A Type Theory for Probability Density Functions

Sooraj Bhat Ashish Agarwal★
Richard Vuduc Alexander Gray

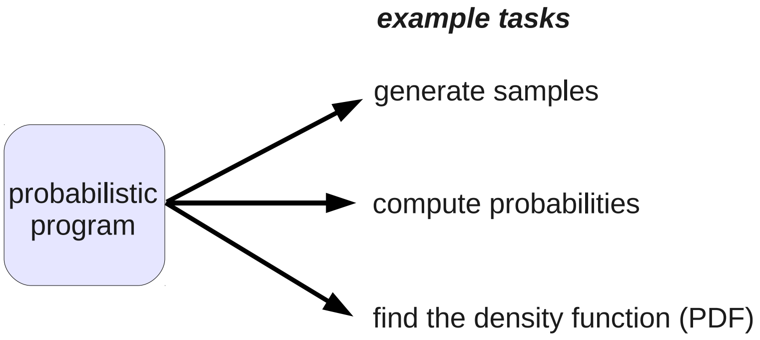
Georgia Institute of Technology
★New York University

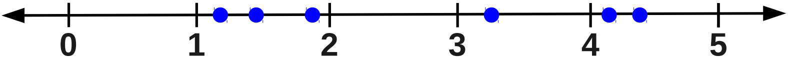
POPL 2012 – January 27, 2012

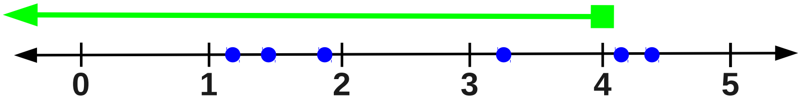
An example probabilistic program

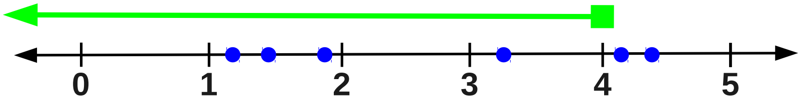
$$
\begin{array}{l}
\textbf{var } z \sim \text{flip } (1/2) \textbf{ in} \\
\quad \textbf{var } x_1 \sim \text{uniform } 1\ 2 \textbf{ in} \\
\quad\quad \textbf{var } x_2 \sim \text{uniform } 3\ 5 \textbf{ in} \\
\quad\quad\quad \text{return } (\textbf{if } z \textbf{ then } x_1 \textbf{ else } x_2)
\end{array}
$$

Using a probabilistic program

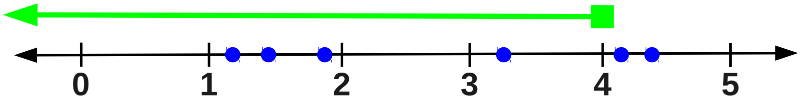



0
1
2
3
4
5

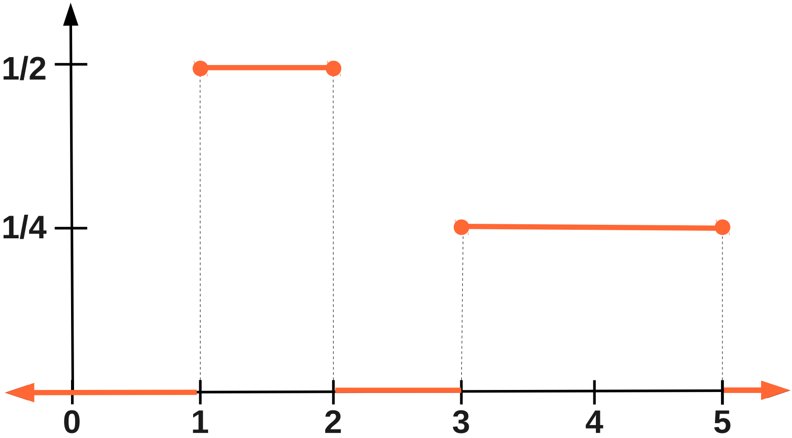

0
1
2
3
4
5



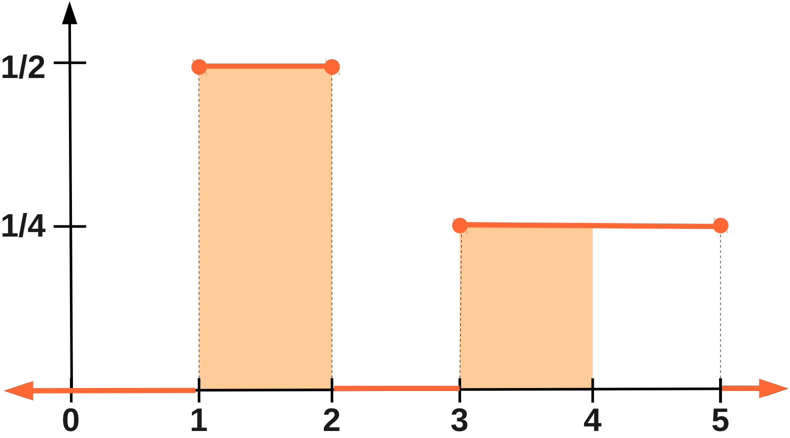
- $\mathbb{P}(A) = 3/4$



- $\mathbb{P}(A) = 3/4$
- how can we compute this?

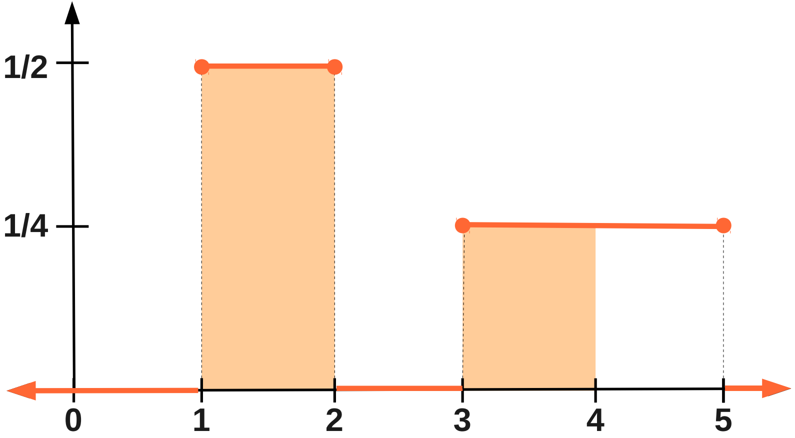

1/2
1/4
0
1
2
3
4
5

$$\mathbb{P}(A) = \int_A f(x)\ dx$$



This is the *probability density function* (PDF).

$$\mathbb{P}(A) = \int_A f(x)\ dx$$



What we want

in:

$$\begin{array}{l}\textbf{var } z \sim \text{flip } (1/2) \textbf{ in} \\ \quad \textbf{var } x_1 \sim \text{uniform } 1\ 2 \textbf{ in} \\ \quad\quad \textbf{var } x_2 \sim \text{uniform } 3\ 5 \textbf{ in} \\ \quad\quad\quad \text{return } (\textbf{if } z \textbf{ then } x_1 \textbf{ else } x_2)\end{array}$$

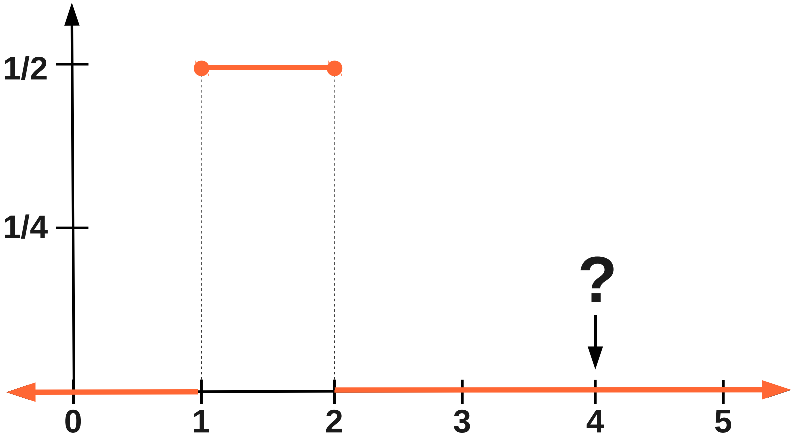
out:

$$\begin{aligned}\mathbf{f(x)} &= \langle \mathbf{1} \leq \mathbf{x} \leq \mathbf{2} \rangle * (\mathbf{1/2}) \\ &\quad + \langle \mathbf{3} \leq \mathbf{x} \leq \mathbf{5} \rangle * (\mathbf{1/4})\end{aligned}$$

A PDF may not exist!

$$\begin{array}{l}
\textbf{var } z \sim \text{flip } (1/2) \textbf{ in} \\
\;\textbf{var } x_1 \sim \text{uniform } 1\ 2 \textbf{ in} \\
\;\;\textbf{var } x_2 \sim \text{return } 4 \textbf{ in} \\
\quad\quad \text{return } (\textbf{if } z \textbf{ then } x_1 \textbf{ else } x_2)
\end{array}$$

A PDF may not exist!



Contributions

First formal syntactic work on PDFs.

- type system ("$\mathbb{P}$ has a PDF")
- compiler for calculating PDFs

Abstract syntax

$$
\begin{array}{lrl}
\text{base types} & \tau ::= & \mathrm{bool} \mid \mathrm{Z} \mid \mathrm{R} \mid \tau_1 \times \tau_2 \\
\text{expressions} & \varepsilon ::= & x \mid l \mid op \;\; \varepsilon_1 ... \varepsilon_n \\
 & & \mid \textbf{if } \varepsilon_1 \textbf{ then } \varepsilon_2 \textbf{ else } \varepsilon_3
\end{array}
$$

Abstract syntax

$$
\begin{array}{rl}
\text{base types} & \tau ::= \mathsf{bool} \mid \mathsf{Z} \mid \mathsf{R} \mid \tau_1 \times \tau_2 \\
\text{expressions} & \varepsilon ::= x \mid l \mid op \;\; \varepsilon_1 \ldots \varepsilon_n \\
& \quad\;\; \mid \textbf{if } \varepsilon_1 \textbf{ then } \varepsilon_2 \textbf{ else } \varepsilon_3 \\
\text{types} & t ::= \tau \mid \mathsf{dist}\ \tau \\
\text{distributions} & e ::= \mathsf{random} \\
& \quad\;\; \mid \mathsf{return}\ \varepsilon \\
& \quad\;\; \mid \textbf{var } x \sim e_1 \textbf{ in } e_2
\end{array}
$$

Abstract syntax

$$
\begin{array}{rrcl}
\text{base types} & \tau & ::= & \mathsf{bool} \mid \mathsf{Z} \mid \mathsf{R} \mid \tau_1 \times \tau_2 \\
\text{expressions} & \varepsilon & ::= & x \mid l \mid op \;\; \varepsilon_1 \ldots \varepsilon_n \\
 & & & \mid \textbf{if } \varepsilon_1 \textbf{ then } \varepsilon_2 \textbf{ else } \varepsilon_3 \\
\text{types} & t & ::= & \tau \mid \mathsf{dist}\ \tau \\
\text{distributions} & e & ::= & \mathsf{random} \\
 & & & \mid \mathsf{return}\ \varepsilon \\
 & & & \mid \textbf{var } x \sim e_1 \textbf{ in } e_2 \\
\text{programs} & p & ::= & \mathsf{pdf}\ e
\end{array}
$$

The power of return+bind

The *probability monad* [Giry '82]

- semantics [Ramsey & Pfeffer POPL'03]
- verification [Audebaud & Paulin-Mohring MPC'06]
- sampling [Park, Pfenning, Thrun POPL'05]
- EDSLs [Erwig & Kollmansberger JFP'05, Kiselyov & Shan '09]

The power of return+bind+random

$$\text{flip } \varepsilon := \\ \quad \textbf{var } u \sim \text{random } \textbf{in} \\ \qquad \text{return } (u \leq \varepsilon)$$

$$\text{uniform } \varepsilon_1\ \varepsilon_2 \ := \\ \quad \textbf{var } u \sim \text{random } \textbf{in} \\ \qquad \text{return } ((\varepsilon_2 - \varepsilon_1) * u + \varepsilon_1)$$

Type system: the obvious strategy

$\textbf{var}\ x \sim \text{random}\ \textbf{in}\ \text{return}\ (2 * x)$

Type system: the obvious strategy

$\textbf{var}\ x \sim \text{random}\ \textbf{in}\ \text{return}\ (2 * x)$

"Has a PDF?"

Type system: the obvious strategy

$\textbf{var}\ x \sim \text{random}\ \textbf{in}\ \text{return}\ (2 * x)$

- random

- $\lambda x\,.\,\text{return}\ (2 * x)$

"Has a PDF?"

Type system: the obvious strategy

var $x \sim$ random **in** return $(2 * x)$

- random
  "Well formed distribution?"
- $\lambda x\,.\,$return $(2 * x)$
  "Has a PDF, for all x?"

"Has a PDF?"

Type system: the obvious strategy

$\textbf{var}\ x \sim \text{random}\ \textbf{in}\ \text{return}\ (2 * x)$

- random
  "Well formed distribution?" **YES**
- $\lambda x\,.\ \text{return}\ (2 * x)$
  "Has a PDF, for all x?"

"Has a PDF?"

Type system: the obvious strategy

var $x \sim$ random **in** return $(2 * x)$

- random
  "Well formed distribution?" **YES**
- λx . return $(2 * x)$
  "Has a PDF, for all x?" **NO**

"Has a PDF?"

Type system: the obvious strategy

var $x \sim$ random **in** return $(2 * x)$

- random
  "Well formed distribution?" **YES**
- λx . return $(2 * x)$
  "Has a PDF, for all x?" **NO**

"Has a PDF?" **NO** too conservative

Type system: refined strategy

var $x \sim$ random **in** return $(2 * x)$

- random
  "Well formed distribution?" **YES**
- λx . return $(2 * x)$

"Has a PDF?"

Type system: refined strategy

$\mathbf{var}\ x \sim \text{random}\ \mathbf{in}\ \text{return}\ (2 * x)$

- random

  "Well formed distribution?" **YES**

- $\lambda x\,.\,\text{return}\ (2 * x)$

  "$\forall N, [\![\text{random}]\!](\{x \mid [\![\text{return}(2{*}x)]\!](N) \neq 0\}) = 0$?"

"Has a PDF?"

Type system: refined strategy

var $x \sim$ random **in** return $(2 * x)$

- random
  "Well formed distribution?" **YES**
- $\lambda x\,.\,$return $(2 * x)$
  "$\forall N, [\![\text{random}]\!](\{x \mid [\![\text{return}(2 * x)]\!](N) \neq 0\}) = 0$?" **YES**

"Has a PDF?"

Type system: refined strategy

var $x \sim$ random **in** return $(2 * x)$

- random
  "Well formed distribution?" **YES**
- λx . return $(2 * x)$
  "$\forall N, [\![\text{random}]\!](\{x \mid [\![\text{return}(2*x)]\!](N) \neq 0\}) = 0$?" **YES**

"Has a PDF?" **YES** too hard to *mechanize*

Type system: our strategy

All non-trivial distributions:

$$e := \begin{array}{l} \textbf{var } x_1 \sim e_1 \textbf{ in} \\ \quad \ddots \\ \qquad \textbf{var } x_n \sim e_n \textbf{ in} \\ \qquad\quad \text{return } \varepsilon \end{array}$$

Type system: our strategy

All non-trivial distributions:

$$e := \begin{array}{l} \textbf{var } x_1 \sim e_1 \textbf{ in} \\ \quad \ddots \\ \qquad \textbf{var } x_n \sim e_n \textbf{ in} \\ \qquad\quad \text{return } \varepsilon \end{array}$$

Inspect

- the *joint distribution* of $e_1, \ldots, e_n$
- the *RV transform*, $\lambda(x_1, \ldots, x_n) \,.\, \varepsilon$

Type system: our strategy

$\textbf{var}\ x \sim \text{random}\ \textbf{in}\ \text{return}\ (2 * x)$

- random
- $\lambda x\,.\,2 * x$

"Has a PDF?"

Type system: our strategy

var $x \sim$ random **in** return $(2 * x)$

- random

  "Has a PDF?"

- $\lambda x \,.\, 2 * x$

  "*Non-nullifying?*"

"Has a PDF?"

Type system: our strategy

var $x \sim$ random **in** return $(2 * x)$

- random

  "Has a PDF?" **YES**

- $\lambda x\,.\,2 * x$

  "*Non-nullifying?*"

"Has a PDF?"

Type system: our strategy

var $x \sim$ random **in** return $(2 * x)$

- random
  "Has a PDF?" **YES**
- $\lambda x \,.\, 2 * x$
  "*Non-nullifying*?" **YES**

"Has a PDF?"

Type system: our strategy

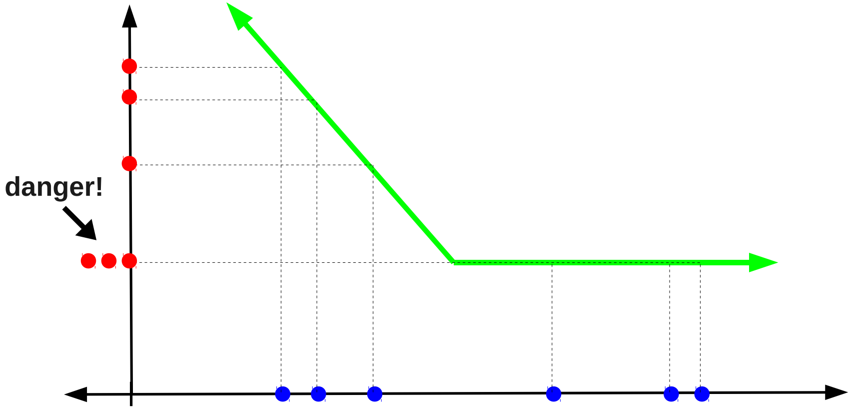
var $x \sim$ random **in** return $(2 * x)$

- random
  "Has a PDF?" **YES**
- $\lambda x \,.\, 2 * x$
  "*Non-nullifying*?" **YES**

"Has a PDF?" **YES**

Intuition for *nullifying* function

$$e' := \textbf{var } x \sim e \textbf{ in } \text{return } (h\ x)$$



Contributions

First formal syntactic work on PDFs.

- type system ("$\mathbb{P}$ has a PDF")
- compiler for calculating PDFs

Compiling PDFs to a usable form

- usable: $\lambda\mathbf{x}\,.\,\mathbf{x} + \mathbf{5}$, $\int_{\mathbf{0}}^{\mathbf{5}} \mathbf{x}^2\, \mathbf{dx}$
- not usable: $\int \mathbf{g}\, \mathbf{d}\mathbb{P}$, $\mathbf{d}\mathbb{P}/\mathbf{d}\mathfrak{L}$

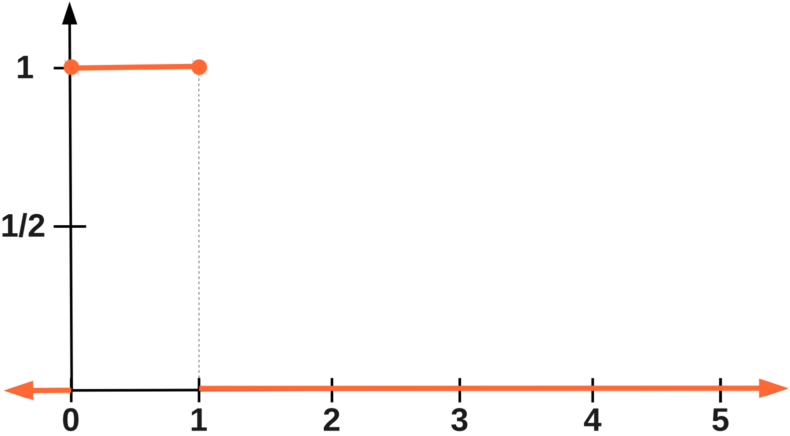
Compiling PDFs to a usable form

- usable: $\lambda \mathbf{x} \,.\, \mathbf{x} + \mathbf{5}$, $\int_{\mathbf{0}}^{\mathbf{5}} \mathbf{x}^2 \, \mathbf{dx}$
- not usable: $\int \mathbf{g} \, \mathbf{d}\mathbb{P}$, $\mathbf{d}\mathbb{P} / \mathbf{d}\mathfrak{L}$

$$\delta ::= \varepsilon \mid \lambda x : \tau \,.\, \delta \mid \delta_1 \; \delta_2 \mid \int \delta$$

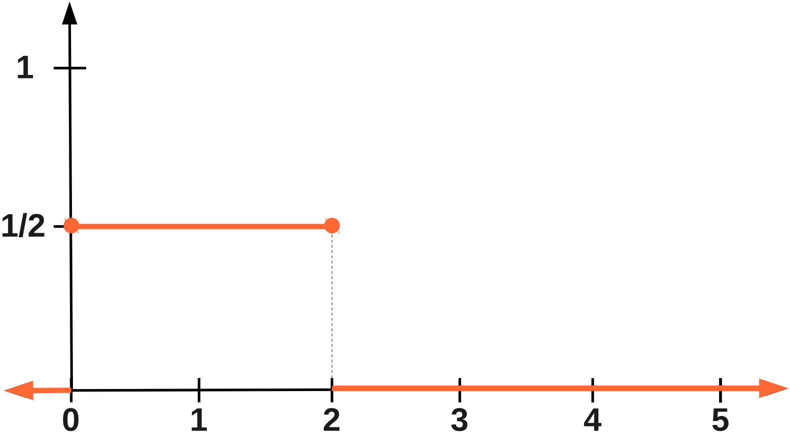
pdf (uniform 3 5)

var $u \sim$ random **in** return $(2 * u + 3)$



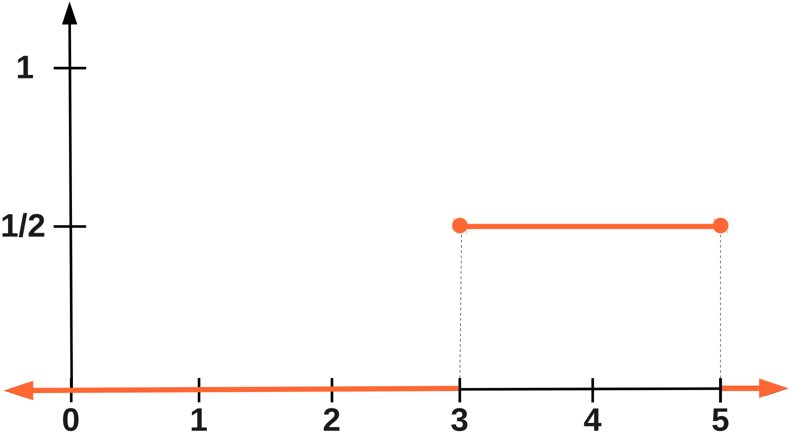
pdf (uniform 3 5)

var $u \sim$ random **in** return $(2 * u + 3)$



pdf (uniform 3 5)

var $u \sim$ random **in** return $(2 * u + 3)$



What we get

in:

$$\textbf{var } z \sim \text{flip } (1/2) \textbf{ in}$$
$$\textbf{var } x_1 \sim \text{uniform } 1\ 2 \textbf{ in}$$
$$\textbf{var } x_2 \sim \text{uniform } 3\ 5 \textbf{ in}$$
$$\text{return } (\textbf{if } z \textbf{ then } x_1 \textbf{ else } x_2)$$

out:

$$\mathbf{f(x) = (1/2) * \langle 1 \leq x \leq 2 \rangle + (1/2) * \langle 3 \leq x \leq 5 \rangle * (1/2)}$$

What we get

in:

$$\begin{aligned}&\textbf{var}\ z \sim \text{flip}\ (1/2)\ \textbf{in}\\ &\ \textbf{var}\ x_1 \sim \text{uniform}\ 1\ 2\ \textbf{in}\\ &\ \ \textbf{var}\ x_2 \sim \text{uniform}\ 3\ 5\ \textbf{in}\\ &\qquad \text{return}\ (\textbf{if}\ z\ \textbf{then}\ x_1\ \textbf{else}\ x_2)\end{aligned}$$

out:

$$\begin{aligned}\mathbf{f(x)} &= \mathbf{(1/2)} * \langle \mathbf{1} \leq \mathbf{x} \leq \mathbf{2} \rangle\\ &+ \mathbf{(1/2)} * \langle \mathbf{3} \leq \mathbf{x} \leq \mathbf{5} \rangle * \mathbf{(1/2)}\end{aligned}$$

More in the paper

- full measure-theoretic details
- PDFs in spaces besides $\mathbb{R}$
- multivariate distributions
- more examples

Future work

Conditional probability.

[Borgstram et al. ESOP'11]

Implementation in Coq.