#Slides

Separation Logics for Probability

(How to deal with REJECTIONS)

A logsem seminar by Heili =)

21/10/24

What is this talk about ??

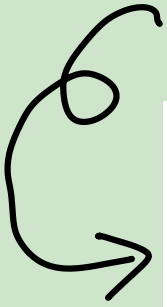
- Logics for reasoning about probability
- Rules + key ideas + examples
- Key idea: Everything is a RESOURCE

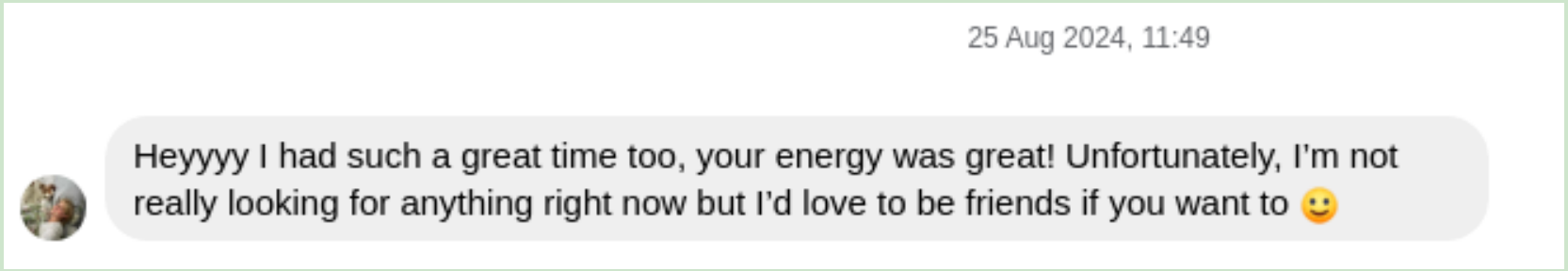
What is this talk about ??

- Logics for reasoning about probability
- Rules + key ideas + examples
- Key idea: Everything is a RESOURCE

What is this talk NOT about ??

- How do you REALLY deal with rejection =(





Why do we care about PROBABILITY

Write your answers in your notebooks now!


Monte Carlo algorithms
Efficient Data structures
Las Vegas algorithms
Why do we care about
PROBABILITY
Primality tests
Machine Learning
(ewww....)
Cryptography
Differential Privacy
and many more!

By probabilistic programs, we mean

randomized programs

(+ higher-order + local state)

$$(\mathrm{rand}\ N, d) \xrightarrow{\frac{1}{N+1}} (n, d)$$

where $n \in \{0, \ldots, N\}$

By probabilistic programs, we mean

randomized programs

(+ higher-order + local state)

$$(\text{rand } N, \sigma) \xrightarrow{\frac{1}{N+1}} (n, \sigma)$$

where $n \in \{0, \ldots, N\}$

Example

$$(\text{rand } 5, \sigma) \xrightarrow{\frac{1}{6}} (n, \sigma)$$

where $0 \leq n \leq 5$

By probabilistic programs, we mean

randomized programs

(+ higher-order + local state)

$$(\text{rand } N, \sigma) \xrightarrow{\frac{1}{N+1}} (n, \sigma)$$

where $n \in \{0, \ldots, N\}$

Example

$$(\text{rand } 5, \sigma) \xrightarrow{\frac{1}{6}} (n, \sigma)$$

where $0 \leq n \leq 5$

NB: We don't consider fancy stuff like inference / continuous distributions

The Motivating Example

Rejection Sampler

$$RS \triangleq \lambda\_.\ \texttt{let } x = \texttt{rand } 2 \texttt{ in}$$
$$\texttt{if } x < 2 \texttt{ then } x \texttt{ else } RS\ ()$$

The Motivating Example

Rejection Sampler

$$RS \triangleq \lambda\_.\ \texttt{let } x = \texttt{rand } 2 \texttt{ in}$$
$$\texttt{if } x < 2 \texttt{ then } x \texttt{ else } RS\ ()$$

Q: What properties does RS have??

* Write your answers in your notebooks now! *

The Motivating Example

Rejection Sampler

$$RS \triangleq \lambda\_.\ \texttt{let}\ x = \texttt{rand}\ 2\ \texttt{in}\ \texttt{if}\ x < 2\ \texttt{then}\ x\ \texttt{else}\ RS\ ()$$

Q: What properties does RS have ??

A:
- Returns 0 or 1 with 50% probability
- Almost sure termination
- Positive almost sure termination

- Contextual equivalence with rand 1
- Logically atomically similar with rand 1 and many more...!

Aim of Today

1. Prove expected time cost of RS

2. Prove error bounds of return value of RS

3.* Prove bounds of favorable return value of RS

Aim of Today

1. Prove expected time cost of RS with Tachis (OOPSLA 2024)
2. Prove error bounds of return value of RS with Eris (ICFP 2024)
3. * Prove bounds of favorable return value of RS with Eris_t (ICFP 2024)

Act I

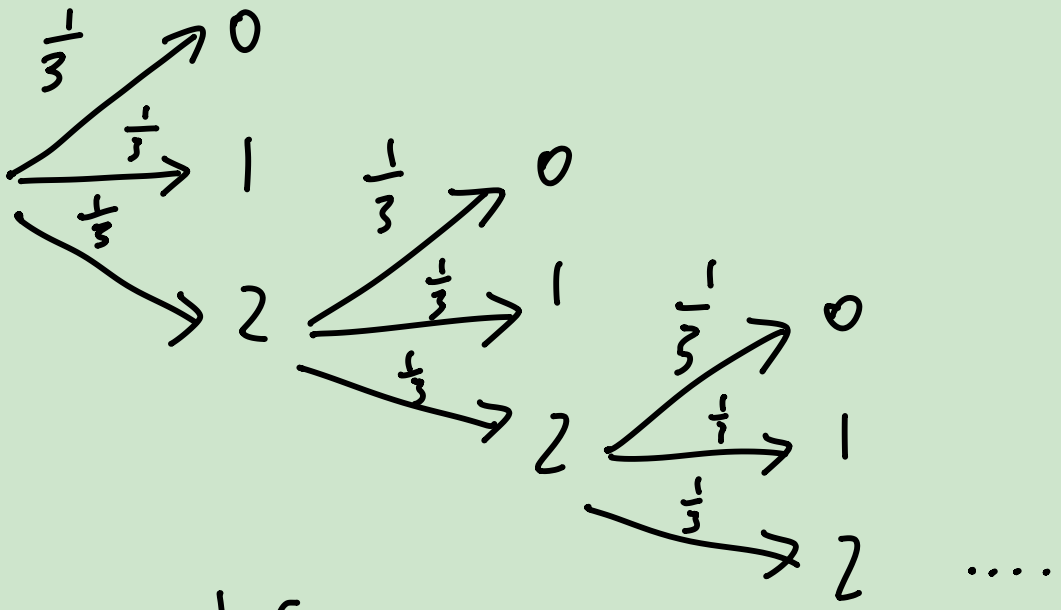
Start of TIME

What is the expected number of "rands" executed in RS ?

$$RS \triangleq \lambda\_.\ \texttt{let } x = \texttt{rand } 2 \texttt{ in}$$
$$\texttt{if } x < 2 \texttt{ then } x \texttt{ else } RS\ ()$$

What is the expected number of "rands" executed in RS ?

$$RS \triangleq \lambda\_.\ \text{let } x = \text{rand } 2 \text{ in } \text{if } x < 2 \text{ then } x \text{ else } RS\ ()$$



$$1 + \tfrac{1}{3}\left(1 + \tfrac{1}{3}\left(1 + \ldots\right)\right)$$

$$= 1 + \tfrac{1}{3} + \tfrac{1}{9} + \tfrac{1}{27} + \ldots$$

$$= 1\left(\tfrac{1}{1-\frac{1}{3}}\right) = 1 \cdot \tfrac{1}{\frac{2}{3}} = \tfrac{3}{2}$$ //

Key idea 1: Computation cost as a RESOURCE!

$⧗_x$ denotes ownership of $x \in \mathbb{R}_{\geq 0}$ cost credits

Credit splitting rule

$$⧗(x_1 + x_2) \dashv\vdash ⧗x_1 * ⧗x_2$$

Key idea 1: Computation cost as a RESOURCE!

$⧗\,x$ denotes ownership of $x \in \mathbb{R}_{\geq 0}$ cost credits

Credit splitting rule

$$⧗(x_1 + x_2) \dashv\vdash ⧗x_1 \;*\; ⧗x_2$$

Adequacy Theorem of Tachis:

$\{⧗x\}\ e\ \{\mathrm{True}\} \Rightarrow$ for any state σ,
expected number of rand executed by $(e, \sigma) \leq x$

Key idea 2 : Expectation-preserving Composition

$$\frac{1 + \sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1}{\{\mathbb{I}\, x_1\}\ \text{rand}\ N\ \{n.\ \mathbb{I}(X_2(n)) + \ulcorner 0 \leq n \leq N \urcorner\}}\ \text{HT-RAND-EXP}$$

Key idea 2 : Expectation-preserving Composition

$$\frac{1 + \sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1}{\{\mathbb{I}\, x_1\}\ \mathrm{rand}\ N\ \{n.\ \mathbb{I}(X_2(n)) * \ulcorner 0 \leq n \leq N \urcorner\}}\ \text{HT-RAND-EXP}$$

cost of rand

average credit remaining

initial credits

This is THE crucial rule of Tachis!

Other rules are similar to 'normal ones', e.g.

$$\{l \mapsto v\}\ !\ l\ \{x.\ l \mapsto v * \ulcorner x = v \urcorner\}$$

DEMO

Hypothesis	Goal
	Expected number of rand executed in RS () $\leq \frac{3}{2}$

DEMO

Hypothesis	Goal
	$\{\frac{3}{2}\}$ RS C) $\{True\}$
	by adequacy theorem of Tachis

DEMO

Hypothesis	Goal
⧗ 3/2	wp RS () {True}
	iIntros

DEMO

Hypothesis	Goal
⧗ 3/2	wp RS () {True}
▷ (⧗ 3/2 −∗ wp RS () {True})	

iLöb that MF

DEMO

Hypothesis

$\blacksquare_{3/2}$

$\blacksquare_{3/2} \multimap$ wp RS () {True}

Goal

wp let x = rand 2 in
if x < 2 then x else RS () {True}

$$1 + \sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1$$

$\{\blacksquare x_1\}$ rand N $\{n.\ \blacksquare(X_2(n)) * \ulcorner 0 \leq n \leq N \urcorner\}$

DEMO

Hypothesis

$\blacksquare X_2(n)$

$\blacksquare \frac{3}{2} \multimap \mathsf{wp}\ \mathsf{RS}()\{\mathsf{True}\}$

$0 \le n \le 2$

$X_2(n) \triangleq \mathsf{if}\ n < 2\ \mathsf{then}\ 0\ \mathsf{else}\ \frac{3}{2}$

Goal

$\mathsf{wp}\ \mathsf{let}\ x = n\ \mathsf{in}\ \mathsf{if}\ x < 2\ \mathsf{then}\ x\ \mathsf{else}\ \mathsf{RS}()\{\mathsf{True}\}$

Note that $1 + \frac{0+0+\frac{3}{2}}{3} = 1 + \frac{1}{2} = \frac{3}{2}$

$$1 + \sum_{n=0}^{N} \frac{X_2(n)}{N+1} \le x_1$$

$$\{\blacksquare x_1\}\ \mathsf{rand}\ N\ \{n.\ \blacksquare(X_2(n)) * \ulcorner 0 \le n \le N \urcorner\}$$

DEMO

Hypothesis	Goal
$\blacksquare X_2(n)$	wp if $n<2$ then n else RS $\{True\}$
$\blacksquare 3/2 \multimap$ wp RS $\{True\}$	
$0 \leq n \leq 2$	
$X_2(n) \triangleq$ if $n<2$ then 0 else $\frac{3}{2}$	
	Continue the rest of the proof by case analysis (left as exercise)

Key Ideas of Tachis

1. Costs can be realized as a RESOURCE ✳
2. Cost credits are distribut across branches
   in an EXPECTATION-PRESERVING MANNER ✳

$$1 + \sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1$$

$$\{\blacksquare\, x_1\}\ \text{rand}\ N\ \{n.\ \blacksquare(X_2(n)) \mid 0 \leq n \leq N\}$$

HT-RAND-EXP

Also in the Tachis paper

- Cost models
  - cost_{app}
  - $\text{cost}_{\text{tick}}$
  - $\text{cost}_{\text{entropy}}$
- Amortized Reasoning
- Many MANY examples
  - Fisher-Yates shuffle
  - Coupon collector
  - Quicksort
  - Meldable heaps

Tachis: Higher-Order Separation Logic with Credits for Expected Costs

PHILIPP G. HASELWARTER, Aarhus University, Denmark
KWING HEI LI, Aarhus University, Denmark
MARKUS DE MEDEIROS, New York University, USA
SIMON ODDERSHEDE GREGERSEN, New York University, USA
ALEJANDRO AGUIRRE, Aarhus University, Denmark
JOSEPH TASSAROTTI, New York University, USA
LARS BIRKEDAL, Aarhus University, Denmark

We present Tachis, a higher-order separation logic to reason about the expected cost of probabilistic programs. Inspired by the uses of time credits for reasoning about the running time of deterministic programs, we introduce a novel notion of probabilistic cost credit. Probabilistic cost credits are a separation logic resource that can be used to pay for the cost of operations in programs, and that can be distributed across all possible branches of sampling instructions according to their weight, thus enabling us to reason about expected cost. The representation of cost credits as separation logic resources gives Tachis a great deal of flexibility and expressivity. In particular, it permits reasoning about amortized expected cost by storing excess credits as potential into data structures to pay for future operations. Tachis further supports a range of cost models, including running time and entropy usage. We showcase the versatility of this approach by applying our techniques to prove upper bounds on the expected cost of a variety of probabilistic algorithms and data structures, including randomized quicksort, hash tables, and meldable heaps. All of our results have been mechanized using Coq, Iris, and the Coquelicot real analysis library.

PAUSE

Press any key to continue ...

Act II

Worlds of ERROR

Context: Many data-structures / algorithms trade accuracy for efficiency (SPEED) (MEMORY)

- Bloom filters
- Hashes
- Miller Rabin

Aim: We want to upper bound the probability for an error result

Context: Many data-structures / algorithms trade accuracy for efficiency (SPEED) (MEMORY)

- Bloom filters
- Hashes
- Miller Rabin

Aim: We want to upper bound the probability for an error result

Example: Suppose the return of the value 1 is an undesirable error result, how can we bound the error probability of RS()

Key idea 1: Computation cost as a RESOURCE!

Key idea 1: ~~computation cost~~ as a RESOURCE!

ERROR

$\$\varepsilon$ denotes ownership of $\varepsilon \in \mathbb{R}_{\geq 0}$ error credits

Credit splitting rule

$\$\varepsilon_1 + \varepsilon_2 \dashv\vdash \$\varepsilon_1 * \$\varepsilon_2$

Key idea 1: ~~Computation cost~~ as a RESOURCE!

ERROR

$↯ε$ denotes ownership of $ε \in \mathbb{R}_{\geq 0}$ error credits

Credit splitting rule

$↯(ε_1 + ε_2) \dashv\vdash ↯ε_1 * ↯ε_2$

New Credit One rule

$↯1 \vdash$ ~~False~~

(Why is this intuitively true???)

Adequacy Theorem of Eris

$\{\!\{\varepsilon\}\!\}\ e\ \{\!\{\phi\}\!\} \Rightarrow$ the probability e returns a value in $\neg\phi \leq \varepsilon$

NB: If e does not terminate, $\{\!\{\varepsilon\}\!\}\ e\ \{\!\{\phi\}\!\}$ holds *TRIVIALLY* !!

Key idea 2 : Expectation-preserving Composition

$$\frac{\sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1}{\{x_1\}\ \mathrm{rand}\ N\ \{n.\ (X_2(n))\ \ulcorner 0 \leq n \leq N \urcorner\}}\ \text{HT-RAND-EXP}$$

This is THE main rule of Eris!

Other rules remain the same, e.g.

$$\{l \mapsto v_1\}\ l \leftarrow v_2\ \{x.\ l \mapsto v_2 * \ulcorner x = () \urcorner\}$$

DEMO

Hypothesis	Goal
	Prob of RS () returning 1 $\leq \frac{1}{2}$

DEMO

Hypothesis	Goal
	$\{\{\frac{1}{2}\}$ RS $()\{v.\ v \neq 1\}$
	By adequacy theorem of Eris

DEMO

Hypothesis	Goal
{½}	wp RS () {v. v ≠ 1}
	; Intros

DEMO

Hypothesis	Goal
$\lozenge \frac{1}{2}$	wp RS () $\{ v.\ v \neq 1 \}$
$\triangleright \left(\lozenge \frac{1}{2} \mathrel{-\!\!*} \text{wp RS () } \{ v.\ v \neq 1 \} \right)$	
	¡Löb that MF!

DEMO

Hypothesis

$\lightning\, 1/2$

$\left(\lightning \frac{1}{2} \mathrel{-\!\!*} \mathsf{wp}\ \mathsf{RS}\ ()\ \{ v.\ v \neq 1 \}\right)$

Goal

$\mathsf{wp}\ \mathsf{let}\ x = \mathsf{rand}\ 2\ \mathsf{in}$
$\quad \mathsf{if}\ x < 2\ \mathsf{then}\ x\ \mathsf{else}\ \mathsf{RS}\ ()$
$\quad \{ v.\ v \neq 1 \}$

$$\sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1$$

$\{ \lightning x_1 \}\ \mathsf{rand}\ N\ \{ n.\ \lightning (X_2(n)) * \ulcorner 0 \leq n \leq N \urcorner \}$

DEMO

Hypothesis

$\$ X_2(n)$

$(\$\frac{1}{2} \mathrel{-\!\!*} \mathsf{wp}\ \mathsf{RS}\ ()\ \{v.\ v \neq 1\})$

$0 \leq n \leq 2$

$X_2 \triangleq \lambda n.$ if $n = 0$ then 0
else if $n = 1$ then 1
else $1/2$

Goal

wp let $x = n$ in
if $x < 2$ then x else $\mathsf{RS}\ ()$
$\{v.\ v \neq 1\}$

Note: $\frac{0+1+\frac{1}{2}}{3} = \frac{\frac{3}{2}}{3} = \frac{1}{2}$

$$\sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1$$

$\{\$x_1\}\ \mathsf{rand}\ N\ \{n.\ \$(X_2(n)) * \ulcorner 0 \leq n \leq N \urcorner\}$

DEMO

Hypothesis	Goal
$\{\!\{ X_2(n) \}\!\}$	wp if $n < 2$ then n else RS () $\{\!\{ v.\ v \neq 1 \}\!\}$
$\left(\{\!\{ \frac{1}{2} \}\!\} \mathrel{-\!\!*} \text{wp RS ()} \{\!\{ v.\ v \neq 1 \}\!\}\right)$	
$0 \leq n \leq 2$	
$X_2 \triangleq \lambda n.$ if $n = 0$ then 0 else if $n = 1$ then 1 else 1/2	Proceed by case analysis (Exercise for the audience :))

Key Ideas of ~~Tachis~~ Eris

1. ~~Costs~~ Error can be realized as a RESOURCE ✳
2. ~~Cost~~ Error credits are distribut across branches in an EXPECTATION-PRESERVING MANNER ✳

$$\frac{\sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1}{\{\lightning x_1\}\ \text{rand}\ N\ \{n.\ \lightning(X_2(n)) \neq \top 0 \leq n \leq N\}} + \lightning 1 \vdash \text{False}$$

PAUSE

Press any key to continue ...

Act III

Principle of Amplification

Recall the adequacy theorem of Eris:

$\{\!\!\{ \varepsilon \}\!\!\}\ e\ \{\!\!\{ \phi \}\!\!\} \Rightarrow$ the probability e returns a value in $\neg\phi \leq \varepsilon$

We know nothing about the probability of e returning a value in ϕ!

e.g. an infinite loop $\Omega \triangleq \mathrm{rec}\, f\, x = f\, x$

What if non-termination is also treated as an error / unfavorable result?

Introducing...

Eris

Adequacy Theorem of $Eris_t$:

$\langle \xi \varepsilon \rangle \, e \, \langle \phi \rangle \Rightarrow$ the probability e returns a value in $\phi \geq 1 - \varepsilon$

Adequacy Theorem of Eris_t :

$\langle \nleq \varepsilon \rangle\ e\ \langle \phi \rangle \Rightarrow$ the probability e returns a value in $\phi \geq 1-\varepsilon$

Eris_t has ALL rules of Eris .....

except

löb induction

I like =
I prayer

Key idea 1 : Error amplification

$$\frac{\varepsilon > 0 \qquad k > 1 \qquad \forall \varepsilon'.\ (\$(k\cdot\varepsilon') \multimap P) * \$\varepsilon' \vdash P}{\$\varepsilon \vdash P}\ \text{IND-ERR-AMP}$$

This is THE main rule of Erise

Key idea 1 : Error amplification

$$\frac{\varepsilon > 0 \qquad k > 1 \qquad \forall \varepsilon'.\ (\$(k \cdot \varepsilon') -\!\!* P) * \$\varepsilon' \vdash P}{\$\varepsilon \vdash P}\ \text{IND-ERR-AMP}$$

Intuition : If you have $\varepsilon > 0$ error credits, and you are able to amplify it by $k > 1$ repeatedly, you will eventually reach $\$1$ error credits!

$$\varepsilon \to k \cdot \varepsilon \to k^2 \cdot \varepsilon \to k^3 \cdot \varepsilon \to \ldots \to k^n \cdot \varepsilon \geq 1$$

This is THE main rule of Erisé

Key Idea 2: Error credits from THIN AIR!

$$\frac{\left(\forall \varepsilon . \langle P * \$\varepsilon * \lceil \varepsilon > 0 \rceil \rangle\ e\ \langle Q \rangle\right)}{\langle P \rangle\ e\ \langle Q \rangle}\ \text{THIN-AIR}$$

To prove any Hoare triple, you can generate some ARBITRARILY small error credit

DEMO

Hypothesis	Goal
	Prob of RS () returning 0 $\leq \frac{1}{2} = 1 - \frac{1}{2}$

DEMO

Hypothesis	Goal
	$\langle \xi \frac{1}{2} \rangle$ RS () $\langle v.\ v=0 \rangle$
	By adequacy theorem of Eriset

DEMO

Hypothesis	Goal
~~½~~	twp RS () ⟨v. v=0⟩
	i Intros

DEMO

Hypothesis	Goal
$\varepsilon > 0$	twp RS () $\langle v.\ v=0 \rangle$
~~ε~~	
~~$1/2$~~	
	Thin-air law =)

DEMO

Hypothesis	Goal
$\varepsilon > 0$	twp RS () ⟨v. v=0⟩
$\$\,\varepsilon$	
$\$\,1/2$	
$(\$(3\cdot\varepsilon) \multimap \$\,1/2 \multimap$ twp RS () ⟨v. v=0⟩)	
$\varepsilon > 0$ $k > 1$ $\varepsilon < 3$	$\forall A.\ (\$(3\cdot k)\varepsilon \multimap A) * \$\,3\varepsilon \vdash A$
	$\$\,3\varepsilon \vdash P$

Error induction rule
(take k=3)

DEMO

Hypothesis	Goal
$\varepsilon > 0$	twp let x = rand 2 in
$\lightning\, \varepsilon$	if $x < 2$ then x else RS ()
$\lightning\, 1/2$	$\langle v.\ v = 0 \rangle$
$(\lightning(3 \cdot \varepsilon) \multimap \lightning 1/2 \multimap$ twp RS () $\langle v.\ v = 0 \rangle)$	$\sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1$
	$\langle \lightning x_1 \rangle$ rand $N \langle n.\ \lightning (X_2(n)) \rangle$ $\ulcorner 0 \leq n \leq N \urcorner$

DEMO

Hypothesis	Goal
$\varepsilon > 0$	twp let $x = n$ in if $x < 2$ then x else RS () $\langle v.\ v = 0 \rangle$
$\lightning X_2(n)$	
$(\lightning(3 \cdot \varepsilon) \twoheadrightarrow \lightning 1/2 \twoheadrightarrow$ twp RS () $\langle v.\ v = 0 \rangle)$	
$X_2 \triangleq \lambda n.$ if $n = 0$ then 0 else if $n = 1$ then 1 else $\frac{1}{2} + 3 \cdot \varepsilon$	$\sum_{n=0}^{N} \frac{X_2(n)}{N+1} \leq x_1$
	$\langle \lightning x_1 \rangle$ rand N $\langle n.\ \lightning(X_2(n)) + \ulcorner 0 \leq n \leq N \urcorner \rangle$

DEMO

Hypothesis	Goal
$\varepsilon > 0$	twp if $n < 2$ then n else RS ()
$\not\downarrow X_2(n)$	$\langle v.\ v = 0 \rangle$
$0 \leq n \leq 2$	
$(\not\downarrow(3 \cdot \varepsilon) \multimap \not\downarrow 1/2 \multimap$ twp RS () $\langle v.\ v = 0 \rangle)$	
$X_2 \triangleq \lambda n.$ if $n = 0$ then 0 else if $n = 1$ then 1 else $\frac{1}{2} + 3 \cdot \varepsilon$	Proceed by case analysis (exercise =))

Key ideas of Eris$_t$

- Error amplification rule
- Error credits from Thin air

$$\frac{\varepsilon > 0 \qquad k > 1 \qquad \forall \varepsilon'.\ (\$(k\cdot\varepsilon') -\!\!* P) * \$\varepsilon' \vdash P}{\$\varepsilon \vdash P}$$

$*$

$$\frac{\forall \varepsilon.\ \langle P * \$\varepsilon * \lceil \varepsilon > 0 \rceil \rangle\ e\ \langle Q \rangle}{\langle P \rangle\ e\ \langle Q \rangle}$$

Also more in the Eris / Eris$_\epsilon$ paper

- Amortized reasoning
- Presampling tapes

$\{\alpha \hookrightarrow (N, n :: ns)\}$
rand(α) N
$\{v.\ \ulcorner v = n \urcorner * \alpha \hookrightarrow (N, ns)\}$

- Many MANY examples
  - Escaping spline
  - WalkSAT
  - Merkle trees

Error Credits: Resourceful Reasoning about Error Bounds for Higher-Order Probabilistic Programs

ALEJANDRO AGUIRRE, Aarhus University, Denmark
PHILIPP G. HASELWARTER, Aarhus University, Denmark
MARKUS DE MEDEIROS, New York University, USA
KWING HEI LI, Aarhus University, Denmark
SIMON ODDERSHEDE GREGERSEN*, New York University, USA
JOSEPH TASSAROTTI, New York University, USA
LARS BIRKEDAL, Aarhus University, Denmark

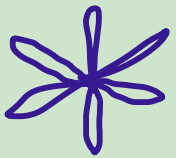
Probabilistic programs often trade accuracy for efficiency, and thus may, with a small probability, return an incorrect result. It is important to obtain precise bounds for the probability of these errors, but existing verification approaches have limitations that lead to error probability bounds that are excessively coarse, or only apply to first-order programs. In this paper we present Eris, a higher-order separation logic for proving error probability bounds for probabilistic programs written in an expressive higher-order language.

Our key novelty is the introduction of *error credits*, a separation logic resource that tracks an upper bound on the probability that a program returns an erroneous result. By representing error bounds as a resource, we recover the benefits of separation logic, including compositionality, modularity, and dependency between errors and program terms, allowing for more precise specifications. Moreover, we enable novel reasoning principles such as expectation-preserving error composition, amortized error reasoning, and error induction.

We illustrate the advantages of our approach by proving amortized error bounds on a range of examples, including collision probabilities in hash functions, which allow us to write more modular specifications for data structures that use them as clients. We also use our logic to prove correctness and almost-sure termination of rejection sampling algorithms. All of our results have been mechanized in the Coq proof assistant using the Iris separation logic framework and the Coquelicot real analysis library.

Main takeaways

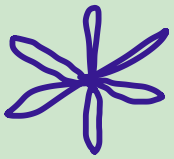
- Errors / cost can be realized as credit
  (amortization / concurrency)
- Expectation preserving rule for rand
- Error amplification induction principle
  (Super powerful!)
- Arbitrary small error credit



Everything formalized in Iris

Main takeaways

- Errors / cost can be realized as credit (amortization / concurrency)
- Expectation preserving rule for rand
- Error amplification induction principle (Super powerful!)
- Arbitrary small error credit

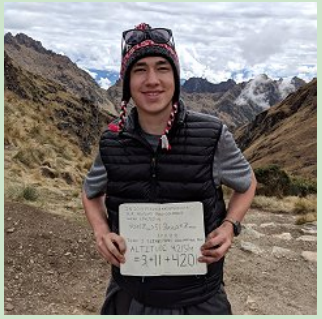


Everything formalized in Iris

Future work

- Concurrency
- Logically randomized atomicity
- Applications to security
- Continuous distributions
- One logic to rule them all

"Lars Probability Mafia"



Many other k3wl logics not discussed in this talk!

Clutch for asynchronous couplings (POPL 2024)

rand 3 ~~~~~ rand 3

Caliper for termination preserving refinement (CJCFP 2024)

Approxis for approximate couplings

rand 3 ~ ε ~ rand 4