Semantic Foundations for Probabilistic Programming

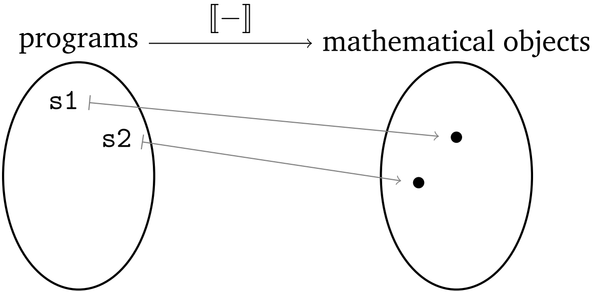
Chris Heunen

THE UNIVERSITY of EDINBURGH informatics

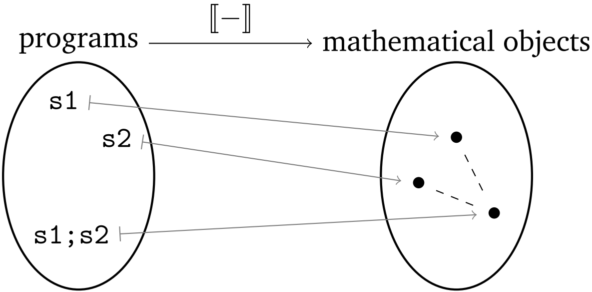
Ohad Kammar, Sam Staton,
Frank Wood, Hongseok Yang

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE

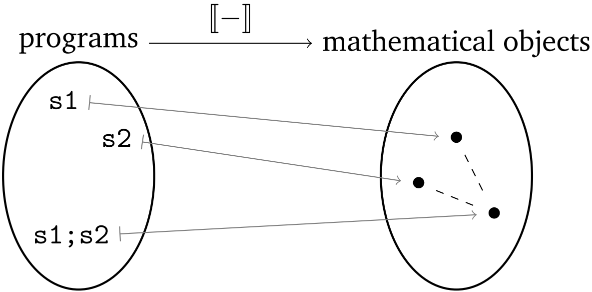
Semantic foundations



Semantic foundations

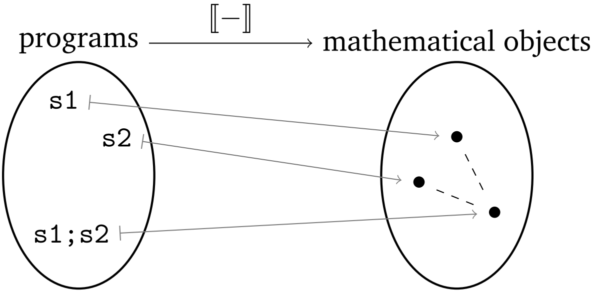


Semantic foundations



- *Operational*: remember implementation details (efficiency)
- *Denotational*: see what program does conceptually (correctness)

Semantic foundations

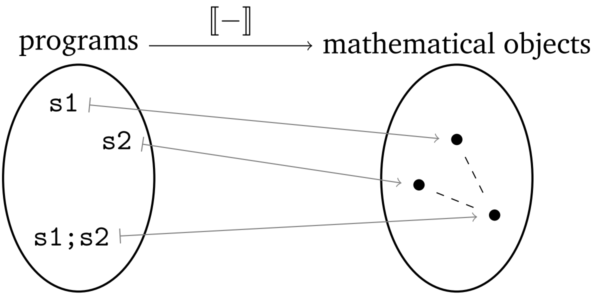


- *Operational*: remember implementation details (efficiency)
- *Denotational*: see what program does conceptually (correctness)

Motivation:

- Ground programmer's unspoken intuitions
- Justify/refute/suggest program transformations
- Understand programming through mathematics

Semantic foundations



- *Operational*: remember implementation details (efficiency)
- *Denotational*: see what program does conceptually (correctness)

Motivation:

- Ground programmer's unspoken intuitions
- Justify/refute/suggest program transformations
- Understand *probability* through *program equations*

Probabilistic programming

$$\mathrm{P}(A \mid B) \;=\; \frac{\mathrm{P}(B \mid A) \times \mathrm{P}(A)}{\mathrm{P}(B)}$$

Probabilistic programming

$$P(A \mid B) \propto P(B \mid A) \times P(A)$$

Probabilistic programming

$$\mathrm{P}(A \mid B) \propto \mathrm{P}(B \mid A) \times \mathrm{P}(A)$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

Probabilistic programming

$$\mathrm{P}(A \mid B) \propto \mathrm{P}(B \mid A) \times \mathrm{P}(A)$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

idealized Anglican = functional programming +

normalize **observe** **sample**

http://www.robots.ox.ac.uk/~fwood/anglican

Overview

- Interpret types as measurable spaces e.g. $[\![\texttt{real}]\!] = \mathbb{R}$
- Interpret (open) terms as kernels
- Interpret closed terms as measures
- Inference normalizes measures posterior $\propto$ likelihood $\times$ prior

[Kozen, "Semantics of probabilistic programs", J Comp Syst Sci, 1981]

Overview

- Interpret types as measurable spaces — e.g. $[\![\texttt{real}]\!] = \mathbb{R}$
- Interpret (open) terms as kernels
- Interpret closed terms as measures
- Inference normalizes measures — posterior $\propto$ likelihood $\times$ prior

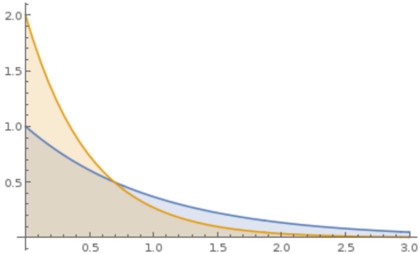
But:

- Commutativity? — Fubini not true for all kernels
- Higher order functions? — $\mathbb{R} \to \mathbb{R}$ not a measurable space
- Extensionality?
- Recursion?

[Kozen, "Semantics of probabilistic programs", J Comp Syst Sci, 1981]
[Aumann, "Borel structures for function spaces", Ill J Math, 1961]

Example

1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay
4. What is the outcome x?



Example

1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay
4. What is the outcome x?

```
let x = sample(bern(0.5)) in
let r = if x then 2.0 else 1.0
observe(0.0 from exp(r));
return x
```

Example

1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay
4. What is the outcome x?

two traces:

	0.5	0.5
`let x = sample(bern(0.5)) in`	`x=true`	`x=false`

```
let x = sample(bern(0.5)) in
let r = if x then 2.0 else 1.0
observe(0.0 from exp(r));
return x
```

Example

1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay
4. What is the outcome x?

two traces:

Code	0.5	0.5
`let x = sample(bern(0.5)) in`	`x=true`	`x=false`
`let r = if x then 2.0 else 1.0`	`r=2.0`	
`observe(0.0 from exp(r));`	`score 2`	
`return x`	`return true`	

Example

1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay
4. What is the outcome x?

two traces:

```
                                           0.5            0.5
let x = sample(bern(0.5)) in               x=true         x=false
let r = if x then 2.0 else 1.0             r=2.0          r=1.0
observe(0.0 from exp(r));                  score 2        score 1
return x                                   return true    return false
```

Example

1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay
4. What is the outcome x?

```
let x = sample(bern(0.5)) in
let r = if x then 2.0 else 1.0
observe(0.0 from exp(r));
return x
```

two traces:

0.5	0.5
x=true	x=false
r=2.0	r=1.0
score 2	score 1
return true	return false

posterior $\propto$ likelihood $\times$ prior

2×0.5: true
1×0.5: false

Example

1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay
4. What is the outcome x?

$\mathrm{P}(\texttt{true}) = 1, \mathrm{P}(\texttt{false}) = 0.5$

```
let x = sample(bern(0.5)) in
let r = if x then 2.0 else 1.0
observe(0.0 from exp(r));
return x
```

two traces:

0.5	0.5
`x=true`	`x=false`
`r=2.0`	`r=1.0`
`score 2`	`score 1`
`return true`	`return false`

posterior $\propto$ likelihood $\times$ prior

2×0.5: true
1×0.5: false

Example

1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay
4. What is the outcome x?

model evidence (score): 1.5

$P(\texttt{true}) = 66\%$, $P(\texttt{false}) = 33\%$

```
let x = sample(bern(0.5)) in
let r = if x then 2.0 else 1.0
observe(0.0 from exp(r));
return x
```

two traces:

0.5	0.5
`x=true`	`x=false`
`r=2.0`	`r=1.0`
`score 2`	`score 1`
`return true`	`return false`

posterior $\propto$ likelihood $\times$ prior

2×0.5: true
1×0.5: false

Example

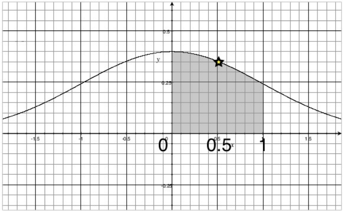
1. Toss a fair coin to get outcome x
2. Set up exponential decay with rate r depending on x
3. Observe immediate decay model evidence (score): 1.5
4. What is the outcome x? $\mathrm{P}(\texttt{true}) = 66\%$, $\mathrm{P}(\texttt{false}) = 33\%$

Programs may also sample continuous distributions so have to deal with uncountable number of traces:

```
let y = sample(gauss(7,2))
```

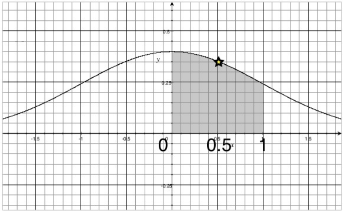
Measure theory

Impossible to sample 0.5 from standard normal distribution
But sample in interval $(0, 1)$ with probability around 0.34



Measure theory

Impossible to sample 0.5 from standard normal distribution
But sample in interval $(0, 1)$ with probability around 0.34



A measurable space is a set X with a family Σ_X of subsets
that is closed under countable unions and complements

A (probability) measure on X is a function $p \colon \Sigma_X \to [0, \infty]$
that satisfies $p(\sum U_n) = \sum p(U_n)$ (and has $p(X) = 1$)

First order language

- Types: $\mathbb{A}, \mathbb{B} ::= \mathbb{R} \mid P(\mathbb{A}) \mid 1 \mid \mathbb{A} \times \mathbb{B} \mid \sum_{i \in I} \mathbb{A}_i$

  real numbers ($\mathbb{R}$); distributions over $\mathbb{A}$ ($P(\mathbb{A})$); finite products ($\mathbb{A} \times \mathbb{B}$); countable sums ($\sum_{i \in I} \mathbb{A}_i$)

  $\texttt{bool} := 1 + 1$

  $\texttt{nat} := \sum_{i \in \mathbb{N}} 1$

First order language

- Types: $\mathbb{A}, \mathbb{B} ::= \mathbb{R} \mid \mathrm{P}(\mathbb{A}) \mid 1 \mid \mathbb{A} \times \mathbb{B} \mid \sum_{i \in I} \mathbb{A}_i$
- Deterministic terms may not sample:
  - variables — $\mathtt{x}, \mathtt{y}, \mathtt{z}$
  - constructors for sums and products — $\mathtt{case}, \mathtt{in}_i, \mathtt{if}, \mathtt{false}, \mathtt{true}$
  - measurable functions — $\mathtt{bern}, \mathtt{exp}, \mathtt{gauss}, \mathtt{dirac}$

$$\vdash_{\mathrm{d}} \mathtt{42.0} : \mathbb{R}$$
$$\vdash_{\mathrm{d}} \mathtt{gauss}(\mathtt{2.0}, \mathtt{7.0}) : \mathrm{P}(\mathbb{R})$$
$$\mathtt{x} : \mathbb{R}, \mathtt{y} : \mathbb{R} \vdash_{\mathrm{d}} \mathtt{x} + \mathtt{y} : \mathbb{R}$$
$$\mathtt{x} : \mathbb{R}, \mathtt{y} : \mathbb{R} \vdash_{\mathrm{d}} x < y : \mathtt{bool}$$

First order language

- Types: $\mathbb{A}, \mathbb{B} ::= \mathbb{R} \mid P(\mathbb{A}) \mid 1 \mid \mathbb{A} \times \mathbb{B} \mid \sum_{i \in I} \mathbb{A}_i$
- Deterministic terms may not sample:
  - variables — `x, y, z`
  - constructors for sums and products — `case`, $\texttt{in}_i$, `if`, `false`, `true`
  - measurable functions — `bern`, `exp`, `gauss`, `dirac`
- Probabilistic terms may sample:
  - sequencing — `return`, `let`
  - constraints — `score`
  - priors — `sample`

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{A}}{\Gamma \vdash_{\mathrm{p}} \texttt{return}(t) \colon \mathbb{A}} \qquad \frac{\Gamma \vdash_{\mathrm{p}} t \colon \mathbb{A} \quad x \colon \mathbb{A} \vdash_{\mathrm{p}} u \colon \mathbb{B}}{\Gamma \vdash_{\mathrm{p}} \texttt{let } x = t \texttt{ in } u \colon \mathbb{B}}$$

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{R}}{\Gamma \vdash_{\mathrm{p}} \texttt{score}(t) \colon 1} \qquad \frac{\Gamma \vdash_{\mathrm{d}} t \colon P(\mathbb{A})}{\Gamma \vdash_{\mathrm{p}} \texttt{sample}(t) \colon \mathbb{A}}$$

First order language

- Types: $\mathbb{A}, \mathbb{B} ::= \mathbb{R} \mid \mathrm{P}(\mathbb{A}) \mid 1 \mid \mathbb{A} \times \mathbb{B} \mid \sum_{i \in I} \mathbb{A}_i$
- Deterministic terms may not sample:
  - variables — `x, y, z`
  - constructors for sums and products — `case`, $\mathtt{in}_i$, `if`, `false`, `true`
  - measurable functions — `bern`, `exp`, `gauss`, `dirac`
  - inference — `norm`
- Probabilistic terms may sample:
  - sequencing — `return`, `let`
  - constraints — `score`
  - priors — `sample`

$$\frac{\Gamma \vdash_{\mathrm{d}} t : \mathbb{A}}{\Gamma \vdash_{\mathrm{p}} \mathtt{return}(t) : \mathbb{A}} \qquad \frac{\Gamma \vdash_{\mathrm{p}} t : \mathbb{A} \quad x : \mathbb{A} \vdash_{\mathrm{p}} u : \mathbb{B}}{\Gamma \vdash_{\mathrm{p}} \mathtt{let}\ x = t\ \mathtt{in}\ u : \mathbb{B}}$$

$$\frac{\Gamma \vdash_{\mathrm{d}} t : \mathbb{R}}{\Gamma \vdash_{\mathrm{p}} \mathtt{score}(t) : 1} \qquad \frac{\Gamma \vdash_{\mathrm{d}} t : \mathrm{P}(\mathbb{A})}{\Gamma \vdash_{\mathrm{p}} \mathtt{sample}(t) : \mathbb{A}}$$

First order semantics

Interpret

- type $\mathbb{A}$ as measurable space $[\![\mathbb{A}]\!]$
- deterministic term $\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{A}$ as measurable function $[\![\Gamma]\!] \to [\![\mathbb{A}]\!]$
- probabilistic term $\Gamma \vdash_{\mathrm{p}} t \colon \mathbb{A}$ as kernel $[\![t]\!] \colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$
  fixing first argument: measure, fixing second argument: measurable

First order semantics

Interpret

- type $\mathbb{A}$ as measurable space $[\![\mathbb{A}]\!]$
- deterministic term $\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{A}$ as measurable function $[\![\Gamma]\!] \to [\![\mathbb{A}]\!]$
- probabilistic term $\Gamma \vDash_{\mathrm{p}} t \colon \mathbb{A}$ as kernel $[\![t]\!] \colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$
  fixing first argument: measure, fixing second argument: measurable

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{R}}{\Gamma \vDash_{\mathrm{p}} \mathtt{score}(t) \colon 1} \qquad [\![\mathtt{score}(t)]\!](\gamma, *) = [\![t]\!](\gamma)$$

First order semantics

Interpret

- type $\mathbb{A}$ as measurable space $[\![\mathbb{A}]\!]$
- deterministic term $\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{A}$ as measurable function $[\![\Gamma]\!] \to [\![\mathbb{A}]\!]$
- probabilistic term $\Gamma \vDash_{\mathrm{p}} t \colon \mathbb{A}$ as kernel $[\![t]\!] \colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$
  fixing first argument: measure, fixing second argument: measurable

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{R}}{\Gamma \vDash_{\mathrm{p}} \mathtt{score}(t) \colon 1} \qquad [\![\mathtt{score}(t)]\!](\gamma, *) = [\![t]\!](\gamma)$$

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathrm{P}(\mathbb{A})}{\Gamma \vDash_{\mathrm{p}} \mathtt{sample}(t) \colon \mathbb{A}} \qquad [\![\mathtt{sample}(t)]\!](\gamma, U) = ([\![t]\!](\gamma))(U)$$

First order semantics

Interpret

- type $\mathbb{A}$ as measurable space $[\![\mathbb{A}]\!]$
- deterministic term $\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{A}$ as measurable function $[\![\Gamma]\!] \to [\![\mathbb{A}]\!]$
- probabilistic term $\Gamma \vDash_{\mathrm{p}} t \colon \mathbb{A}$ as kernel $[\![t]\!] \colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$
  fixing first argument: measure, fixing second argument: measurable

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{R}}{\Gamma \vDash_{\mathrm{p}} \mathtt{score}(t) \colon 1} \qquad [\![\mathtt{score}(t)]\!](\gamma, *) = [\![t]\!](\gamma)$$

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathrm{P}(\mathbb{A})}{\Gamma \vDash_{\mathrm{p}} \mathtt{sample}(t) \colon \mathbb{A}} \qquad [\![\mathtt{sample}(t)]\!](\gamma, U) = ([\![t]\!](\gamma))(U)$$

$$\frac{\Gamma \vDash_{\mathrm{p}} t \colon \mathbb{A} \quad x \colon \mathbb{A} \vDash_{\mathrm{p}} u \colon \mathbb{B}}{\Gamma \vDash_{\mathrm{p}} \mathtt{let}\ x = t\ \mathtt{in}\ u \colon \mathbb{B}} \qquad [\![\mathtt{let}\ x = t\ \mathtt{in}\ u]\!](\gamma, U) = \int_{[\![\mathbb{A}]\!]} [\![u]\!](\gamma, x, U)\, [\![t]\!](\gamma, \mathrm{d}x)$$

First order semantics

Interpret

- type $\mathbb{A}$ as measurable space $[\![\mathbb{A}]\!]$
- deterministic term $\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{A}$ as measurable function $[\![\Gamma]\!] \to [\![\mathbb{A}]\!]$
- probabilistic term $\Gamma \vDash_{\mathrm{p}} t \colon \mathbb{A}$ as kernel $[\![t]\!] \colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$
  fixing first argument: measure, fixing second argument: measurable

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathbb{R}}{\Gamma \vDash_{\mathrm{p}} \mathtt{score}(t) \colon 1} \qquad [\![\mathtt{score}(t)]\!](\gamma, *) = [\![t]\!](\gamma)$$

$$\frac{\Gamma \vdash_{\mathrm{d}} t \colon \mathrm{P}(\mathbb{A})}{\Gamma \vDash_{\mathrm{p}} \mathtt{sample}(t) \colon \mathbb{A}} \qquad [\![\mathtt{sample}(t)]\!](\gamma, U) = ([\![t]\!](\gamma))(U)$$

$$\frac{\Gamma \vDash_{\mathrm{p}} t \colon \mathbb{A} \quad x \colon \mathbb{A} \vDash_{\mathrm{p}} u \colon \mathbb{B}}{\Gamma \vDash_{\mathrm{p}} \mathtt{let}\ x = t\ \mathtt{in}\ u \colon \mathbb{B}} \qquad [\![\mathtt{let}\ x = t\ \mathtt{in}\ u]\!] = \int_{[\![\mathbb{A}]\!]} [\![u]\!] \mathrm{d} [\![t]\!]$$

Example

$$\left[\!\!\left[\begin{array}{l}\texttt{let x = sample(bern(0.5)) in}\\ \texttt{let r = if x then 2.0 else 1.0}\\ \texttt{observe(0.0 from exp(r));}\\ \texttt{return x}\end{array}\right]\!\!\right]$$

The meaning of a program returning values in X is a measure on X

$$\begin{array}{rll} \emptyset & \text{has measure} & 0.0 \\ \{\texttt{true}\} & \text{has measure} & 1.0 = 0.5 \times 2.0 \\ \{\texttt{false}\} & \text{has measure} & 0.5 = 0.5 \times 1.0 \\ \{\texttt{true}, \texttt{false}\} & \text{has measure} & 1.5 \end{array}$$

Normalization: **posterior** $\propto$ **likelihood** $\times$ **prior**

$$\frac{\Gamma \vdash_{\mathrm{p}} t : \mathbb{A}}{\Gamma \vdash_{\mathrm{d}} \texttt{norm}(t) : \mathbb{R} \times \mathrm{P}(\mathbb{A}) \; + \underbrace{1 + 1}}$$

model evidence

errors

normalized posterior

Normalization: **posterior** $\propto$ **likelihood** $\times$ **prior**

$$\frac{\Gamma \vdash_{\mathrm{p}} t \colon \mathbb{A}}{\Gamma \vdash_{\mathrm{d}} \texttt{norm}(t) \colon \mathbb{R} \times \mathrm{P}(\mathbb{A}) \; + \underbrace{1 + 1}}$$

model evidence

errors

normalized posterior

Interpretation of probabilistic term is kernel $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ so

fixing first argument gives measure

$$\frac{[\![t]\!](\gamma, -)}{[\![t]\!](\gamma, [\![\mathbb{A}]\!])}$$

is normalized probability measure

Normalization: **posterior** $\propto$ **likelihood** $\times$ **prior**

$$\frac{\Gamma \vdash_{\mathrm{p}} t \colon \mathbb{A}}{\Gamma \vdash_{\mathrm{d}} \mathtt{norm}(t) \colon \mathbb{R} \times \mathrm{P}(\mathbb{A}) \; + \underbrace{1 + 1}}$$

model evidence

errors
(constant is 0 or ∞)

normalized posterior

Interpretation of probabilistic term is kernel $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ so

fixing first argument gives measure

$$\frac{[\![t]\!](\gamma, -)}{[\![t]\!](\gamma, [\![\mathbb{A}]\!])}$$

is normalized probability measure

normalizing constant is model evidence

Normalization: **posterior** $\propto$ **likelihood** $\times$ **prior**

$$\frac{\Gamma \vdash_{\mathrm{p}} t : \mathbb{A}}{\Gamma \vdash_{\mathrm{d}} \texttt{norm}(t) : \mathbb{R} \times \mathrm{P}(\mathbb{A}) + \underbrace{1 + 1}}$$

model evidence

errors
(constant is 0 or ∞)

normalized posterior

$$\left[\!\!\left[\begin{array}{l} \texttt{let x = sample(bern(0.5)) in} \\ \texttt{let r = if x then 2.0 else 1.0} \\ \texttt{observe(0.0 from exp(r));} \\ \texttt{return x} \end{array} \right]\!\!\right] = \begin{pmatrix} \texttt{true}: & 2.0 \times 0.5 \\ \texttt{false}: & 1.0 \times 0.5 \end{pmatrix}$$

Normalization: posterior $\propto$ likelihood $\times$ prior

$$\frac{\Gamma \vdash_{\mathrm{p}} t : \mathbb{A}}{\Gamma \vdash_{\mathrm{d}} \texttt{norm}(t) : \mathbb{R} \times \mathrm{P}(\mathbb{A}) + \underbrace{1 + 1}}$$

model evidence

normalized posterior

errors
(constant is 0 or ∞)

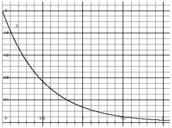
$$\left[\!\left[\begin{array}{l} \texttt{norm(} \\ \quad \texttt{let x = sample(bern(0.5)) in} \\ \quad \texttt{let r = if x then 2.0 else 1.0} \\ \quad \texttt{observe(0.0 from exp(r));} \\ \quad \texttt{return x)} \end{array} \right]\!\right] = \mathrm{in}_1\big(1.5,\ \texttt{bern}(0.66)\big)$$

Example: sequential Monte Carlo

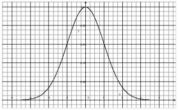
$$\left[\!\!\left[\begin{array}{l}\texttt{norm(let x=t}\\ \texttt{\ \ \ \ \ \ in u)}\end{array}\right]\!\!\right] = \left[\!\!\left[\begin{array}{l}\texttt{norm(let (e,d) = norm(t) in}\\ \texttt{\ \ \ \ \ \ score(e); let x=sample(d)}\\ \texttt{\ \ \ \ \ \ in u)}\end{array}\right]\!\!\right]$$

Example: importance sampling

$$[\![\quad \texttt{sample(exp(2))} \quad]\!]$$

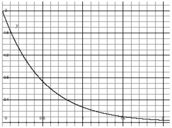


$$= \left[\!\!\left[\begin{array}{l} \texttt{let x = sample(gauss(0,1)))} \\ \texttt{score(exp-pdf(2,x) / gauss-pdf(0,1,x));} \\ \texttt{return x} \end{array} \right]\!\!\right]$$

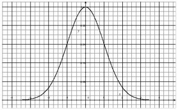


Example: importance sampling

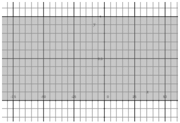
```
⟦ sample(exp(2)) ⟧
```



```
= ⟦ let x = sample(gauss(0,1))
    score(exp-pdf(2,x) / gauss-pdf(0,1,x));
    return x ⟧
```

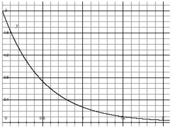


```
= ⟦ let x = sample(gauss(0,1))
    score(1 / gauss-pdf(0,1,x));
    score(exp-pdf(2,x));
    return x ⟧
```

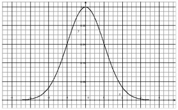


Example: importance sampling

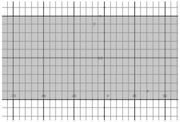
```
⟦norm( sample(exp(2)) )⟧
```



```
= ⟦norm(
    let x = sample(gauss(0,1)))
    score(exp-pdf(2,x) / gauss-pdf(0,1,x));
    return x  )⟧
```



```
≠ ⟦norm( norm(
    let x = sample(gauss(0,1)))
    score(1 / gauss-pdf(0,1,x));  );
    score(exp-pdf(2,x));
    return x )⟧
```



Don't normalize as you go

Commutativity

Reordering lines is very useful program transformation

$$\left[\!\left[\begin{array}{l}\texttt{let x=t in}\\ \texttt{let y=u in}\\ \texttt{v}\end{array}\right]\!\right] = \left[\!\left[\begin{array}{l}\texttt{let y=u in}\\ \texttt{let x=t in}\\ \texttt{v}\end{array}\right]\!\right]$$

Commutativity

Reordering lines is very useful program transformation

$$\left[\!\left[\begin{array}{l}\texttt{let x=t in}\\ \texttt{let y=u in}\\ \texttt{v}\end{array}\right]\!\right] = \left[\!\left[\begin{array}{l}\texttt{let y=u in}\\ \texttt{let x=t in}\\ \texttt{v}\end{array}\right]\!\right]$$

amounts to Fubini's theorem

$$\int_{[\![\mathbb{A}]\!]}\int_{[\![\mathbb{B}]\!]} [\![v]\!]\,\mathrm{d}[\![u]\!]\,\mathrm{d}[\![t]\!] = \int_{[\![\mathbb{B}]\!]}\int_{[\![\mathbb{A}]\!]} [\![v]\!]\,\mathrm{d}[\![t]\!]\,\mathrm{d}[\![u]\!]$$

Commutativity

Reordering lines is very useful program transformation

$$\left[\!\left[\begin{array}{l}\texttt{let x=t in}\\ \texttt{let y=u in}\\ \texttt{v}\end{array}\right]\!\right] = \left[\!\left[\begin{array}{l}\texttt{let y=u in}\\ \texttt{let x=t in}\\ \texttt{v}\end{array}\right]\!\right]$$

amounts to Fubini's theorem

$$\int_{[\![\mathbb{A}]\!]}\int_{[\![\mathbb{B}]\!]} [\![v]\!] \, \mathrm{d}[\![u]\!] \, \mathrm{d}[\![t]\!] = \int_{[\![\mathbb{B}]\!]}\int_{[\![\mathbb{A}]\!]} [\![v]\!] \, \mathrm{d}[\![t]\!] \, \mathrm{d}[\![u]\!]$$

Not true for arbitrary kernels, only for s-finite kernels

kernel is s-finite when countable sum of bounded ones

$k \colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ bounded if $\exists n \forall \gamma \forall U \colon k(\gamma, U) < n$

Commutativity

Reordering lines is very useful program transformation

$$\left[\!\!\left[\begin{array}{l}\texttt{let x=t in}\\ \texttt{let y=u in}\\ \texttt{v}\end{array}\right]\!\!\right] = \left[\!\!\left[\begin{array}{l}\texttt{let y=u in}\\ \texttt{let x=t in}\\ \texttt{v}\end{array}\right]\!\!\right]$$

amounts to Fubini's theorem

$$\int_{[\![\mathbb{A}]\!]}\int_{[\![\mathbb{B}]\!]} [\![v]\!]\,\mathrm{d}[\![u]\!]\,\mathrm{d}[\![t]\!] = \int_{[\![\mathbb{B}]\!]}\int_{[\![\mathbb{A}]\!]} [\![v]\!]\,\mathrm{d}[\![t]\!]\,\mathrm{d}[\![u]\!]$$

Not true for arbitrary kernels, only for s-finite kernels
kernel is s-finite when countable sum of bounded ones
$k\colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty]$ bounded if $\exists n \forall \gamma \forall U \colon k(\gamma, U) < n$

- kernel k is s-finite iff it can be built from sub-probability distributions, `score`, and *binding*
  $k \gg= l$ is $(\gamma, V) \mapsto \int_{[\![\mathbb{A}]\!]} l(\gamma, x, V) k(\gamma, \mathrm{d}x)$
- measurable spaces and s-finite kernels form distributive symmetric monoidal category

Commutativity

Reordering lines is very useful program transformation

$$\left[\!\!\left[\begin{array}{l}\texttt{let x=t in}\\ \texttt{let y=u in}\\ \texttt{v}\end{array}\right]\!\!\right] = \left[\!\!\left[\begin{array}{l}\texttt{let y=u in}\\ \texttt{let x=t in}\\ \texttt{v}\end{array}\right]\!\!\right]$$

amounts to Fubini's theorem

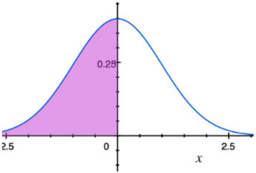
$$\int_{[\![\mathbb{A}]\!]} \int_{[\![\mathbb{B}]\!]} [\![v]\!] \,\mathrm{d}[\![u]\!] \,\mathrm{d}[\![t]\!] = \int_{[\![\mathbb{B}]\!]} \int_{[\![\mathbb{A}]\!]} [\![v]\!] \,\mathrm{d}[\![t]\!] \,\mathrm{d}[\![u]\!]$$

Not true for arbitrary kernels, only for s-finite kernels
kernel is s-finite when countable sum of bounded ones
$k \colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty]$ bounded if $\exists n \forall \gamma \forall U \colon k(\gamma, U) < n$

Interpret terms as s-finite kernels

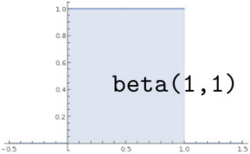
Example: facts about distributions

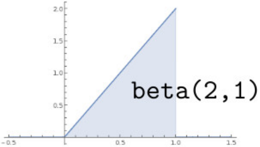
$$\left[\!\!\left[\begin{array}{l}\texttt{let x = sample(gauss(0.0,1.0))}\\ \texttt{in return (x<0)}\end{array}\right]\!\!\right] = [\![\,\texttt{sample(bern(0.5))}\,]\!]$$



Example: conjugate priors

$$\left[\!\!\left[\begin{array}{l}\texttt{let x = sample(beta(1,1))}\\ \texttt{in observe(bern(x), true);}\\ \texttt{return x}\end{array}\right]\!\!\right] = \left[\!\!\left[\begin{array}{l}\texttt{observe(bern(0.5), true);}\\ \texttt{let x = sample(beta(2,1))}\\ \texttt{in return x}\end{array}\right]\!\!\right]$$





Higher order functions

Allow probabilistic terms as input/output for other terms

```
(define (ibp-stick-breaking-process concentration base-measure)
  (let ((sticks (mem (lambda j (random-beta 1.0 concentration))))
        (atoms  (mem (lambda j (base-measure)))))
    (lambda ()
      (let loop ((j 1) (dualstick (sticks 1))}
        (append (if (flip dualstick)              ;; with prob. dualstick
                    (atoms j)                     ;;   add feature j
                    '())                          ;; otherwise, next stick
                (loop (+ j 1) (* dualstick (sticks (+ j 1)))) ))))))
```

[Roy et al, “A stochastic programming perspective on nonparametric Bayes”, ICML 2008]

Higher order functions

Allow probabilistic terms as input/output for other terms

```
(define (ibp-stick-breaking-process concentration base-measure)
  (let ((sticks (mem (lambda j (random-beta 1.0 concentration))))
        (atoms  (mem (lambda j (base-measure)))))
    (lambda ()
      (let loop ((j 1) (dualstick (sticks 1))}
        (append (if (flip dualstick)          ;; with prob. dualstick
                    (atoms j)                 ;;   add feature j
                    '())                      ;; otherwise, next stick
                (loop (+ j 1) (* dualstick (sticks (+ j 1)))) ))))))
```

$$\mathbb{A}, \mathbb{B} ::= \mathbb{R} \mid \mathrm{P}(\mathbb{A}) \mid 1 \mid \mathbb{A} \times \mathbb{B} \mid \sum_{i \in I} \mathbb{A}_i \mid \mathbb{A} \to \mathbb{B}$$

[Roy et al, “A stochastic programming perspective on nonparametric Bayes”, ICML 2008]

Higher order functions

Allow probabilistic terms as input/output for other terms

```
(define (ibp-stick-breaking-process concentration base-measure)
  (let ((sticks (mem (lambda j (random-beta 1.0 concentration))))
        (atoms  (mem (lambda j (base-measure)))))
    (lambda ()
      (let loop ((j 1) (dualstick (sticks 1))}
        (append (if (flip dualstick)          ;; with prob. dualstick
                    (atoms j)                 ;;   add feature j
                    '())                      ;; otherwise, next stick
                (loop (+ j 1) (* dualstick (sticks (+ j 1)))) ))))))
```

$$\mathbb{A}, \mathbb{B} ::= \mathbb{R} \mid \mathrm{P}(\mathbb{A}) \mid 1 \mid \mathbb{A} \times \mathbb{B} \mid \sum_{i \in I} \mathbb{A}_i \mid \mathbb{A} \to \mathbb{B}$$

$\mathbb{R} \to \mathbb{R}$ is not a measurable space

[Roy et al, "A stochastic programming perspective on nonparametric Bayes", ICML 2008]

[Aumann, "Borel structures for function spaces", Ill J Math, 1961]

Higher order functions

Allow probabilistic terms as input/output for other terms

```
(define (ibp-stick-breaking-process concentration base-measure)
  (let ((sticks (mem (lambda j (random-beta 1.0 concentration))))
        (atoms  (mem (lambda j (base-measure)))))
    (lambda ()
      (let loop ((j 1) (dualstick (sticks 1))}
        (append (if (flip dualstick)              ;; with prob. dualstick
                    (atoms j)                     ;;   add feature j
                    '())                          ;; otherwise, next stick
                (loop (+ j 1) (* dualstick (sticks (+ j 1)))) ))))))
```

$$\mathbb{A}, \mathbb{B} ::= \mathbb{R} \mid \mathrm{P}(\mathbb{A}) \mid 1 \mid \mathbb{A} \times \mathbb{B} \mid \sum_{i \in I} \mathbb{A}_i \mid \mathbb{A} \to \mathbb{B}$$

$\mathbb{R} \to \mathbb{R}$ is not a measurable space
Easy to handle operationally.
What to do denotationally?

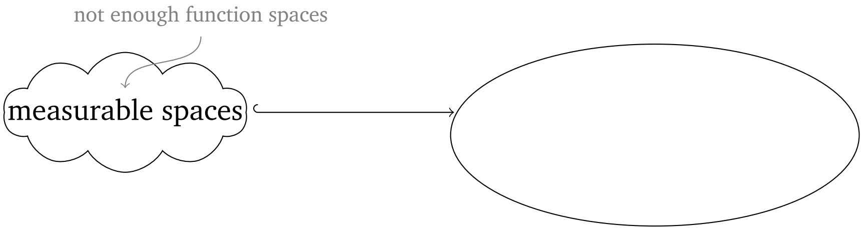
[Roy et al, "A stochastic programming perspective on nonparametric Bayes", ICML 2008]

[Aumann, "Borel structures for function spaces", Ill J Math, 1961]

[Borgström et al, "Measure transformer semantics for Bayesian machine learning", ESOP2011]

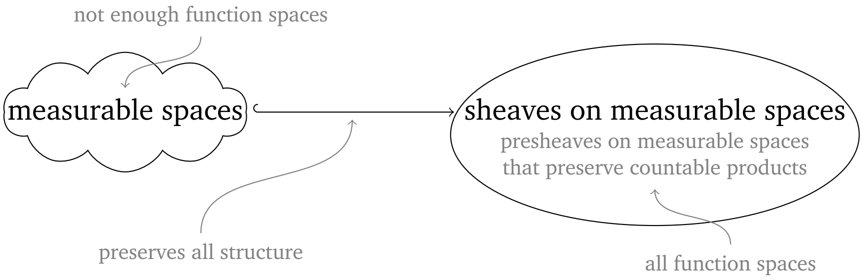
Higher order semantics

Use category theory to extend measure theory



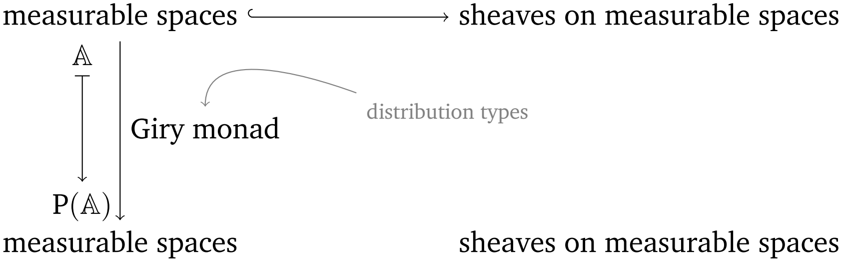
Higher order semantics

Use category theory to extend measure theory



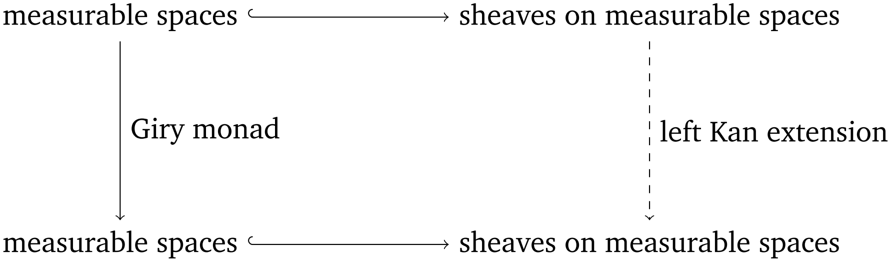
Higher order semantics

Use category theory to extend measure theory



Higher order semantics

Use category theory to extend measure theory



[Power, “Generic models for computational effects”, Th Comp Sci 2006]

Higher order semantics

Use category theory to extend measure theory

$$\text{measurable spaces} \hookrightarrow \text{sheaves on measurable spaces}$$

- $[\![1 \to (\mathbb{R} \to \mathbb{R})]\!]$ consists of random functions   measurable $\Omega \times \mathbb{R} \to \mathbb{R}$
- All definable functions $\mathbb{R} \to \mathbb{R}$ are measurable   “Church-Turing”
- Denotational and operational semantics match   soundness & adequacy

Extensionality

Not extensional: $1 \xrightarrow{p} \mathbb{A} \underset{g}{\overset{f}{\rightrightarrows}} \mathbb{B}$ for all $p \;\not\Rightarrow\; f = g$

Solution: restrict to subcategory that *is* extensional

Extensionality

Not extensional: $1 \xrightarrow{p} \mathbb{A} \overset{f}{\underset{g}{\rightrightarrows}} \mathbb{B}$ for all $p \not\Rightarrow f = g$
Solution: restrict to subcategory that *is* extensional

A quasi-measurable space is a set X with $M_X \subseteq [\mathbb{R} \to X]$ satisfying

- if $f \colon \mathbb{R} \to \mathbb{R}$ is measurable and $g \in M$, then $gf \in M$
- if $f \colon \mathbb{R} \to X$ is constant, then $f \in M$
- if $f \colon \mathbb{R} \to \mathbb{N}$ is measurable and $g_n \in M$, then $[g_n]f \in M$ $\quad t \mapsto g_{f(t)}(t)$

morphisms are functions $f \colon X \to Y$ with $g \in M_X \Rightarrow fg \in M_Y$

Extensionality

Not extensional: $1 \xrightarrow{p} \mathbb{A} \underset{g}{\overset{f}{\rightrightarrows}} \mathbb{B}$ for all $p \not\Rightarrow f = g$
Solution: restrict to subcategory that *is* extensional

A quasi-measurable space is a set X with $M_X \subseteq [\mathbb{R} \to X]$ satisfying

- if $f : \mathbb{R} \to \mathbb{R}$ is measurable and $g \in M$, then $gf \in M$
- if $f : \mathbb{R} \to X$ is constant, then $f \in M$
- if $f : \mathbb{R} \to \mathbb{N}$ is measurable and $g_n \in M$, then $[g_n]f \in M$ $\quad t \mapsto g_{f(t)}(t)$

morphisms are functions $f : X \to Y$ with $g \in M_X \Rightarrow fg \in M_Y$

Example: X measurable space, M_X measurable functions $\mathbb{R} \to X$
morphism $X \to Y$ is measurable function

Extensionality

Not extensional: $1 \xrightarrow{p} \mathbb{A} \overset{f}{\underset{g}{\rightrightarrows}} \mathbb{B}$ for all $p \not\Rightarrow f = g$

Solution: restrict to subcategory that *is* extensional

A quasi-measurable space is a set X with $M_X \subseteq [\mathbb{R} \to X]$ satisfying

- if $f \colon \mathbb{R} \to \mathbb{R}$ is measurable and $g \in M$, then $gf \in M$
- if $f \colon \mathbb{R} \to X$ is constant, then $f \in M$
- if $f \colon \mathbb{R} \to \mathbb{N}$ is measurable and $g_n \in M$, then $[g_n]f \in M$ $t \mapsto g_{f(t)}(t)$

morphisms are functions $f \colon X \to Y$ with $g \in M_X \Rightarrow fg \in M_Y$

Example: X measurable space, M_X measurable functions $\mathbb{R} \to X$

morphism $X \to Y$ is measurable function

Theorem: this gives cartesian closed category with countable sums

Corollary: if term t has first order type, then $[\![t]\!]$ is measurable

even if t involves higher order functions

Extensionality

Not extensional: $1 \xrightarrow{p} \mathbb{A} \underset{g}{\overset{f}{\rightrightarrows}} \mathbb{B}$ for all $p \not\Rightarrow f = g$

Solution: restrict to subcategory that *is* extensional

A quasi-measurable space is a set X with $M_X \subseteq [\mathbb{R} \to X]$ satisfying

- if $f \colon \mathbb{R} \to \mathbb{R}$ is measurable and $g \in M$, then $gf \in M$
- if $f \colon \mathbb{R} \to X$ is constant, then $f \in M$
- if $f \colon \mathbb{R} \to \mathbb{N}$ is measurable and $g_n \in M$, then $[g_n]f \in M$ $\quad t \mapsto g_{f(t)}(t)$

A measure on (X, M_X) is a measure μ on $\mathbb{R}$ with a function $f \in M$

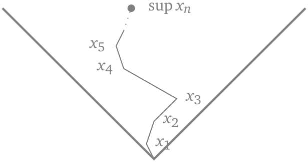
Proposition: measures on $[X \to Y]$ are random functions

measurable map $\mathbb{R} \times X \to Y$ modulo measure on $\mathbb{R}$

Recursion

No recursion / least fixed points
Idea: restrict to presheaves over domains

An ω-complete partial order has suprema of increasing sequences
morphisms preserve suprema of increasing sequences and infima



A quasi-measurable space is ordered when X is an ωcpo and M is closed under pointwise increasing suprema

Example: Any ωcpo, e.g. [0,1]   take M all measurable functions $\mathbb{R} \to X$
where X has the Borel σ-algebra on the Lawson topology

Theorem: this gives a cartesian closed category with countable sums

Example: von Neumann's trick

$$\left[\!\!\left[\begin{array}{l}\texttt{let g = bern(0.66) in}\\ \texttt{letrec f() = (let x = sample(g)}\\ \qquad\qquad\quad\texttt{let y = sample(g)}\\ \qquad\qquad\quad\texttt{if x=y then f()}\\ \qquad\qquad\qquad\quad\texttt{else return x)}\\ \texttt{in f()}\end{array}\right]\!\!\right] \stackrel{?}{=} [\![\texttt{sample(bern(0.5))}]\!]$$

Conclusion

Foundational semantics for probabilistic programming:

- continuous distributions
- soft constraints
- commutativity
- higher order functions
- recursion

can verify/suggest program transformations. Approximations?