A Type Theory for Probability Density Functions

Sooraj Bhat    Ashish Agarwal★
Richard Vuduc   Alexander Gray

Georgia Institute of Technology
★New York University

An example probabilistic program

\begin{verbatim}
var z \sim \text{flip} (1/2) in
var x_1 \sim \text{uniform} 1 2 in
var x_2 \sim \text{uniform} 3 5 in
return (if z then x_1 else x_2)
\end{verbatim}
Using a probabilistic program

example tasks

- generate samples
- compute probabilities
- find the density function (PDF)
\[ P(A) = \frac{3}{4} \]
\( \mathbb{P}(A) = 3/4 \)

how can we compute this?
\[ P(A) = \int_A f(x) \, dx \]
This is the *probability density function* (PDF).

\[ P(A) = \int_A f(x) \, dx \]
What we want

\[
\begin{align*}
\text{in:} & \quad \text{var } z & \sim & \text{flip } (1/2) \text{ in} \\
& \text{var } x_1 & \sim & \text{uniform } 1 \ 2 \ \text{in} \\
& \text{var } x_2 & \sim & \text{uniform } 3 \ 5 \ \text{in} \\
& \text{return } (\text{if } z \text{ then } x_1 \text{ else } x_2) \\
\text{out:} & \quad f(x) = \langle 1 \leq x \leq 2 \rangle \ast (1/2) \\
& & + & \langle 3 \leq x \leq 5 \rangle \ast (1/4)
\end{align*}
\]
A PDF may not exist!

\[
\begin{align*}
\text{var } z &\sim \text{ flip } (1/2) \text{ in } \\
\text{var } x_1 &\sim \text{ uniform } 1 2 \text{ in } \\
\text{var } x_2 &\sim \text{ return } 4 \text{ in } \\
\text{return } (\text{if } z \text{ then } x_1 \text{ else } x_2)
\end{align*}
\]
A PDF may not exist!
Contributions

First formal syntactic work on PDFs.

- type system ("P has a PDF")
- compiler for calculating PDFs
Abstract syntax

base types \( \tau ::= \text{bool} \mid \mathbb{Z} \mid \mathbb{R} \mid \tau_1 \times \tau_2 \)

expressions \( \varepsilon ::= x \mid l \mid \text{op} \ \varepsilon_1 \ldots \varepsilon_n \)
\( \mid \text{if } \varepsilon_1 \text{ then } \varepsilon_2 \text{ else } \varepsilon_3 \)
Abstract syntax

base types \[ \tau ::= \text{bool} \mid Z \mid R \mid \tau_1 \times \tau_2 \]

expressions \[ \varepsilon ::= x \mid l \mid \text{op} \; \varepsilon_1 \ldots \varepsilon_n \]
\[ \mid \text{if} \; \varepsilon_1 \; \text{then} \; \varepsilon_2 \; \text{else} \; \varepsilon_3 \]

types \[ t ::= \tau \mid \text{dist} \; \tau \]

distributions \[ e ::= \text{random} \]
\[ \mid \text{return} \; \varepsilon \]
\[ \mid \text{var} \; x \sim e_1 \; \text{in} \; e_2 \]
Abstract syntax

Base types  \( \tau ::= \text{bool} \mid \mathbb{Z} \mid \mathbb{R} \mid \tau_1 \times \tau_2 \)

Expressions  \( \varepsilon ::= x \mid l \mid \text{op} \ \varepsilon_1 \ldots \varepsilon_n \)
\( \quad | \text{if } \varepsilon_1 \text{ then } \varepsilon_2 \text{ else } \varepsilon_3 \)

Types  \( t ::= \tau \mid \text{dist } \tau \)

Distributions  \( e ::= \text{random} \)
\( \quad | \text{return } \varepsilon \)
\( \quad | \text{var } x \sim e_1 \text{ in } e_2 \)

Programs  \( p ::= \text{pdf } e \)
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

\[ \text{flip } \varepsilon := \]
\[ \text{var } u \sim \text{random in } \]
\[ \text{return } (u \leq \varepsilon) \]

\[ \text{uniform } \varepsilon_1 \varepsilon_2 := \]
\[ \text{var } u \sim \text{random in } \]
\[ \text{return } ((\varepsilon_2 - \varepsilon_1) * u + \varepsilon_1) \]
Type system: the obvious strategy

\texttt{var } x \sim \text{random } \textbf{in} \text{ return } (2 \times x)
Type system: the obvious strategy

\text{var } x \sim \text{random in } \text{return } (2 \times x)

“Has a PDF?”
Type system: the obvious strategy

\texttt{var} \ x \ \sim \ \text{random} \ \textbf{in} \ \text{return} \ (2 \ast x)

- \ \text{random}

- \ \lambda x . \ \text{return} \ (2 \ast x)

“Has a PDF?”
Type system: the obvious strategy

```plaintext
var x \sim\text{random} \text{ in return } (2 \ast x)
```

- random
  - “Well formed distribution?”
- \(\lambda x \cdot \text{return } (2 \ast x)\)
  - “Has a PDF, for all \(x\)?”

“Has a PDF?”
Type system: the obvious strategy

\texttt{var } x \sim \text{ random } \texttt{ in } \text{ return } (2 \ast x)

\begin{itemize}
  \item random
    \begin{itemize}
      \item \texttt{``Well formed distribution?'' } YES
    \end{itemize}
  \item \texttt{\lambda x \ return} (2 \ast x)
    \begin{itemize}
      \item \texttt{``Has a PDF, for all } x\texttt{?''}
    \end{itemize}
\end{itemize}

\texttt{``Has a PDF?''}
Type system: the obvious strategy

var x ∼ 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

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

- \texttt{random}
  - “Well formed distribution?” \textbf{YES}

- \texttt{\lambda x . return (2 * x)}
  - “Has a PDF, for all \( x \)” \textbf{NO}

“Has a PDF?” \textbf{NO} \hspace{1cm} too conservative
\textbf{Type system: refined strategy}

\texttt{var} \( x \sim \text{random} \ \textbf{in} \ \text{return} \ (2 \times x) \)

- \texttt{random}
  
  \textbf{“Well formed distribution?”} \ YES

- \( \lambda x \ . \ \text{return} \ (2 \times x) \)

\textbf{“Has a PDF?”}
**Type system: refined strategy**

\texttt{var} \( x \sim \text{random} \textbf{ in } \text{return} \ (2 \times x) \)

- \text{random}
  
  "Well formed distribution?" \ YES

- \( \lambda x \cdot \text{return} \ (2 \times x) \)
  
  "\( \forall N, [\text{random}](\{ x \mid [\text{return}(2x)](N) \neq 0 \}) = 0 ? \)"

"Has a PDF?"
Type system: refined strategy

\begin{verbatim}
var x \sim \text{random} \text{ in return } (2 \times x)
\end{verbatim}

- random
  “Well formed distribution?” \hspace{1cm} YES

- $\lambda x \cdot \text{return } (2 \times x)$
  “\forall N, [\text{random}](\{x \mid [\text{return}(2 \times x)](N) \neq 0\}) = 0?” \hspace{1cm} YES

“Has a PDF?”
\textbf{Type system: refined strategy}

\texttt{var } \texttt{x } \sim \texttt{random in return (2 } \ast \texttt{x)}

\begin{itemize}
  \item \texttt{random} \texttt{``Well formed distribution?'' YES}
  \item \texttt{\lambda x \ . \ return (2 } \ast \texttt{x)} \texttt{``\forall N, [random]({\{x \mid [return(2* x)](N) \neq 0\}}) = 0?'' YES}
\end{itemize}

``Has a PDF?'' YES too hard to mechanize
Type system: our strategy

All non-trivial distributions:

\[
\begin{align*}
\text{var } x_1 & \sim e_1 \text{ in } \\
\vdots \\
\text{e := } & \\
\text{var } x_n & \sim e_n \text{ in } \\
\text{return } \varepsilon
\end{align*}
\]
Type system: our strategy

All non-trivial distributions:

\[ \text{var } x_1 \sim e_1 \text{ in } \]
\[ \ldots \]
\[ e := \text{var } x_n \sim e_n \text{ in } \]
\[ \text{return } \varepsilon \]

Inspect

- the \textit{joint distribution} of \( e_1, \ldots, e_n \)
- the \textit{RV transform}, \( \lambda(x_1, \ldots, x_n) \cdot \varepsilon \)
Type system: our strategy

\texttt{var} \ x \ \sim \ \text{random} \ \textbf{in} \ \text{return} \ (2 \times x)

- random

- $\lambda x \cdot 2 \times x$

"Has a PDF?"
Type system: our strategy

\[ \text{var } x \sim \text{random in return } (2 \times x) \]

- random
  - “Has a PDF?”
- \( \lambda x \cdot 2 \times x \)
  - “Non-nullifying?”

“Has a PDF?”
Type system: our strategy

\texttt{var x \sim random in return (2 \ast x)}

- \texttt{random}
  - “Has a PDF?” YES
- \texttt{\lambda x \cdot 2 \ast x}
  - “Non-nullifying?”

“Has a PDF?”
Type system: our strategy

\texttt{var x \sim \text{random in return } (2 \times x)}

- \text{random}
  - "Has a PDF?" YES
- \(\lambda x \cdot 2 \times x\)
  - "Non-nullifying?" YES

"Has a PDF?"
Type system: our strategy

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

- \ \text{random}
  - \ "Has a PDF?" \ YES
- \ \textbf{\lambda}x. \ 2 * x
  - \ "Non-nullifying?" \ YES

"Has a PDF?" \ YES
Intuition for *nullifying* function

\[ e' := \texttt{var } x \sim e \texttt{ in return } (h \ x) \]
Contributions

- First formal syntactic work on PDFs.
  - type system ("\(P\) has a PDF")
  - compiler for calculating PDFs
Compiling PDFs to a usable form

usable: \( \lambda x . x + 5, \int_0^5 x^2 \, dx \)

not usable: \( \int g \, dP, \frac{dP}{d\mathcal{L}} \)
Compiling PDFs to a usable form

usable: $\lambda x \cdot x + 5$, $\int_0^5 x^2 \, dx$

not usable: $\int g \, dP$, $dP/d\mathcal{L}$

$$\delta ::= \varepsilon \mid \lambda x : \tau \cdot \delta \mid \delta_1 \, \delta_2 \mid \int \delta$$
var \( u \sim \) random in return \((2 \ast u + 3)\)
\texttt{var } u \sim \text{ random in return } (2 \ast u + 3)
\texttt{var } u \sim \text{ random in return } (2 \times u + 3)
What we get

\begin{align*}
\text{in:} & \quad \text{var } z \sim \text{flip } (1/2) \text{ in } \\
& \quad \text{var } x_1 \sim \text{uniform } 1 2 \text{ in } \\
& \quad \text{var } x_2 \sim \text{uniform } 3 5 \text{ in } \\
& \quad \text{return } (\text{if } z \text{ then } x_1 \text{ else } x_2) \\
\text{out:} & \quad f(x) = (1/2) \ast \langle 1 \leq x \leq 2 \rangle \\
& \quad + (1/2) \ast \langle 3 \leq x \leq 5 \rangle \ast (1/2)
\end{align*}
What we get

\[\begin{align*}
\text{in:} & \quad \text{var } z & \sim & \text{flip } (1/2) \text{ in} \\
& \quad \text{var } x_1 & \sim & \text{uniform } 1 \ 2 \text{ in} \\
& \quad \text{var } x_2 & \sim & \text{uniform } 3 \ 5 \text{ in} \\
& \quad \text{return } (\text{if } z \text{ then } x_1 \text{ else } x_2) \\
\text{out:} & \quad f(x) &= (1/2) \ast \langle 1 \leq x \leq 2 \rangle \\
& & + & (1/2) \ast \langle 3 \leq x \leq 5 \rangle \ast (1/2)
\end{align*}\]
More in the paper

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

Conditional probability.

Implementation in Coq.

[Borgstram et al. ESOP’11]