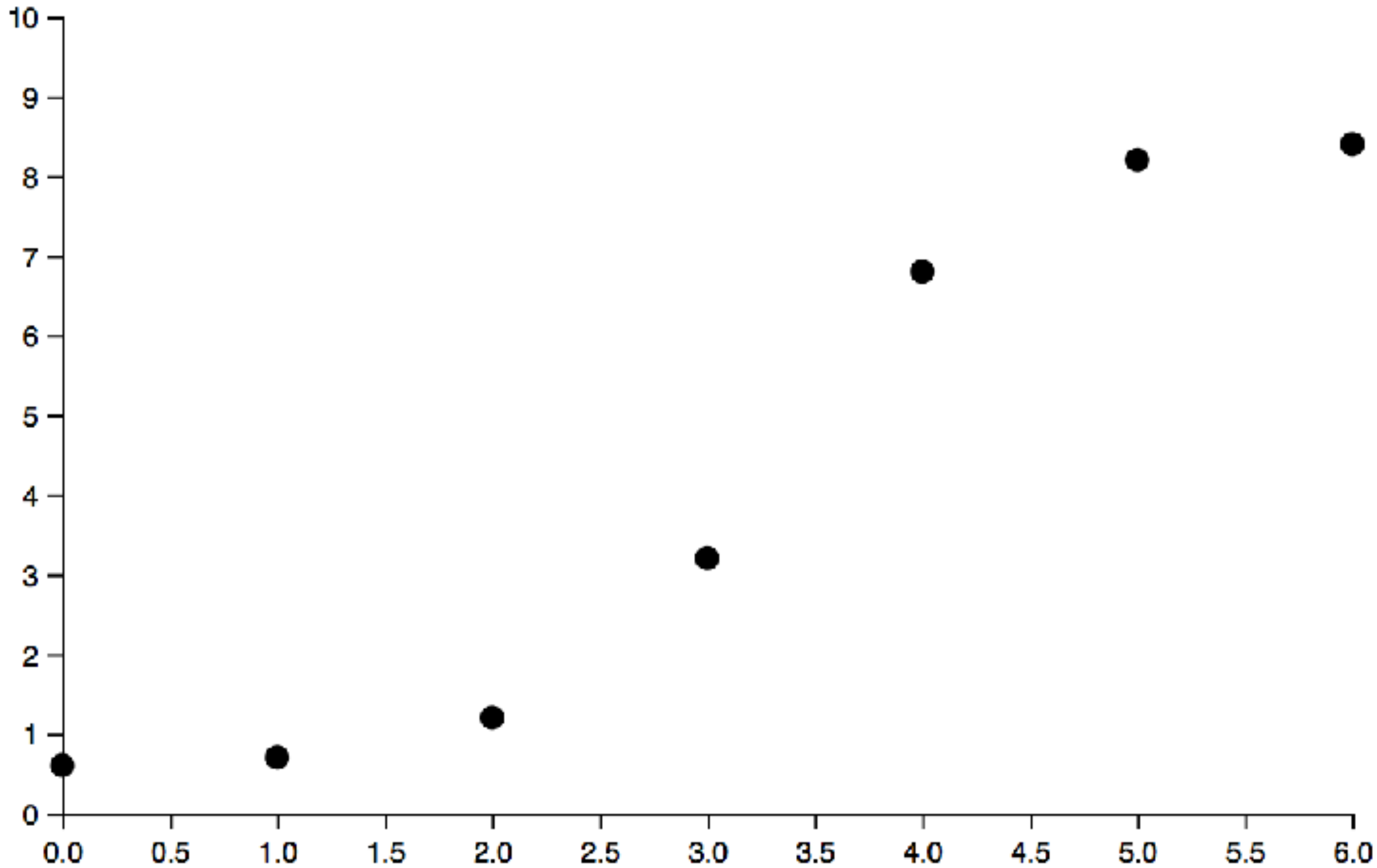
CS492: Probabilistic Programming

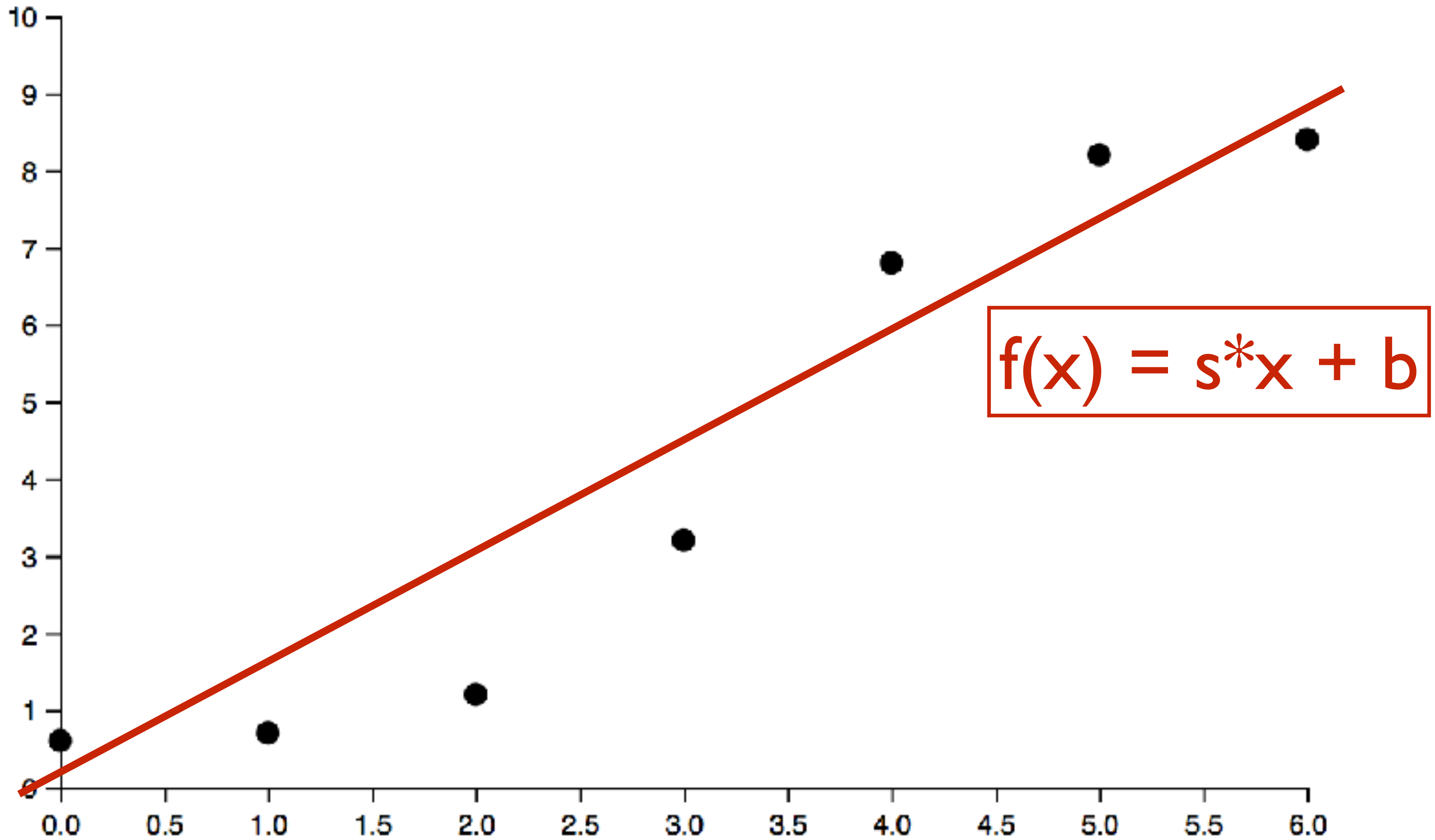
Denotational Semantics of Probabilistic Programs

Hongseok Yang
KAIST

Line fitting



Line fitting



Anglican program

```
(defquery lin-regression []
```

Anglican program

```
(defquery lin-regression []
  (let [s (sample (normal 0 2))
        b (sample (normal 0 6))
        f  (fn [x] (+ (* s x) b))]
```

Anglican program

```
(defquery lin-regression []
  (let [s (sample (normal 0 2))
        b (sample (normal 0 6))
        f  (fn [x] (+ (* s x) b))]

    (observe (normal (f 0) .5) .6)
    (observe (normal (f 1) .5) .7)
    (observe (normal (f 2) .5) 1.2)
    (observe (normal (f 3) .5) 3.2)
    (observe (normal (f 4) .5) 6.8)
    (observe (normal (f 5) .5) 8.2)
    (observe (normal (f 6) .5) 8.4)
```

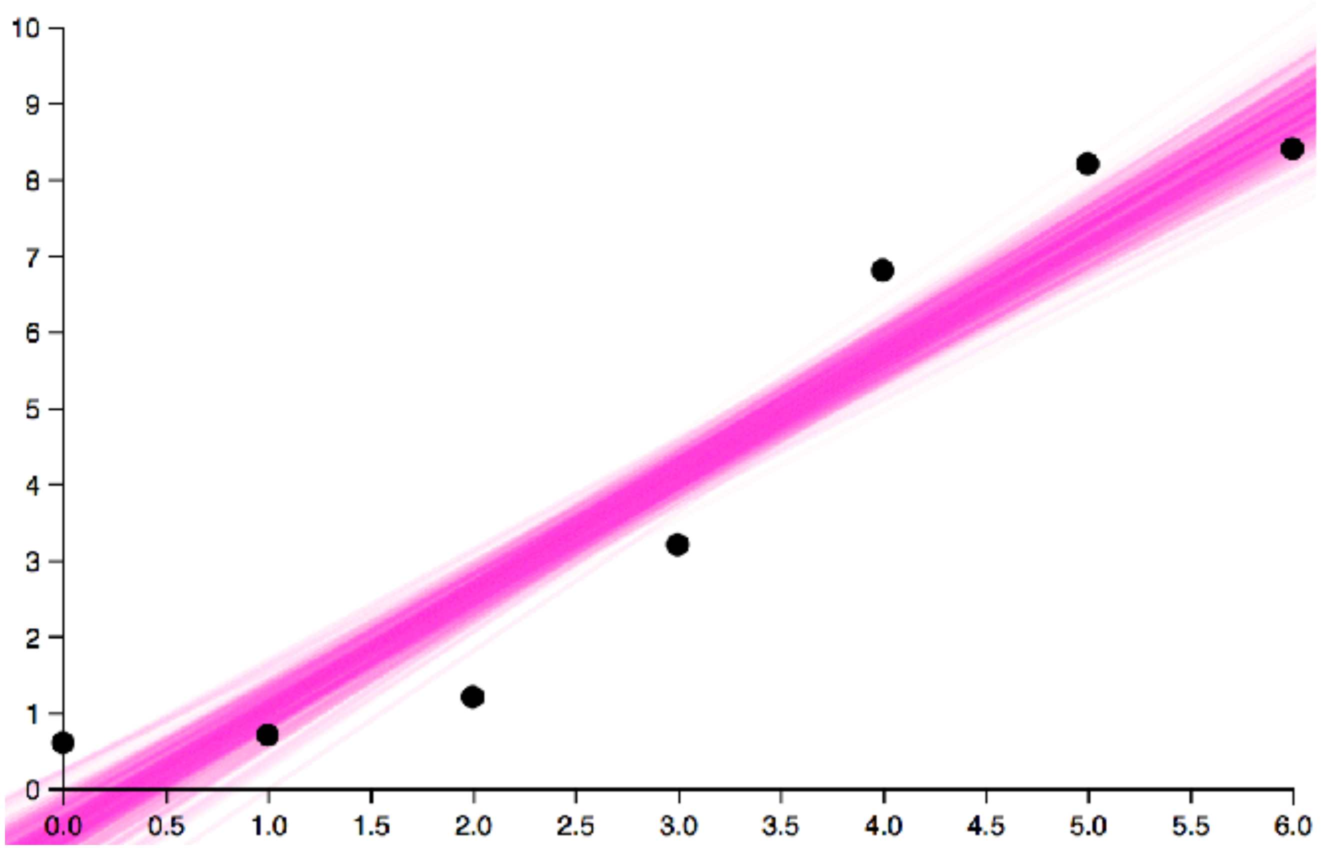
Anglican program

```
(defquery lin-regression []
  (let [s (sample (normal 0 2))
        b (sample (normal 0 6))
        f  (fn [x] (+ (* s x) b))]

    (observe (normal (f 0) .5) .6)
    (observe (normal (f 1) .5) .7)
    (observe (normal (f 2) .5) 1.2)
    (observe (normal (f 3) .5) 3.2)
    (observe (normal (f 4) .5) 6.8)
    (observe (normal (f 5) .5) 8.2)
    (observe (normal (f 6) .5) 8.4)

    f))
```

Samples from posterior



```
(defquery lin-regression []
  (let [s (sample (normal 0 2))
        b (sample (normal 0 6))
        f  (fn [x] (+ (* s x) b))]

    (observe (normal (f 0) .5) .6)
    (observe (normal (f 1) .5) .7)
    (observe (normal (f 2) .5) 1.2)
    (observe (normal (f 3) .5) 3.2)
    (observe (normal (f 4) .5) 6.8)
    (observe (normal (f 5) .5) 8.2)
    (observe (normal (f 6) .5) 8.4)

    f))
```

[Q] Which posterior?

```
(defquery lin-regression []
  (let [s (sample (normal 0 2))
        b (sample (normal 0 6))
        f  (fn [x] (+ (* s x) b))]

    (observe (normal (f 0) .5) .6)
    (observe (normal (f 1) .5) .7)
    (observe (normal (f 2) .5) 1.2)
    (observe (normal (f 3) .5) 3.2)
    (observe (normal (f 4) .5) 6.8)
    (observe (normal (f 5) .5) 8.2)
    (observe (normal (f 6) .5) 8.4)

    f))
```

[Q] Which posterior?

Inference algo. gives only approximation.

```
(defquery lin-regression []
  (let [s (sample (normal 0 2))
        b (sample (normal 0 6))
        f (fn [x] (+ (* s x) b))]

    (observe (normal (f 0) .5) .6)
    (observe (normal (f 1) .5) .7)
    (observe (normal (f 2) .5) 1.2)
    (observe (normal (f 3) .5) 3.2)
    (observe (normal (f 4) .5) 6.8)
    (observe (normal (f 5) .5) 8.2)
    (observe (normal (f 6) .5) 8.4)

    f))
```

[Q] Which posterior?

Inference algo. gives only approximation.

Should define distr. on functions. Not easy.

Foundational question

Measure theory provides a foundation for modern probability theory.

But it doesn't support higher-order fns well.

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

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

```
(defquery lin-regression []
  (let [s (sample (normal 0 2))
        b (sample (normal 0 6))
        f  (fn [x] (+ (* s x) b))]

    (observe (normal (f 0) .5) .6)
    (observe (normal (f 1) .5) .7)
    (observe (normal (f 2) .5) 1.2)
    (observe (normal (f 3) .5) 3.2)
    (observe (normal (f 4) .5) 6.8)
    (observe (normal (f 5) .5) 8.2)
    (observe (normal (f 6) .5) 8.4)

    f))
```

[Q] Which posterior?

Inference algo. gives only approximation.

Should define distr. on functions. Not easy.

Denotational semantics: Compositional method. Answers a deep Q.

Learning outcome

- Can define a denotational semantics for a simple programming language.
- Can use measure theory to interpret prob. prog. with continuous distributions.
- Can use quasi-Borel space to interpret higher-order prob. prog. with conditioning.

References

1. A convenient category for higher-order probability theory. Heunen et al. LICS'17.
2. Commutative semantics for probabilistic programs. Staton. ESOP'17.
3. Reynolds's "Theories of Programming Languages".
4. Billingsley's "Probability and Measure".

Plan for the rest

1. Denotational semantics.
   PL with discrete random choices.
2. Baby measure theory.
   PL with cont. distribution.
3. Quasi-Borel space (QBS).
   PL with cont. distr. & higher-order (HO) fns.
4. SFinKer monad on QBS.
   PL with cont. distr., HO fns & conditioning.

Plan for the rest

1. Denotational semantics.
   PL with discrete random choices.

2. Baby measure theory.
   PL with cont. distribution.

3. Quasi-Borel space (QBS).
   PL with cont. distr. & higher-order (HO) fns.

4. SFinKer monad on QBS.
   PL with cont. distr., HO fns & conditioning.

Denotational semantics

- Defines formal meanings of programs.
- Interprets type and expr. mathematically.
  - Type as space (e.g. set, measurable space).
  - Expr. as good function between spaces.
- Used for justifying compiler optimisation, inference algorithms and language constructs.

Denotational semantics

- Defines formal meanings of programs.
- Interprets type and expr. mathematically.
  - Type as space (e.g. set, measurable space).
  - Expr. as good function between spaces.
- Used for justifying compiler optimisation, inference algorithms and language constructs.

First-order PL with discrete random choices

```
t ::= bool | rational | dist[bool] | dist[rational]

e ::= c | x | (p e ... e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | ...

p ::= sample | flip | poisson | and | + | ...
```

First-order PL with discrete random choices

```
t  ::=  bool | rational | dist[bool] | dist[rational]

e  ::=  c | x | (p e ... e) | (let [x e] e) | (if e e e)

c  ::=  true | false | 0 | 1 | ...

p  ::=  sample | flip | poisson | and | + | ...
```

First-order PL with discrete random choices

t ::= bool | rational | dist[bool] | dist[rational]

e ::= c | x | (p e … e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | …

p ::= sample | flip | poisson | and | + | …

No function type.

First-order PL with discrete random choices

t ::= bool | rational | dist[bool] | dist[rational]

e ::= c | x | (p e … e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | …

p ::= sample | flip | poisson | and | + | …

No function type.

No (fn [x … x] e) case.

First-order PL with discrete random choices

t ::= bool | rational | dist[bool] | dist[rational]

e ::= c | x | (p e ... e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | ...

p ::= sample | flip | poisson | and | + | ...

No function type.

No (fn [x ... x] e) case.

First-order PL with discrete random choices

t ::= bool | rational | dist[bool] | dist[rational]

e ::= c | x | (p e ... e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | ...

p ::= sample | flip | poisson | and | + | ...

No function type.
No (fn [x ... x] e) case.
Only primitive functions can be applied.

First-order PL with discrete random choices

t ::= bool | rational | dist[bool] | dist[rational]

e ::= c | x | (p e ... e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | ...

p ::= sample | flip | poisson | and | + | ...

First-order PL with discrete random choices

```
t ::= bool | rational | dist[bool] | dist[rational]

e ::= c | x | (p e ... e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | ...

p ::= sample | flip | poisson | and | + | ...
```

First-order PL with discrete random choices

```
t  ::=  bool | rational | dist[bool] | dist[rational]

e  ::=  c | x | (p e … e) | (let [x e] e) | (if e e e)

c  ::=  true | false | 0 | 1 | …

p  ::=  sample | flip | poisson | and | + | …
```

First-order PL with discrete random choices

```
t ::= bool | rational | dist[bool] | dist[rational]
e ::= c | x | (p e … e) | (let [x e] e) | (if e e e)
c ::= true | false | 0 | 1 | …
p ::= sample | flip | poisson | and | + | …
```

[Q] Denotational semantics of this PL?

First-order PL with discrete random choices

t ::= bool | rational | dist[bool] | dist[rational]

e ::= c | x | (p e ... e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | ...

p ::= sample | flip | poisson | and | + | ...

[Q] Denotational semantics of this PL?

Interpret type as set and expr. as function.

Types mean sets

$[\![\mathtt{t}]\!]$ is the meaning of $\mathtt{t}$.

$[\![\mathtt{bool}]\!] = \ldots$

$[\![\mathtt{rational}]\!] = \ldots$

$[\![\mathtt{dist[bool]}]\!] = \ldots$

$[\![\mathtt{dist[rational]}]\!] = \ldots$

Types mean sets

$[\![\mathtt{t}]\!]$ is the meaning of $\mathtt{t}$.

$[\![\mathtt{bool}]\!] = \mathbb{B} = \{\mathrm{tt}, \mathrm{ff}\}$

$[\![\mathtt{rational}]\!] = \ldots$

$[\![\mathtt{dist[bool]}]\!] = \ldots$

$[\![\mathtt{dist[rational]}]\!] = \ldots$

Types mean sets

$[\![t]\!]$ is the meaning of t.

$[\![\mathtt{bool}]\!] = \mathbb{B} = \{tt, ff\}$

$[\![\mathtt{rational}]\!] = \ldots$

$[\![\mathtt{dist[bool]}]\!] = \{p{:}\mathbb{B}\to[0,1] \mid p(tt)+p(ff)=1\}$

$[\![\mathtt{dist[rational]}]\!] = \ldots$

Types mean sets

$[\![\mathtt{t}]\!]$ is the meaning of $\mathtt{t}$.

$[\![\mathtt{bool}]\!] = \mathbb{B} = \{tt, ff\}$

$[\![\mathtt{rational}]\!] = \ldots$

$[\![\mathtt{dist[bool]}]\!] = \{p{:}\mathbb{B}\to[0,1] \mid p(tt)+p(ff)=1\}$

$[\![\mathtt{dist[rational]}]\!] = \ldots$

[Q] Fill in $\ldots$

Types mean sets

$[\![t]\!]$ is the meaning of t.

$[\![\mathrm{bool}]\!] = \mathbb{B} = \{tt, ff\}$

$[\![\mathrm{rational}]\!] = \mathbb{Q} = \{0, 1, -1/3, 1/7, \ldots\}$

$[\![\mathrm{dist}[\mathrm{bool}]]\!] = \{p{:}\mathbb{B} \to [0,1] \mid p(tt)+p(ff)=1\}$

$[\![\mathrm{dist}[\mathrm{rational}]]\!] = \{p{:}\, \mathbb{Q} \to [0,1] \mid \Sigma_r p(r)=1\}$

[Q] Fill in . . .

Types mean sets

$[\![\mathtt{t}]\!]$ is the meaning of $\mathtt{t}$.

$[\![\mathtt{bool}]\!] = \mathbb{B} = \{tt, ff\}$

$[\![\mathtt{rational}]\!] = \mathbb{Q} = \{0, 1, -1/3, 1/7, \ldots\}$

$[\![\mathtt{dist[bool]}]\!] = \{p{:}\mathbb{B} \rightarrow [0,1] \mid p(tt)+p(ff)=1\}$

$[\![\mathtt{dist[rational]}]\!] = \{p{:}\,\mathbb{Q} \rightarrow [0,1] \mid \Sigma_r p(r)=1\}$

[Q] Fill in . . .

Types mean sets

$[\![\mathtt{t}]\!]$ is the meaning of $\mathtt{t}$.

$[\![\mathtt{bool}]\!] = \mathbb{B} = \{\mathrm{tt}, \mathrm{ff}\}$

$[\![\mathtt{rational}]\!] = \mathbb{Q} = \{0, 1, -1/3, 1/7, \ldots\}$

$[\![\mathtt{dist[bool]}]\!] = \mathrm{DiscProb}([\![\mathtt{bool}]\!])$

$[\![\mathtt{dist[rational]}]\!] = \mathrm{DiscProb}([\![\mathtt{rational}]\!])$

[Q] Fill in . . .

Typed expressions

$$x_1{:}t_1,\ x_2{:}t_2,\ \ldots,\ x_n{:}t_n \vdash e : t$$

Typed expressions

$$\overbrace{\mathtt{x_1{:}t_1,\ x_2{:}t_2,\ \ldots,\ x_n{:}t_n}}^{\text{typing context } \Gamma} \vdash \mathtt{e : t}$$

- Γ is a finite map from variables to types.

Typed expressions

$$\underbrace{\mathtt{x_1{:}t_1,\ x_2{:}t_2,\ \ldots,\ x_n{:}t_n}}^{\text{typing context }\Gamma} \vdash \mathtt{e : t}$$

- Γ is a finite map from variables to types.
- Denotes a $\mathtt{t}$-typed expr. $\mathtt{e}$ under Γ.

Typed expressions

$$\underbrace{x_1{:}t_1,\ x_2{:}t_2,\ \ldots,\ x_n{:}t_n}_{\text{typing context } \Gamma} \vdash e : t$$

- Γ is a finite map from variables to types.
- Denotes a t-typed expr. e under Γ.

x:bool, y:bool $\vdash$ (if x y y): bool

x:bool, y:rational $\vdash$ (if x y y) : rational

x:rational $\vdash$ (sample (flip x)) : bool

Typed expressions mean fns to discrete prob. distr.

Typed expressions mean fns to discrete prob. distr.

$[\![x_1{:}t_1, \ldots, x_n{:}t_n \vdash e : t]\!]$

Typed expressions mean fns to discrete prob. distr.

$[\![x_1{:}t_1, \ldots, x_n{:}t_n \vdash e : t]\!]$

$= g : [\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] \to \mathrm{DiscProb}([\![t]\!])$

Typed expressions mean fns to discrete prob. distr.

$$[\![x_1{:}t_1, \ldots, x_n{:}t_n \vdash e : t]\!]$$
$$= g : [\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] \to \mathrm{DiscProb}([\![t]\!])$$

1. $[\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] =$
   $\{$map η from $\{x_1,\ldots,x_n\} \mid \eta(x_i) \in [\![t_i]\!]$ for all $i\}$

Typed expressions mean fns to discrete prob. distr.

$[\![x_1{:}t_1, \ldots, x_n{:}t_n \vdash e : t]\!]$

$= g : [\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] \to \mathrm{DiscProb}([\![t]\!])$

1. $[\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] =$
   $\{\text{map } \eta \text{ from } \{x_1,\ldots,x_n\} \mid \eta(x_i) \in [\![t_i]\!] \text{ for all } i\}$

2. $\mathrm{DiscProb}(A) = \{p{:}A \to [0,1] \mid \sum_a p(a) = 1\}$

Typed expressions mean fns to discrete prob. distr.

$$[\![x_1{:}t_1, \ldots, x_n{:}t_n \vdash e : t]\!]$$
$$= g : [\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] \to \mathrm{DiscProb}([\![t]\!])$$

1. $[\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] =$
   $\{\text{map } \eta \text{ from } \{x_1,\ldots,x_n\} \mid \eta(x_i) \in [\![t_i]\!] \text{ for all } i\}$
2. $\mathrm{DiscProb}(A) = \{p{:}A \to [0,1] \mid \sum_a p(a) = 1\}$

[Q] Any problem with DiscProb($[\![\mathtt{dist[bool]}]\!]$)?

Typed expressions mean fns to discrete prob. distr.

$$[\![x_1{:}t_1, \ldots, x_n{:}t_n \vdash e : t]\!]$$
$$= g : [\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] \to \mathrm{DiscProb}([\![t]\!])$$

1. $[\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] =$
   $\{\text{map } \eta \text{ from } \{x_1,\ldots,x_n\} \mid \eta(x_i) \in [\![t_i]\!] \text{ for all } i\}$

2. $\mathrm{DiscProb}(A) = \{p{:}A \to [0,1] \mid \sum_a p(a) = 1\}$

[Q] Any problem with $\mathrm{DiscProb}([\![\mathtt{dist[bool]}]\!])$?

Typed expressions mean fns to discrete prob. distr.

$[\![x_1{:}t_1, \ldots, x_n{:}t_n \vdash e : t]\!]$
$= g : [\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] \to \mathrm{DiscProb}([\![t]\!])$

1. $[\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] =$
   $\{\text{map } \eta \text{ from } \{x_1,\ldots,x_n\} \mid \eta(x_i) \in [\![t_i]\!] \text{ for all } i\}$

2. $\mathrm{DiscProb}(A) = \{p{:}A \to [0,1] \mid \sum_a p(a) = 1\}$

   p(a)=0 except for countably many a's and

[Q] Any problem with $\mathrm{DiscProb}([\![\mathtt{dist[bool]}]\!])$?

Typed expressions mean fns to discrete prob. distr.

$[\![x_1{:}t_1, \ldots, x_n{:}t_n \vdash e : t]\!]$

$= g : [\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] \to \mathrm{DiscProb}([\![t]\!])$

1. $[\![x_1{:}t_1, \ldots, x_n{:}t_n]\!] =$
   $\{\text{map } \eta \text{ from } \{x_1,\ldots,x_n\} \mid \eta(x_i) \in [\![t_i]\!] \text{ for all } i\}$

2. $\mathrm{DiscProb}(A) = \{p{:}A \to [0,1] \mid \sum_a p(a) = 1\}$

   p(a)=0 except for countably many a's and

[Q] Define the interpretation recursively.

Compiler optimisation

Show the following equations:

$[\![\Gamma \vdash (\texttt{if true } e_1\ e_2) : t]\!] = [\![\Gamma \vdash e_1 : t]\!]$

$[\![\Gamma \vdash (\texttt{sample (flip (+ 0.1 0.2))}) : \texttt{bool}]\!]$
$= [\![\Gamma \vdash (\texttt{sample (flip 0.3)}) : \texttt{bool}]\!]$

Plan for the rest

1. Denotational semantics.
   PL with discrete random choices.

2. Baby measure theory.
   PL with cont. distribution.

3. Quasi-Borel space (QBS).
   PL with cont. distr. & higher-order (HO) fns.

4. SFinKer monad on QBS.
   PL with cont. distr., HO fns & conditioning.

First-order PL with discrete random choices

```
t  ::=  bool | rational | dist[bool] | dist[rational]

e  ::=  c | x | (p e ... e) | (let [x e] e) | (if e e e)

c  ::=  true | false | 0 | 1 | ...

p  ::=  sample | flip | poisson | and | + | ...
```

First-order PL with discrete and continuous random choices

t ::= bool | ~~rational~~ real | dist[bool] | dist[~~rational~~ real]

e ::= c | x | (p e … e) | (let [x e] e) | (if e e e)

c ::= true | false | 0 | 1 | …

p ::= sample | flip | poisson | and | + | …
| normal | uniform-continuous | …

How to define denotational semantics?

Types as spaces and expressions as functions.

How to define denotational semantics?

Types as spaces and expressions as functions.

$$\mathtt{t} \;::=\; \mathrm{bool} \mid \mathrm{real} \mid \mathrm{dist[bool]} \mid \mathrm{dist[real]}$$

[Try] Interpret $[\![\mathtt{t}]\!]$ as a set.

How to define denotational semantics?

Types as spaces and expressions as functions.

$$\mathtt{t} \ ::= \ \mathtt{bool} \mid \mathtt{real} \mid \mathtt{dist[bool]} \mid \mathtt{dist[real]}$$

[Try] Interpret $[\![\mathtt{t}]\!]$ as a set. Then, we get stuck since $[\![\mathtt{dist[real]}]\!]$ can't be DiscProb($[\![\mathtt{real}]\!]$).

How to define denotational semantics?

Types as spaces and expressions as functions.

$$\mathtt{t} \ ::= \ \mathtt{bool} \mid \mathtt{real} \mid \mathtt{dist[bool]} \mid \mathtt{dist[real]}$$

[Try] Interpret $[\![\mathtt{t}]\!]$ as a set. Then, we get stuck since $[\![\mathtt{dist[real]}]\!]$ can't be DiscProb($[\![\mathtt{real}]\!]$).

[Sol] Use measure theory. $[\![\mathtt{t}]\!]$ as a measurable space, and $[\![\Gamma \vdash \mathtt{e} : \mathtt{t}]\!]$ as a measurable function.

How to specify prob. p?

How to specify prob. p?

$X = \{0, 1, 2\}$.

Define $p : X \to [0,1]$. E.g., $p = [0.4, 0.4, 0.2]$.

Lifted $p : 2^X \to [0,1]$ by $p(A) = \sum_{x \in A} p(x)$.

How to specify prob. p?

$X = \mathbb{R}.$

Define $p : X \to [0,1]$.

Lifted $p : 2^X \to [0,1]$ by $p(A) = \sum_{x \in A} p(x)$.

How to specify prob. p?

$X = \mathbb{R}$.

Define $p : X \rightarrow [0,1]$.

Lifted $p : 2^X \rightarrow [0,1]$ by $p(A) = \sum_{x \in A} p(x)$.

Uncountable sum.
Typically ∞.

How to specify prob. p?

$X = \mathbb{R}$.

~~Define $p : X \to [0,1]$.~~

~~Lifted~~ Define $p : 2^X \to [0,1]$ by ~~$p(A) = \sum_{x \in A} p(x)$.~~

How to specify prob. p?

$X = \mathbb{R}$.

~~Define $p : X \to [0,1]$~~

~~Lifted $p : 2^X \to [0,1]$ by $p(A) = \sum_{x \in A} p(x)$.~~

~~Define~~

Pick a good collection $\Sigma \subseteq 2^X$.

Define $p : \Sigma \to [0,1]$ with some care.

How to specify prob. p?

$X = \mathbb{R}$.

~~Define $p : X \to [0,1]$~~

~~Define~~ Lifted ~~$p : 2^X \to [0,1]$ by $p(A) = \sum_{x \in A} p(x)$.~~

σ-algebra

Pick a good collection $\Sigma \subseteq 2^X$.

Define $p : \Sigma \to [0,1]$ with some care.

probability measure

Let $\Sigma \subseteq 2^X$.

Σ is a <u>σ-algebra</u> if it contains X, and is closed under countable union and set subtraction.

(X, Σ) is a <u>measurable space</u> if Σ is a σ-algebra.

$p : \Sigma \to [0,1]$ is a <u>probability measure</u> if $p(X)=1$ and $p(\biguplus_{n\in\mathbb{N}} A_n) = \sum_{n\in\mathbb{N}} p(A_n)$ for all disjoint A_n's.

(X,Σ,p) is a <u>probability space</u> if …

[Q] What are not measurable spaces?

1. $(\mathbb{B}, 2^{\mathbb{B}})$.
2. $(\mathbb{B} \mathrm{x} \mathbb{B}, \{ A\mathrm{x}B \mid A \in 2^{\mathbb{B}} \text{ and } B \in 2^{\mathbb{B}} \})$.
3. $(\mathbb{R}, \{ A \subseteq \mathbb{R} \mid A \text{ or } (\mathbb{R}\text{-}A) \text{ countable} \})$.
4. $(\mathbb{R}, \{ (r,s] \mid r<s \})$.

[Q] What are not measurable spaces?

1. $(\mathbb{B}, 2^{\mathbb{B}})$.
2. $(\mathbb{B} \times \mathbb{B}, \{ A \times B \mid A \in 2^{\mathbb{B}} \text{ and } B \in 2^{\mathbb{B}} \})$.
3. $(\mathbb{R}, \{ A \subseteq \mathbb{R} \mid A \text{ or } (\mathbb{R} - A) \text{ countable} \})$.
4. $(\mathbb{R}, \{ (r,s] \mid r < s \})$.

[Q] Convert them to measurable spaces.

1. $(\mathbb{B}, 2^{\mathbb{B}})$.
2. $(\mathbb{B}\mathrm{x}\mathbb{B}, \{ A\mathrm{x}B \mid A \in 2^{\mathbb{B}} \text{ and } B \in 2^{\mathbb{B}} \})$.
3. $(\mathbb{R}, \{ A \subseteq \mathbb{R} \mid A \text{ or } (\mathbb{R}\text{-}A) \text{ countable} \})$.
4. $(\mathbb{R}, \{ (r,s] \mid r<s \})$.

[Q] Convert them to measurable spaces.

1. $(\mathbb{B}, 2^{\mathbb{B}})$.
2. $(\mathbb{B}\times\mathbb{B}, \sigma\{ A\times B \mid A\in 2^{\mathbb{B}} \text{ and } B\in 2^{\mathbb{B}} \})$.
3. $(\mathbb{R}, \{ A\subseteq\mathbb{R} \mid A \text{ or } (\mathbb{R}-A) \text{ countable} \})$.
4. $(\mathbb{R}, \sigma\{ (r,s] \mid r<s \})$.

Closure exists.
$\sigma(\Pi)$ smallest σ-algebra containing Π.

(X, Σ), (Y, Θ) - mBle spaces.

Product σ-algebra: $\Sigma \otimes \Theta = \sigma\{A x B \mid A \in \Sigma, B \in \Theta\}$.

Product space: $(X,\Sigma) x_m (Y, \Theta) = (X x Y, \Sigma \otimes \Theta)$.

Borel σ-algebra on $\mathbb{R}$: $\mathfrak{B} = \sigma\{(r,s] \mid r<s\}$.

Borel space: $(\mathbb{R}, \mathfrak{B})$.

$(X, \Sigma), (Y, \Theta)$ - mBle spaces.

$Pr(\Sigma) = \ldots$

Probability space: $Pr(X,\Sigma) = (Pr(X), Pr(\Sigma))$

[Q] What is $Pr(\Sigma)$?

(X, Σ), (Y, Θ) - mBle spaces.

$Pr(\Sigma) = \sigma\{ \{p \mid p(A)<r\} \mid A\in\Sigma, r\in\mathbb{R} \}$.

Probability space: $Pr(X,\Sigma) = (Pr(X), Pr(\Sigma))$

[Q] What is $Pr(\Sigma)$?

Types mean mBle spaces

Types mean mBle spaces

$$[\![\mathtt{bool}]\!] = (\mathbb{B}, 2^{\mathbb{B}})$$

$$[\![\mathtt{dist[bool]}]\!] = \mathsf{Pr}([\![\mathtt{bool}]\!])$$

Types mean mBle spaces

$$[\![\texttt{bool}]\!] = (\mathbb{B}, 2^{\mathbb{B}})$$

$$[\![\texttt{real}]\!] = \ \ldots$$

$$[\![\texttt{dist[bool]}]\!] = \mathrm{Pr}([\![\texttt{bool}]\!])$$

$$[\![\texttt{dist[real]}]\!] = \ \ldots$$

[Q] Fill in ...

Types mean mBle spaces

$$[\![\mathtt{bool}]\!] = (\mathbb{B}, 2^{\mathbb{B}})$$

$$[\![\mathtt{real}]\!] = (\mathbb{R}, \mathfrak{B})$$

$$[\![\mathtt{dist[bool]}]\!] = \mathrm{Pr}([\![\mathtt{bool}]\!])$$

$$[\![\mathtt{dist[real]}]\!] = \mathrm{Pr}([\![\mathtt{real}]\!])$$

[Q] Fill in …

Types mean mBle spaces

$$[\![\mathtt{x}_1:\mathtt{t}_1, \ldots, \mathtt{x}_n:\mathtt{t}_n]\!] = (X, \Sigma)$$

Types mean mBle spaces

$$[\![\mathtt{x}_1 : \mathtt{t}_1, \ldots, \mathtt{x}_n : \mathtt{t}_n]\!] = (X, \Sigma)$$

where

$(X_i, \Sigma_i) = [\![\mathtt{t}_i]\!]$

$X = \ldots$

$\Sigma = \ldots$

Types mean mBle spaces

$$[\![\mathtt{x}_1{:}\mathtt{t}_1, \ldots, \mathtt{x}_n{:}\mathtt{t}_n]\!] = (X, \Sigma)$$

where

$(X_i, \Sigma_i) = [\![\mathtt{t}_i]\!]$

$X = \{\text{map } \eta \text{ from } \{\mathtt{x}_1, \ldots, \mathtt{x}_n\} \mid \eta(\mathtt{x}_i) \in X_i \text{ for all } i\}$

$\Sigma = \ \ldots$

Types mean mBle spaces

$$[\![\mathtt{x}_1{:}\mathtt{t}_1, \ldots, \mathtt{x}_n{:}\mathtt{t}_n]\!] = (X, \Sigma)$$

where

$(X_i, \Sigma_i) = [\![\mathtt{t}_\mathtt{i}]\!]$

$X = \{\text{map } \eta \text{ from } \{\mathtt{x}_1,\ldots,\mathtt{x}_n\} \mid \eta(\mathtt{x}_\mathtt{i}) \in X_i \text{ for all i}\}$

$\Sigma = \ldots$

[Q] Fill in ...

Types mean mBle spaces

$$[\![\mathtt{x}_1{:}\mathtt{t}_1, \ldots, \mathtt{x}_n{:}\mathtt{t}_n]\!] = (X, \Sigma)$$

where

$(X_i, \Sigma_i) = [\![\mathtt{t}_i]\!]$

$X = \{\text{map } \eta \text{ from } \{\mathtt{x}_1,\ldots,\mathtt{x}_n\} \mid \eta(\mathtt{x}_i) \in X_i \text{ for all } i\}$

$\Sigma = \sigma\{ \{\eta \mid \eta(\mathtt{x}_i) \in A_i \text{ for all } i\} \mid A_i \in \Sigma_i \text{ for all } i\}$

[Q] Fill in …

(X, Σ), (Y, Θ) - mBle spaces.

$f:X \to Y$ is measurable (denoted $f:X \to_m Y$) if $f^{-1}(A) \in \Sigma$ for all $A \in \Theta$.

(X, Σ), (Y, Θ) - mBle spaces.

$f: X \to Y$ is <u>measurable</u> (denoted $f: X \to_m Y$) if $f^{-1}(A) \in \Sigma$ for all $A \in \Theta$.

Exprs mean mBle fns

$[\![\Gamma \vdash \mathtt{e} : \mathtt{t}]\!]$ is a mBle fn from $[\![\Gamma]\!]$ to $\mathrm{Pr}[\![\mathtt{t}]\!]$.

Exprs mean mBle fns

$[\![\Gamma \vdash \mathtt{e} : \mathtt{t}]\!]$ is a mBle fn from $[\![\Gamma]\!]$ to $\Pr[\![\mathtt{t}]\!]$.

$[\![\mathtt{y{:}real} \vdash \mathtt{(sample\ (norm\ y\ 1))} : \mathtt{real}]\!]$

Exprs mean mBle fns

$[\![\Gamma \vdash \mathtt{e} : \mathtt{t}]\!]$ is a mBle fn from $[\![\Gamma]\!]$ to Pr$[\![\mathtt{t}]\!]$.

$[\![\mathtt{y{:}real} \vdash \mathtt{(sample\ (norm\ y\ 1))} : \mathtt{real}]\!]\eta(A)$

Exprs mean mBle fns

$[\![\Gamma \vdash \mathtt{e} : \mathtt{t}]\!]$ is a mBle fn from $[\![\Gamma]\!]$ to $\mathrm{Pr}[\![\mathtt{t}]\!]$.

$$[\![\mathtt{y{:}real} \vdash (\mathtt{sample}\ (\mathtt{norm}\ \mathtt{y}\ \mathtt{1})) : \mathtt{real}]\!]\eta(A)$$

$$= \int_A \text{density-norm}(s \mid \eta(y), 1)\ ds.$$

Exprs mean mBle fns

$[\![\Gamma \vdash \mathtt{e} : \mathtt{t}]\!]$ is a mBle fn from $[\![\Gamma]\!]$ to Pr$[\![\mathtt{t}]\!]$.

$[\![\mathtt{y{:}real} \vdash \mathtt{(sample\ (norm\ y\ 1))} : \mathtt{real}]\!]\eta(A)$

$= \int_A$ density-norm(s | η(y), 1) ds.

Defined recursively. Complex but doable.

Plan for the rest

1. Denotational semantics.
   PL with discrete random choices.

2. Baby measure theory.
   PL with cont. distribution.

3. Quasi-Borel space (QBS).
   PL with cont. distr. & higher-order (HO) fns.

4. SFinKer monad on QBS.
   PL with cont. distr., HO fns & conditioning.