Semantic models of higher order probability theory

Sam Staton, Oxford

partly jointly with Hongseok Yang, Ohad Kammar, Chris Heunen, Frank Wood…

Motivation

What is a general notion of statistical model?

How can we understand the fundamental structure?

Aims of semantic models:

1. Fundamental;
2. Towards applications in Machine Learning.

Bayes' law

$$P(x \mid d) = \frac{P(d \mid x) \times P(x)}{P(d)}$$

Bayes' law

$$P(x \mid d) = \frac{P(d \mid x) \times P(x)}{\Sigma_y P(d \mid y) \times P(y)}$$

Bayes' law

$$P(x \mid d) \propto P(d \mid x) \times P(x)$$

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

Idealized Anglican

$$P(x \mid d) \propto P(d \mid x) \times P(x)$$

Posterior $\propto$ Likelihood $\times$ Prior

Idealized Anglican = sequential programming +

normalize **observe** **sample**

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

Idealized Anglican

$$P(x \mid d) \;\propto\; P(d \mid x) \times P(x)$$

$$\text{Posterior} \;\propto\; \text{Likelihood} \times \text{Prior}$$

Idealized Anglican = sequential programming +

normalize **observe** **sample**

other languages:
Church, Venture…
Hakaru … Stan…

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

Example

1. A call centre operator doesn't know what day it is.
2. He knows: weekends: avg 3 calls per hour.
   weekdays: avg 10 calls per hour.
3. He notices a 15 minute gap between calls.
4. **Is it the weekend?**

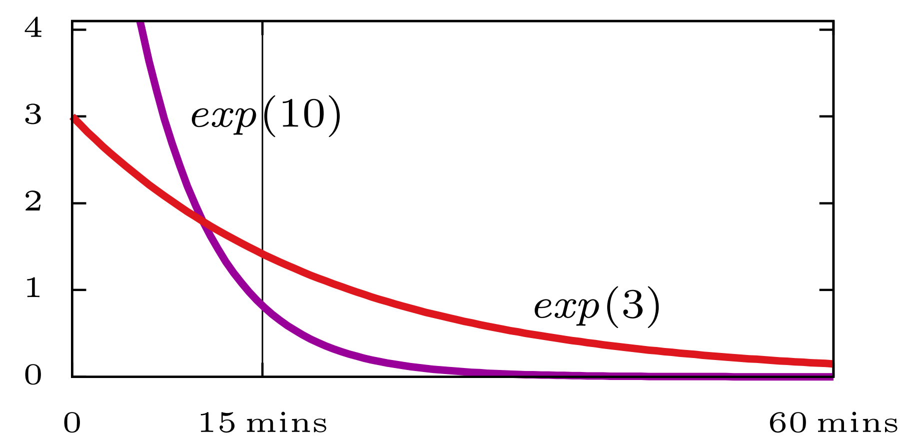
Example

1. A call centre operator doesn't know what day it is.
2. He knows: weekends: avg 3 calls per hour.
   weekdays: avg 10 calls per hour.
3. He notices a 15 minute gap between calls.
4. **Is it the weekend?**

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 0.25 from exp-dist(rate);
  return(weekend) )
```

Example

1. A call centre operator doesn’t know what day it is.
2. He knows: weekends: avg 3 calls per hour.
   weekdays: avg 10 calls per hour.
3. He notices a 15 minute gap between calls.
4. **Is it the weekend?**




```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 0.25 from exp-dist(rate);
  return(weekend) )
```

Motivation

$$P(x \mid d) \;\propto\; P(d \mid x) \times P(x)$$

$$\text{Posterior} \;\propto\; \text{Likelihood} \times \text{Prior}$$

Idealized Anglican = sequential programming +

normalize **observe** **sample**

Aims of semantic models:

1. Justify program transformations;
2. Understand complex probabilistic models through semantics.

Main theorem: Commutativity

$$\boxed{\begin{array}{l}\mathsf{let}\, x = t \,\mathsf{in} \\ \mathsf{let}\, y = u \,\mathsf{in} \\ v\end{array}} = \boxed{\begin{array}{l}\mathsf{let}\, y = u \,\mathsf{in} \\ \mathsf{let}\, x = t \,\mathsf{in} \\ v\end{array}}$$

where x not free in u,
y not free in t

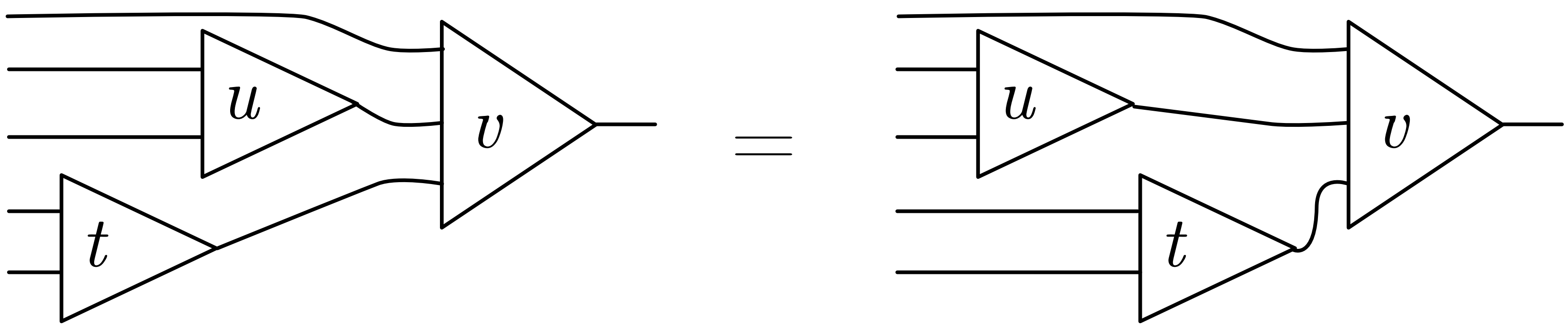
Aims of semantics:

1. Justify program transformations;
2. Understand complex probabilistic models through semantics.

Main theorem: Commutativity

$$\boxed{\begin{array}{l}\mathsf{let}\; x = t\; \mathsf{in}\\ \mathsf{let}\; y = u\; \mathsf{in}\\ v\end{array}} \quad = \quad \boxed{\begin{array}{l}\mathsf{let}\; y = u\; \mathsf{in}\\ \mathsf{let}\; x = t\; \mathsf{in}\\ v\end{array}}$$

where x not free in u,
y not free in t



connections with multicategories, monoidal categories etc..

Overview of the rest of part 1

Compositional probability theory;
a compositional theory of impropriety

- Summary of semantics
- Examples
- s-finite kernels and
  proof of commutativity.

Summary of semantics

To begin: a first order typed language.

Types:

distributions over A

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

countable sums

real numbers

finite products

Terms: sequencing: `let x = t in u`

`normalize` `observe` `sample`

Summary of semantics

For now: a first order typed language without recursion.

Types interpreted as measurable spaces.

Closed terms of type A
interpreted as measures on $[\![A]\!]$.

Summary of semantics

Types interpreted as measurable spaces.

A **measurable space** (X, Σ_X) is a set X together with set Σ_X of subsets closed under countable unions and complements.

Summary of semantics

Types interpreted as measurable spaces.

e.g. $[\![\texttt{real}]\!] = (\mathbb{R}, \Sigma_{\mathbb{R}})$

where $\Sigma_{\mathbb{R}}$ = Borel sets

generated by intervals e.g. (-2.2 , 4.1)
& countable unions & complements

A **measurable space** (X, Σ_X) is
a set X together with set Σ_X of
subsets closed under countable
unions and complements.

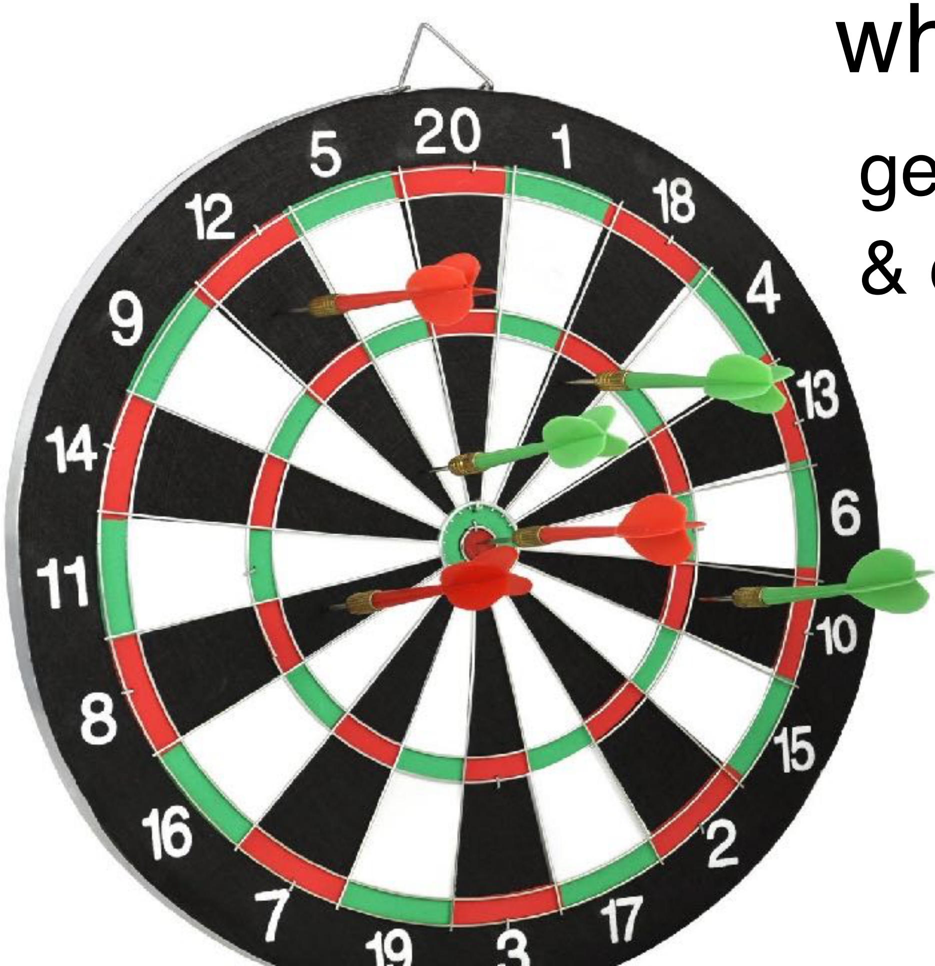
Summary of semantics

Types interpreted as measurable spaces.

e.g. $[\![\texttt{real}]\!] = (\mathbb{R}, \Sigma_{\mathbb{R}})$

where $\Sigma_{\mathbb{R}}$ = Borel sets

generated by intervals e.g. (-2.2 , 4.1)
& countable unions & complements



A **measurable space** (X, Σ_X) is a set X together with set Σ_X of subsets closed under countable unions and complements.

Summary of semantics

Types interpreted as measurable spaces.

e.g. $[\![\texttt{real}]\!] = (\mathbb{R}, \Sigma_{\mathbb{R}})$

Closed terms interpreted as measures

Open terms $\Gamma \vdash t : \mathbb{A}$ are interpreted as kernels

$$[\![t]\!] : [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty].$$

measurable in $[\![\Gamma]\!]$

countably additive in $\Sigma_{[\![\mathbb{A}]\!]}$

$\biguplus \mapsto +$

Summary of semantics

Types interpreted as measurable spaces.

e.g. $[\![\texttt{real}]\!] = (\mathbb{R}, \Sigma_{\mathbb{R}})$

Closed terms interpreted as measures

Open terms interpreted as kernels

Sequencing is integration

$$[\![\mathsf{let}\, x = t\, \mathsf{in}\, u]\!](\gamma, U) \stackrel{\mathrm{def}}{=} \int [\![u]\!](\gamma, x, U)\ [\![t]\!](\gamma, \mathrm{d}x)$$

Summary of semantics

Types interpreted as measurable spaces.

e.g. `real` type interpreted as
$\mathbb{R}$ with Borel sets

Closed terms interpreted as measures

Open terms interpreted as kernels

Sequencing is integration

see Kozen, 1981
Börgstrom, Gordon et al., ESOP 2011
Shan, Ramsey, POPL 2017

$$\text{“}\ [\![\mathsf{let}\, x = t\, \mathsf{in}\, u]\!] = \int [\![u]\!]\ \mathrm{d}[\![t]\!]\ \text{”}$$

Summary of semantics

Closed terms interpreted as measures

Sequencing is integration

$$\text{“}\,[\![\mathsf{let}\, x = t \,\mathsf{in}\, u]\!] = \int [\![u]\!]\ \mathrm{d}[\![t]\!]\,\text{”}$$

Key Theorem: Commutativity.

$$\boxed{\begin{array}{l}\mathsf{let}\, x = t \,\mathsf{in}\\ \mathsf{let}\, y = u \,\mathsf{in}\\ v\end{array}} = \boxed{\begin{array}{l}\mathsf{let}\, y = u \,\mathsf{in}\\ \mathsf{let}\, x = t \,\mathsf{in}\\ v\end{array}}$$

Proof summary:
Probabilistic programs
≈ s-finite kernels.

$$\int\Big(\int [\![\mathtt{v}]\!]\ \mathrm{d}[\![\mathtt{u}]\!]\Big)\ \mathrm{d}[\![\mathtt{t}]\!] \;=\; \int\Big(\int [\![\mathtt{v}]\!]\,\mathrm{d}[\![\mathtt{t}]\!]\Big)\ \mathrm{d}[\![\mathtt{u}]\!]$$

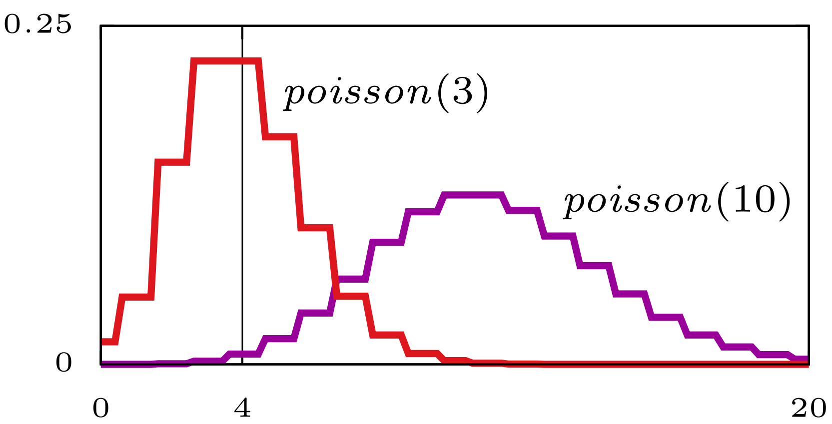
Overview of the rest of part 1

Compositional probability theory;
a compositional theory of impropriety

- Summary of semantics
- Examples
- s-finite kernels and
  proof of commutativity.

Worked examples

1. A call centre operator doesn't know what day it is.
2. He knows: weekends: avg 3 calls per hour.
   weekdays: avg 10 calls per hour.
3. He notices 4 calls in a given hour.
4. Is it the weekend?



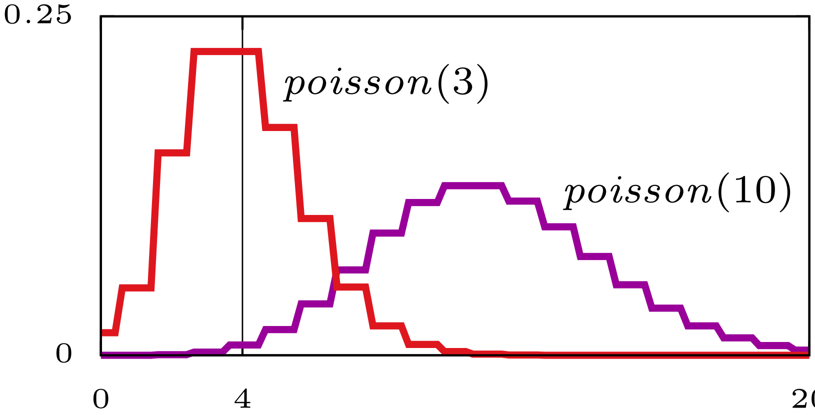

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

Worked examples

Unnormalized posterior:

```
m(weekend=true)  = 2/7 × 0.168 = 0.048

m(weekend=false) = 5/7 × 0.019 = 0.014
```




```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

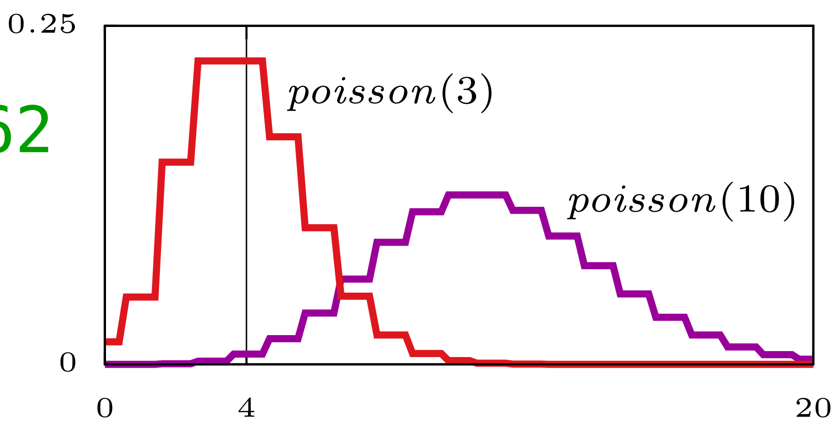
Worked examples

Unnormalized posterior:

```
m(weekend=true)  = 2/7 × 0.168 = 0.048
m(weekend=false) = 5/7 × 0.019 = 0.014
```

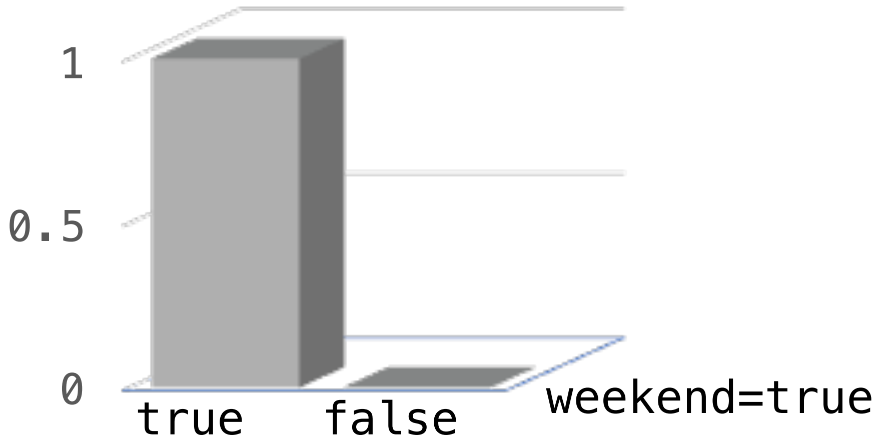
Normalized posterior:

```
p(weekend=true) = 0.048 / 0.062
                = 0.78
```




```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

Worked examples



⟦return(weekend)⟧

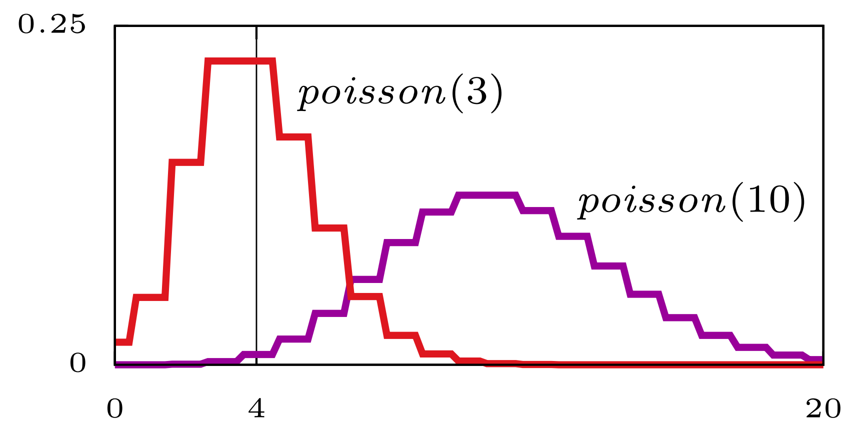
Open terms $\Gamma \vdash t\colon \mathbb{A}$ are interpreted as kernels

$$[\![t]\!]\colon\ [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty].$$

measurable in $[\![\Gamma]\!]$

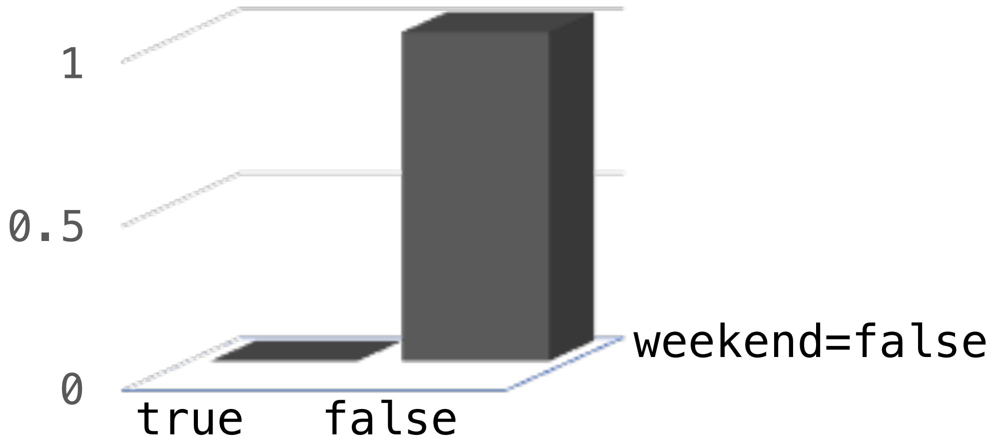
countably additive in $\Sigma_{[\![\mathbb{A}]\!]}$

$\biguplus \mapsto +$




```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

Worked examples



$[\![$return(weekend)$]\!]$

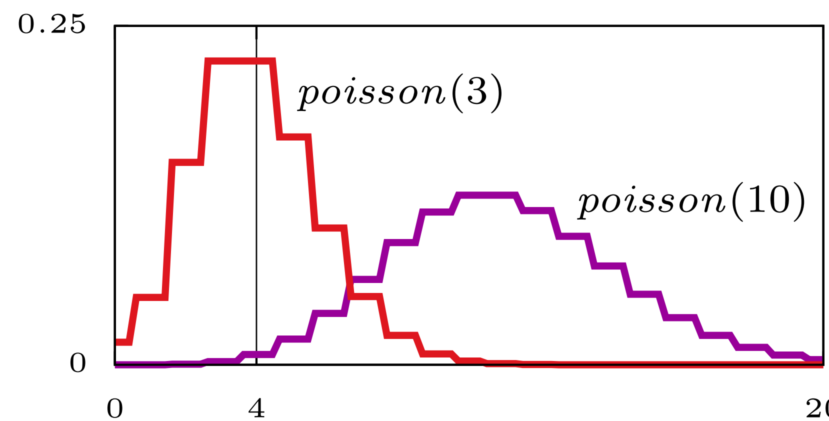
Open terms $\Gamma \vdash t\colon \mathbb{A}$ are interpreted as kernels

$$[\![t]\!]\colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty].$$

measurable in $[\![\Gamma]\!]$

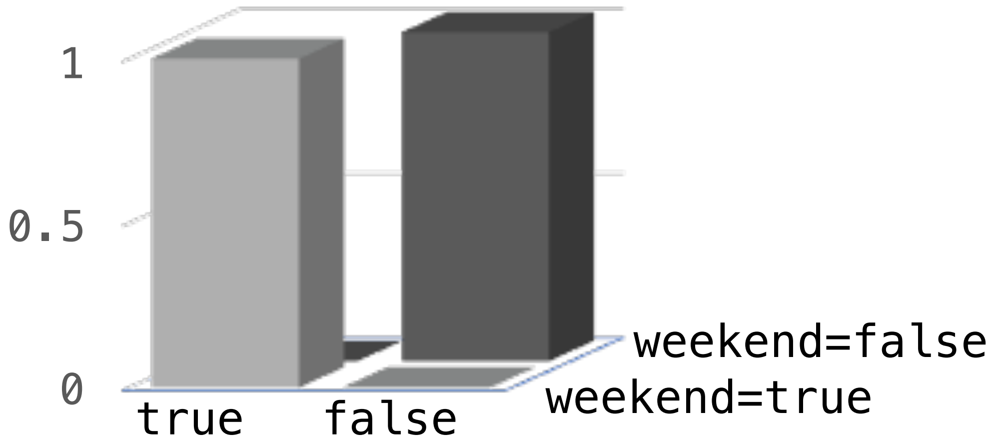
countably additive in $\Sigma_{[\![\mathbb{A}]\!]}$

$\uplus \mapsto +$




```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

Worked examples



$[\![$return(weekend)$]\!]$

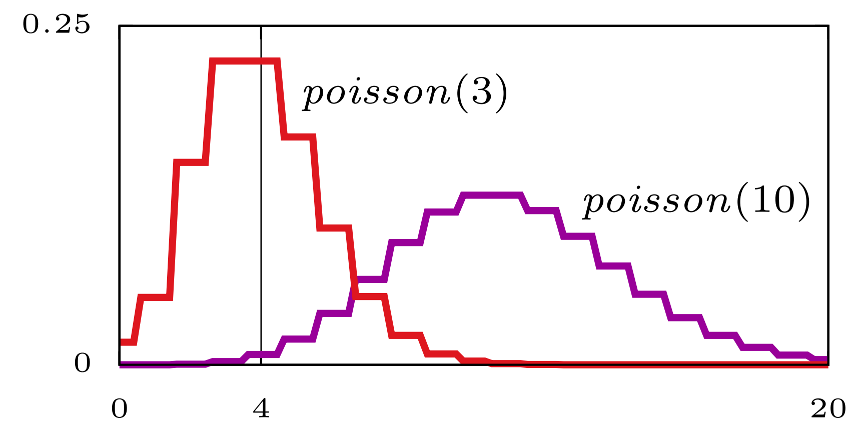
Open terms $\Gamma \vdash t\colon \mathbb{A}$ are interpreted as kernels

$$[\![t]\!]\colon [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty].$$

measurable in $[\![\Gamma]\!]$

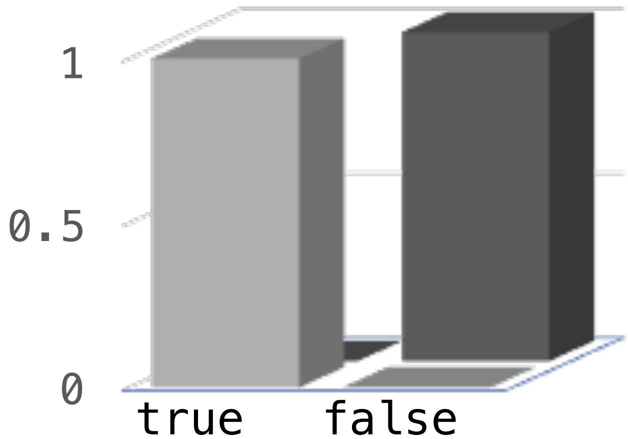
countably additive in $\Sigma_{[\![\mathbb{A}]\!]}$

$\uplus \mapsto +$

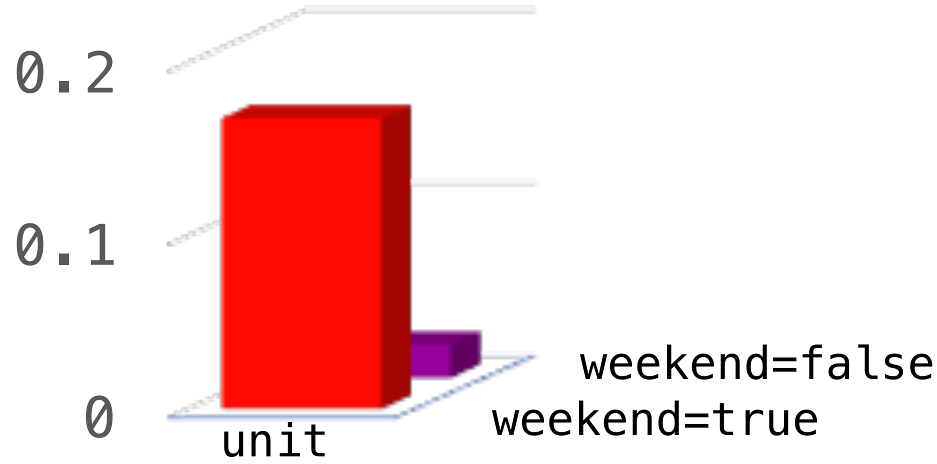



```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

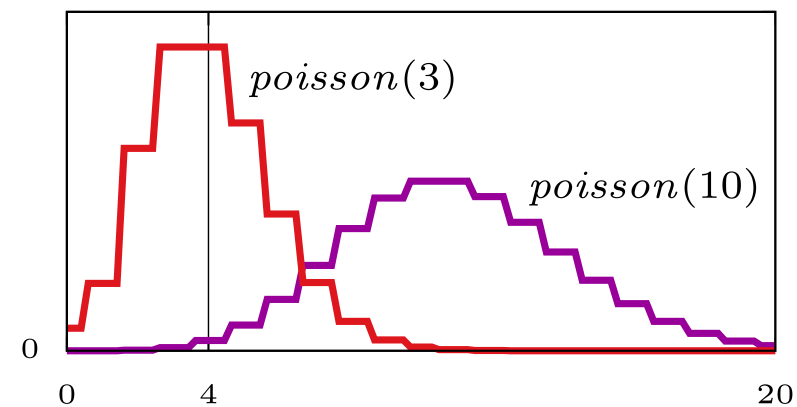
Worked examples



⟦return(weekend)⟧

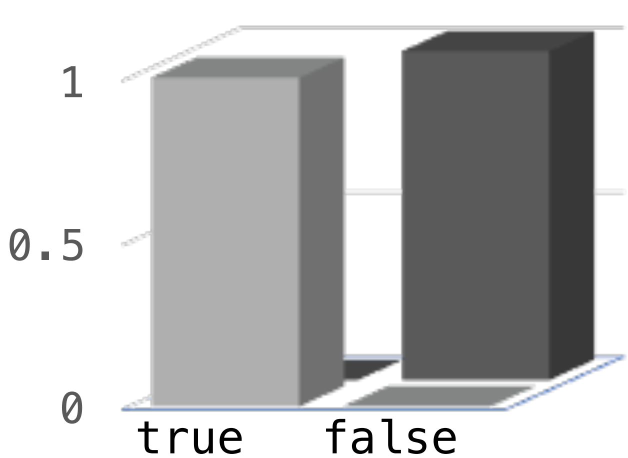


⟦let rate = … in
observe 4 from …⟧

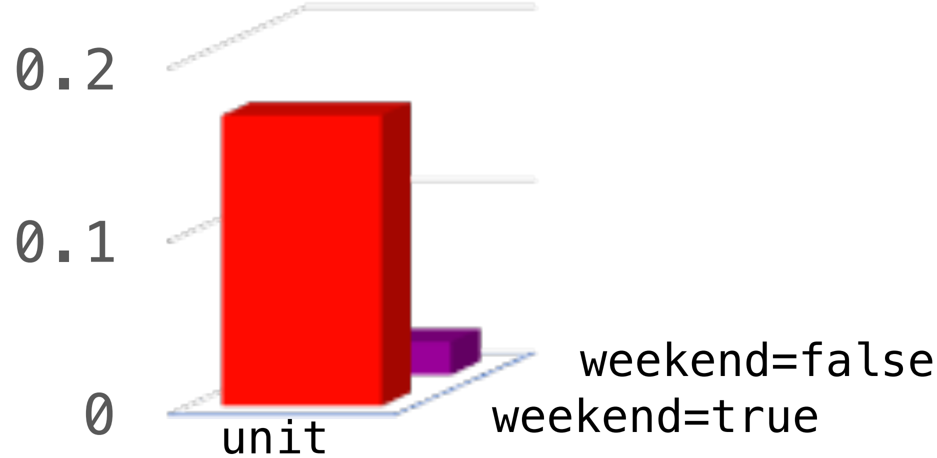



```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

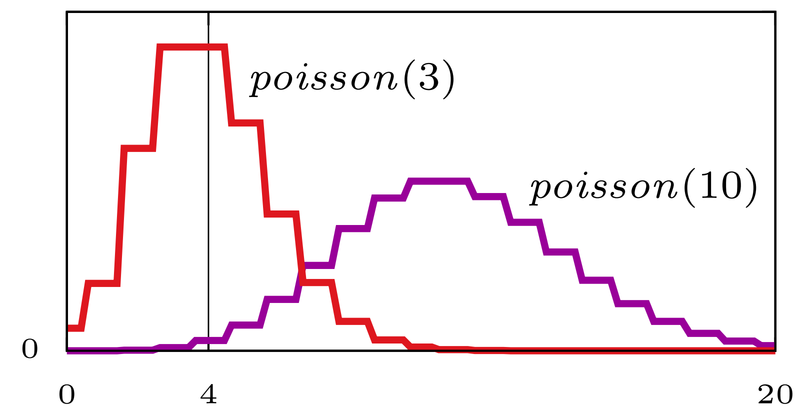
Worked examples



⟦return(weekend)⟧



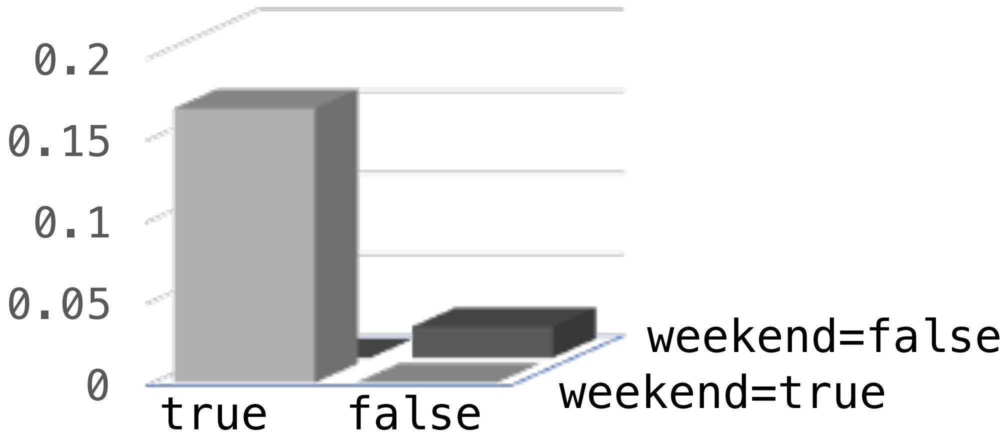
⟦let rate = … in
observe 4 from …⟧



$$[\![\text{let } x = t \text{ in } u]\!] = \int_{[\![\mathbb{A}]\!]} [\![u]\!]\, \mathrm{d}[\![t]\!]$$

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

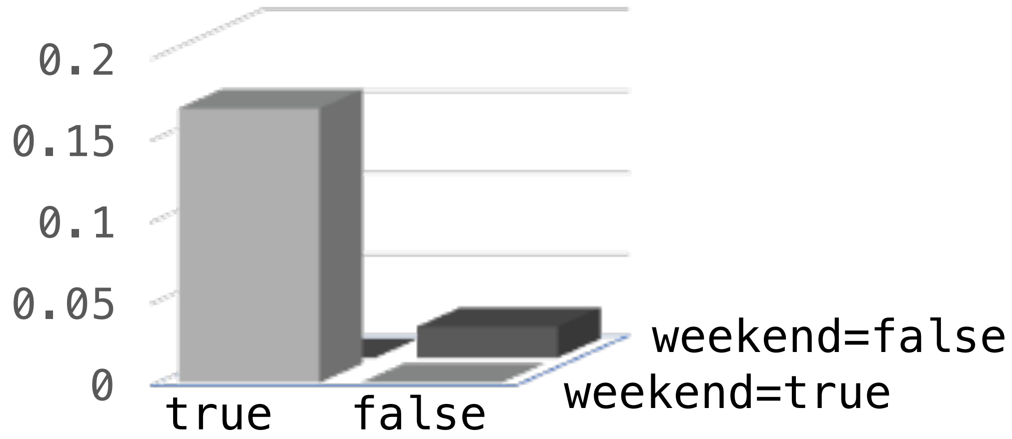
Worked examples



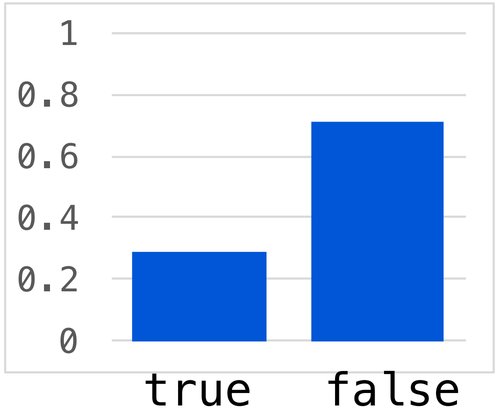

```
⟦let rate = … in
 observe 4 from …;
 return(weekend)⟧
```

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

Worked examples



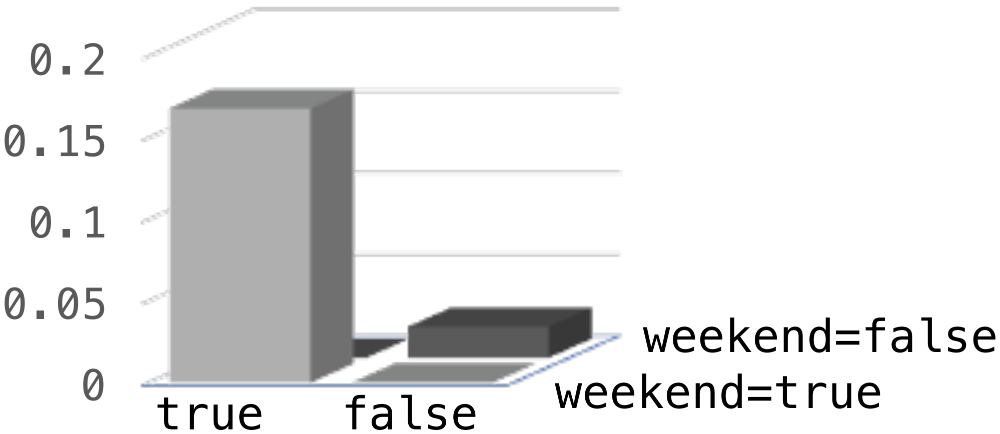
⟦let rate = … in
observe 4 from …;
return(weekend)⟧



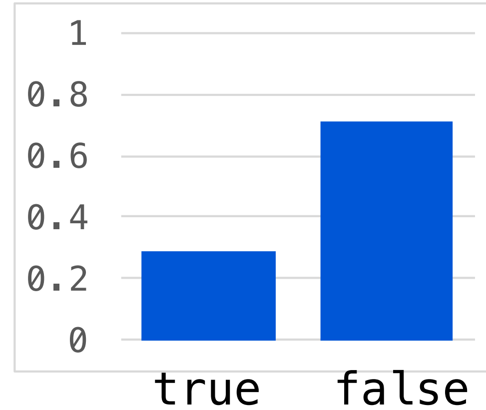
⟦sample(bernoulli(2/7)⟧

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

Worked examples



⟦let rate = … in
observe 4 from …;
return(weekend)⟧

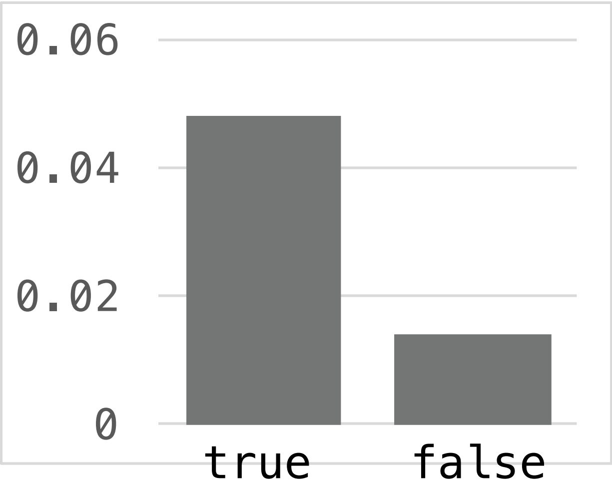


⟦sample(bernoulli(2/7))⟧

$$[\![\text{let } x = t \text{ in } u]\!] = \int_{[\![\mathbb{A}]\!]} [\![u]\!] \,\mathrm{d}[\![t]\!]$$

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

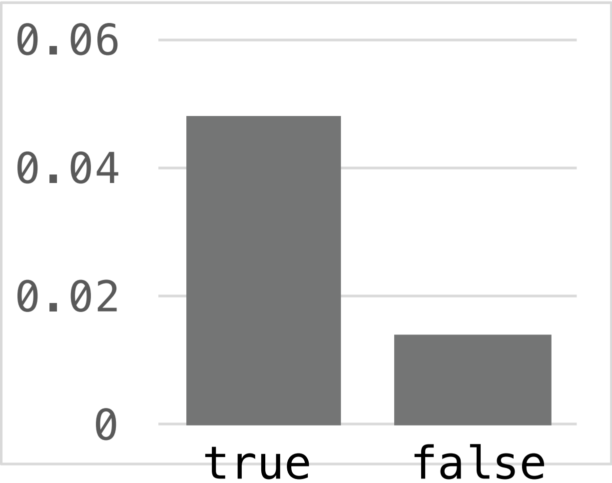
Worked examples



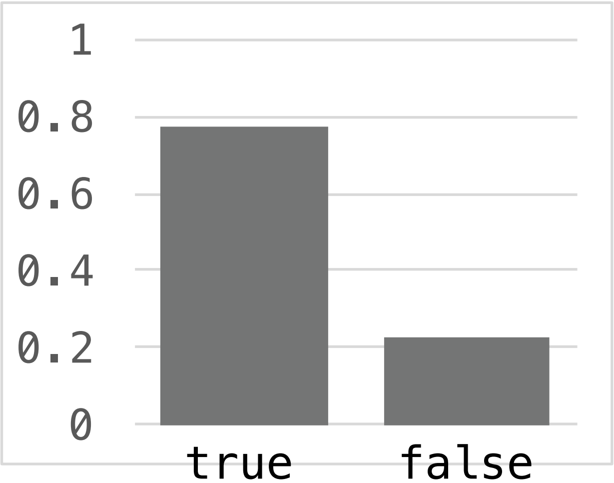

```
⟦let weekend = … in
 let rate = … in
 observe 4 from …;
 return(weekend)⟧
```

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

Worked examples



⟦let weekend = … in
let rate = … in
observe 4 from …;
return(weekend)⟧

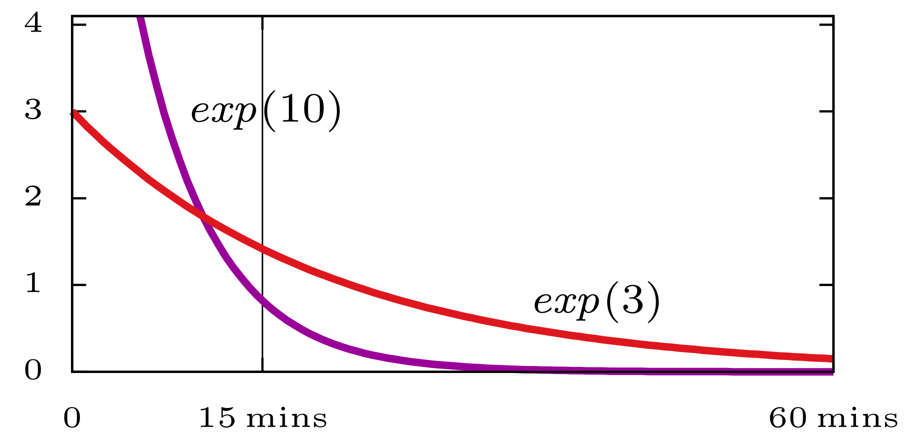


⟦normalize(
let weekend = … in
let rate = … in
observe 4 from …;
return(weekend))⟧

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 4 from poisson(rate);
  return(weekend) )
```

Worked examples

1. A call centre operator doesn't know what day it is.
2. He knows: weekends: avg 3 calls per hour.
   weekdays: avg 10 calls per hour.
3. He notices a 15 minute gap between calls.
4. Is it the weekend?




```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 0.25 from exp-dist(rate);
  return(weekend) )
```

Worked examples

Unnormalized posterior:

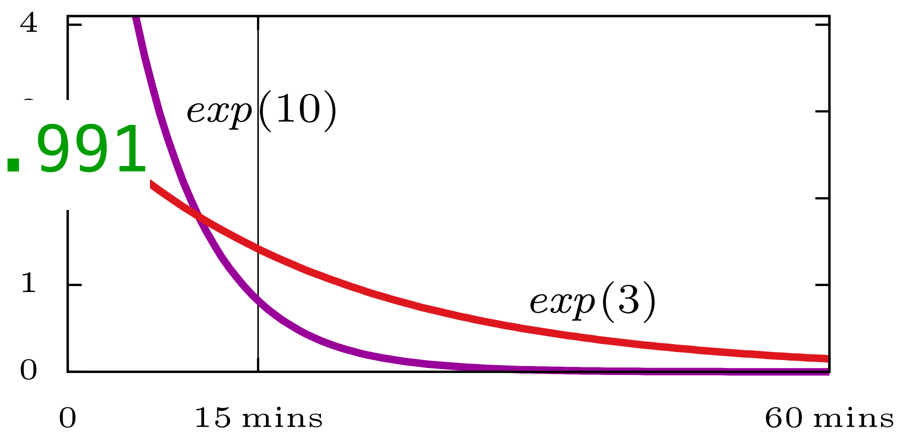
```
m(weekend=true)  = 2/7 × 1.42 = 0.405

m(weekend=false) = 5/7 × 0.82 = 0.586
```

Normalized posterior:

```
p(weekend=true) = 0.405 / 0.991

                = 0.408
```



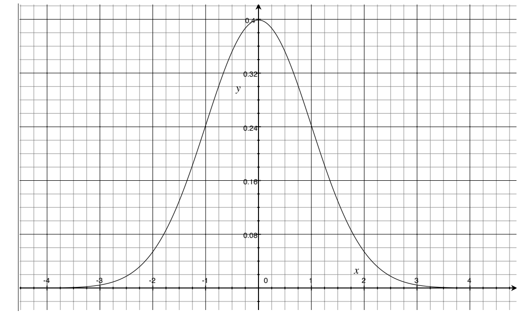

```
normalize(
  let weekend = sample(bernoulli(2/7)) in
  let rate = if weekend then 3 else 10 in
  observe 0.25 from exp-dist(rate);
  return(weekend) )
```

Worked examples

1. A call centre operator doesn't know what time it is.
2. He knows how the avg num of calls varies with time.
3. He notices a 15 minute gap between calls.
4. What time is it?

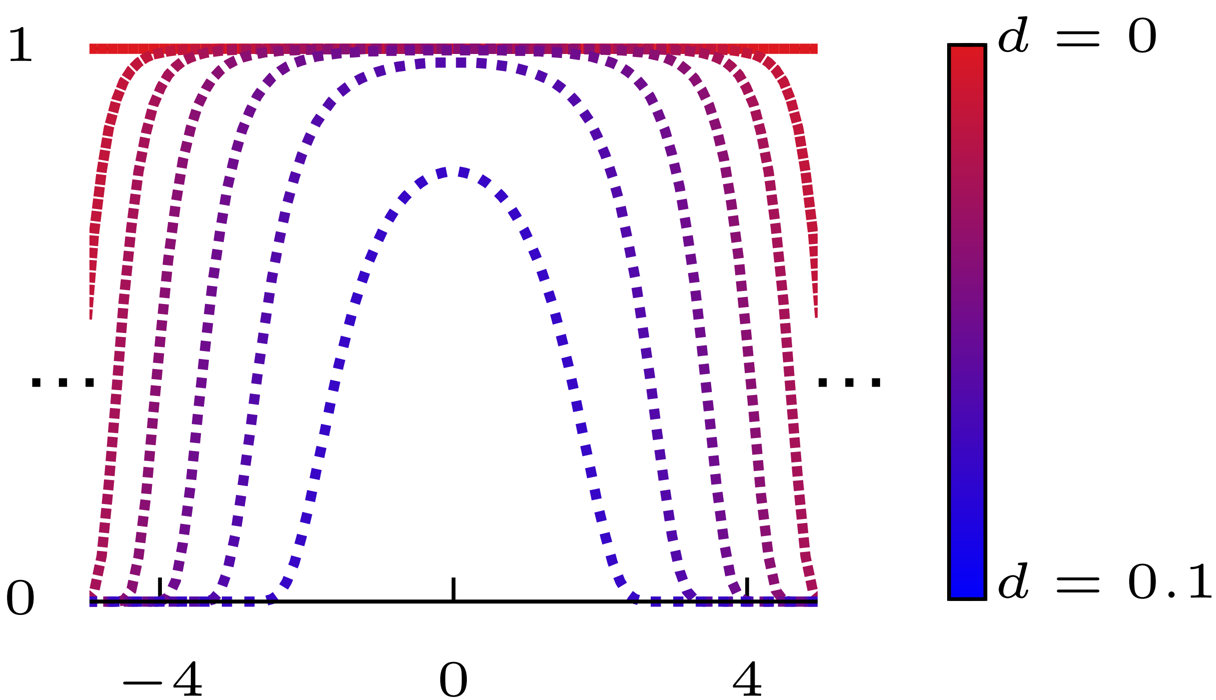
```
normalize(
  let time = sample(uniform(0,24)) in
  let rate = f(time) in
  observe 0.25 from exp-dist(rate);
  return(weekend) )
```

Improper posteriors



```
let x = sample(Normal(0,1)) in
let r = 1/f(x) in
observe d from exp-dist(r);
return x
```

where $\texttt{f}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$



As $d \to 0$, the normalizing constant tends to ∞.
"The posterior is improper."

Overview of the rest of part 1

Compositional probability theory;
a compositional theory of impropriety

- Summary of semantics
- Examples
- s-finite kernels and proof of commutativity.

Overview of the rest of part 1

Compositional probability theory; a compositional theory of impropriety

- Summary of semantics
- Examples
- **s-finite kernels and proof of commutativity.**

A technical problem:

Defn. A kernel is a function $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty]$ such that for all $U \in \Sigma_{[\![\mathbb{A}]\!]}$, $[\![t]\!](-,U) : [\![\Gamma]\!] \to [0,\infty]$ is measurable and $[\![t]\!](\gamma,-) : \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty]$ is additive.

A technical problem:

Defn. A kernel is a function $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$

such that for all $U \in \Sigma_{[\![\mathbb{A}]\!]}$, $[\![t]\!](-, U) : [\![\Gamma]\!] \to [0, \infty]$

is measurable and …

A technical problem:

Defn. A kernel is a function $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty]$ such that for all $U \in \Sigma_{[\![\mathbb{A}]\!]}$, $[\![t]\!](-,U) : [\![\Gamma]\!] \to [0,\infty]$ is measurable and …

Sequencing is integration:

$$[\![\mathsf{let}\, x = t \,\mathsf{in}\, u]\!](\gamma, U) \stackrel{\text{def}}{=} \int [\![u]\!](\gamma, x, U)\ [\![t]\!](\gamma, \mathrm{d}x)$$

A technical problem:

Defn. A kernel is a function $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$

such that for all $U \in \Sigma_{[\![\mathbb{A}]\!]}$, $[\![t]\!](-, U) : [\![\Gamma]\!] \to [0, \infty]$

is measurable and …

Sequencing is integration:

$$[\![\mathsf{let}\, x = t \,\mathsf{in}\, u]\!](\gamma, U) \stackrel{\text{def}}{=} \int [\![u]\!](\gamma, x, U)\ [\![t]\!](\gamma, \mathrm{d}x)$$

Is this measurable in γ?

A technical problem:

Defn. A kernel is a function $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ such that for all $U \in \Sigma_{[\![\mathbb{A}]\!]}$, $[\![t]\!](-, U) : [\![\Gamma]\!] \to [0, \infty]$ is measurable and …

Sequencing is integration:

$$[\![\mathsf{let}\, x = t \,\mathsf{in}\, u]\!](\gamma, U) \stackrel{\text{def}}{=} \int [\![u]\!](\gamma, x, U)\ [\![t]\!](\gamma, \mathrm{d}x)$$

Is this measurable in γ?

Answer: **unknown** (if t and u are allowed to range over arbitrary kernels.)

Commutativity

Terms are interpreted as kernels.

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

reordering lines

- Useful for program transformations.
- Essence of probability theory.

Commutativity

Terms are interpreted as kernels.

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

reordering lines

Roughly amounts to

$$\int\Big(\int [\![\texttt{v}]\!]\,\mathrm{d}[\![\texttt{u}]\!]\Big)\,\mathrm{d}[\![\texttt{t}]\!] \;=\; \int\Big(\int [\![\texttt{v}]\!]\,\mathrm{d}[\![\texttt{t}]\!]\Big)\,\mathrm{d}[\![\texttt{u}]\!]$$

Fubini's theorem

Commutativity

Terms are interpreted as kernels.

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

reordering lines

Roughly amounts to

$$\int\Big(\int [\![\mathtt{v}]\!]\;\mathrm{d}[\![\mathtt{u}]\!]\Big)\;\mathrm{d}[\![\mathtt{t}]\!] \;=\; \int\Big(\int [\![\mathtt{v}]\!]\;\mathrm{d}[\![\mathtt{t}]\!]\Big)\;\mathrm{d}[\![\mathtt{u}]\!]$$

Fubini's theorem

Warning: Fubini's theorem does not hold for arbitrary kernels.

Commutativity

Terms are interpreted as kernels.

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

reordering lines

Roughly amounts to

$$\int\Big(\int [\![\mathtt{v}]\!]\ \mathrm{d}[\![\mathtt{u}]\!]\Big)\ \mathrm{d}[\![\mathtt{t}]\!] \;=\; \int\Big(\int [\![\mathtt{v}]\!]\ \mathrm{d}[\![\mathtt{t}]\!]\Big)\ \mathrm{d}[\![\mathtt{u}]\!]$$

Fubini's theorem

Warning: Fubini's theorem does not hold for arbitrary kernels.

Fubini's theorem *does* hold for **s-finite** kernels.

Commutativity

Terms are interpreted as **s-finite** kernels.

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

reordering lines

Roughly amounts to

$$\int\Big(\int [\![\mathtt{v}]\!]\ \mathrm{d}[\![\mathtt{u}]\!]\Big)\ \mathrm{d}[\![\mathtt{t}]\!] \;=\; \int\Big(\int [\![\mathtt{v}]\!]\ \mathrm{d}[\![\mathtt{t}]\!]\Big)\ \mathrm{d}[\![\mathtt{u}]\!]$$

Fubini's theorem

Warning: Fubini's theorem does not hold for arbitrary kernels.

Fubini's theorem *does* hold for **s-finite** kernels.

s-finite kernels

Definition. A kernel $k : [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ is **bounded** if $\exists n.\ \forall \gamma.\ \forall U.\ k(\gamma, U) < n$.

see Börgstrom, Gordon et al., ESOP 2011

Definition. A kernel $k : [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ is **s-finite** if it is a countable sum of bounded kernels.

$$\int \Big(\int [\![\mathtt{v}]\!]\ \mathrm{d}[\![\mathtt{u}]\!] \Big)\ \mathrm{d}[\![\mathtt{t}]\!] \ = \ \int \Big(\int [\![\mathtt{v}]\!]\ \mathrm{d}[\![\mathtt{t}]\!] \Big)\ \mathrm{d}[\![\mathtt{u}]\!]$$

Fubini's theorem

Warning: Fubini's theorem does not hold for arbitrary kernels.

Fubini's theorem *does* hold for **s-finite** kernels.

s-finite kernels

Definition. A kernel $k : [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ is **bounded** if $\exists n.\ \forall \gamma.\ \forall U.\ k(\gamma, U) < n$.

Definition. A kernel $k : [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ is **s-finite** if it is a countable sum of bounded kernels.

Warning: Fubini's theorem does not hold for arbitrary kernels.

Fubini's theorem *does* hold for **s-finite** kernels.

Staton, ESOP 2017

s-finite kernels

Definition. A kernel $k : [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ is **bounded** if $\exists n.\ \forall \gamma.\ \forall U.\ k(\gamma, U) < n$.

Definition. A kernel $k : [\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$ is **s-finite** if it is a countable sum of bounded kernels.

Theorem. A kernel is s-finite if and only if it is definable.

Warning: Fubini's theorem does not hold for arbitrary kernels.

Fubini's theorem *does* hold for **s-finite** kernels.

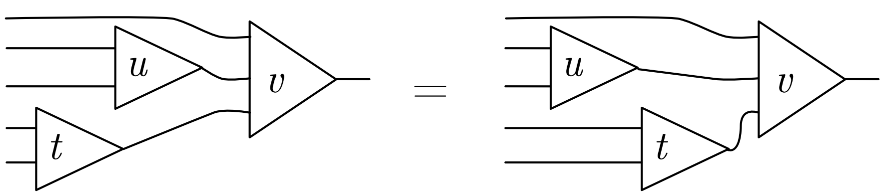
Staton, ESOP 2017

Commutativity

Theorem.

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

$$\int\!\Big(\int [\![\mathtt{v}]\!]\,\mathrm{d}[\![\mathtt{u}]\!]\Big)\,\mathrm{d}[\![\mathtt{t}]\!] = \int\!\Big(\int [\![\mathtt{v}]\!]\,\mathrm{d}[\![\mathtt{t}]\!]\Big)\,\mathrm{d}[\![\mathtt{u}]\!]$$



Fubini's theorem *does* hold for **s-finite** kernels.

A technical problem:

Defn. A kernel is a function $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0, \infty]$

such that for all $U \in \Sigma_{[\![\mathbb{A}]\!]}$, $[\![t]\!](-, U) : [\![\Gamma]\!] \to [0, \infty]$

is measurable and …

Sequencing is integration:

$$[\![\mathsf{let}\, x = t \,\mathsf{in}\, u]\!](\gamma, U) \stackrel{\text{def}}{=} \int [\![u]\!](\gamma, x, U)\ [\![t]\!](\gamma, \mathrm{d}x)$$

Is this measurable in γ?

Staton, ESOP 2017

A technical problem:

Defn. A kernel is a function $[\![\Gamma]\!] \times \Sigma_{[\![\mathbb{A}]\!]} \to [0,\infty]$

such that for all $U \in \Sigma_{[\![\mathbb{A}]\!]}$, $[\![t]\!](-,U) : [\![\Gamma]\!] \to [0,\infty]$

is measurable and …

Sequencing is integration:

$$[\![\text{let}\, x = t \,\text{in}\, u]\!](\gamma, U) \stackrel{\text{def}}{=} \int [\![u]\!](\gamma, x, U)\ [\![t]\!](\gamma, \mathrm{d}x)$$

Is this measurable in γ?

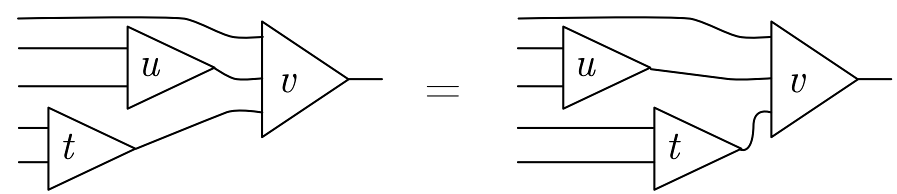
Answer: **yes** (if t and u are probabilistic programs,

they are s-finite kernels.)

Summary

Theorem.

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



- Probabilistic programs have a measure-theoretic semantics.
- Commutativity holds.
- Probabilistic programs are s-finite kernels.

Overview

- **Part 1:** Compositional probability theory; a compositional theory of impropriety
- **<u>Part 2:</u>** Higher order functions; random functions as measures on a space of functions.

Overview

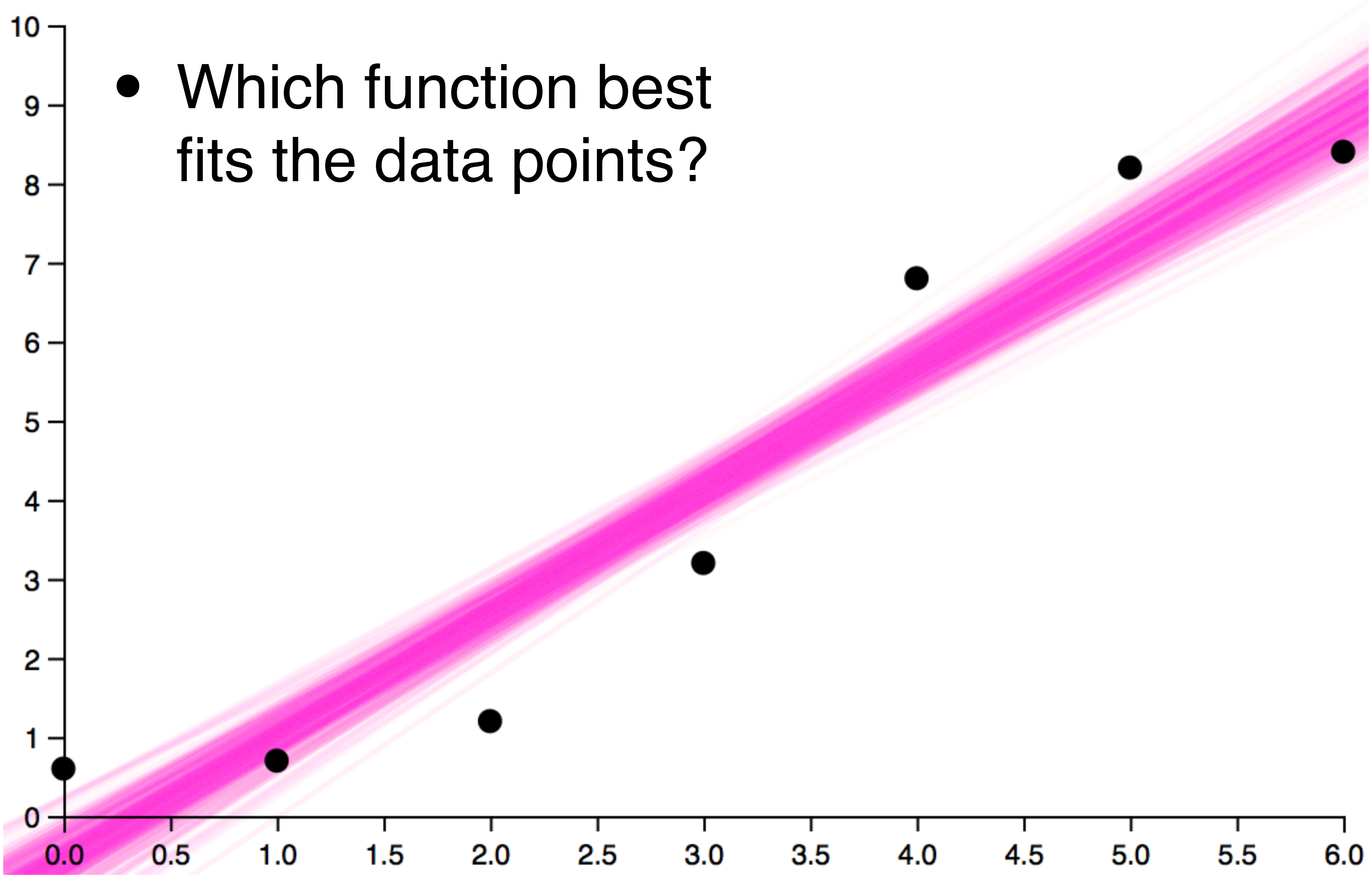
- **Summary of semantics**
- Examples
- s-finite kernels and
  proof of commutativity.

Overview of part 2

Higher order functions;
random functions as measures on a space of functions.

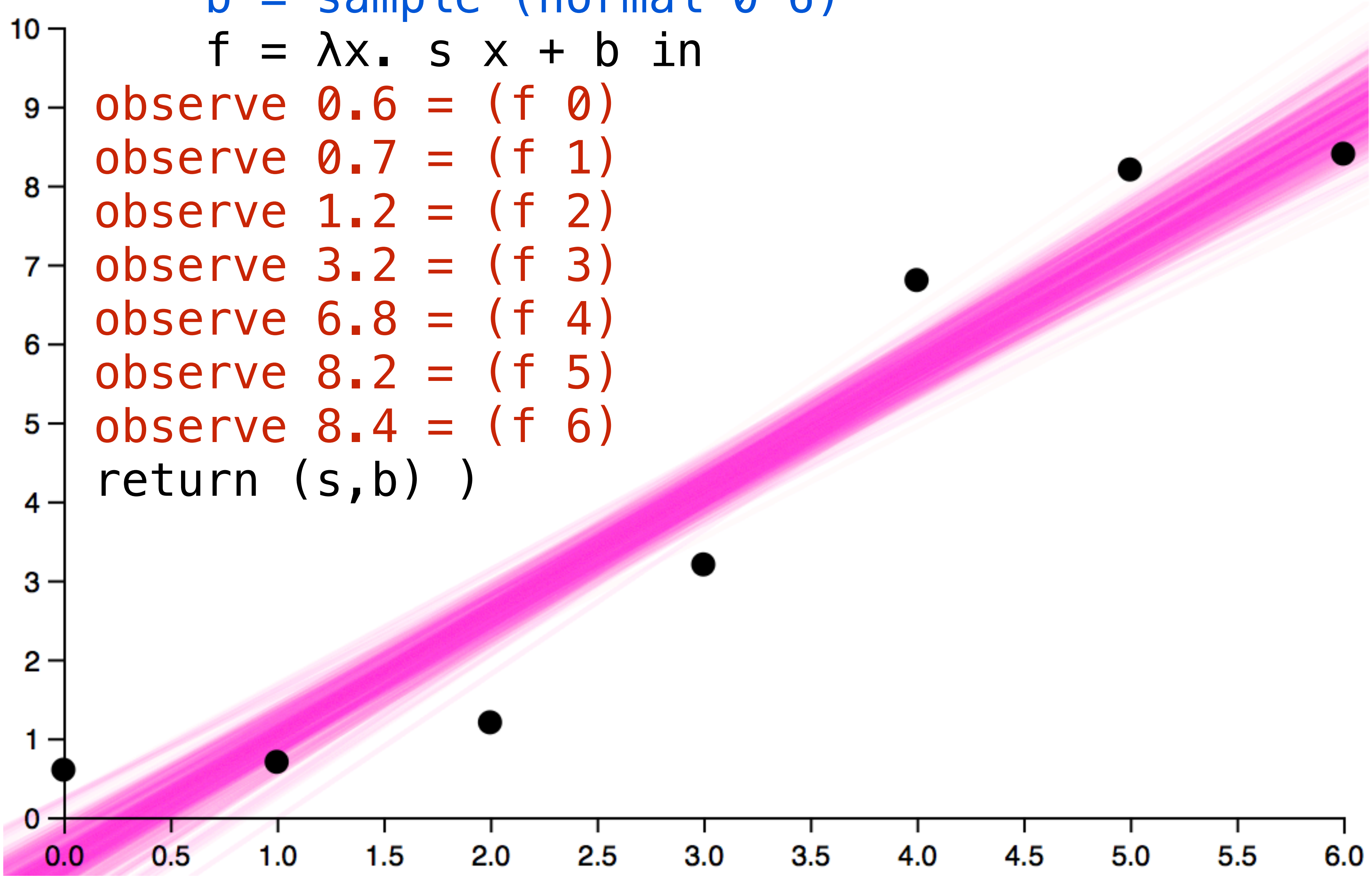
- Examples of regression and higher order functions
- Quasi-Borel spaces

Regression

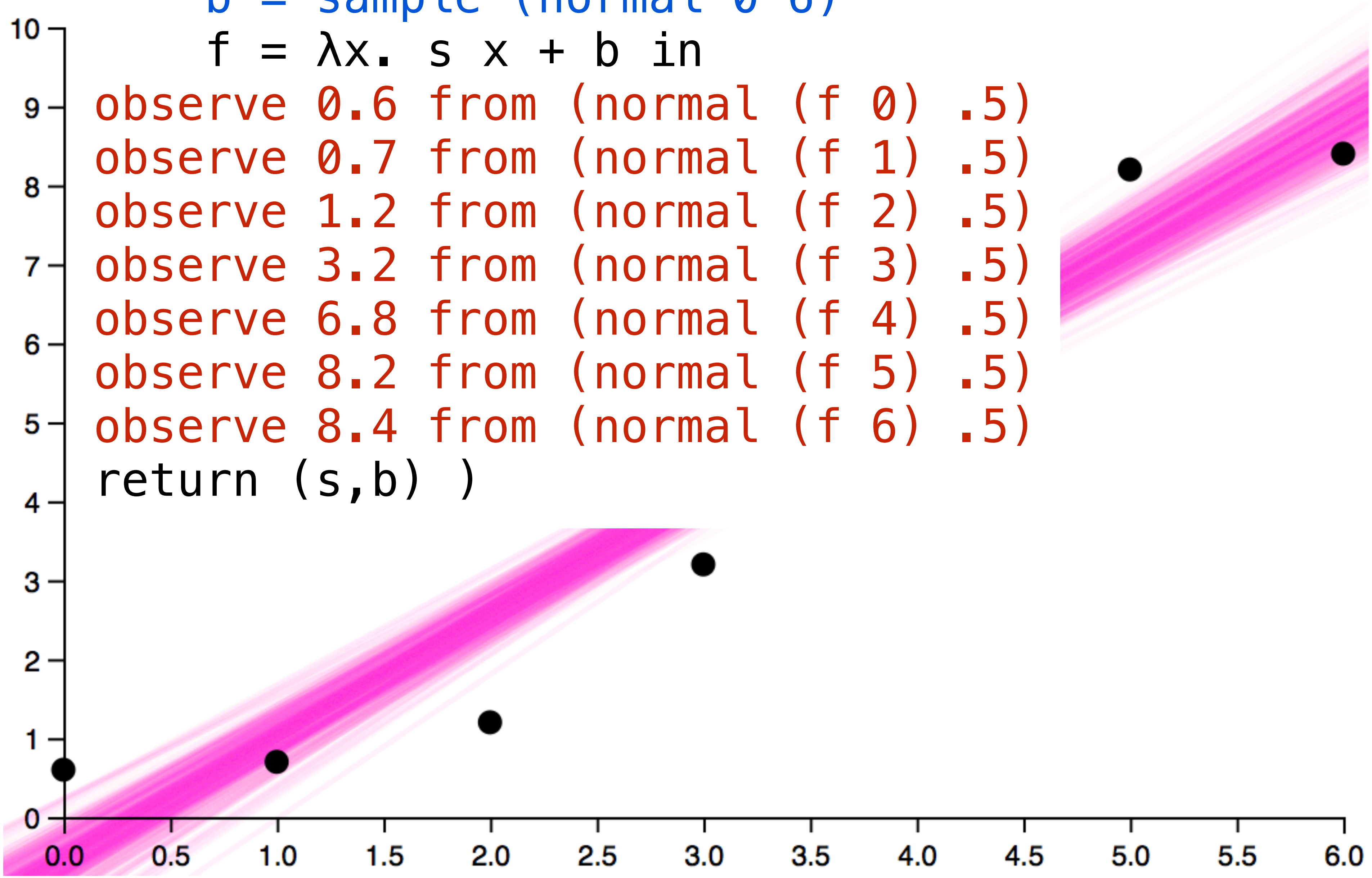


- Which function best fits the data points?

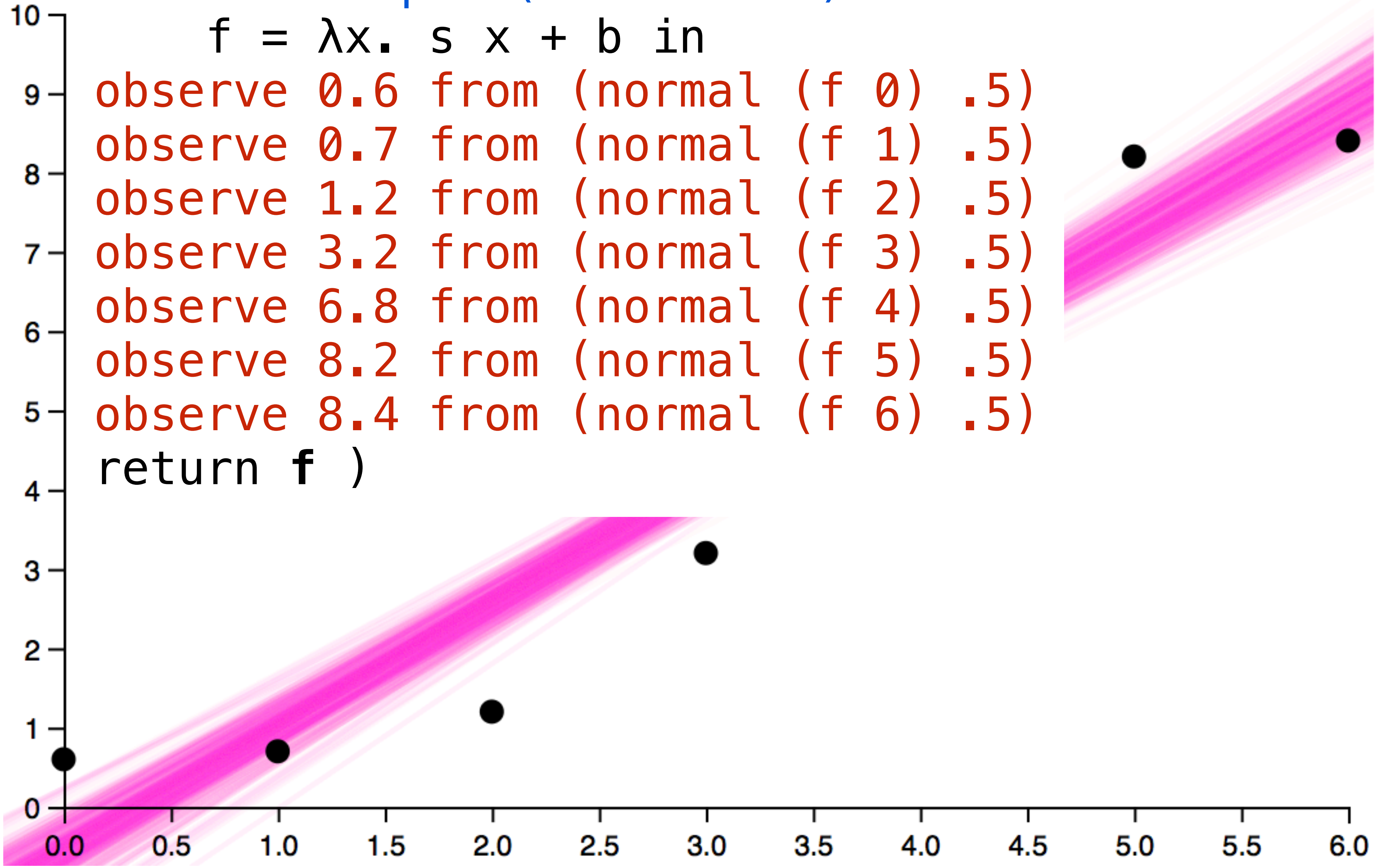
```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      f = λx. s x + b in
  observe 0.6 = (f 0)
  observe 0.7 = (f 1)
  observe 1.2 = (f 2)
  observe 3.2 = (f 3)
  observe 6.8 = (f 4)
  observe 8.2 = (f 5)
  observe 8.4 = (f 6)
  return (s,b) )
```



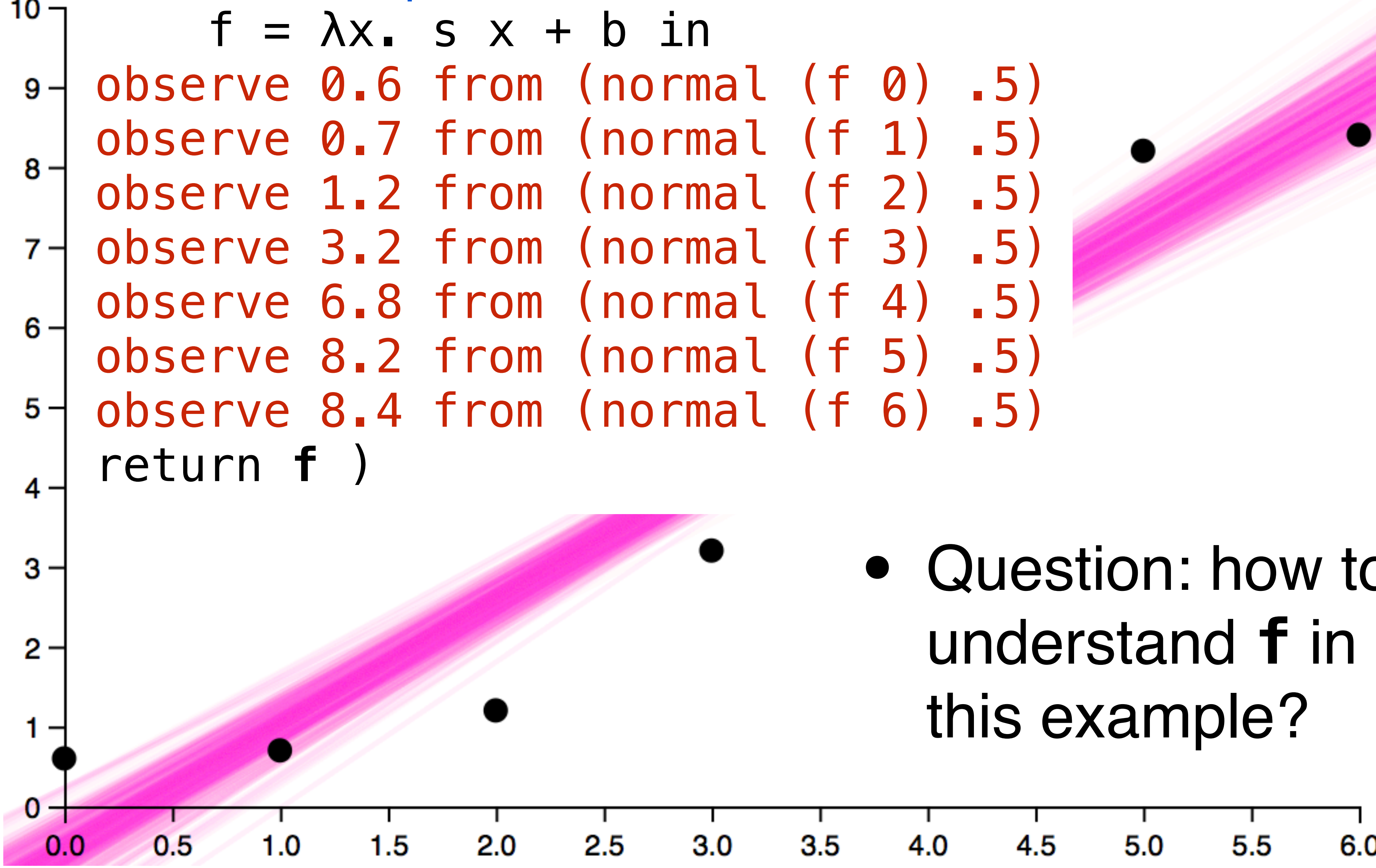
```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      f = λx. s x + b in
  observe 0.6 from (normal (f 0) .5)
  observe 0.7 from (normal (f 1) .5)
  observe 1.2 from (normal (f 2) .5)
  observe 3.2 from (normal (f 3) .5)
  observe 6.8 from (normal (f 4) .5)
  observe 8.2 from (normal (f 5) .5)
  observe 8.4 from (normal (f 6) .5)
  return (s,b) )
```



```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      f = λx. s x + b in
  observe 0.6 from (normal (f 0) .5)
  observe 0.7 from (normal (f 1) .5)
  observe 1.2 from (normal (f 2) .5)
  observe 3.2 from (normal (f 3) .5)
  observe 6.8 from (normal (f 4) .5)
  observe 8.2 from (normal (f 5) .5)
  observe 8.4 from (normal (f 6) .5)
  return f )
```

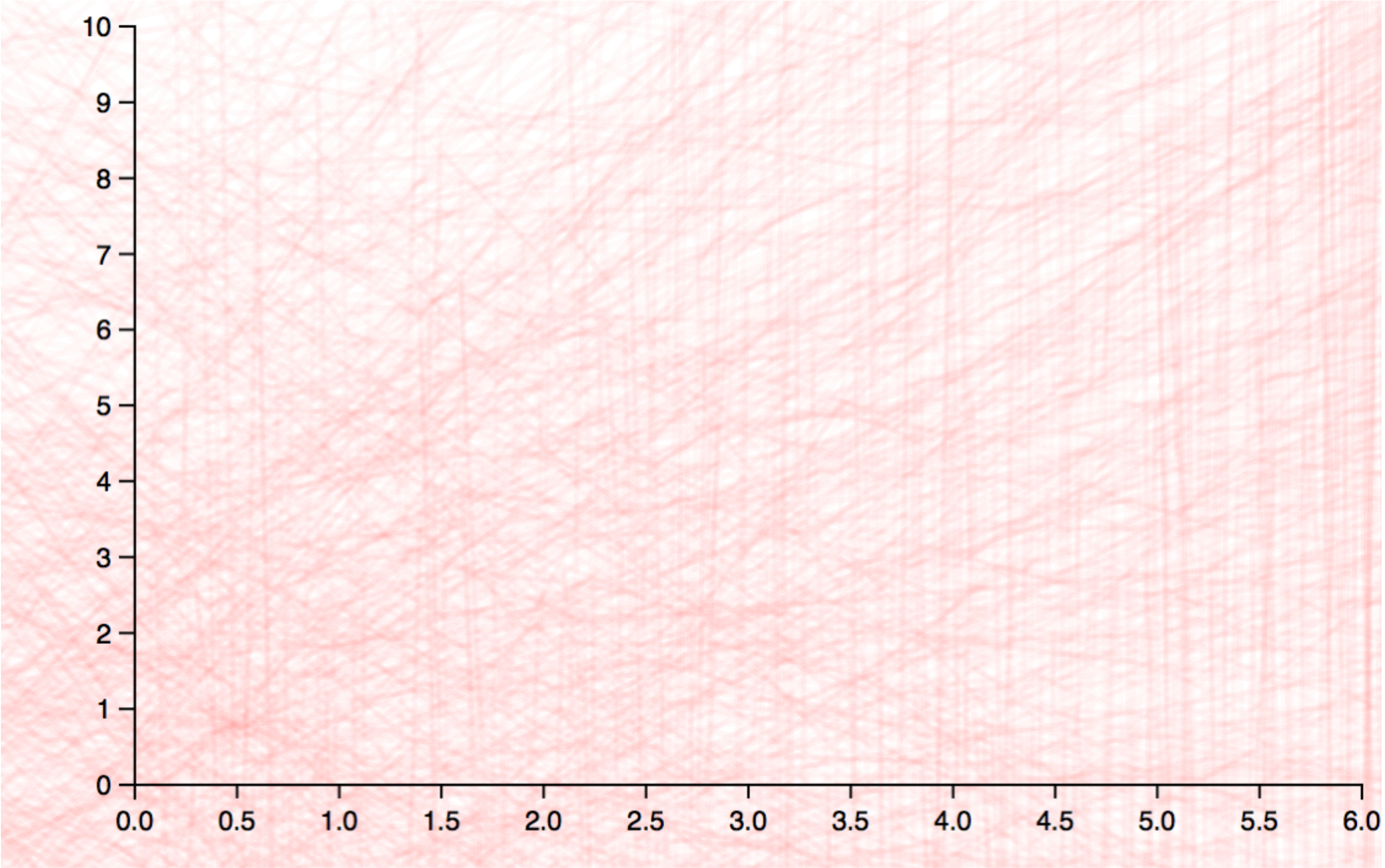


```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      f = λx. s x + b in
  observe 0.6 from (normal (f 0) .5)
  observe 0.7 from (normal (f 1) .5)
  observe 1.2 from (normal (f 2) .5)
  observe 3.2 from (normal (f 3) .5)
  observe 6.8 from (normal (f 4) .5)
  observe 8.2 from (normal (f 5) .5)
  observe 8.4 from (normal (f 6) .5)
  return f )
```



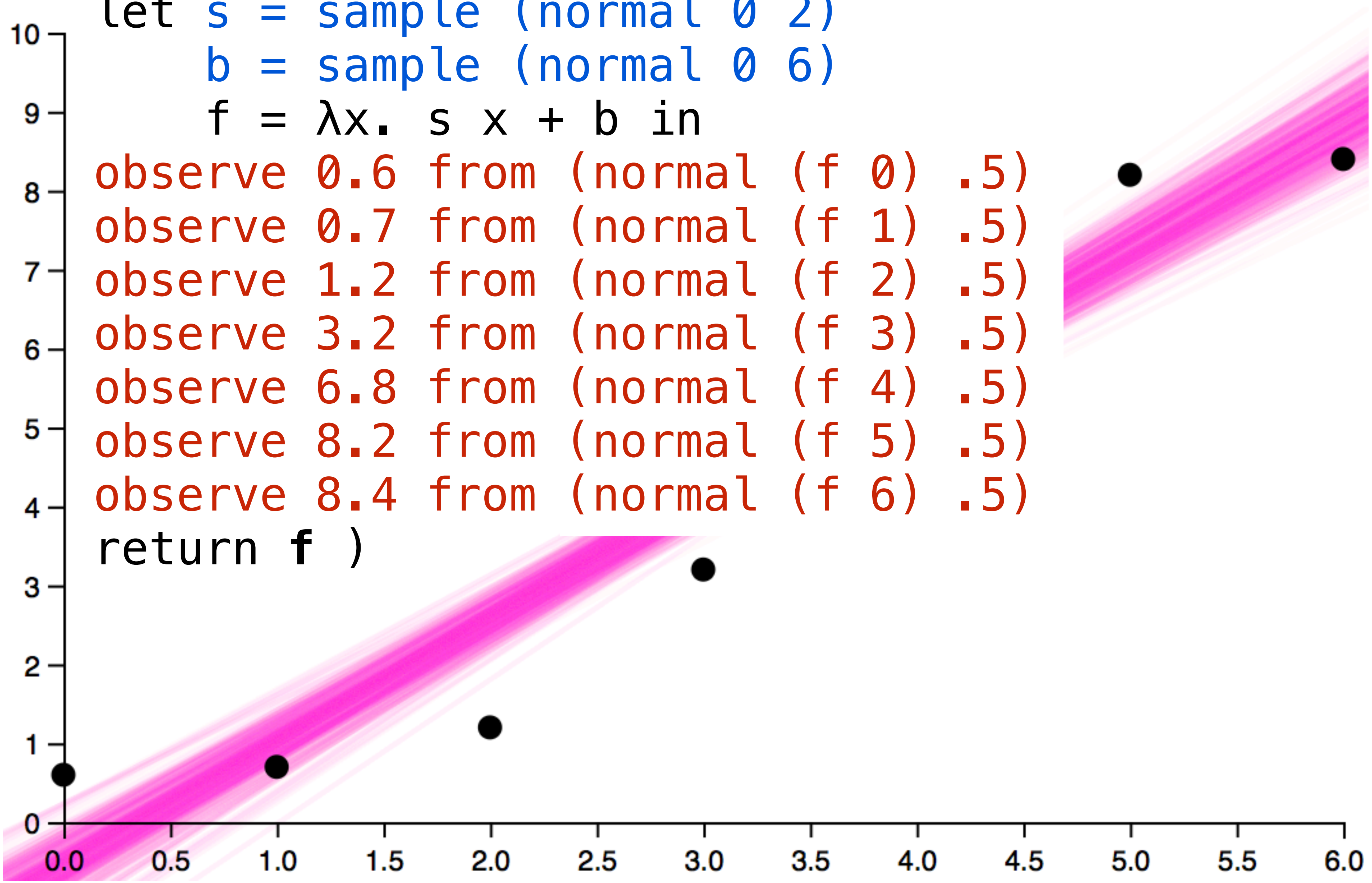
- Question: how to understand **f** in this example?

```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      f = λx. s x + b in
  return f )
```

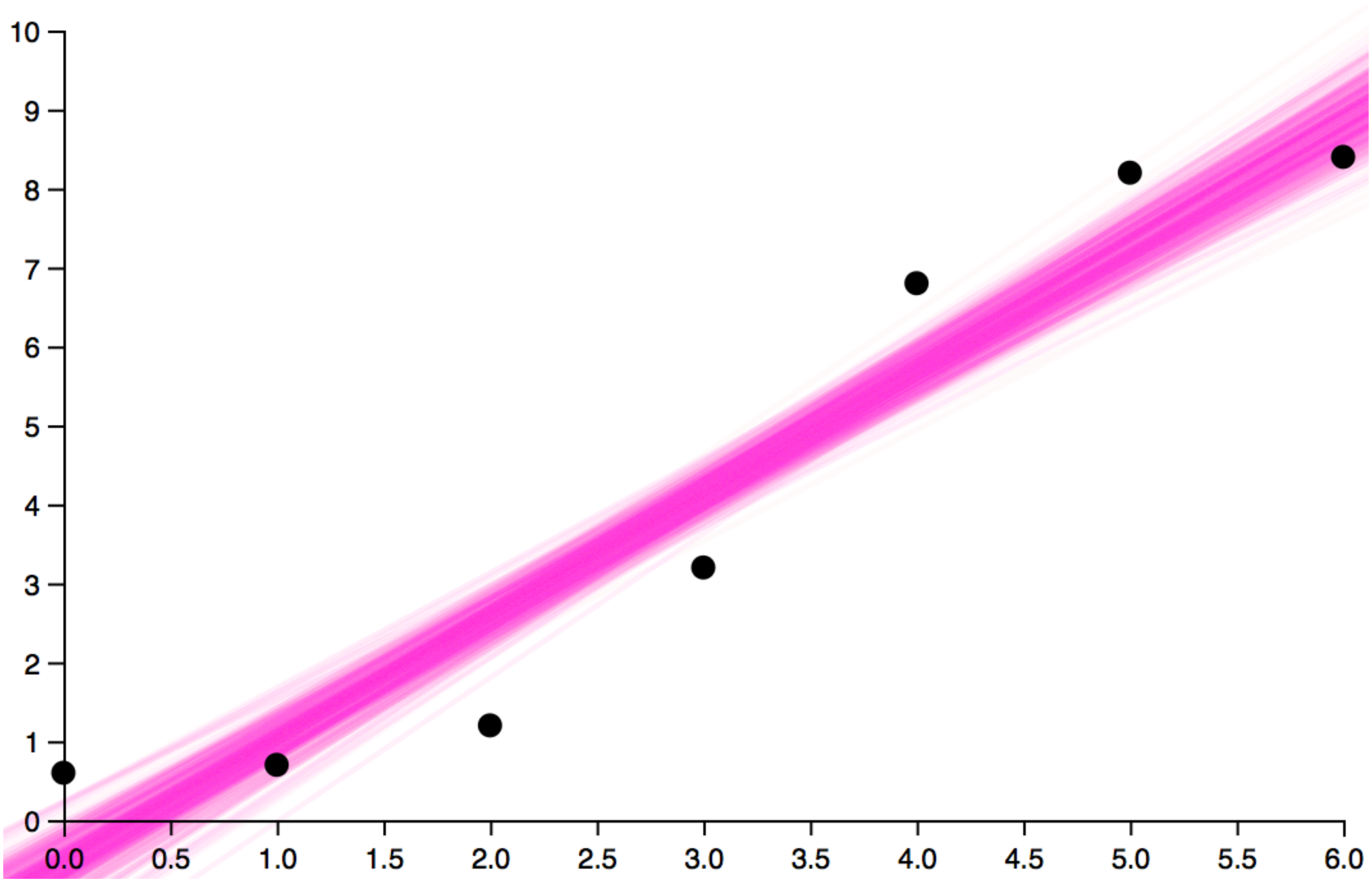


Samples from the prior

```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      f = λx. s x + b in
  observe 0.6 from (normal (f 0) .5)
  observe 0.7 from (normal (f 1) .5)
  observe 1.2 from (normal (f 2) .5)
  observe 3.2 from (normal (f 3) .5)
  observe 6.8 from (normal (f 4) .5)
  observe 8.2 from (normal (f 5) .5)
  observe 8.4 from (normal (f 6) .5)
  return f )
```



Samples from the posterior



Samples from the posterior

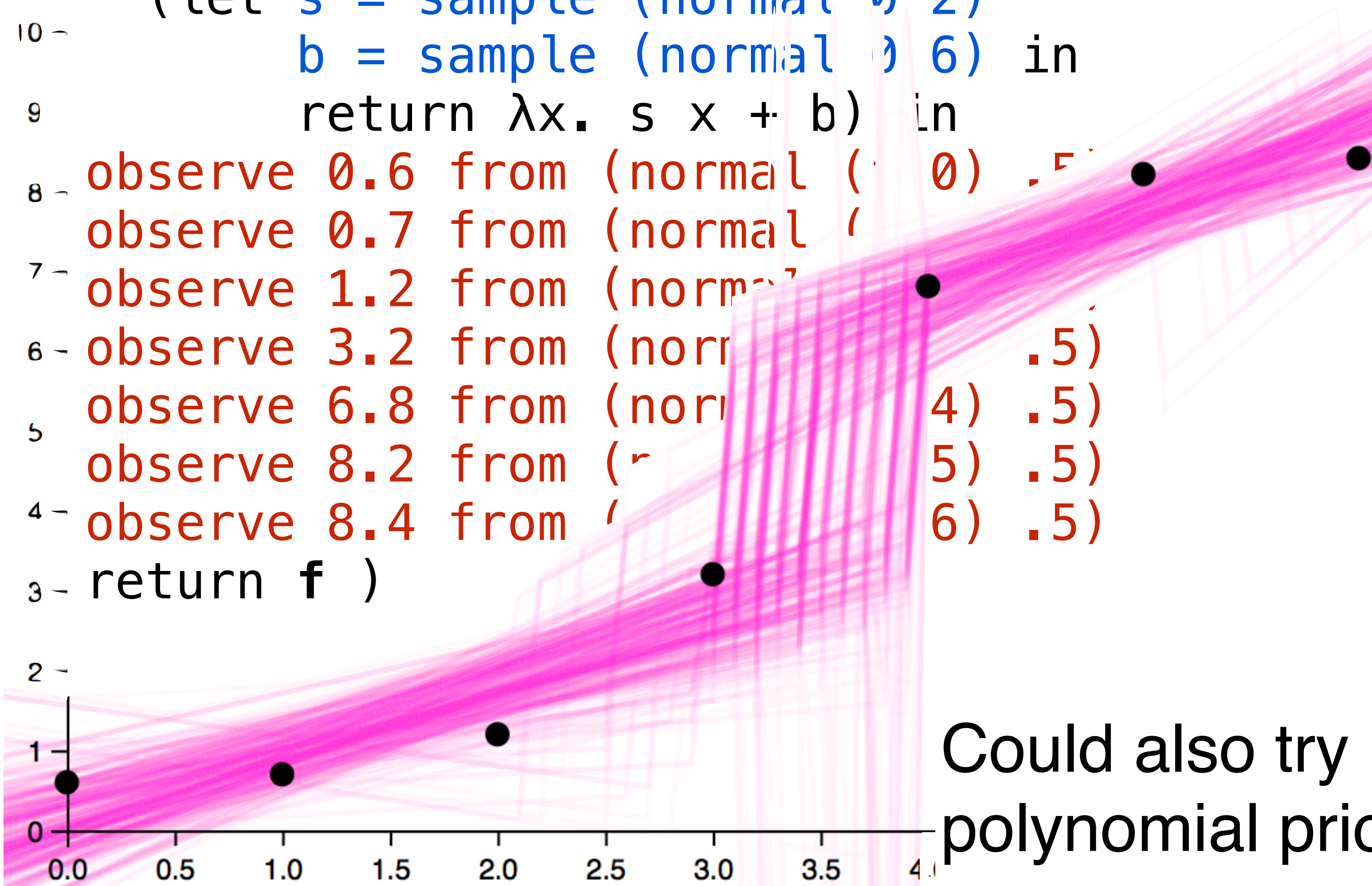
```
normalize(
  let f =
    (let s = sample (normal 0 2)
         b = sample (normal 0 6) in
         return λx. s x + b) in
  observe 0.6 from (normal (f 0) .5)
  observe 0.7 from (normal (f 1) .5)
  observe 1.2 from (normal (f 2) .5)
  observe 3.2 from (normal (f 3) .5)
  observe 6.8 from (normal (f 4) .5)
  observe 8.2 from (normal (f 5) .5)
  observe 8.4 from (normal (f 6) .5)
  return f )
```

More higher-order functions

```
normalize(
  let f = add-change-points
    (let s = sample (normal 0 2)
         b = sample (normal 0 6) in
         return λx. s x + b) in
  observe 0.6 from (normal (f 0) .5)
  observe 0.7 from (normal (f 1) .5)
  observe 1.2 from (normal (f 2) .5)
  observe 3.2 from (normal (f 3) .5)
  observe 6.8 from (normal (f 4) .5)
  observe 8.2 from (normal (f 5) .5)
  observe 8.4 from (normal (f 6) .5)
  return f )
```

Posterior

```
normalize(
  let f = add-change-points
    (let s = sample (normal 0 2)
         b = sample (normal 0 6) in
         return λx. s x + b) in
  observe 0.6 from (normal (  0) .5)
  observe 0.7 from (normal (
  observe 1.2 from (norm
  observe 3.2 from (nor           .5)
  observe 6.8 from (nor         4) .5)
  observe 8.2 from (            5) .5)
  observe 8.4 from (            6) .5)
  return f )
```



Could also try polynomial priors, or programs as priors.

Posterior

Technical problem

Measure theory doesn't support HO fns well.

$$\mathrm{ev} : (\mathbb{R} \to_{\mathrm{m}} \mathbb{R}) \times \mathbb{R} \to \mathbb{R}, \qquad \mathrm{ev}(f, x) = f(x).$$

[Aumann 61] ev is not measurable no matter which σ-algebra is used for $\mathbb{R} \to_{\mathrm{m}} \mathbb{R}$.

[Corollary] The category of measurable spaces is not cartesian closed.

Overview of part 2

Higher order functions;
random functions as measures on a space of functions.

- Examples of regression and higher order functions
- **Quasi-Borel spaces**

What about higher-order functions?

$$\mathbb{R} \to \mathbb{R}$$

*this is **not** a measurable space [Aumann 1961]*

Easy to deal with operationally. But denotationally?

Staton, Yang, Heunen, Kammar, Wood, LICS 2016
Börgstrom, Dal Lago, Gordon, Szymczak, ICFP 2016

What about function types?

Measurable spaces	$\xrightarrow{\text{embedding}}$	Bigger class of spaces
Models first order language with sample, score		Models **higher order** language with sample, score

Theorem. Adequacy.

What about function types?

Standard Borel spaces	embedding →	**Bigger class of spaces**

Models
first order
language with
sample,
score

Models
higher order
language with
sample,
score

Theorem. Adequacy.

What about function types?

Standard Borel spaces	$\xrightarrow{\text{embedding}}$	**Bigger class of spaces**

Models first order language with sample, score

Problem: $\mathbb{R}\to\mathbb{R}$ doesn't exist

i.e. $\mathrm{Hom}(-\times\mathbb{R}\ ,\ \mathbb{R}) : \mathbf{Meas}^{\mathrm{op}}\to\mathbf{Set}$

isn't representable

What about function types?

Staton, Yang, Heunen, Kammar, Wood, LICS 2016
See also Power, TCS 2006

$$\boxed{\textbf{Standard Borel spaces}} \xrightarrow[\text{embedding}]{\text{Yoneda}} \boxed{\textbf{Sheaves on measurable spaces}}$$

Models
first order
language with
sample,
score

What about function types?

Staton, Yang, Heunen, Kammar, Wood, LICS 2016
see also Power, TCS 2006

Standard Borel spaces	$\xrightarrow[\text{embedding}]{\text{Yoneda}}$	Sheaves on measurable spaces

Models first order language with sample, score

(Functors $\mathbf{Meas}^{\mathrm{op}} \to \mathbf{Set}$ that preserve countable products)

Staton, Yang, Heunen, Kammar, Wood, LICS 2016
see also Power, TCS 2006

What about function types?

Standard Borel spaces —Yoneda embedding→ **Sheaves on measurable spaces**

‘Quasi-Borel spaces’

Models first order language with sample, score

What about function types?

Staton, Yang, Heunen, Kammar, Wood, LICS 2016
see also Power, TCS 2006

Standard Borel spaces

Yoneda embedding

Sheaves on measurable spaces

Models first order language with sample, score

'Quasi-Borel spaces'

"Random elements first."

Random elements α in X

$$\alpha : \Omega \to X$$

- X - set of values.
- Ω - set of random seeds.
- Random seed generator.

Random elements α in X
in classical measure theory

$$\alpha : \Omega \to X$$

is a random element
if $\alpha^{-1}(A) \in \Sigma$ for all $A \in \Theta$

- X - set of values.
- Ω - set of random seeds.
- Random seed generator.

1. $\Sigma \subseteq 2^{\Omega}$, $\Theta \subseteq 2^{X}$
2. $\mu : \Sigma \to [0,1]$

Random elements α in X
in quasi-Borel spaces

$$\alpha : \Omega \to X$$

- X - set of values.
- $\Omega = \mathbb{R}$ - set of random seeds.
- Random seed generator.

Random elements α in X
in quasi-Borel spaces

$$\alpha : \Omega \to X$$

is a random variable
if $\alpha \in M$

- X - set of values.
- $\Omega = \mathbb{R}$ - set of random seeds.
- Random seed generator.

1. $\mathbb{R}$ as random source
2. Borel subsets $\mathfrak{B} \subseteq 2^{\mathbb{R}}$
3. $M \subseteq [\mathbb{R} \to X]$

Heunen, Kammar, Staton, Yang, LICS 2017

A ccc of measurable functions

Defn. A quasi-Borel space is a pair (X,M) where

- X is a set
- $M \subseteq [\mathbb{R} \to X]$ s.t.

Heunen, Kammar, Staton, Yang, LICS 2017

A ccc of measurable functions

Defn. A quasi-Borel space is a pair (X,M) where

- X is a set
- $M \subseteq [\mathbb{R} \to X]$

such that

- if $f : \mathbb{R} \to \mathbb{R}$ measurable and $g \in M$ then $gf \in M$.
- piecewise combination: if $\mathbb{R} = \uplus_{i \in \mathbb{N}} R_i$ with R_i Borel and $\alpha_1, \alpha_2, \ldots \in M$, then $\uplus_{i \in \mathbb{N}} (\alpha_i \cap (R_i \times X)) \in M$.
- all constant functions are in M

A morphism $(X,M) \to (Y,N)$ is a function $f : X \to Y$ such that $g \in M$ implies $fg \in N$

A ccc of measurable functions

Heunen, Kammar, Staton, Yang, LICS 2017

Defn. A quasi-Borel space is a pair (X,M) where

- X is a set
- $M \subseteq [\mathbb{R}\to X]$ s.t. ...

Example: X is a standard Borel measurable space, $M \subseteq [\mathbb{R}\to X]$ comprises the measurable functions.

Then 'morphism' = 'measurable function'.

A morphism $(X,M) \to (Y,N)$ is a function $f : X \to Y$ such that $g \in M$ implies $fg \in N$

Heunen, Kammar, Staton, Yang, LICS 2017

A ccc of measurable functions

Defn. A quasi-Borel space is a pair (X,M) where

- X is a set
- $M \subseteq [\mathbb{R} \to X]$ s.t. ...

Example: X is a standard Borel measurable space, $M \subseteq [\mathbb{R} \to X]$ comprises the measurable functions.

Proposition. The category of quasi-Borel spaces is cartesian closed with countable sums.

Corollary. If a term t has first order type then $[\![t]\!]$ is a measurable function.

Heunen, Kammar, Staton, Yang, LICS 2017

Standard Borel spaces $\longrightarrow$ **'Quasi-Borel spaces'**

What is a measure on $X \to Y$?

Heunen, Kammar, Staton, Yang, LICS 2017

Standard Borel spaces	$\longrightarrow$	'Quasi-Borel spaces'

Defn. A **quasi-Borel space** is a pair (X,M) where

- X is a set
- $M \subseteq [\mathbb{R} \to X]$ s.t. ...

Defn. A **measure** on a quasi-Borel space is a pair

$(\mu\ ,\ f)$

a measure on $\mathbb{R}$

a f^n $f : \mathbb{R} \to X$ in M

(modulo inducing the same integration operator)

Heunen, Kammar, Staton, Yang, LICS 2017

Standard Borel spaces $\longrightarrow$ **'Quasi-Borel spaces'**

Proposition. A measure on $[X \to Y]$ is a pair

$$(\mu\ ,\ f)$$

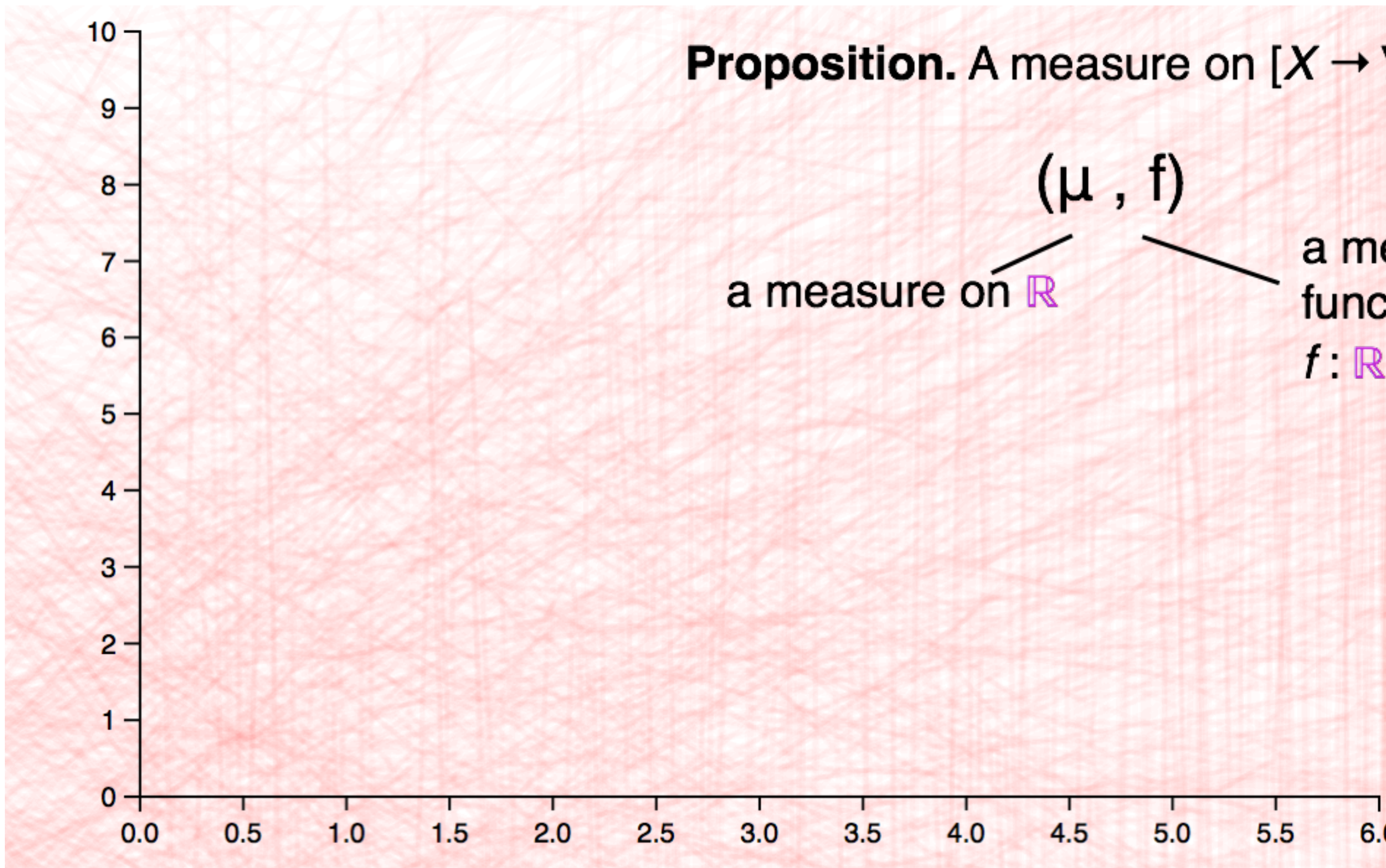
a measure on $\mathbb{R}$

a measurable function $f : \mathbb{R} \times X \to Y$

– a 'random function'.

Heunen, Kammar, Staton, Yang, LICS 2017

```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      g = λx. s x + b in
  return g )
```



Proposition. A measure on $[X \to Y]$ is a pair

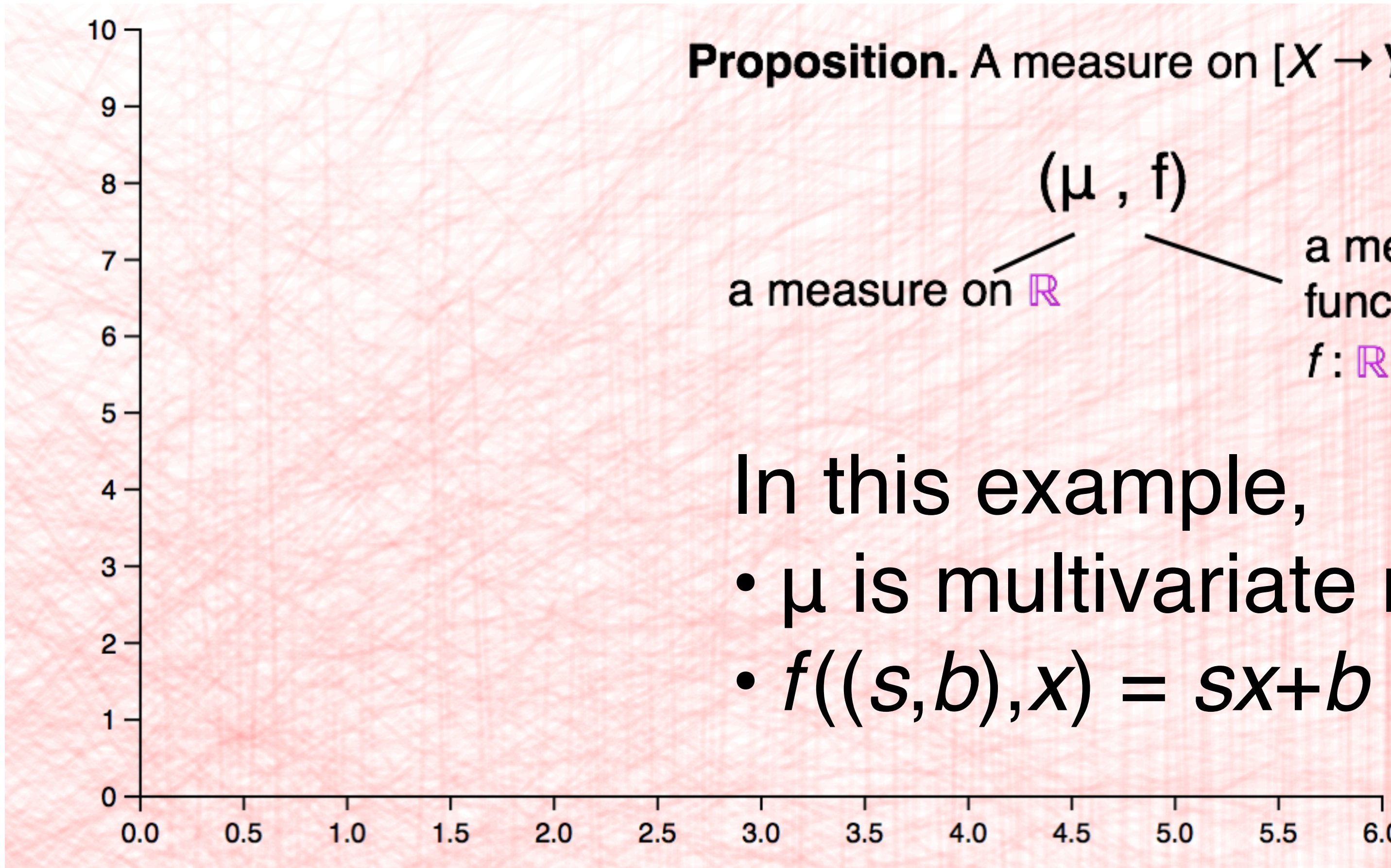
$(\mu\ ,\ f)$

a measure on $\mathbb{R}$

a measurable function $f : \mathbb{R} \times X \to Y$

Heunen, Kammar, Staton, Yang, LICS 2017

```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      g = λx. s x + b in
  return g )
```



Proposition. A measure on $[X \to Y]$ is a pair

$$(\mu\ ,\ f)$$

a measure on $\mathbb{R}$

a measurable function $f : \mathbb{R} \times X \to Y$

In this example,

- μ is multivariate normal,
- $f((s,b),x) = sx+b$

NB $\mathbb{R}\cong\mathbb{R}\times\mathbb{R}$

Quasi-Borel spaces work well:

- Simple theorems e.g. “randomization lemmas” can be stated in the internal logic;
- A version of de Finetti’s theorem for exchangeable sequences holds;
- One can justify higher order inference algorithms, (Metropolis-Hastings, Sequential Monte Carlo) jww Scibior, Vakar, Kai, Ostermann, Moss, Ghahramani

Motivation

What is a general notion of statistical model?

How can we understand the fundamental structure?

Aims of semantic models:

1. Fundamental;
2. Towards applications in Machine Learning.

What's next:

We're now looking at
"exchangeable random structures"
towards a general theory of Aldous-Hoover results and the connection with abstract types.

- Discussions with Ackerman, Freer, Roy, Bloom-Reddy.

Aims of semantic models:

1. Fundamental;
2. Towards applications in Machine Learning.

Overview

- **Part 1:** Compositional probability theory; a compositional theory of impropriety
- **Part 2:** Higher order functions; random functions as measures on a space of functions.