Conditioning and density, mathematically and computationally

Chung-chieh Shan (with Wazim Mohammed Ismail)

INDIANA UNIVERSITY

Mathematical Foundations of Programming Semantics
June 14, 2014

This work is supported by DARPA grant FA8750-14-2-0007.
Probabilistic programming

Alice beat Bob at a game. Is she better than him at it?

Generative story
Alice beat Bob at a game. Is she better than him at it?

Generative story

\[
a \leftarrow \text{normal } 10 \ 3
\]
Probabilistic programming

Alice beat Bob at a game. Is she better than him at it?

Generative story

\[
a \leftarrow \text{normal } 10 3
\]
\[
b \leftarrow \text{normal } 10 3
\]
Probabilistic programming

Alice beat Bob at a game. Is she better than him at it?

Generative story

\[ a \leftarrow \text{normal} \ 10 \ 3 \]
\[ b \leftarrow \text{normal} \ 10 \ 3 \]
\[ l \leftarrow \text{normal} \ 0 \ 2 \]
Alice beat Bob at a game. Is she better than him at it?

**Generative story**

\[
a \leftarrow \text{normal} \ 10 \ 3  \\
b \leftarrow \text{normal} \ 10 \ 3  \\
l \leftarrow \text{normal} \ 0 \ 2
\]

**Observed effect**

condition (a - b > l)
Alice beat Bob at a game. Is she better than him at it?

**Generative story**

\[
\begin{align*}
a & \gets \text{normal } 10 \ 3 \\
b & \gets \text{normal } 10 \ 3 \\
l & \gets \text{normal } 0 \ 2 \\
\end{align*}
\]

**Observed effect**

\[
\text{condition } (a-b > l)
\]

**Hidden cause**

\[
\text{return } (a > b)
\]
Alice beat Bob at a game. Is she better than him at it?

**Generative story**

\[
a \leftarrow \text{normal } 10 \ 3 \\
b \leftarrow \text{normal } 10 \ 3 \\
l \leftarrow \text{normal } 0 \ 2
\]

**Observed effect**

condition \((a-b > l)\)

**Hidden cause**

\[
\text{return } (a > b)
\]

**Denoted measure:**

\[
\lambda c. \int da \int db \int dl \langle a - b > l \rangle c(a > b) \\
N(10,3) \ N(10,3) \ N(0,2)
\]
Importance sampling

Alice beat Bob at a game. Is she better than him at it?

Generative story

\[ a \leftarrow \text{normal } 10 \ 3 \]
\[ b \leftarrow \text{normal } 10 \ 3 \]

weight \((1 + \text{erf}(\frac{a-b}{2\sqrt{2}}))/2\)

Hidden cause

return \((a > b)\)
Importance sampling

Alice beat Bob at a game. Is she better than him at it?

Generative story

\[ a \leftarrow \text{normal} \ 10 \ 3 \]
\[ b \leftarrow \text{normal} \ 10 \ 3 \]
\[ \text{weight} \ \frac{1 + \text{erf} \left( \frac{a-b}{2\sqrt{2}} \right)}{2} \]

Hidden cause

\[ \text{return} \ (a > b) \]

Denoted measure:

\[ \lambda_c \int da \int db \ \frac{1 + \text{erf} \left( \frac{a-b}{2\sqrt{2}} \right)}{2} \ c(a > b) \]

\[ N(10,3) \quad N(10,3) \]
Conditioning on a real number

The bike was moving near location 9. Where’s it now?

Generative story
The bike was moving near location 9. Where’s it now?

Generative story

\[ x \leftarrow \text{normal } 10 3 \]
Conditioning on a real number

The bike was moving near location 9. Where’s it now?

**Generative story**

```
x <- normal 10 3
m <- normal x 1
```
Conditioning on a real number

The bike was moving near location 9. Where’s it now?

Generative story

\[
x \leftarrow \text{normal} \ 10 \ 3 \\
m \leftarrow \text{normal} \ x \ 1 \\
x' \leftarrow \text{normal} \ (x+5) \ 2
\]

Denoted measure:
\[
\lambda c. \int dx \int dm \int dx' \ c(x, m, x') \\
N(10,3) \ N(x,1) \ N(x+5,2)
\]
Conditioning on a real number

The bike was moving near location 9. Where’s it now?

**Generative story**

\[ x \leftarrow \text{normal } 10 \ 3 \]
\[ m \leftarrow \text{normal } x \ 1 \]
\[ x' \leftarrow \text{normal } (x+5) \ 2 \]

**Observed effect**

condition \( (m = 9) \)

**Denoted measure:**

\[ \lambda c. \int dx \int dm \int dx' \ c(x, m, x') \]

\[ N(10,3) \ N(x,1) \ N(x+5,2) \]
Conditioning on a real number

The bike was moving near location 9. Where’s it now?

Generative story

\[
x \leftarrow \text{normal} \quad 10 \quad 3 \\
m \leftarrow \text{normal} \quad x \quad 1 \\
x' \leftarrow \text{normal} \quad (x+5) \quad 2
\]

Observed effect

condition (m = 9)

Denoted measure:

\[
\lambda c. \quad \int dm \quad \int dx \quad \int dx' \quad c(x, m, x') \\
N(10, \sqrt{10}) \quad N(\frac{9}{10} m + \frac{1}{10} \ 10, \sqrt{\frac{9}{10}}) \quad N(x+5,2)
\]
The bike was moving near location 9. Where’s it now?

Generative story

x <- normal 10 3  \quad m <- normal 10 \sqrt{10} \quad \text{let } m = 9
m <- normal x 1  \quad x <- normal \left( \frac{9}{10} m + \frac{1}{10} 10 \right) \sqrt{\frac{9}{10}}
x' <- normal (x+5) 2

Observed effect

\text{condition } (m = 9)

Denoted \textbf{conditional} measure:

\lambda m. \lambda c. \int dx \int dx' c(x, m, x')
\quad \quad N\left( \frac{9}{10} m + \frac{1}{10} 10, \sqrt{\frac{9}{10}} \right) \quad N(x+5,2)
Conditioning on a real number

The bike was moving near location 9. Where’s it now?

Generative story

\[
\begin{align*}
x & \leftarrow \text{normal} \frac{9}{10} \sqrt{\frac{9}{10}} \\
x' & \leftarrow \text{normal} (x+5) 2
\end{align*}
\]

Denoted marginal measure:

\[
\lambda c. \quad \int dx \quad \int dx' \ c(x, x')
\]

\[
\mathcal{N} \left( \frac{9}{10}, \sqrt{\frac{9}{10}} \right) \quad \mathcal{N} (x+5, 2)
\]
Conditioning on a real number

The bike was moving near location 9. Where’s it now?

Generative story

\[ x \leftarrow \text{normal} \left( \frac{9}{10}, \sqrt{\frac{9}{10}} \right) \]
\[ x' \leftarrow \text{normal} \left( x+5, 2 \right) \]

Hidden cause

return \( x' \)

Denoted \textbf{marginal} measure:

\[
\lambda c. \quad \int dx \quad \int dx' \ c(x') \\
N\left( \frac{9}{10}, \sqrt{\frac{9}{10}} \right) \quad N(x+5,2)
\]
Conditioning on a real number

The bike was moving near location 9. Where’s it now?

Generative story

\[
x' \leftarrow \text{normal} \frac{141}{10} \sqrt{\frac{49}{10}}
\]

Hidden cause

return \(x'\)

Denoted **marginal** measure:

\[
\lambda c. \int dx' \, c(x')
\]

\[
N\left(\frac{141}{10}, \sqrt{\frac{49}{10}}\right)
\]
Importance sampling

Each sample has an *importance weight*
Importance sampling

Each sample has an *importance weight*

**Generative story**

\[
x \leftarrow \text{normal } 10 \ 3 \\
m \leftarrow \text{normal } x \ 1 \\
x' \leftarrow \text{normal } (x+5) \ 2
\]

**Observed effect**

condition \((m = 9)\)
Importance sampling

Each sample has an importance weight:
How much did we rig our random choices to avoid rejection?

Generative story

\[ x \leftarrow \text{normal } 10 \ 3 \]

weight \[ e^{-\frac{(9-x)^2}{2}/\sqrt{2\pi}} \]

\[ x' \leftarrow \text{normal } (x+5) \ 2 \]

Denoted measure:

\[ \lambda c. \int dx \frac{e^{-\frac{(9-x)^2}{2}}}{\sqrt{2\pi}} \]

\[ N(10,3) \]

\[ \int dx' c(x, m, x') \]

\[ N(x+5,2) \]
Probability or density?

Generative story

x <- flip True False
m <- (if x then flip 0 1
    else normal 0 1)

Observed effect

condition (m = 1)

Hidden cause

return x
Generative story

\[ x \leftarrow \text{flip True False} \]

weight (if \( x \) then \( \frac{1}{2} \)
\[ \text{else } \frac{e^{-1/2}}{\sqrt{2\pi}} \])

Hidden cause

\[ \text{return } x \]
Ambient measures

Let $\mu$ be a measure over pairs $(x, m)$.

$$\mu = \ldots \gets \ldots = \lambda c. \int \ldots c(x, m)$$

$$\ldots \gets \ldots$$
$$\ldots$$

return $(x, m)$

Condition $\mu$ on $m$: Express $\mu$ by choosing $m$ first! (easy when $m$ is drawn from a fixed measure)

$$\mu = m \gets \ldots = \lambda c. \int dm \int \ldots c(x, m)$$

$$\ldots \gets \ldots$$
$$\ldots$$

return $(x, m)$

It doesn’t matter what the ambient measure $\tau$ is, but there must be one, such as Lebesgue.
Ambient measures

Let $\mu$ be a measure over pairs $(x, m)$.

\[
\mu = \ldots \leftarrow \ldots = \lambda c. \int \ldots c(x, m)
\]

\[
\ldots \leftarrow \ldots
\]

\[
\ldots
\]

return $(x, m)$

Condition $\mu$ on $m = \text{Express } \mu$ by choosing $m$ first!
(easy when $m$ is drawn from a fixed measure)

\[
\mu = m \leftarrow \ldots = \lambda c. \int dm \int \ldots c(x, m)
\]

\[
\ldots \leftarrow \ldots
\]

\[
\ldots
\]

return $(x, m)$

It doesn’t matter what the ambient measure $\tau$ is, but there must be one, such as Lebesgue.
Ambient measures

Let $\mu$ be a measure over pairs $(x, m)$.

$$
\mu = \ldots \leftarrow \ldots = \lambda c. \int \cdots c(x, m)
$$

$$
\ldots \leftarrow \ldots
$$

$$
\ldots
$$

$$
\text{return } (x, m)
$$

**Condition $\mu$ on $m$** = Express $\mu$ by choosing $m$ first!
(easy when $m$ is drawn from a fixed measure)

$$
\nu(m) = m \leftarrow \tau
$$

$$
\ldots \leftarrow \ldots
$$

$$
\ldots
$$

$$
\text{return } (x, m)
$$

It doesn’t matter what the ambient measure $\tau$ is, but there must be one, such as Lebesgue.
Discrete importance sampling revisited

\[
a \leftarrow \text{normal } 10 \ 3 \\
b \leftarrow \text{normal } 10 \ 3 \\
l \leftarrow \text{normal } 0 \ 2 \\
w \leftarrow \text{dirac } (a-b > l)
\]

\[
\text{return } (a > b)
\]

\[
\text{probability density (absolute continuity) w.r.t. counting}
\]
Discrete importance sampling revisited

\[ \begin{align*}
    a & \leftarrow \text{normal } 10 \ 3 \\
    b & \leftarrow \text{normal } 10 \ 3 \\
    l & \leftarrow \text{normal } 0 \ 2 \\
    w & \leftarrow \text{dirac } (a-b > l) \\
    \text{return } (a > b)
\end{align*} \]
Continuous importance sampling revisited

\[
x \leftarrow \text{normal} \ 10 \ 3
\]
\[
m \leftarrow \text{normal} \ x \ 1
\]
\[
x' \leftarrow \text{normal} \ (x+5) \ 2
\]
\[
\ldots
\]

\[
x \leftarrow \text{normal} \ 10 \ 3
\]
\[
m \leftarrow \text{lebesgue}
\]
\[
\text{weight} \ e^{-\frac{(m-x)^2}{2}/\sqrt{2\pi}}
\]
\[
x' \leftarrow \text{normal} \ (x+5) \ 2
\]
\[
\ldots
\]

probability density (absolute continuity) w.r.t. lebesgue
Continuous importance sampling revisited

\[
x \leftarrow \text{normal } 10 3
\]
\[
m \leftarrow \text{normal } x 1
\]
\[
x' \leftarrow \text{normal } (x+5) 2
\]
\[
\ldots
\]

\[
\text{weight } e^{-\frac{(m-x)^2}{2}} / \sqrt{2\pi}
\]

\[
x \leftarrow \text{normal } 10 3
\]
\[
m \leftarrow \text{lebesgue}
\]
\[
x' \leftarrow \text{normal } (x+5) 2
\]
\[
\ldots
\]

probability density (absolute continuity) w.r.t. lebesgue
Discrete+continuous importance sampling revisited

\[
\begin{align*}
x & \leftarrow \text{flip True False} \\
m & \leftarrow (\text{if } x \text{ then flip 0 1 else normal 0 1}) \\
& \cdots \\

x & \leftarrow \text{flip True False} \\
m & \leftarrow (x' \leftarrow \text{flip True False} \\
& \quad \text{if } x' \text{ then flip 0 1 else normal 0 1}) \\
\text{weight} & \leftarrow (\text{if } x \text{ then if } m==0 || m==1 \text{ then 2 else 0} \\
& \quad \text{else if } m==0 || m==1 \text{ then 0 else 2}) \\
& \cdots
\end{align*}
\]

probability density (absolute continuity) w.r.t. marginal
Discrete+continuous importance sampling revisited

x <- flip True False
m <- (if x then flip 0 1 else normal 0 1)
...

x <- flip True False
m <- (x' <- flip True False
    if x' then flip 0 1 else normal 0 1)
weight (if x then if m==0 || m==1 then 2 else 0
    else if m==0 || m==1 then 0 else 2)
...

probability density (absolute continuity) w.r.t. marginal
Borel’s paradox

Requires change of variable:

\[ z \leftarrow \text{uniform} \quad -1 \quad 1 \]
\[ \theta \leftarrow \text{uniform} \quad -\pi/2 \quad \pi/2 \]
\[ y \leftarrow \text{dirac} \quad \sqrt{1 - z^2} \sin \theta \]
\[ \quad \ldots \]

\[ z \leftarrow \text{uniform} \quad -1 \quad 1 \]
\[ y \leftarrow \text{uniform} \quad -\sqrt{1 - z^2} \quad \sqrt{1 - z^2} \]
\[ \theta \leftarrow \text{dirac} \quad \arcsin(y/\sqrt{1 - z^2}) \]
\[ \text{weight} \quad 2/\cos \theta \]
\[ \quad \ldots \]
Borel’s paradox

Requires change of variable:

\[
\begin{align*}
  z &\sim \text{uniform} \quad -1 \quad 1 \\
  \theta &\sim \text{uniform} \quad -\pi/2 \quad \pi/2 \\
  y &\sim \text{dirac} \quad \sqrt{1 - z^2} \sin \theta \\
  \theta &\sim \text{dirac} \quad \arcsin(y/\sqrt{1 - z^2}) \\
  \text{weight} &\quad 2/\cos \theta
\end{align*}
\]
Another desideratum

Generative story

\[ x \leftarrow \text{normal} \ 0 \ 1 \]
\[ m \leftarrow \text{dirac} \ (x \ * \ 2) \]

Observed effect

condition \ (m = 32) \]

Hidden cause

return \ (x \ == \ 16) \]

Should give \texttt{True} with 100\% probability.
Another desideratum

Generative story

\begin{align*}
x & \leftarrow \text{normal} \ 0 \ 1 \\
m & \leftarrow \text{dirac} \ (x \ / \ 2)
\end{align*}

Observed effect

condition (m = 32)

Hidden cause

return (x == 64)

Should give True with 100\% probability.
Interim summary

**Conditioning means disintegration.**

(Chang & Pollard)

*If* the conditioned choice has a density with respect to an ambient measure, *then* we can transform the program locally into importance sampling.

- Simple ambient measures: counting, lebesgue
- Complex ambient measures: marginal, ...

How to infer an ambient measure?
How to calculate density with respect to it?

(Pfeffer, Bhat et al.)
Interim summary

**Conditioning means disintegration.**

(Chang & Pollard)

*If* the conditioned choice has a density with respect to an ambient measure, *then* we can transform the program locally into importance sampling.

▶ Simple ambient measures:
  - counting, lebesgue
▶ Complex ambient measures:
  - marginal, ...

How to infer an ambient measure?
How to calculate density with respect to it?

(Pfeffer, Bhat et al.)
Density calculator demo

\[ x \leftarrow \text{uniform } 0 \quad 1 \]
\[ x + x + \text{uniform } 0 \quad 1 \quad \updownarrow \]
\[ \int_0^1 \langle 0 < t - x - x < 1 \rangle \, dx \quad \updownarrow \]
\[ \begin{cases} 
0 & \text{if } t < 0 \\
\frac{t}{2} & \text{if } t < 1 \\
\frac{1}{2} & \text{if } t < 2 \\
\frac{3}{2} - \frac{t}{2} & \text{if } t < 3 \\
0 & \text{if } 3 \leq t 
\end{cases} \]
Density calculator demo

\[ \exp \left( \log \left( \text{uniform } 0 \ 1 \right) + \log \left( \text{uniform } 0 \ 1 \right) \right) \]
\[ \downarrow \]

\[ \begin{cases} \int_0^1 \langle 0 < e^{\ln t - \ln x} < 1 \rangle e^{\ln t - \ln x} \, dx / t & \text{if } 0 < t \\ 0 & \text{otherwise} \end{cases} \]
\[ \downarrow \]

\[ \begin{cases} -\ln t & \text{if } 0 < t < 1 \\ 0 & \text{otherwise} \end{cases} \]
Density calculator demo

\[
\begin{align*}
x & \leftarrow \text{uniform } 0 \text{ 1} \\
& (\text{if } x < .2 \text{ then } 0 \text{ else uniform } 0 \text{ 1}) \\
& \quad + (\text{if } x > .7 \text{ then } 1 \text{ else uniform } 0 \text{ 1})
\end{align*}
\]

Pfeffer’s approximate algorithm randomly samples from one term then calculates density of the other term.
Mochastic vs stochastic lambda calculus

Mochastic/stochastic only

\[ x \leftarrow \text{normal } 0 \ 1 \]
\[ \text{return (exp x + 1)} \]

\[ x \leftarrow \text{normal } 0 \ 1 \]
\[ y \leftarrow \text{normal } 0 \ 1 \]
\[ \text{return (x + y)} \]

\[ x \leftarrow \text{flip True False} \]
\[ \text{if x} \]
\[ \text{then flip 0 1} \]
\[ \text{else normal 0 1} \]

\[ x \leftarrow \text{normal } 0 \ 1 \]
\[ \text{return (x + x)} \]

Stochastic

\[ \text{exp (normal } 0 \ 1 \) + 1 \]

\[ \text{normal } 0 \ 1 + \text{normal } 0 \ 1 \]

\[ \text{if flip True False} \]
\[ \text{then flip 0 1} \]
\[ \text{else normal 0 1} \]

\[ \text{none} \]
Stochastic lambda calculus: expectation

\[
\begin{align*}
\left[ \text{uniform } 0 \ 1 \right] & \quad \rho \ c = \int_{0}^{1} c(t) \ dt \\
\left[ \exp \ \varepsilon \right] & \quad \rho \ c = \left[ \varepsilon \right] \rho \left( \lambda x. \ c(e^{x}) \right)
\end{align*}
\]
Stochastic lambda calculus: density

\[ \text{[uniform 0 1]} \rho c = \int_0^1 c(t) \, dt \]
\[ \text{[exp } \epsilon \text{]} \rho c = \text{[} \epsilon \text{]} \rho (\lambda x. c(e^x)) \]

**Goal:** Given \( \epsilon \), find \( D(\epsilon) \) such that
\[ \text{[} \epsilon \text{]} \rho c = \int D(\epsilon) \rho t \times c(t) \, dt \quad \text{for all } \rho, c. \]

**Approach:** Define \( D(\epsilon) \) by recursion on \( \epsilon \).
Stochastic lambda calculus: density

\[
\begin{align*}
[\text{uniform } 0 \ 1] \ \rho \ c &= \int_0^1 c(t) \, dt \\
[\exp \ \epsilon] \ \rho \ c &= [\epsilon] \ \rho \ (\lambda x. \ c(e^x))
\end{align*}
\]

**Goal:** Given \( \epsilon \), find \( D(\epsilon) \) such that

\[
[\epsilon] \ \rho \ c = \int D(\epsilon) \ \rho \ t \times c(t) \, dt \quad \text{for all } \rho, c.
\]

One base case:

\[
D(\text{uniform } 0 \ 1) \ \rho \ t = \langle 0 < t < 1 \rangle
\]
Stochastic lambda calculus: density

\[ \text{[uniform } 0 \ 1]\ \rho \ c = \int_0^1 c(t) \, dt \]
\[ \text{[exp } \varepsilon]\ \rho \ c = \varepsilon \rho (\lambda x. c(e^x)) \]

**Goal:** Given \( \varepsilon \), find \( D(\varepsilon) \) such that

\[ \varepsilon \rho \ c = \int D(\varepsilon) \rho \ t \times c(t) \, dt \quad \text{for all } \rho, c. \]

One recursive case:

\[ D(\exp \varepsilon) \rho \ t = ??? \]

\[ \exp \varepsilon \rho \ c = \int_{-\infty}^{\infty} ??? \times c(t) \, dt \]
Stochastic lambda calculus: density

\[ \text{[uniform 0 1]} \rho c = \int_0^1 c(t) \, dt \]

\[ \text{[exp } \epsilon \text{]} \rho c = \text{[} \epsilon \text{]} \rho (\lambda x. c(e^x)) \]

**Goal:** Given \( \epsilon \), find \( D(\epsilon) \) such that

\[ \text{[} \epsilon \text{]} \rho c = \int D(\epsilon) \rho t \times c(t) \, dt \quad \text{for all } \rho, c. \]

One recursive case:

\[ D(\text{exp } \epsilon) \rho t = ??? \]

\[ \text{[exp } \epsilon \text{]} \rho c = \int_{-\infty}^{\infty} ??? \times c(t) \, dt \]

\[ \|
\text{[} \epsilon \text{]} \rho (\lambda x. c(e^x)) = \int_{-\infty}^{\infty} D(\epsilon) \rho x \times c(e^x) \, dx
\]
Stochastic lambda calculus: density

\[ \text{uniform 0 1] } \rho \ c = \int_0^1 c(t) \, dt \]
\[ \text{exp } \varepsilon \] \rho \ c = [\varepsilon] \rho (\lambda x. c(e^x))

**Goal:** Given \( \varepsilon \), find \( D(\varepsilon) \) such that

\[ [\varepsilon] \rho \ c = \int D(\varepsilon) \rho \ t \times c(t) \, dt \quad \text{for all } \rho, c. \]

One recursive case:

\[ D(\text{exp } \varepsilon) \rho \ t = \langle t > 0 \rangle \frac{D(\varepsilon) \rho \ (\ln t)}{t} \]
\[ [\text{exp } \varepsilon] \rho \ c = \int_{-\infty}^{\infty} \langle t > 0 \rangle \frac{D(\varepsilon) \rho \ (\ln t)}{t} \times c(t) \, dt \]
\[ [\varepsilon] \rho (\lambda x. c(e^x)) = \int_{-\infty}^{\infty} D(\varepsilon) \rho \ x \times c(e^x) \, dx \]
Stochastic lambda calculus: density

\[ [\varepsilon_1 + \varepsilon_2] \rho c = [\varepsilon_1] \rho (\lambda x_1. [\varepsilon_2] \rho (\lambda x_2. c(x_1 + x_2))) \]
\[ = [\varepsilon_2] \rho (\lambda x_2. [\varepsilon_1] \rho (\lambda x_1. c(x_1 + x_2))) \]

Randomized algorithm for binary operators

\[ D(\varepsilon_1 + \varepsilon_2) \rho t = [\varepsilon_1] \rho (\lambda x. D(\varepsilon_2) \rho (t - x)) \]
\[ = [\varepsilon_2] \rho (\lambda x. D(\varepsilon_1) \rho (t - x)) \]
\[ [42] \rho c = c(42) \]
\[ [\text{var}] \rho c = c(\rho \text{var}) \]
\[ [\text{var} \leftarrow \varepsilon_1; \varepsilon_2] \rho c = [\varepsilon_1] \rho (\lambda x. [\varepsilon_2] (\rho\{\text{var} \mapsto x\}) c) \]

Three strategies for bound variables

\[
D(\text{bool}_\text{var}) \rho t = \langle \rho \text{ bool}_\text{var} = t \rangle \\
D(\text{var} \leftarrow \varepsilon_1; \varepsilon_2) \rho t = D(\varepsilon_2\{\text{var} \mapsto \varepsilon_1\}) \rho t \\
\text{if } \varepsilon_2 \text{ uses } \text{var} \text{ at most once} \\
D(\text{var} \leftarrow \varepsilon_1; \varepsilon_2) \rho t = [\varepsilon_1] \rho (\lambda x. D(\varepsilon_2) (\rho\{\text{var} \mapsto x\}) (t))
\]
Probabilistic programming
- Denote measure by generative story
- Run backwards to infer cause from effect

Mathematical reasoning
- Define conditioning as disintegration
- Perform importance sampling
- Derive density calculator

Need semantics and inference for loops
- Bag of words
- Brownian motion
- Probabilistic context-free grammars
Come to Indiana University to create essential abstractions and practical languages for clear, robust and efficient programs.

Dan Friedman
relational & logic languages, meta-circularity & reflection

Amr Sabry
quantum computing, type theory, information effects

Jeremy Siek
gradual typing, mechanized metatheory, high performance

Ryan Newton
streaming, distributed & GPU DSLs, Haskell deterministic parallelism

Chung-chieh Shan
probabilistic programming, semantics

Sam Tobin-Hochstadt
types for untyped languages, contracts, languages for the Web

Check out our work: Boost Libraries · Build to Order BLAS · C++ Concepts · Chapel Generics · HANSEI · JavaScript Modules · Racket & Typed Racket · miniKanren · LVars · monad-par · meta-par · WaveScript

http://lambda.soic.indiana.edu/