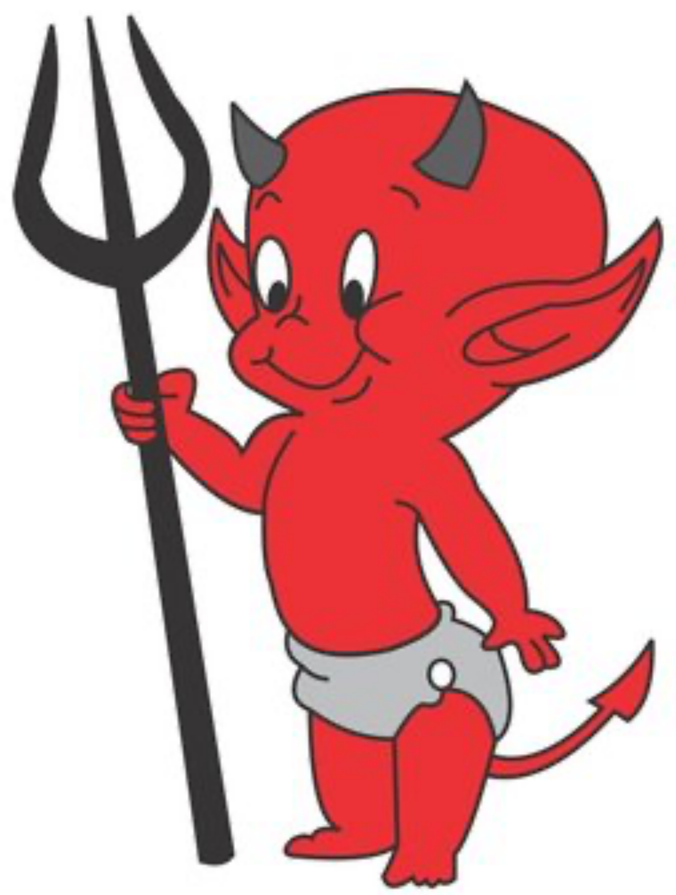
Pomsets

Event Structures

Boolean Algebra

Control-Flow Graphs

Back to the devil



Probability

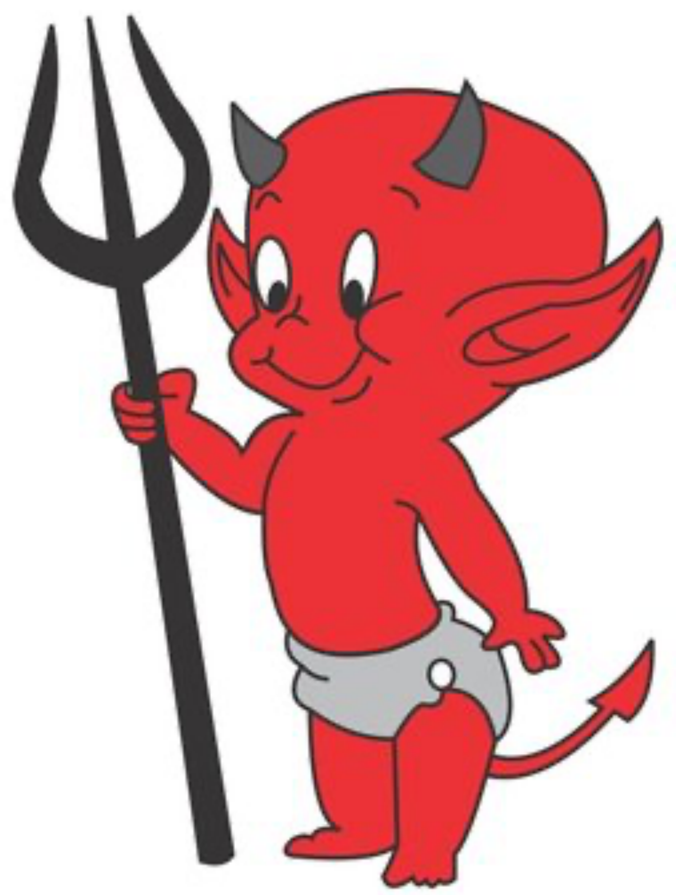
+

Concurrency

+

Control Flow

Back to the devil



Probability

+

Concurrency

+

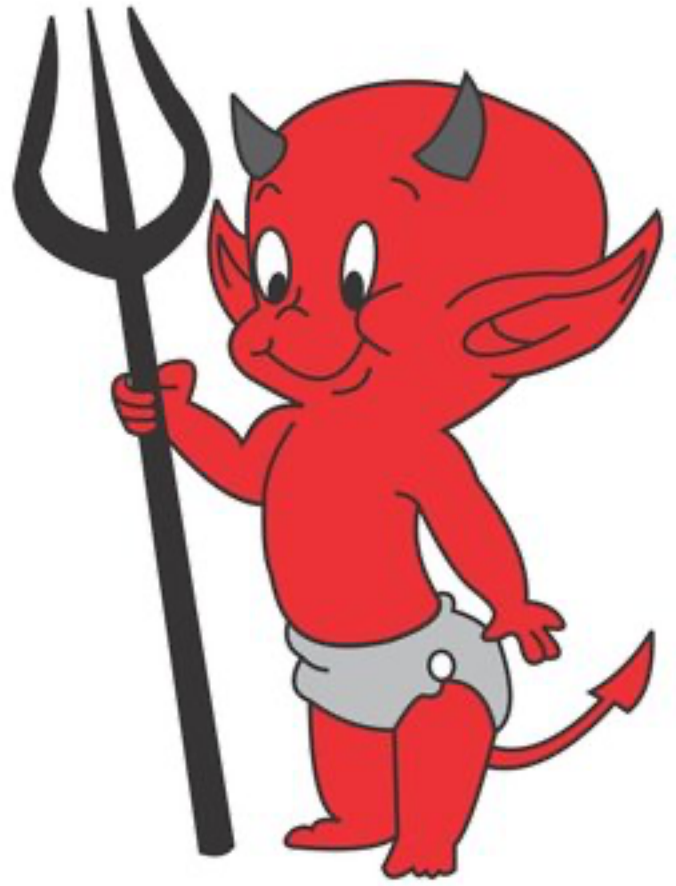
Control Flow

Pomset

with

Formulae

Back to the devil



Probability

+

Concurrency

+

Control Flow

Pomset

with

Formulae

Nodes abstract

probabilistic actions

Pomset with formulae

$$\texttt{if } b_1\ \{a_1\}\ \texttt{else } \{\texttt{if } b_2\ \{a_2\}\ \texttt{else } \{a_3\}\}$$

$$\begin{tikzcd}
 & z_1 & & z_2 \\
y_1 & & y_2 \arrow[ul] \arrow[ur] & \\
 & x \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

$\varphi(z_1) = \neg x \wedge y_2$

$\varphi(z_2) = \neg x \wedge \neg y_2$

$\varphi(y_1) = x$

$\varphi(y_2) = \neg x$

$\varphi(x) = \mathsf{true}$

PCOL: probabilistic outcome logic

$$
\begin{aligned}
C ::= {} & \mathbf{skip} \\
\mid {} & C_1 ; C_2 \\
\mid {} & C_1 \mid C_2 \\
\mid {} & \mathbf{if}\ b\ \{C_1\}\ \mathbf{else}\ \{C_2\} \\
\mid {} & \mathbf{while}\ b\ \{C\} \\
\mid {} & a
\end{aligned}
$$

PCOL: probabilistic outcome logic

$$\llbracket - \rrbracket : \mathit{cmd} \to \mathit{pom}$$

$$\begin{aligned}
\llbracket \mathtt{skip} \rrbracket &\triangleq \langle \mathsf{fork} \rangle \\
\llbracket C_1 ; C_2 \rrbracket &\triangleq \llbracket C_1 \rrbracket \fatsemi \llbracket C_2 \rrbracket \\
\llbracket C_1 \mid C_2 \rrbracket &\triangleq \llbracket C_1 \rrbracket \parallel \llbracket C_2 \rrbracket \\
\llbracket \mathtt{if}\ b\ \{C_1\}\ \mathtt{else}\ \{C_2\} \rrbracket &\triangleq \mathsf{guard}(b, \llbracket C_1 \rrbracket, \llbracket C_2 \rrbracket) \\
\llbracket \mathtt{while}\ b\ \{C\} \rrbracket &\triangleq \mathsf{lfp}\left(\Phi_{\langle C, b \rangle}\right) \\
\llbracket a \rrbracket &\triangleq \langle a \rangle
\end{aligned}$$

PCOL: probabilistic outcome logic

$$[\![-]\!] : \mathit{cmd} \to \mathit{pom}$$

$$
\begin{aligned}
[\![\texttt{skip}]\!] &\triangleq \langle \mathsf{fork} \rangle \\
[\![C_1; C_2]\!] &\triangleq [\![C_1]\!] \fatsemi [\![C_2]\!] \\
[\![C_1 \mid C_2]\!] &\triangleq [\![C_1]\!] \parallel [\![C_2]\!] \\
[\![\texttt{if } b\ \{C_1\} \texttt{ else } \{C_2\}]\!] &\triangleq \mathsf{guard}(b, [\![C_1]\!], [\![C_2]\!]) \\
[\![\texttt{while } b\ \{C\}]\!] &\triangleq \mathsf{lfp}\left(\Phi_{\langle C,b\rangle}\right) \\
[\![a]\!] &\triangleq \langle a \rangle
\end{aligned}
$$

New pomset operations

PCOL: probabilistic outcome logic

$$[\![-]\!] : cmd \to pom$$

New pomset operations

$$
\begin{aligned}
[\![\mathbf{skip}]\!] &\triangleq \langle \mathsf{fork} \rangle \\
[\![C_1; C_2]\!] &\triangleq [\![C_1]\!] \fatsemi [\![C_2]\!] \\
[\![C_1 \mid C_2]\!] &\triangleq [\![C_1]\!] \parallel [\![C_2]\!] \\
[\![\mathbf{if}\ b\ \{C_1\}\ \mathbf{else}\ \{C_2\}]\!] &\triangleq \mathsf{guard}(b, [\![C_1]\!], [\![C_2]\!]) \\
[\![\mathbf{while}\ b\ \{C\}]\!] &\triangleq \mathsf{lfp}\left(\Phi_{\langle C, b\rangle}\right) \\
[\![a]\!] &\triangleq \langle a \rangle
\end{aligned}
$$

Getting this as a fixpoint is where a lot of the work is!

PCOL: probabilistic outcome logic

$$\varphi ::= \top \mid \bot \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \bigoplus_{X \sim d(E)} \varphi \mid \varphi * \psi \mid \lceil P \rceil$$

PCOL: probabilistic outcome logic

$$\varphi ::= \top \mid \bot \mid \varphi \land \psi \mid \varphi \lor \psi \mid \bigoplus_{X \sim d(E)} \varphi \mid \varphi * \psi \mid \lceil P \rceil$$

$$\dfrac{}{I \vdash \langle \varphi \rangle \ \mathbf{skip} \ \langle \varphi \rangle}\ \textsc{Skip} \qquad \dfrac{I \vdash \langle \varphi \rangle\, C_1\, \langle \vartheta \rangle \quad I \vdash \langle \vartheta \rangle\, C_2\, \langle \psi \rangle}{I \vdash \langle \varphi \rangle\, C_1 \fatsemi C_2\, \langle \psi \rangle}\ \textsc{Seq} \qquad \dfrac{\forall 0 \le k \le n-1. \quad I \vdash \langle \varphi_k \rangle\, C\, \langle \varphi_{k+1} \rangle}{I \vdash \langle \varphi_0 \rangle\ \mathbf{for}\ n\ \mathbf{do}\ C\ \langle \varphi_n \rangle}\ \textsc{For}$$

$$\dfrac{\varphi \Rightarrow \lceil b \mapsto \mathsf{true} \rceil \qquad I \vdash \langle \varphi \rangle\, C_1\, \langle \psi \rangle}{I \vdash \langle \varphi \rangle\ \mathbf{if}\ b\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\ \langle \psi \rangle}\ \textsc{If1} \qquad \dfrac{\varphi \Rightarrow \lceil b \mapsto \mathsf{false} \rceil \qquad I \vdash \langle \varphi \rangle\, C_2\, \langle \psi \rangle}{I \vdash \langle \varphi \rangle\ \mathbf{if}\ b\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\ \langle \psi \rangle}\ \textsc{If2}$$

$$\dfrac{I \vdash \langle \varphi_1 \rangle\, C_1\, \langle \psi_1 \rangle \qquad I \vdash \langle \varphi_2 \rangle\, C_2\, \langle \psi_2 \rangle \qquad \mathsf{precise}(\psi_1, \psi_2)}{I \vdash \langle \varphi_1 * \varphi_2 \rangle\, C_1 \parallel C_2\, \langle \psi_1 * \psi_2 \rangle}\ \textsc{Par}$$

$$\dfrac{(\varphi * \lceil x \mapsto E \rceil) \Rightarrow \lceil e \mapsto E' \rceil}{I \vdash \langle \varphi * \lceil x \mapsto E \rceil \rangle\ x := e\ \langle \varphi * \lceil x \mapsto E' \rceil \rangle}\ \textsc{Assign} \qquad \dfrac{(\varphi * \lceil x \mapsto E \rceil) \Rightarrow \lceil e \mapsto E' \rceil}{I \vdash \langle \varphi * \lceil x \mapsto E \rceil \rangle\ x :\approx d(e)\ \langle \varphi * (x \sim d(E')) \rangle}\ \textsc{Samp}$$

PCOL: probabilistic outcome logic

$$\varphi ::= \top \mid \bot \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \bigoplus_{X \sim d(E)} \varphi \mid \varphi * \psi \mid \lceil P \rceil$$

$$\frac{}{I \vdash \langle \lceil x \mapsto - \rceil \rangle \; x :\approx d \; \langle x \sim d \rangle} \textsc{Samp}$$

Back to the example

$$y := 0 \,\mathbin{;}\, \left(\; x_1 := y \,\mathbin{;}\, x_2 :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right) \mathbin{;}\, z := \mathsf{xor}(x_1, x_2) \;\;\Big\|\;\; y := 1 \; \right)$$

Back to the example

$$y := 0 \,⨟\, \Big(\; x_1 := y \,⨟\, x_2 :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right) ⨟\, z := \mathsf{xor}(x_1, x_2) \;\Big\|\; y := 1 \;\Big)$$

$$\{\; \top \;\}\; C \;\{z \sim \mathrm{Ber}(1/2)\}$$

Back to the example

$$y := 0 \,\mathbin{;}\, \Big(\; x_1 := y \,\mathbin{;}\, x_2 :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right) \mathbin{;}\, z := \mathsf{xor}(x_1, x_2) \;\Big\|\; y := 1 \;\Big)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle\lceil y \mapsto Y * (Y = 0 \lor Y = 1) * x_1 \mapsto X \rceil\rangle\; x_1 := y \;\langle\lceil x_1 \mapsto Y * y \mapsto Y * (Y = 0 \lor Y = 1)\rceil\rangle}\;\textsc{Assign}}{\langle\lceil y \mapsto Y * (Y = 0 \lor Y = 1) * x_1 \mapsto X \rceil\rangle\; x_1 := y \;\langle\lceil x_1 \in \{0,1\}\rceil * \lceil y \in \{0,1\}\rceil\rangle}\;\textsc{Consequence}}{\langle\lceil \mathsf{own}(x_1)\rceil * \lceil y \in \{0,1\}\rceil\rangle\; x_1 := y \;\langle\lceil x_1 \in \{0,1\}\rceil * \lceil y \in \{0,1\}\rceil\rangle}\;\textsc{Exists} \times 2}{y \in \{0,1\} \vdash \langle\lceil \mathsf{own}(x_1)\rceil\rangle\; x_1 := y \;\langle\lceil x_1 \in \{0,1\}\rceil\rangle}\;\textsc{Atom}$$

Value of assignment is non-det!

Back to the example

$$y \coloneqq 0 \fatsemi \Big(\ x_1 \coloneqq y \fatsemi x_2 :\approx \mathbf{Ber}\left(\tfrac{1}{2}\right) \fatsemi z \coloneqq \mathsf{xor}(x_1, x_2)\ \Big\|\ y \coloneqq 1\ \Big)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle \lceil y \mapsto Y * (Y = 0 \lor Y = 1) * x_1 \mapsto X \rceil \rangle\ x_1 \coloneqq y\ \langle \lceil x_1 \mapsto Y * y \mapsto Y * (Y = 0 \lor Y = 1) \rceil \rangle}\ \textsc{Assign}}{\langle \lceil y \mapsto Y * (Y = 0 \lor Y = 1) * x_1 \mapsto X \rceil \rangle\ x_1 \coloneqq y\ \langle \lceil x_1 \in \{0,1\} \rceil * \lceil y \in \{0,1\} \rceil \rangle}\ \textsc{Consequence}}{\langle \lceil \mathsf{own}(x_1) \rceil * \lceil y \in \{0,1\} \rceil \rangle\ x_1 \coloneqq y\ \langle \lceil x_1 \in \{0,1\} \rceil * \lceil y \in \{0,1\} \rceil \rangle}\ \textsc{Exists} \times 2}{y \in \{0,1\} \vdash \langle \lceil \mathsf{own}(x_1) \rceil \rangle\ x_1 \coloneqq y\ \langle \lceil x_1 \in \{0,1\} \rceil \rangle}\ \textsc{Atom}$$

Value of assignment is non-det!

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\langle \lceil \mathsf{own}(x_2) \rceil \rangle\ x_2 :\approx \mathbf{Ber}\,(1/2)\ \langle (x_2 \sim \mathbf{Ber}\,(1/2)) \rangle}\ \textsc{Samp}}{\langle \lceil x_1 \mapsto X \rceil * \lceil \mathsf{own}(x_2, z) \rceil \rangle\ x_2 :\approx \mathbf{Ber}\,(1/2)\ \langle \lceil x_1 \mapsto X \rceil * (x_2 \sim \mathbf{Ber}\,(1/2)) * \lceil \mathsf{own}(z) \rceil \rangle}\ \textsc{Frame} \qquad (\bigstar)}{\langle \lceil x_1 \mapsto X \rceil * \lceil \mathsf{own}(x_2, z) \rceil \rangle\ x_2 :\approx \mathbf{Ber}\,(1/2) \fatsemi z \coloneqq \mathsf{xor}(x_1, x_2)\ \langle z \sim \mathbf{Ber}\,(1/2) \rangle}\ \textsc{Seq}}{\langle \lceil x_1 \in \{0,1\} \rceil * \lceil \mathsf{own}(x_2, z) \rceil \rangle\ x_2 :\approx \mathbf{Ber}\,(1/2) \fatsemi z \coloneqq \mathsf{xor}(x_1, x_2)\ \langle z \sim \mathbf{Ber}\,(1/2) \rangle}\ \textsc{Exists}$$

Value of z is a fair coin!

Probabilistic Concurrent Programs

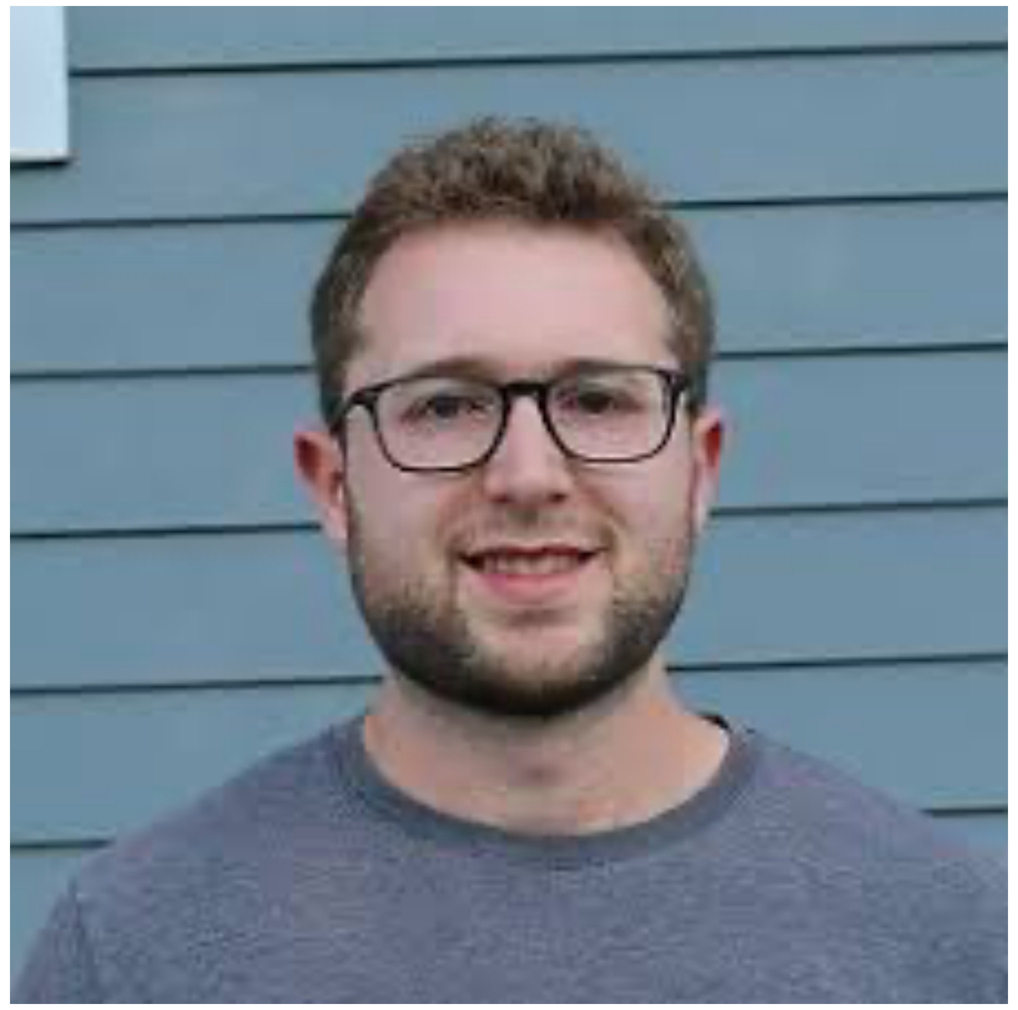
- *Randomized distributed systems have been studied for decades*
- *Interactions between randomization and concurrency are subtle*
- *Formal models ensure correctness*
- *Program Logics enable compositional proofs*

Probabilistic Concurrent Programs

- *Randomized distributed systems have been studied for decades*
- *Interactions between randomization and concurrency are subtle*
- *Formal models ensure correctness*
- *Program Logics enable compositional proofs*

Our work builds a **new denotational model** and **program logics** — work in progress, more on the way!

Probabilistic Concurrent Outcome Logic



Noam Zilberstein

https://www.cs.cornell.edu/~noamz/