Has there been work on embedding PP in theorem proving?

Can we combine the heap manipulation and probabilistic programming?

Want to conclude with big challenges in probabilistic programming.

Perspectives

Credits "John" for a lot of this, don't know who that is.

Question: Does it behave like diamond?

It gets hairy, don't wanna say much more.

Probability" Lilac paper, with title "A modal separation logic for conditional independence?"

Core semantic challenges

Usability

It is important to be able to pull back x

Definitions are in the paper.

That gives me my interpretation for disjoint union.

A nice aspect there is we've managed to take well studied measure factorization structure.

Disjoint union is a partial function that takes two probability spaces and tries to combine them into one, preserving the

Independent combination is a function.

Today, we're gonna try to get intuition about props like x \tilde{\text{~}} y.

Can we end props in smaller context to a larger context.

Do we get all the same properties from separation logic?

Bernoulli 1/2

x and y are independent random variables each following

Now, let's consider an analogous program

Separation Logic: has been quite helpful for imperative heap programs.

Separation Logic for Heap Programs

This is ongoing work, started with a POPL 2020 paper from Bartha, Hsu, Liao, "Probabilistic Separation Logic"

Probabilistic Programming

Program Logics for Probabilistic Programs

Motivation: Analogy between probability and mutable state have more in common than you think.

Probability and mutable state have more in common than you think.

Open questions there.

It is hard to make denotational arguments about prob, even

Paper: Staton

IF we add higher-order functions, recursion, other stuff, adds

Let us do this

I showed you a very simple language, no loops, which does

something that looks like a probability distribution

I want something that takes a term, and out comes a prob,

Defining different structures could be quite useful.

we needed for our examples

Interesting question

Conditioning needs to be in our logic

Something perpendicular maybe.

Have to be clever about it.

a function from events to 0,1

Behaves very strangely due to real numbers, can always carve out from it.

It's a large set of possible intervals I can carve out for my new

my program

I like this one.

They'll often just output a wrong answer instead of failing

No debuggers, profilers, ide tooling

Scalable one will require a synthesis of lots of ideas.

It's a big barrier to using them in real programs

We have this underlying hardness that's difficult to avoid

Helpful that a better deductive logic, better sampling will

Hopeful that a better deductive logic, better sampling will

We have this underlying hardness that's difficult to avoid

Interesting question

Conditioning needs to be in our logic

Something perpendicular maybe.

Have to be clever about it.

a function from events to 0,1

Behaves very strangely due to real numbers, can always carve out from it.

It's a large set of possible intervals I can carve out for my new

my program