Semantic models of higher order Bayesian inference

Sam Staton, Oxford

based on:

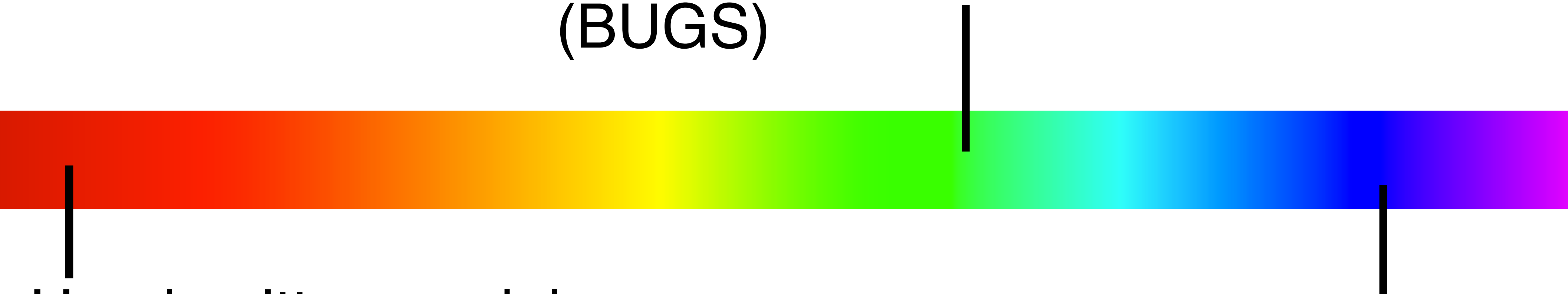
ESOP 2017. *Sam Staton.*

LICS 2017: *Chris Heunen, Ohad Kammar, Sam Staton, Hongseok Yang.*

POPL 2018: Adam *Scibior, Ohad Kammar, Matthijs Vákár, Sam Staton, Hongseok Yang, Yufei Cai, Klaus Ostermann, Sean Moss, Chris Heunen, Zoubin Ghahramani.*

A spectrum of modelling methods

Restricted PPLs as interfaces to particular inference algorithms (BUGS)



Hand-written models & inference for particular problems (e.g. R package)

General purpose PPLs with interchangeable inference algorithms (e.g. Anglican)

Motivation

What is a semantic foundation for probabilistic programming?

How can it help us with:

- expressivity of languages?
- validity of inference algorithms?
- validity & meaning of programs/models?

Overview

- **Part 1:** Illustrations of key ideas.
  - **Simple example,** semantic approaches
  - Bayesian regression and h.o. functions
- **Part 2:** From new foundations to modular and valid inference algorithms.
- **Part 3:** What next?

Probabilistic programming

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

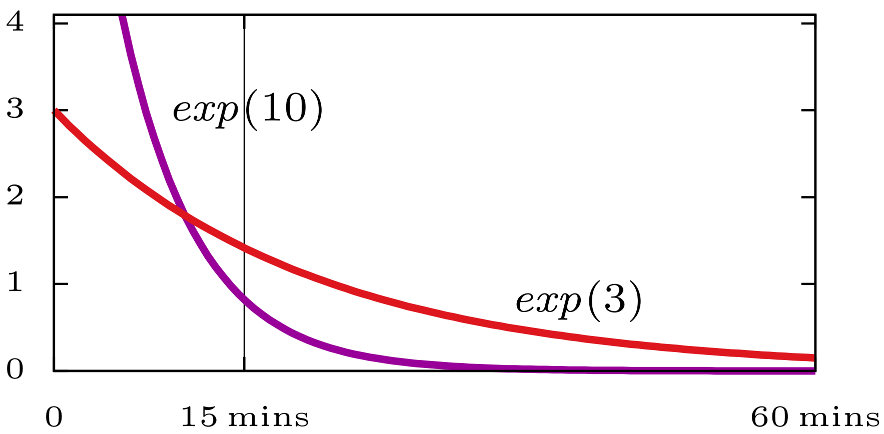
$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

probabilistic programming =
sequential programming +

normalize **observe** **sample**

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) )
```

Overview

- **Part 1:** Illustrations of key ideas.
  - Simple example, **semantic approaches**
  - Bayesian regression and h.o. functions
- **Part 2:** From new foundations to modular and valid inference algorithms.
- **Part 3:** What next?

Staton, ESOP 2017
Kozen 1981

What do Prob. Prog's mean?

The meaning of a program of type A
is a measure on A (typically unnormalized).

```
let x = sample(μ) in k(x)
```

means $\int k(x)\ \mu(dx)$

```
observe x from μ ; k
```

means $\mathrm{pdf}_{\mu}(x) \times k$

```
normalize(k)
```

means $k/(\int k)$

Staton, ESOP 2017

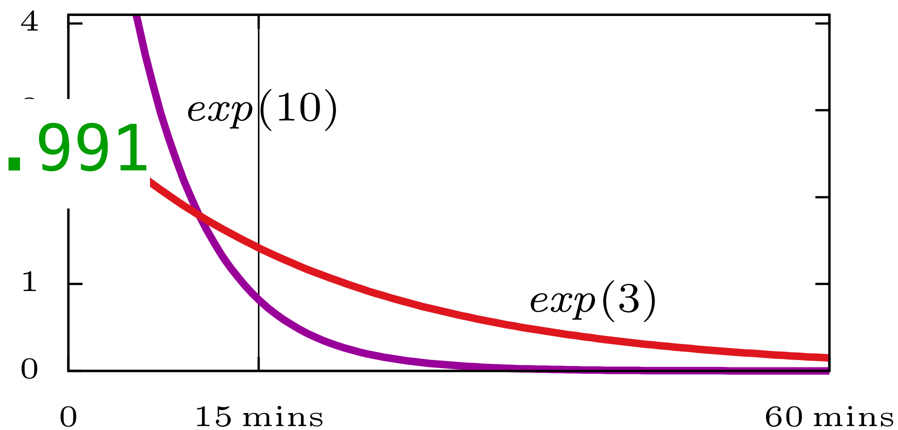
Example

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) )
```

Example

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?

Unnormalized posterior:

$$m(U) = \int_U f(t)\, e^{-0.25 f(t)} / 24 \, dt$$

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

How do we run prob prog's?

Monte Carlo simulation:

1. run many times;
2. each run gives a result & importance weight

`sample(μ)` *means* sample from μ, then run k

`observe x from μ` *means* multiply current weight by $\mathrm{pdf}_{\mu}(\mathbf{x})$

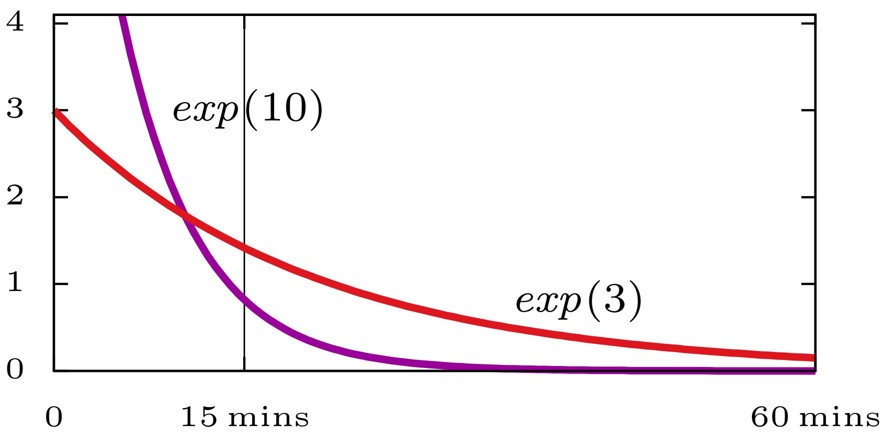
Normalization constant = average weight

Example

Two possible traces:

```
weekend=true (prob 2/7)
weight=1.42
result=true

weekend=false (prob 5/7)
weight=0.82
result=false
```




```
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)
```

How do we run prob prog's?

Monte Carlo simulation:

1. run many times;
2. each run gives a result & importance weight

Concerns:

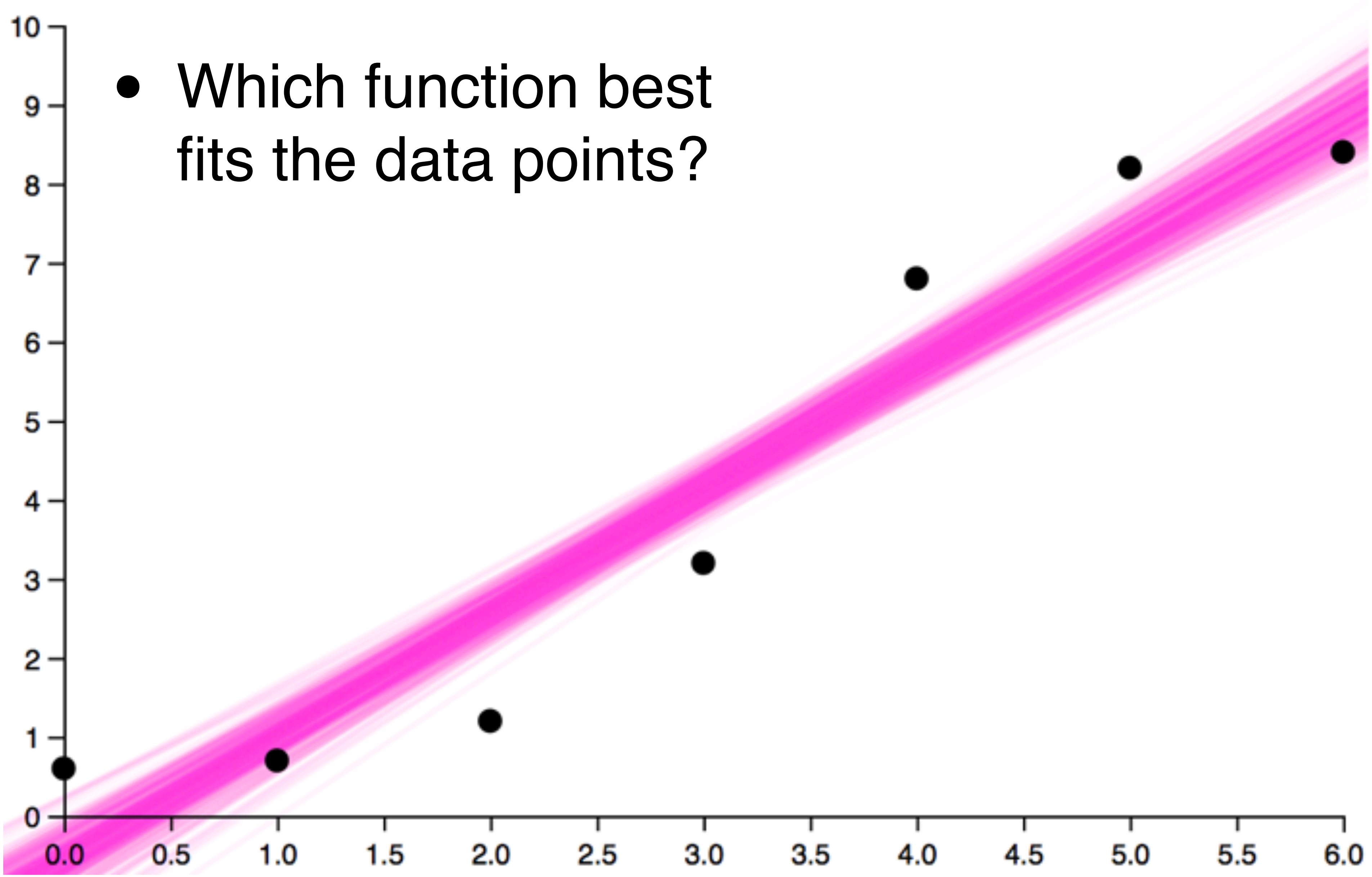
1. too much time spent on low weight traces
   solution: SMC (sequential Monte Carlo)
2. resampling everything each time is costly
   solution: MCMC, MHG

Overview

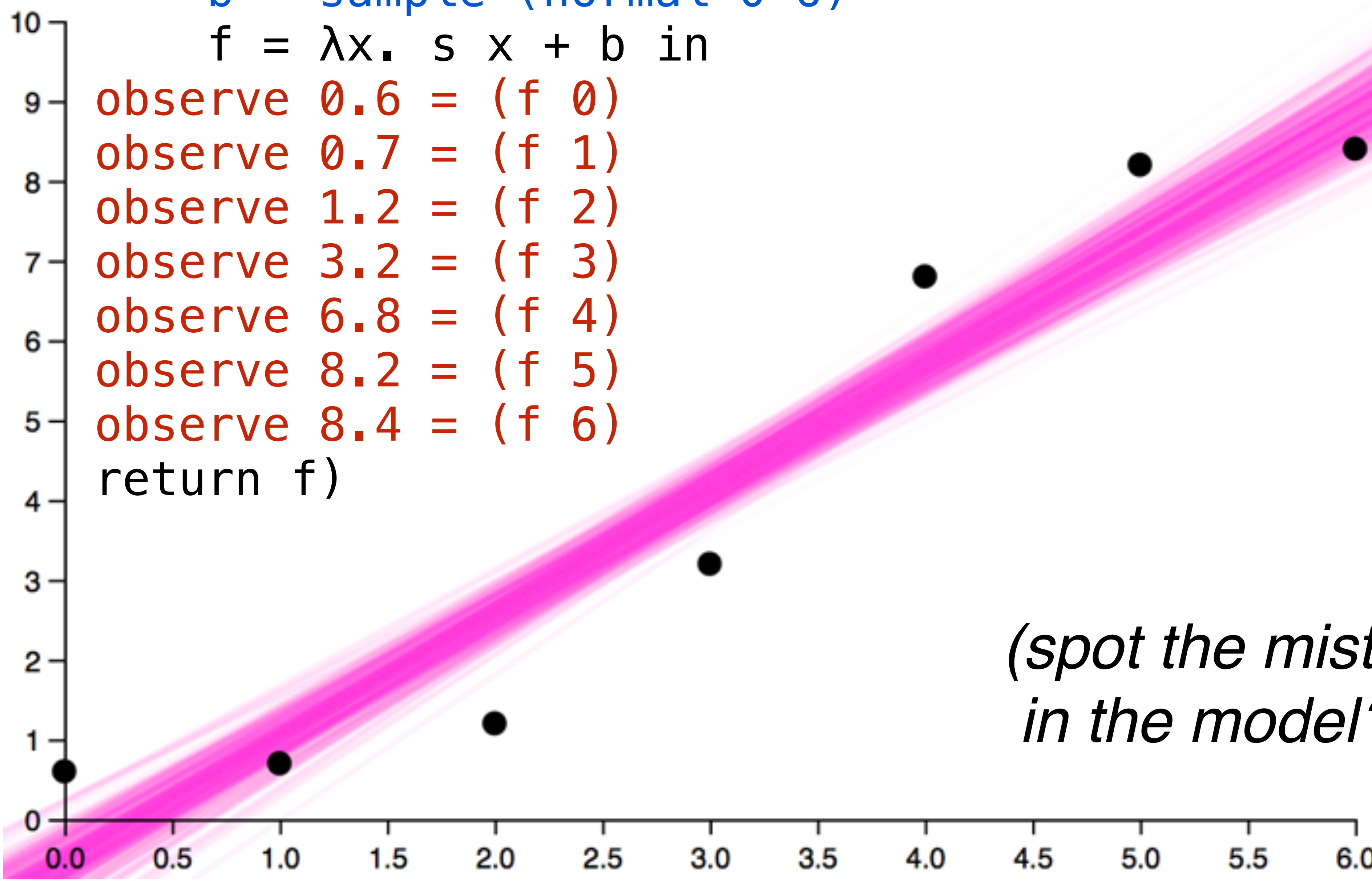
- **Part 1:** Illustrations of key ideas.
  - Simple example, semantic approaches
  - **Bayesian regression and h.o. functions**
- **Part 2:** From new foundations to modular and valid inference algorithms.
- **Part 3:** What next?

Bayesian regression

- Which function best fits the data points?

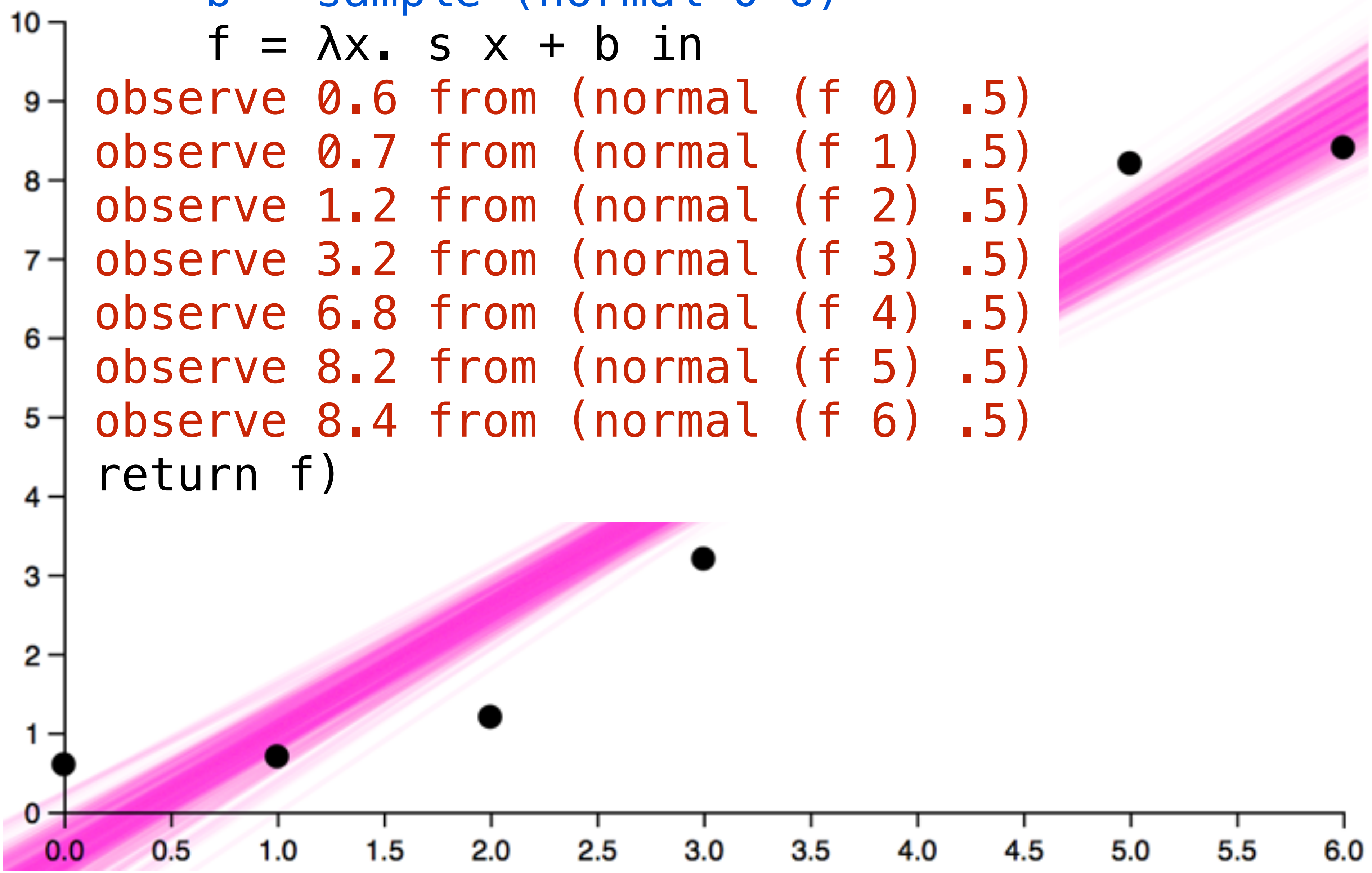


```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      f = λx. s x + b in
  observe 0.6 = (f 0)
  observe 0.7 = (f 1)
  observe 1.2 = (f 2)
  observe 3.2 = (f 3)
  observe 6.8 = (f 4)
  observe 8.2 = (f 5)
  observe 8.4 = (f 6)
  return f)
```

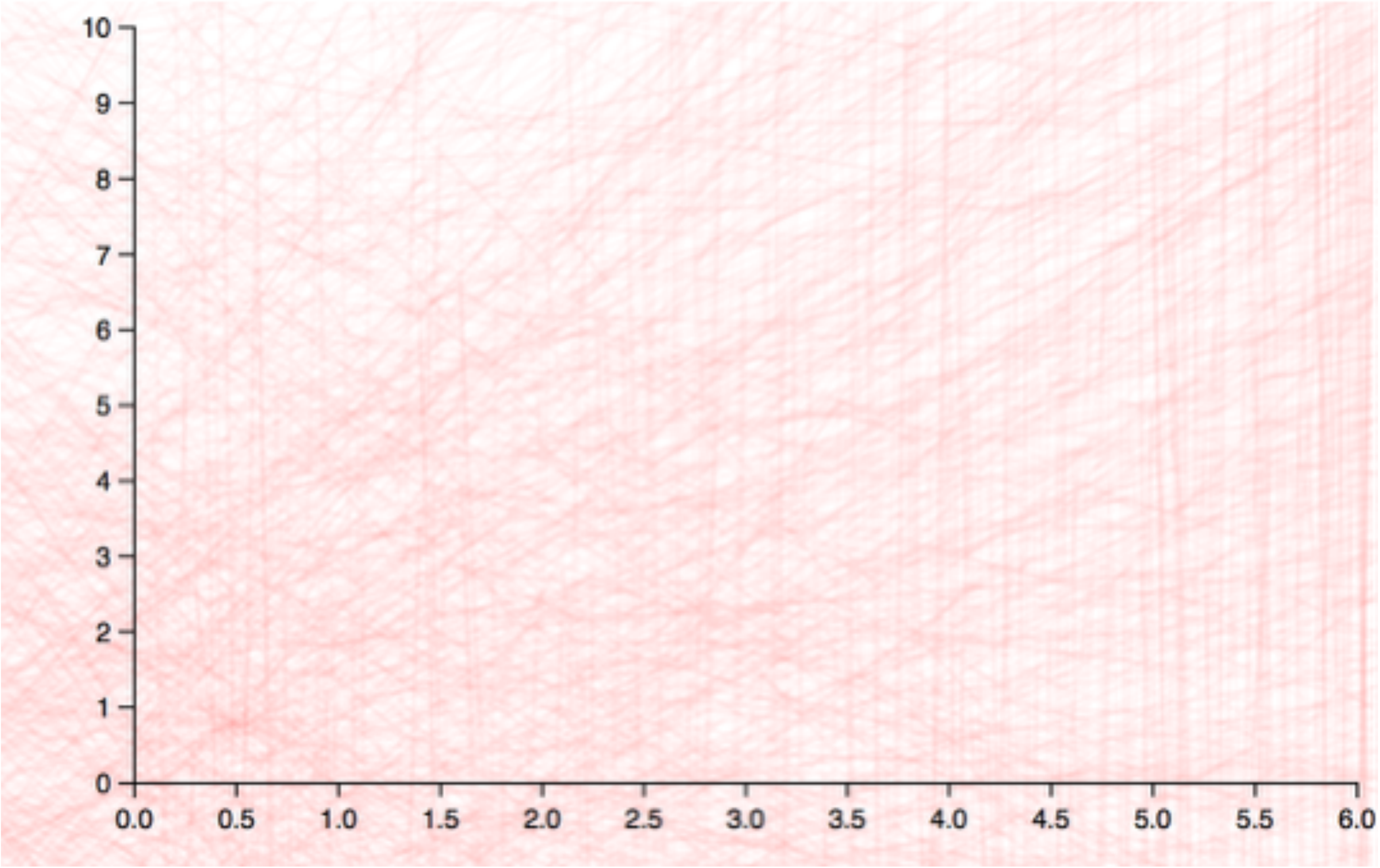


(spot the mistake in the model?)

```
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)
```

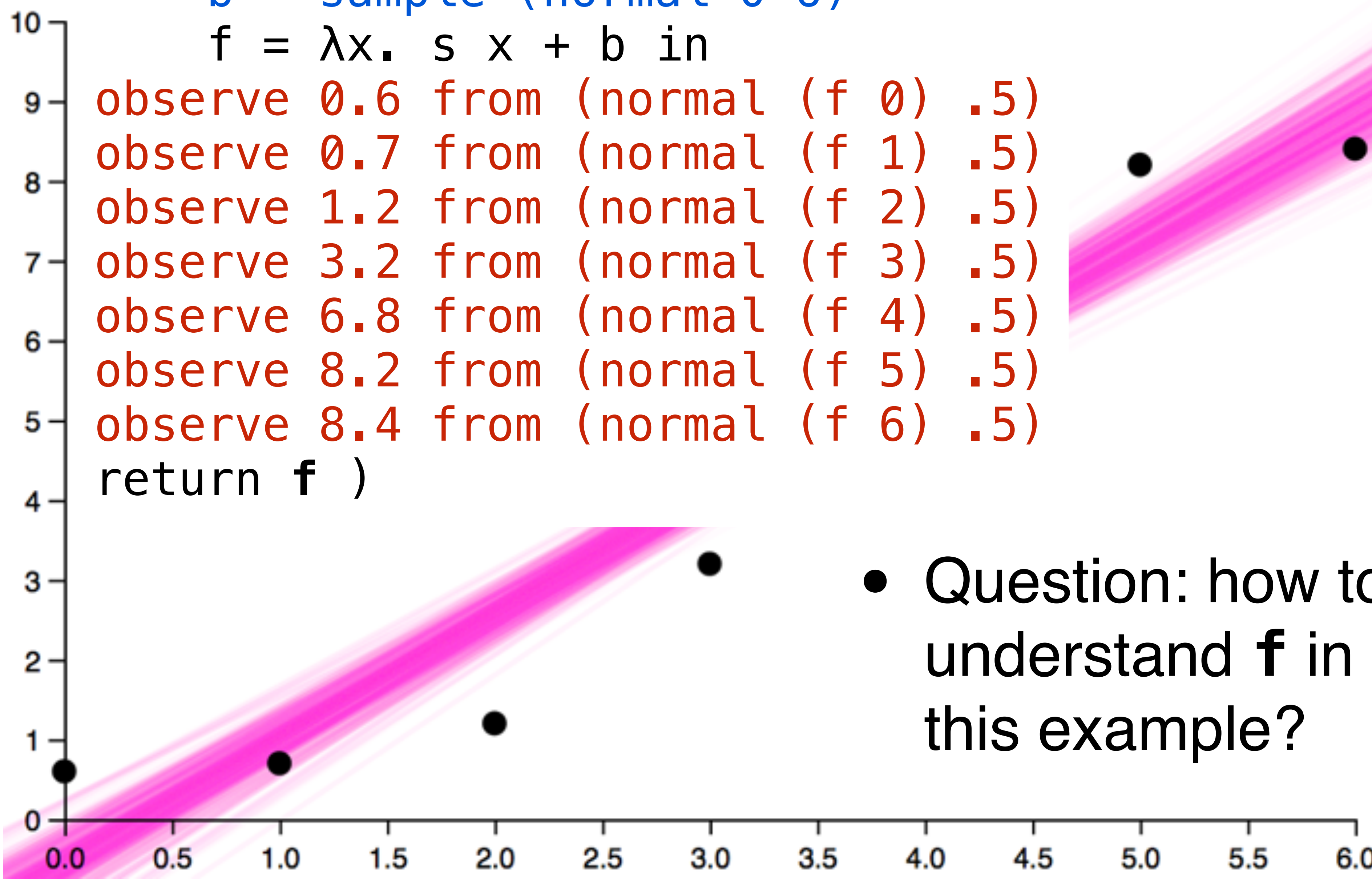


```
normalize(
  let s = sample (normal 0 2)
      b = sample (normal 0 6)
      f = λx. s x + b in
  return f )
```



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 )
```



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

Technical problem

Measure theory doesn't support HO fns well.

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

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

Corollary. Measurable spaces don't fully support higher order functions. *(Not Cartesian closed.)*

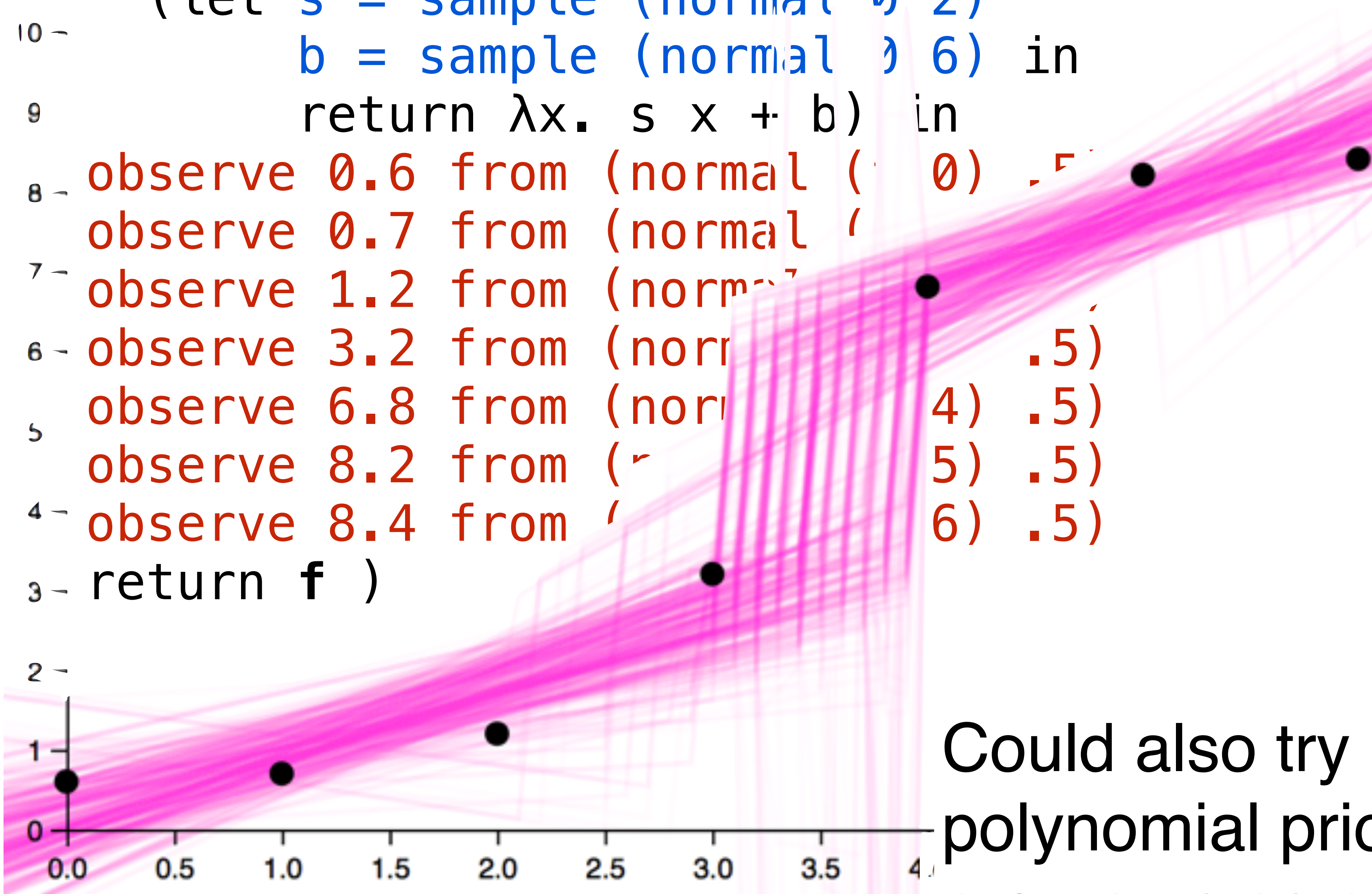
```
normalize(
  let f =
    (let s = sample (normal 0 2)
         b = sample (normal 0 6) in
         return λx. s x + b) in
  observe 0.6 from (normal (f 0) .5)
  observe 0.7 from (normal (f 1) .5)
  observe 1.2 from (normal (f 2) .5)
  observe 3.2 from (normal (f 3) .5)
  observe 6.8 from (normal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

Piecewise linear functions

```
normalize(
  let f = piecewise
    (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 )
```



Posterior

Could also try polynomial priors, or programs as priors.

Motivation

What is a semantic foundation for probabilistic programming?

How can it help us with:

- expressivity of languages?
- validity of inference algorithms?
- validity & meaning of programs/models?

Overview

- **Part 1:** Illustrations of key ideas.
- **Part 2: From new foundations to modular and valid inference algorithms.**
- **Part 3:** What next?

How do we run prob prog’s?

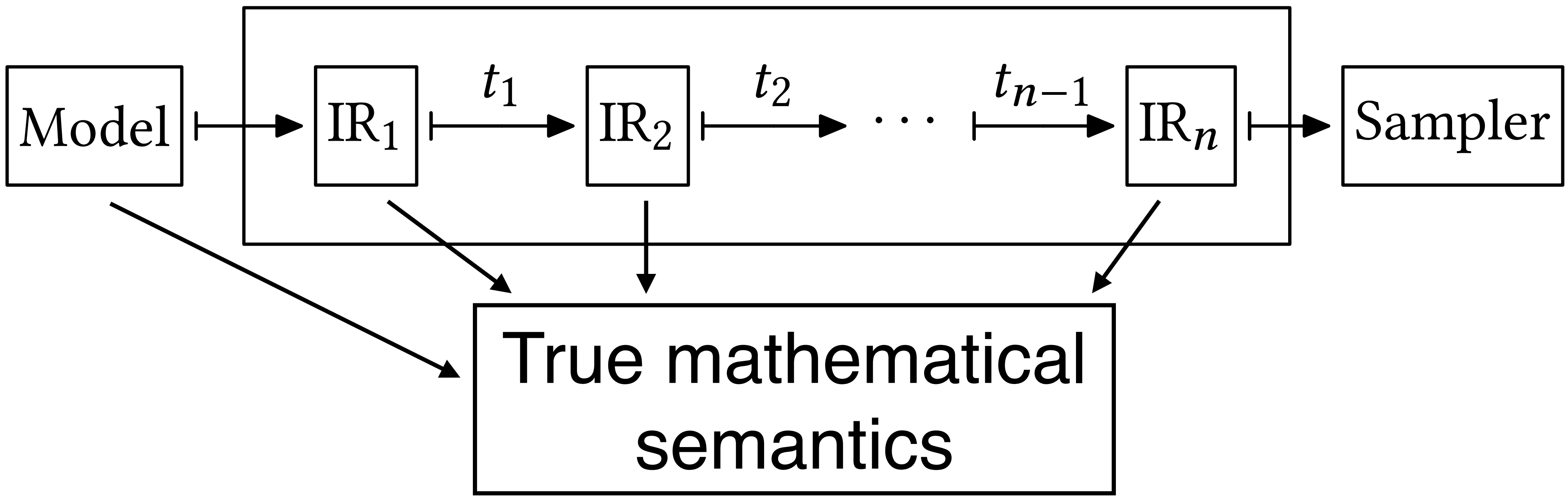
Monte Carlo simulation:

1. run many times;
2. each run gives a result & importance weight

Concerns:

1. too much time spent on low weight traces
   solution: SMC (sequential Monte Carlo)
2. resampling everything each time is costly
   solution: MCMC, MHG

Modular inference algorithms



- exact inference is intractable
- approximate inference algorithms work by manipulating intermediate representations

Overview

- **Part 1:** Illustrations of key ideas.
- **Part 2:** From new foundations to modular and valid inference algorithms.
  - **A synthetic measure theory?**
  - Quasi-Borel spaces
  - Modular & valid inference algorithms
- **Part 3:** What next?

Synthetic measure theory

Kock TAC 2012

Ścibior, Kammar, Vákár, Staton, Yang, Cai, Ostermann, Moss, Heunen, Ghahramani POPL 2018

What's a mathematical universe for probabilistic programming?

What's a synthetic measure theory?

- Want h.o. functions and natural numbers.
  Cartesian closed category with sums.
- Want a space of measures M(X) on every space X.
  A commutative monad M.
- Want M(1) to behave like [0,∞]
  $M(0)=1$, $M(X+Y)=M(X)\times M(Y)$.

Synthetic measure theory

Kock TAC 2012
Ścibior, Kammar, Vákár, Staton, Yang, Cai, Ostermann, Moss, Heunen, Ghahramani POPL 2018

Cart closed category with + & a commutative additive monad.

Dictionary:

+ve scalars	$[0,1]$	$:= \mathrm{M}\,\mathbb{1}$
pushforward	$f_*\underline{\mu}$	$:= (\mathrm{M}\,f)(\underline{\mu})$
Dirac measure	$\underline{\delta}_x$	$:= \mathbf{return}(x)$
Integration	$\oint_X f(x)\underline{\mu}(\mathrm{d}x)$	$:= \underline{\mu} \gg= f$
Product meas.	$\underline{\mu} \otimes \underline{\nu}$	$:= \oint_X \left(\oint_Y \underline{\delta}_{(x,y)}\underline{\nu}(\mathrm{d}y)\right)\underline{\mu}(\mathrm{d}x)$
Expectation	$\mathbb{E}^A_{x\sim\underline{\mu}}[f(x)]$	$:= \underline{\mu} \gg= f$

Problem: classical measure theory is not a model!

Overview

- **Part 1:** Illustrations of key ideas.
- **Part 2:** From new foundations to modular and valid inference algorithms.
  - A synthetic measure theory?
  - **Quasi-Borel spaces**
  - Modular & valid inference algorithms
- **Part 3:** What next?

A semantic model

Standard Borel spaces $\xrightarrow{\text{embedding}}$ **Quasi-Borel spaces**

Models
first order
language with
sample,
score

Models
higher order
language with
sample,
score

Slogan: Random elements come first.

Theorem. Adequacy.

Random elements

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

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

Heunen, Kammar, Staton, Yang, LICS 2017

Quasi-Borel spaces

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 $fg \in N$

Heunen, Kammar, Staton, Yang, LICS 2017

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. ...

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

Proposition. Quasi-Borel spaces include standard Borel spaces fully faithfully.

Proposition. The set of morphisms again forms a quasi-Borel space: we have higher order functions.

Synthetic measure theory

What's the mathematical universe for probabilistic programming?

What's a synthetic measure theory?

- Want h.o. functions and natural numbers. ✔
  Cartesian closed category with sums.
- Want a space of measures M(X) on every space X.
  A commutative monad M.
- Want M(1) to behave like $[0,\infty]$
  $M(0)=1$, $M(X+Y)=M(X)\times M(Y)$.

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

(μ, f)

a σ-finite measure on $\mathbb{R}$

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

(modulo inducing the same integration operator)

Synthetic measure theory

What's the mathematical universe for probabilistic programming?

What's a synthetic measure theory?

- Want h.o. functions and natural numbers.
  Cartesian closed category with sums. ✔
- Want a space of measures M(X) on every space X.
  A commutative monad M. ✔
- Want M(1) to behave like [0,∞]
  M(0)=1, M(X+Y)=M(X)×M(Y). ✔

Heunen, Kammar, Staton, Yang, LICS 2017

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

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

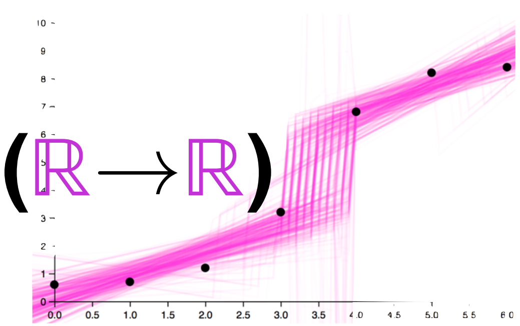
$$(\mu\ ,\ f)$$

a measure on $\mathbb{R}$

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

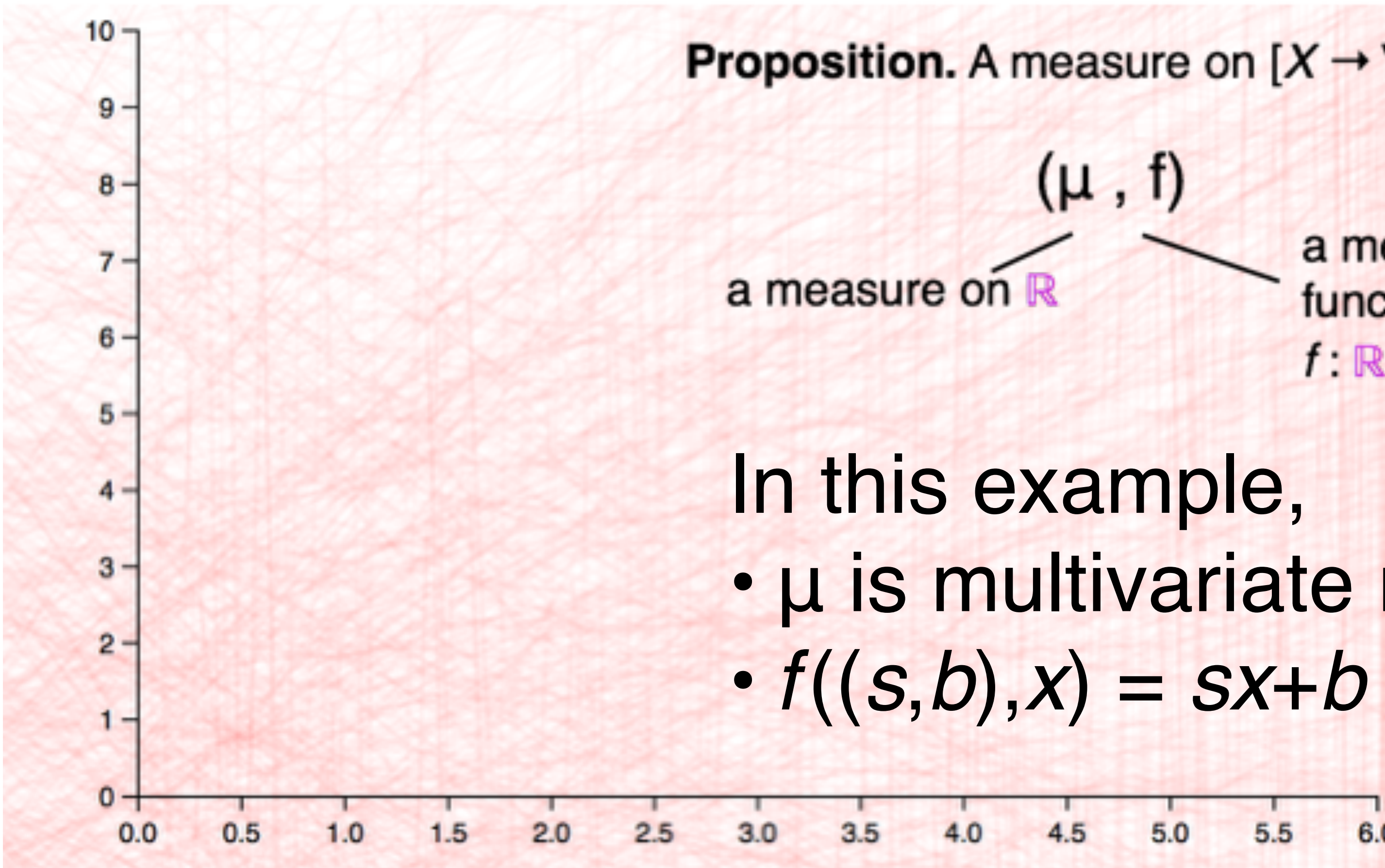
– a 'random function'.

Example: `piecewise` **:** $\mathbf{M}(\mathbb{R}\to\mathbb{R}) \to \mathbf{M}(\mathbb{R}\to\mathbb{R})$



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,

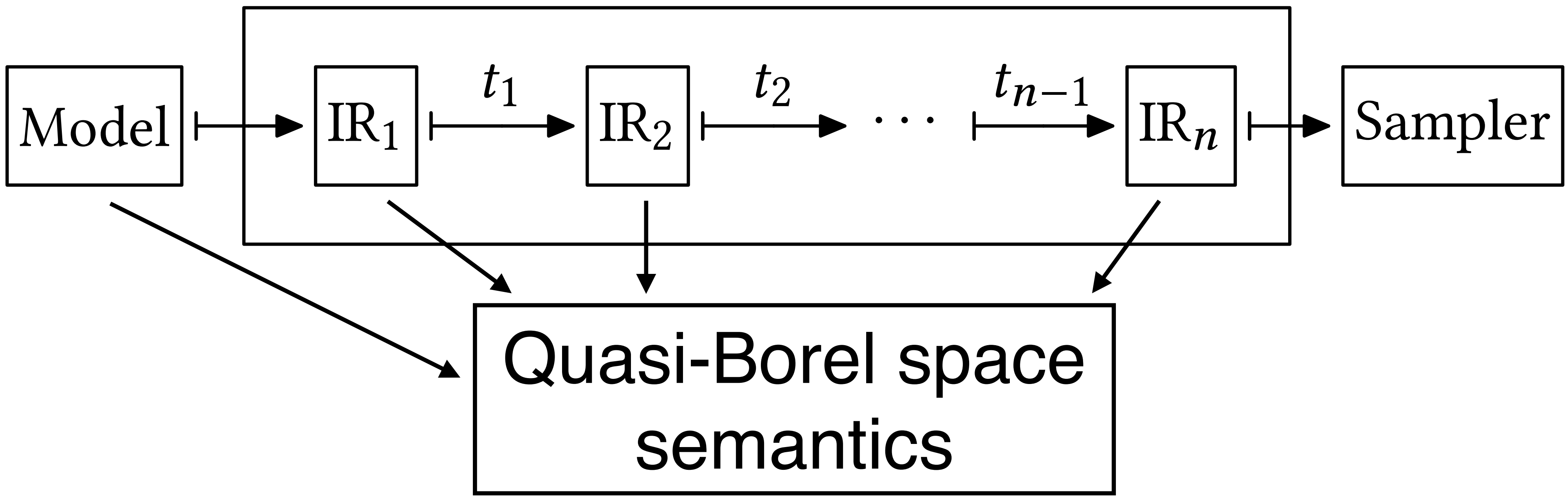
- μ is multivariate normal,
- $f((s,b),x) = sx+b$

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

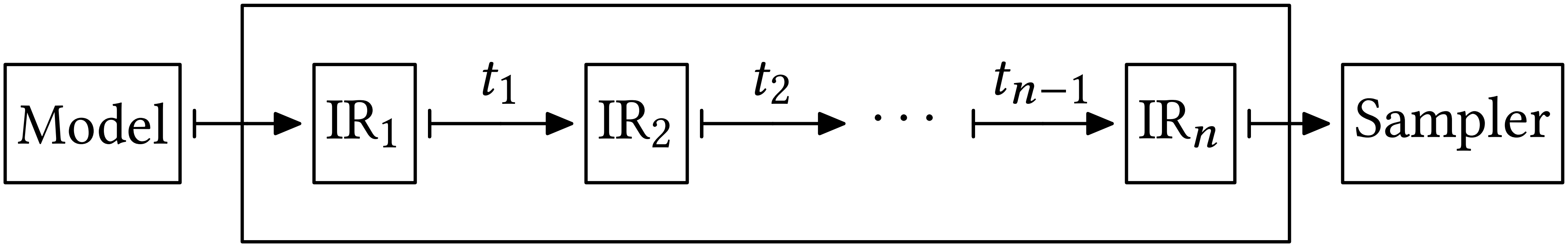
Overview

- **Part 1:** Illustrations of key ideas.
- **Part 2:** From new foundations to modular and valid inference algorithms.
  - A synthetic measure theory?
  - Quasi-Borel spaces
  - **Modular & valid inference algorithms**
- **Part 3:** What next?

Modular inference algorithms



Modular inference algorithms

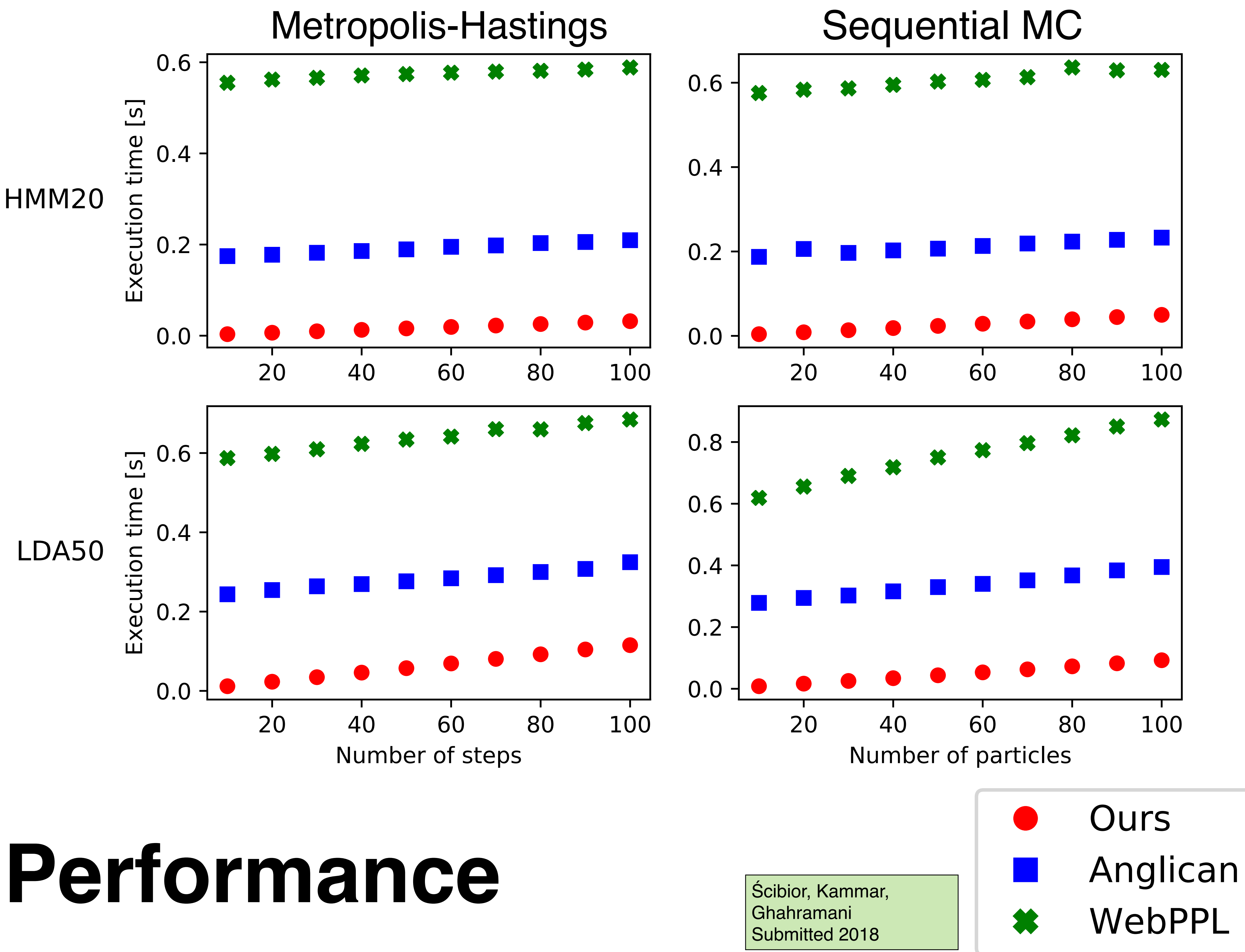


Example IR (intermediate representation):
[0,1]-indexed decision trees.

$$\mathrm{Sam}\ \alpha = \{\mathrm{Return}\ \alpha \mid \mathrm{Sample}\ ([0,1] \to\ \mathrm{Sam}\ \alpha)\}$$

Manipulations of this structure are higher-order functions.

Theorem: MHG works in quasi-Borel spaces.



Performance

Ścibior, Kammar, Ghahramani
Submitted 2018

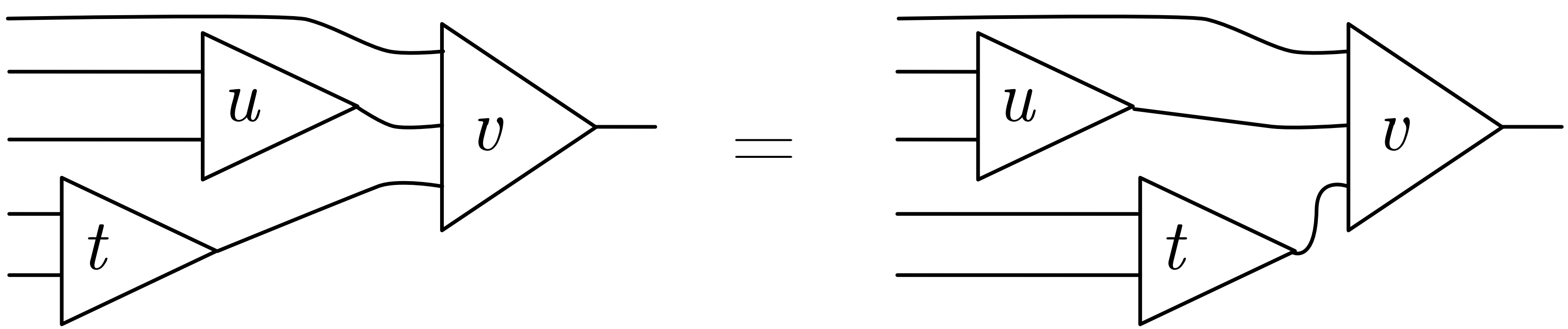
Overview

- **Part 1:** Illustrations of key ideas.
- **Part 2:** From new foundations to modular and valid inference algorithms.
  - A synthetic measure theory
  - Quasi-Borel spaces as a model
  - Modular & valid inference algorithms
- **Part 3: What next?**

Commutativity

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

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



$$\int \left(\int v(x,y)\, u(\mathrm{d}y) \right) \, t(\mathrm{d}x) = \int \left(\int v(x,y)\, t(\mathrm{d}x) \right) \, u(\mathrm{d}y)$$

Commutativity=exchangeability?

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

- **Church** considers user defined ‘exchangeable random primitives’ – new commutative constructions.
- Perhaps these make new models of synthetic measure theory
  – just as 1980s ideas in Bayesian non-parametric came out of non-standard analysis

Overview

- **Part 1:** Illustrations of key ideas.
- **Part 2:** From new foundations to modular and valid inference algorithms.
  - A synthetic measure theory
  - Quasi-Borel spaces are a model
  - Modular & valid inference algorithms
- **Part 3: What next?** Exchangeability and commutativity in non-parametric models