Semantics of probabilistic programming

Fredrik Dahlqvist

Work in progress with Dexter Kozen and help from Vincent Danos, Ilias Garnier and Alexandra Silva

London, 15 November 2018

Warning

What is probabilistic programming?

```
(defquery example
  (let [x (sample (normal 0 1))]
    (observe (normal x 1) 0.5)
    (> x 1)))
```

Denotational and Operational semantics

Operational semantics

Denotational semantics

Denotational and Operational semantics

Operational semantics

- Step-by-step execution of the program

Denotational semantics

Denotational and Operational semantics

Operational semantics

- Step-by-step execution of the program
- Sampling *actually* occurs

Denotational semantics

Denotational and Operational semantics

Operational semantics

- Step-by-step execution of the program
- Sampling *actually* occurs
- Statistical properties emerge after many 'runs'

Denotational semantics

Denotational and Operational semantics

Operational semantics

- Step-by-step execution of the program
- Sampling *actually* occurs
- Statistical properties emerge after many ‘runs’

Denotational semantics

- Mathematical meaning of the program

Denotational and Operational semantics

Operational semantics

- Step-by-step execution of the program
- Sampling *actually* occurs
- Statistical properties emerge after many ‘runs’

Denotational semantics

- Mathematical meaning of the program
- Sampling = distribution

Denotational and Operational semantics

Operational semantics

- Step-by-step execution of the program
- Sampling *actually* occurs
- Statistical properties emerge after many 'runs'

Denotational semantics

- Mathematical meaning of the program
- Sampling = distribution
- Statistical properties immediately available

Denotational and Operational semantics

Operational semantics

- Step-by-step execution of the program
- Sampling *actually* occurs
- Statistical properties emerge after many 'runs'

Denotational semantics

- Mathematical meaning of the program
- Sampling = distribution
- Statistical properties immediately available

$\Leftrightarrow$

Probabilistic Adequacy

Denotational Semantics

Denotational semantics [Kozen ’81]: the main ideas

- Probabilistic programs transform probabilities.

Denotational semantics [Kozen '81]: the main ideas

- Probabilistic programs transform probabilities.
- Probabilities are measures. Measures form vector spaces

$$\mathcal{M}(X) = \text{ all `finite' measures}$$

Denotational semantics [Kozen ’81]: the main ideas

- Probabilistic programs transform probabilities.
- Probabilities are measures. Measures form vector spaces

$$\mathcal{M}(X) = \text{ all ‘finite’ measures}$$

- Probabilistic programs transform measures *linearly*

$$\mathcal{M}(X) \xrightarrow{\mathtt{p}} \mathcal{M}(X)$$

Denotational semantics [Kozen '81]: the main ideas

- Probabilistic programs transform probabilities.
- Probabilities are measures. Measures form vector spaces

$$\mathcal{M}(X) = \text{ all 'finite' measures}$$

- Probabilistic programs transform measures *linearly*

$$\mathcal{M}(X) \xrightarrow{\mathtt{p}} \mathcal{M}(X)$$

- Measures are normed (Banach space) $\Rightarrow$ can study convergence.

Denotational semantics [Kozen '81]: the main ideas

- Probabilistic programs transform probabilities.
- Probabilities are measures. Measures form vector spaces

$$\mathcal{M}(X) = \text{ all 'finite' measures}$$

- Probabilistic programs transform measures *linearly*

$$\mathcal{M}(X) \xrightarrow{\mathtt{p}} \mathcal{M}(X)$$

- Measures are normed (Banach space) $\Rightarrow$ can study convergence.
- Measures are (partially) ordered $\Rightarrow$ can study fixpoints (`while` loops)

Denotational semantics [Kozen '81]: the main ideas

- Probabilistic programs transform probabilities.
- Probabilities are measures. Measures form vector spaces

$$\mathcal{M}(X) = \text{ all 'finite' measures}$$

- Probabilistic programs transform measures *linearly*

$$\mathcal{M}(X) \xrightarrow{\mathtt{p}} \mathcal{M}(X)$$

- Measures are normed (Banach space) $\Rightarrow$ can study convergence.
- Measures are (partially) ordered $\Rightarrow$ can study fixpoints (`while` loops)
- Measures belong to a *monoidal closed category* $\Rightarrow$ higher-order.

Assignments

```
{
 x:=0.5
}
```

Assignments

```
{
 x:=0.5
}
```

$$\frac{\vdash 0.5 : \mathtt{real}}{\mathtt{x} : \mathtt{real} \vdash \mathtt{x} \coloneqq 0.5 : \mathtt{real}}$$

Assignments

```
{
 x:=0.5
}
```

$$\frac{\vdash 0.5 : \mathtt{real}}{\mathtt{x} : \mathtt{real} \vdash \mathtt{x} := 0.5 : \mathtt{real}}$$

$$\frac{[\![\ ]\!] = \mathbb{R} \xrightarrow{[\![0.5]\!]=1 \mapsto \delta_{0.5}} [\![\mathtt{real}]\!] = \mathcal{M}\mathbb{R}}{[\![\mathtt{real}]\!] = \mathcal{M}\mathbb{R} \xrightarrow[\mu \mapsto \mu(\mathbb{R})\delta_{0.5}]{} [\![\mathtt{real}]\!] = \mathcal{M}\mathbb{R}}$$

Some initial observations

Some initial observations

- Ground types $\mathtt{T}$ (e.g. `real`) are interpreted by $\mathcal{M}[\![\mathtt{T}]\!]$

Some initial observations

- Ground types `T` (e.g. `real`) are interpreted by $\mathcal{M}[\![\mathtt{T}]\!]$
- Very similar to 'flat domains' in Scott's denotational semantics

Some initial observations

- Ground types $\mathtt{T}$ (e.g. $\mathtt{real}$) are interpreted by $\mathcal{M}[\![\mathtt{T}]\!]$
- Very similar to 'flat domains' in Scott's denotational semantics
- $\mathcal{M}$ turns any partial map into a total linear operator

Some initial observations

- Ground types $\texttt{T}$ (e.g. $\texttt{real}$) are interpreted by $\mathcal{M}[\![\texttt{T}]\!]$
- Very similar to ‘flat domains’ in Scott’s denotational semantics
- $\mathcal{M}$ turns any partial map into a total linear operator
- Again, similar to monotone maps between ‘flat domains’ and

Some initial observations

- Ground types $\mathtt{T}$ (e.g. $\mathtt{real}$) are interpreted by $\mathcal{M}[\![\mathtt{T}]\!]$
- Very similar to 'flat domains' in Scott's denotational semantics
- $\mathcal{M}$ turns any partial map into a total linear operator
- Again, similar to monotone maps between 'flat domains' and
- Rule of thumb: denotational semantics will be one $\mathcal{M}$-level up

Sampling

```
{
 x:=sample(normal(0,1))
}
```

Sampling

```
{
 x:=sample(normal(0,1))
}
```

$$\dfrac{\dfrac{\vdash \mathtt{normal}(0,1) : \mathtt{M}\,\mathtt{real}}{\vdash \mathtt{sample}(\mathtt{normal}(0,1)) : \mathtt{real}}}{\mathtt{x} : \mathtt{real} \vdash \mathtt{x} := \mathtt{sample}(\mathtt{normal}(0,1)) : \mathtt{real}}$$

Sampling

```
{
  x:=sample(normal(0,1))
}
```

$$\dfrac{\dfrac{\vdash \mathtt{normal}(0,1) : \mathtt{M}\,\mathtt{real}}{\vdash \mathtt{sample}(\mathtt{normal}(0,1)) : \mathtt{real}}}{\mathtt{x} : \mathtt{real} \vdash \mathtt{x} := \mathtt{sample}(\mathtt{normal}(0,1)) : \mathtt{real}}$$

$$\begin{tikzcd}
\llbracket\ \rrbracket = \mathbb{R} \arrow[r, "1 \mapsto \delta_{\mathcal{N}(0,1)}"] \arrow[dr, "\llbracket \mathtt{sample}(\mathtt{normal}(0,1)) \rrbracket"'] & \llbracket \mathtt{M}\,\mathtt{real} \rrbracket = \mathcal{M}^2\mathbb{R} \arrow[d, "\mu \mapsto \int_{B^+(\mathcal{M}\mathbb{R})} x \, d\mu(x)"] \\
 & \llbracket \mathtt{real} \rrbracket = \mathcal{M}\mathbb{R}
\end{tikzcd}$$

Some observation about sampling

Some observation about sampling

- The transformation $\mathcal{M}V \to V$ is completely generic: it is given by the *Bochner integral* $\mu \mapsto \int_{B^+(V)} x \, d\mu(x)$

Some observation about sampling

- The transformation $\mathcal{M}V \to V$ is completely generic: it is given by the *Bochner integral* $\mu \mapsto \int_{B^+(V)} x \, d\mu(x)$
- Denotationally $[\![\texttt{sample}(\texttt{normal}(0,1))]\!]$ is proportional to $\mathcal{N}(0,1)$ as expected.

Some observation about sampling

- The transformation $\mathcal{M}V \to V$ is completely generic: it is given by the *Bochner integral* $\mu \mapsto \int_{B^+(V)} x \, d\mu(x)$
- Denotationally $[\![\mathtt{sample}(\mathtt{normal}(0,1))]\!]$ is proportional to $\mathcal{N}(0,1)$ as expected.
- Bochner integrals are an essential part of the mathematical universe allowing higher-order functions.

Higher-order functions

```
{
 fn x. normal(x,y)
}
```

Higher-order functions

```
{
 fn x. normal(x,y)
}
```

$$\frac{\mathtt{x} : \mathtt{real}, \mathtt{y} : \mathtt{real} \vdash \mathtt{normal}(\mathtt{x}, \mathtt{y}) : \mathtt{M}\,\mathtt{real}}{\mathtt{y} : \mathtt{real} \vdash \mathtt{fn}\ \mathtt{x}.\ \mathtt{normal}(\mathtt{x}, \mathtt{y}) : \mathtt{real} \to \mathtt{M}\,\mathtt{real}}$$

Higher-order functions

```
{
 fn x. normal(x,y)
}
```

$$\frac{\mathtt{x} : \mathtt{real}, \mathtt{y} : \mathtt{real} \vdash \mathtt{normal}(\mathtt{x}, \mathtt{y}) : \mathtt{M}\,\mathtt{real}}{\mathtt{y} : \mathtt{real} \vdash \mathtt{fn}\ \mathtt{x}.\ \mathtt{normal}(\mathtt{x}, \mathtt{y}) : \mathtt{real} \to \mathtt{M}\,\mathtt{real}}$$

$$\frac{[\![\mathtt{real}]\!] \otimes [\![\mathtt{real}]\!] = \mathcal{M}\mathbb{R} \otimes \mathcal{M}\mathbb{R} \longrightarrow [\![\mathtt{M}\,\mathtt{real}]\!] = \mathcal{M}^2\mathbb{R}}{\mathcal{M}\mathbb{R} \xrightarrow[{[\![\mathtt{fn}\ \mathtt{x}.\ \mathtt{normal}(\mathtt{x},\mathtt{y})]\!]}]{} \mathcal{L}_r(\mathcal{M}\mathbb{R}, \mathcal{M}^2\mathbb{R})}$$

A computer science perspective on tensor products

A computer science perspective on tensor products

- Given a map f in two arguments U, V into W, we want to *curry*

$$U \to [V, W] \qquad V \to [U, W]$$

A computer science perspective on tensor products

- Given a map f in two arguments U, V into W, we want to *curry*

$$U \to [V, W] \qquad V \to [U, W]$$

- These morphisms must be linear transformations

A computer science perspective on tensor products

- Given a map f in two arguments U, V into W, we want to *curry*

$$U \to [V, W] \qquad V \to [U, W]$$

- These morphisms must be linear transformations
- This means our map f is linear in U and V *separately* (bilinear)

$$f(\lambda u + \lambda' u', v) = \lambda f(u, v) + \lambda' f(u', v)$$

A computer science perspective on tensor products

- Given a map f in two arguments U, V into W, we want to *curry*

$$U \to [V, W] \qquad V \to [U, W]$$

- These morphisms must be linear transformations
- This means our map f is linear in U and V *separately* (bilinear)

$$f(\lambda u + \lambda' u', v) = \lambda f(u, v) + \lambda' f(u', v)$$

- This does not mean that it is *jointly* linear:

$$f(\lambda(u, v)) = f(\lambda u, \lambda v) = \lambda f(u, \lambda v) = \lambda^2 f(u, v)$$

A computer science perspective on tensor products

- Given a map f in two arguments U, V into W, we want to *curry*

$$U \to [V, W] \qquad V \to [U, W]$$

- These morphisms must be linear transformations
- This means our map f is linear in U and V *separately* (bilinear)

$$f(\lambda u + \lambda' u', v) = \lambda f(u, v) + \lambda' f(u', v)$$

- This does not mean that it is *jointly* linear:

$$f(\lambda(u, v)) = f(\lambda u, \lambda v) = \lambda f(u, \lambda v) = \lambda^2 f(u, v)$$

- So $f : U \times V \to W$ is *not* linear!

A computer science perspective on tensor products

- Given a map f in two arguments U, V into W, we want to *curry*

$$U \to [V, W] \qquad V \to [U, W]$$

- These morphisms must be linear transformations
- This means our map f is linear in U and V *separately* (bilinear)

$$f(\lambda u + \lambda' u', v) = \lambda f(u, v) + \lambda' f(u', v)$$

- This does not mean that it is *jointly* linear:

$$f(\lambda(u, v)) = f(\lambda u, \lambda v) = \lambda f(u, \lambda v) = \lambda^2 f(u, v)$$

- So $f : U \times V \to W$ is *not* linear!
- BUT: $\hat{f} : U \otimes V \to W$ is.

Denotational semantics: conclusion

- Typed language accommodating many important classical and probabilistic constructs

Denotational semantics: conclusion

- Typed language accommodating many important classical and probabilistic constructs
- Very powerful semantics in terms of ordered Banach space

Denotational semantics: conclusion

- Typed language accommodating many important classical and probabilistic constructs
- Very powerful semantics in terms of ordered Banach space
- Advanced but completely mainstream mathematics

Denotational semantics: conclusion

- Typed language accommodating many important classical and probabilistic constructs
- Very powerful semantics in terms of ordered Banach space
- Advanced but completely mainstream mathematics
- Many ‘moral’ similarities with Scott’s semantics

Operational Semantics

Operational semantics: discrete case

```
{
 x=sample(bernoulli(0.2))
}
```

Operational semantics: discrete case

```
{
 x=sample(bernoulli(0.2))
}
```

$$
\begin{tikzcd}
 & \Sigma[\mathtt{x} \mapsto 0] \vdash 1 \\
\Sigma \vdash \mathtt{x} := \mathtt{sample}(\mathtt{bernoulli}(0.5)) \arrow[ur, "\Downarrow_{0.2,\mathtt{unit}}"] \arrow[dr, "\Downarrow_{0.8,\mathtt{unit}}"'] & \\
 & \Sigma[\mathtt{x} \mapsto 1] \vdash 1
\end{tikzcd}
$$

Operational semantics: continuous case

```
{
 x=sample(normal(0,1))
}
```

Operational semantics: continuous case

```
{
 x=sample(normal(0,1))
}
```

$$
\begin{tikzcd}
 & \Sigma[\mathtt{x} \mapsto 3.1416] \vdash 1 \\
\Sigma \vdash \mathtt{x} := \mathtt{sample}(\mathtt{normal}(0,1)) \arrow[ur, "\Downarrow_{0,\mathtt{unit}}"] \arrow[dr, "\Downarrow_{0,\mathtt{unit}}"'] & \vdots \\
 & \Sigma[\mathtt{x} \mapsto -1.4142] \vdash 1
\end{tikzcd}
$$

Operational semantics: continuous case

```
{
 x=sample(normal(0,1))
}
```

Operational semantics: continuous case

```
{
 x=sample(normal(0,1))
}
```

$$(\Sigma, \mathtt{seed}) \vdash \mathtt{x} = \mathtt{sample}(\mathtt{normal}(0, 1)) \Downarrow_{\mathtt{unit}}$$
$$(\Sigma[\mathtt{x} \mapsto [\![\mathtt{normal}(0, 1)]\!](\mathtt{seed})], \mathtt{seed} + 1) \vdash 1$$

where

$$[\![\mathtt{normal}(0, 1)]\!] : \mathbb{N} \to \mathbb{R}$$

with certain properties

Probabilistic Adequacy

Probabilistic adequacy: general aim

Probabilistic adequacy: general aim

- Ideally, the step-by-step execution of the model ‘agrees’ with its intended mathematical meaning

Probabilistic adequacy: general aim

- Ideally, the step-by-step execution of the model ‘agrees’ with its intended mathematical meaning
- But what does it mean for a probabilistic program?

Probabilistic adequacy: general aim

- Ideally, the step-by-step execution of the model ‘agrees’ with its intended mathematical meaning
- But what does it mean for a probabilistic program?
- Operationally: program = empirical process.
- Empirical distribution for $A \subseteq [\![\Sigma]\!]$

$$\mathbb{P}_n(A) = \frac{1}{n}\sum_{i=1}^{n} I_A(X_i)$$

Probabilistic adequacy: general aim

- Ideally, the step-by-step execution of the model 'agrees' with its intended mathematical meaning
- But what does it mean for a probabilistic program?
- Operationally: program = empirical process.
- Empirical distribution for $A \subseteq [\![\Sigma]\!]$

$$\mathbb{P}_n(A) = \frac{1}{n} \sum_{i=1}^{n} I_A(X_i)$$

- Probabilistic adequacy:

Does the empirical distribution converge to the denotational semantics? If yes, how fast?

Concentration of measure – an example

- Consider the hamming cube $\{0, 1\}^n$

Concentration of measure – an example

- Consider the hamming cube $\{0, 1\}^n$
- Metric space with $d((x_1, \ldots, x_n), (y_1, \ldots, y_n)) = \frac{1}{n} \sum_i (x_i + y_i \mod 2)$

Concentration of measure – an example

- Consider the hamming cube $\{0, 1\}^n$
- Metric space with $d((x_1, \ldots, x_n), (y_1, \ldots, y_n)) = \frac{1}{n} \sum_i (x_i + y_i \mod 2)$
- Measured space with counting measure $\mu(A) = \frac{\#A}{2^n}$

Concentration of measure – an example

- Consider the hamming cube $\{0, 1\}^n$
- Metric space with $d((x_1, \ldots, x_n), (y_1, \ldots, y_n)) = \frac{1}{n} \sum_i (x_i + y_i \mod 2)$
- Measured space with counting measure $\mu(A) = \frac{\#A}{2^n}$
- Consider the function $f : \{0, 1\}^n \to \mathbb{R}, (x_1, \ldots, x_n) \mapsto \frac{1}{n} \sum_i x_i$

Concentration of measure – an example

- Consider the hamming cube $\{0, 1\}^n$
- Metric space with $d((x_1, \ldots, x_n), (y_1, \ldots, y_n)) = \frac{1}{n}\sum_i (x_i + y_i \mod 2)$
- Measured space with counting measure $\mu(A) = \frac{\#A}{2^n}$
- Consider the function $f : \{0, 1\}^n \to \mathbb{R}$, $(x_1, \ldots, x_n) \mapsto \frac{1}{n}\sum_i x_i$
- The median of f is $\frac{1}{2}$: $\mu\{x \mid f(x) \leqslant \frac{1}{2}\} = \mu\{x \mid f(x) \geqslant \frac{1}{2}\}$

Concentration of measure – an example

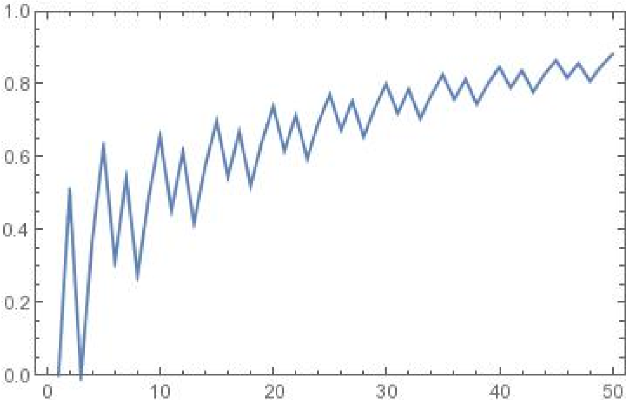
- Consider the hamming cube $\{0, 1\}^n$
- Metric space with $d((x_1, \dots, x_n), (y_1, \dots, y_n)) = \frac{1}{n} \sum_i (x_i + y_i \mod 2)$
- Measured space with counting measure $\mu(A) = \frac{\#A}{2^n}$
- Consider the function $f : \{0, 1\}^n \to \mathbb{R}$, $(x_1, \dots, x_n) \mapsto \frac{1}{n} \sum_i x_i$
- The median of f is $\frac{1}{2}$: $\mu\{x \mid f(x) \leqslant \frac{1}{2}\} = \mu\{x \mid f(x) \geqslant \frac{1}{2}\}$
- How far are we away from the median on average?

$$A_f(\varepsilon, n) := \left\{ (x_1, \dots, x_n) \mid \left| f(x_1, \dots, x_n) - \frac{1}{2} \right| < \varepsilon \right\}$$

$$\mu(A_f(\varepsilon, n)) = \frac{1}{2^n} \sum_{k=\lceil n\varepsilon \rceil}^{\lfloor n\varepsilon \rfloor} \binom{n}{k}$$

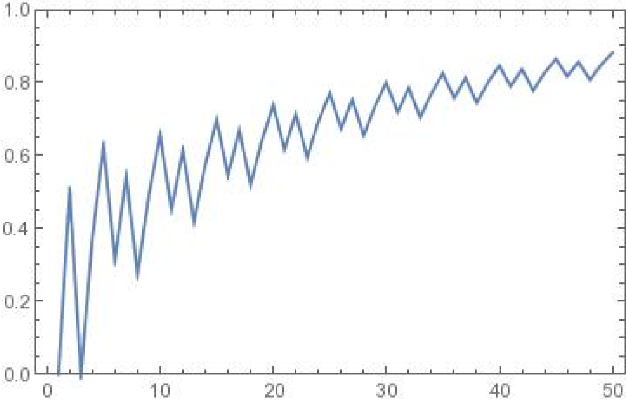
Concentration of measure – an example

- For $\varepsilon = \frac{1}{10}$ this is what $\mu(A_f(\varepsilon), n)$ varies as n increases:



Concentration of measure – an example

- For $\varepsilon = \frac{1}{10}$ this is what $\mu(A_f(\varepsilon), n)$ varies as n increases:



- This is a general pattern in *metric measured spaces*

Probabilistic adequacy and concentration of measure

```
{
 x=sample(bernoulli(0.5))
}
```

Probabilistic adequacy and concentration of measure

```
{
 x=sample(bernoulli(0.5))
}
```

- Denotationally $[\![\mathtt{x} = \mathtt{sample}(\mathtt{bernoulli}(0.5))]\!](\mu) \propto \mathit{Bern}(0.5)$

Probabilistic adequacy and concentration of measure

```
{
 x=sample(bernoulli(0.5))
}
```

- Denotationally $[\![\mathtt{x} = \mathtt{sample}(\mathtt{bernoulli}(0.5))]\!](\mu) \propto \mathit{Bern}(0.5)$
- The function f computes the empirical probability of $\Sigma = [\mathtt{x} \mapsto 1]$

Probabilistic adequacy and concentration of measure

```
{
 x=sample(bernoulli(0.5))
}
```

- Denotationally $[\![\mathtt{x} = \mathtt{sample}(\mathtt{bernoulli}(0.5))]\!](\mu) \propto \mathit{Bern}(0.5)$
- The function f computes the empirical probability of $\Sigma = [\mathtt{x} \mapsto 1]$
- The convergence of $\mu(A_f)(\varepsilon, n)$ given above shows that the empirical probability (multiple runs of the program) converges with the denotational semantics.

Probabilistic adequacy and concentration of measure

```
{
 x=sample(bernoulli(0.5))
}
```

- Denotationally $[\![\texttt{x} = \texttt{sample}(\texttt{bernoulli}(0.5))]\!](\mu) \propto \mathit{Bern}(0.5)$
- The function f computes the empirical probability of $\Sigma = [\texttt{x} \mapsto 1]$
- The convergence of $\mu(A_f)(\varepsilon, n)$ given above shows that the empirical probability (multiple runs of the program) converges with the denotational semantics.
- Moreover, the rate of convergence can also be bounded ($\sqrt{n}$)

Thank you.