Semantics of Higher-order Probabilistic Programs

Hongseok Yang
University of Oxford

Joint work with Chris Heunen, Ohad Kammar, Sam Staton, Frank Wood

What issues did we encounter in defining the denotational semantics of an idealised Anglican?

Continuous distribution and soft constraints

```
let x=sample(normal(0,1)) in
obs(normal(x,1),2); return(x)
```

```
let x=sample(normal(0,1)) in
obs(normal(x,1),2); return(x)
```

```
let x=sample(normal(0,1)) in
obs(normal(x,1),2); return(x)
```

$$[\![\; \boxed{\begin{array}{l}\texttt{let x=sample(normal(0,1)) in} \\ \texttt{obs(normal(x,1),2); return(x)}\end{array}} \;]\!] \in M(R)$$

Which monad M should we pick?

$[\![$
```
let x=sample(normal(0,1)) in
obs(normal(x,1),2); return(x)
```
$]\!] \in M(R)$

Which monad M should we pick?

$p(x) \times p(y=2 \mid x)$

1. Lazy semantics (importance sampling):

$$M(T) = \mathrm{Prob}(R_{\geq 0} \times T)$$

$$[\![\ \boxed{\texttt{let x=sample(normal(0,1)) in obs(normal(x,1),2); return(x)}}\ ]\!] \in M(R)$$

```
let x=sample(normal(0,1)) in
obs(normal(x,1),2); return(x)
```

Which monad M should we pick?

$$p(x) \times p(y{=}2 \mid x) \;=\; p(y{=}2) \times p(x \mid y{=}2)$$

1. Lazy semantics (importance sampling):

$$M(T) = Prob(R_{\geq 0} \times T)$$

2. Eager semantics:

$$M(T) = R_{\geq 0} \times Prob(T)$$

$[\![$
```
let x=sample(normal(0,1)) in
obs(normal(x,1),2); return(x)
```
$]\!] \in M(R)$

Which monad M should we pick?

$$p(x) \times p(y=2 \mid x) = p(y=2) \times p(x \mid y=2)$$

1. Lazy semantics (importance sampling):

$$M(T) = Prob(R_{\geq 0} \times T)$$

2. Eager semantics:

$$M(T) = R_{\geq 0} \times Prob(T)$$

```
let x=sample(exponential(1)) in
obs(normal(0,exp(-x)),0);
return(x)
```

$[\![$

```
let x=sample(exponential(1)) in
obs(normal(0,exp(-x)),0);
return(x)
```

$]\!]_E$

$\notin R_{\geq 0} \times Prob(R)$

Because p(y=0) = ∞ and p(x|y=0) undefined.

$$[\![$$

```
let x=sample(exponential(1)) in
obs(normal(0,exp(-x)),0);
return(x)
```

$$]\!]_E \notin R_{\geq 0} \times Prob(R)$$

Because $p(y=0) = \infty$ and $p(x|y=0)$ undefined.

- $p(x) = \exp(-x)$.
- $p(y=0 \mid x) = 1 / (c \times \exp(-x))$.
- $p(y=0) = \int p(y=0|x)p(x)dx = \int 1/c \, dx = \infty$.

$[\![$
```
let x=sample(exponential(1)) in
obs(normal(0,exp(-x)),0);
return(x)
```
$]\!]_E$

$\notin R_{\geq 0} \times Prob(R)$

Because $p(y=0) = \infty$ and $p(x|y=0)$ undefined.

- $p(x) = exp(-x)$.
- $p(y=0 \mid x) = 1 / (c \times exp(-x))$.
- $p(y=0) = \int p(y=0|x)p(x)dx = \int 1/c \, dx = \infty$.

We tried $(R_{\geq 0} \times Prob(R) + 1 + 1)$, but failed.

In the lazy semantics, conjugate-prior equations fail.

$$[\![\ \texttt{let x=sample(beta(1,1)) in obs(bern(x),true); return(x)}\ ]\!]$$

$$\neq$$

$$[\![\ \texttt{obs(bern(0.5),true); sample(beta(2,1))}\ ]\!]$$

In the lazy semantics, conjugate-prior equations fail.
But can be recovered with explicit normalisation (or nested query).

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

$[\![\mathrm{norm}]\!] : \mathrm{Prob}(R_{\geq 0} \times T) \to (R_{\geq 0} \times \mathrm{Prob}(T) + 1 + 1)$

Continuous densities

Density object

```
let x=sample(exponential(1)) in
obs(normal(0,exp(-x)),0);
return(x)
```

- Represents a probability density.
- Supports sample and obs methods.
- In Anglican, the densities of these objects are usually continuous functions.

$$[\![\text{Dens}[R]]\!] = \{\ f : R \rightarrow R_{\geq 0}\ \mid\ f \text{ is continuous and } \int f(x)dx = 1\}$$

$$[\![\text{sample}]\!]\ :\ [\![\text{Dens}[R]]\!] \rightarrow P(R_{\geq 0} \times R)$$

$$[\![\text{obs}]\!] :\ [\![\text{Dens}[R]]\!] \times [\![R]\!] \rightarrow P(R_{\geq 0} \times ())$$

$$[\![\mathrm{Dens}[R]]\!] = \{\, f : R \to R_{\geq 0} \;\mid\; f \text{ is continuous and } \int f(x)dx = 1 \,\}$$

$$[\![\mathrm{sample}]\!] \;:\; [\![\mathrm{Dens}[R]]\!] \to P(R_{\geq 0} \times R)$$

$$[\![\mathrm{obs}]\!] : [\![\mathrm{Dens}[R]]\!] \times [\![R]\!] \to P(R_{\geq 0} \times ())$$

[Q] Is $[\![\mathrm{obs}]\!]$ measurable?

Non-measurability of ev

$$ev : (R \rightarrow_m R) \times R \rightarrow R, \qquad ev(f,x) = f(x)$$

[Aumann 61 & Halmos] ev is not measurable no matter which σ-algebra is used for $R \rightarrow_m R$.

Non-measurability of ev

$$ev : (R \to_m R) \times R \to R, \qquad ev(f,x) = f(x)$$

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

But:

1. $ev_r = ev(-,r) : (R \to_m R) \to R$ is measurable.

Non-measurability of ev

$$ev : (R \to_m R) \times R \to R, \qquad ev(f,x) = f(x)$$

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

But:

1. $ev_r = ev(-,r) : (R \to_m R) \to R$ is measurable.
2. $ev : (R \to_c R) \times R \to R$ is measurable.

$$[\![\mathrm{Dens}[R]]\!] = \{\, f : R \to R_{\geq 0} \;\mid\; f \text{ is continuous and } \int f(x)dx = 1 \,\}$$

$$[\![\mathrm{sample}]\!] \;:\; [\![\mathrm{Dens}[R]]\!] \to P(R_{\geq 0} \times R)$$

$$[\![\mathrm{obs}]\!] : [\![\mathrm{Dens}[R]]\!] \times [\![R]\!] \to P(R_{\geq 0} \times ())$$

[Q] Is $[\![\mathrm{obs}]\!]$ measurable? [A] Yes.

Higher-order function

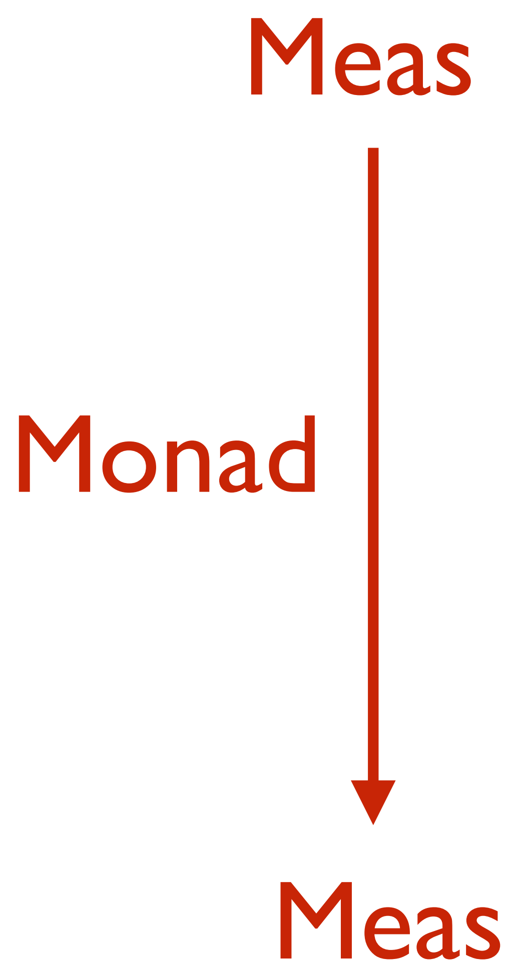
Difficulty

$$ev : (R \to_m R) \times R \to R, \qquad ev(f,x) = f(x)$$

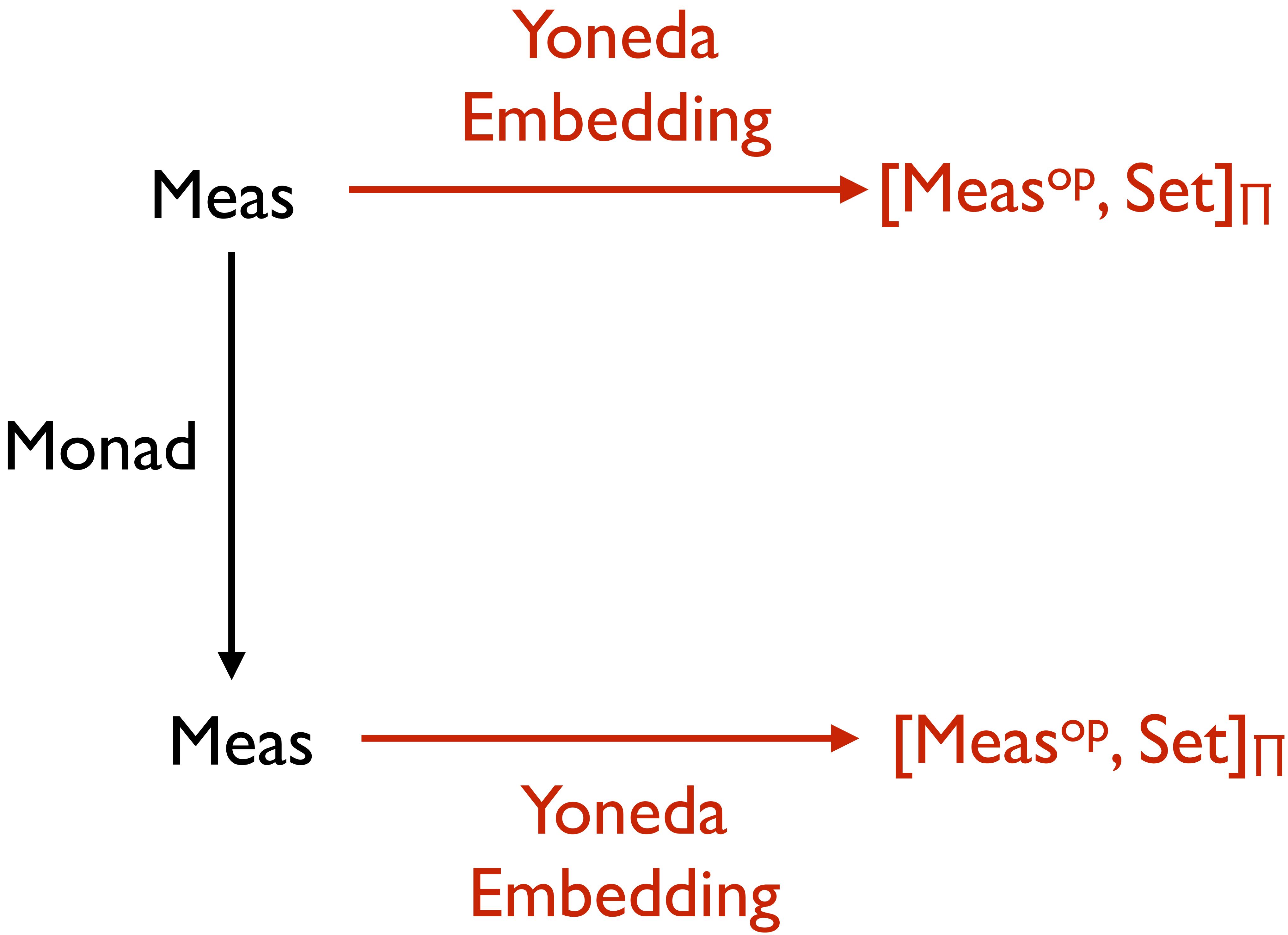
[Aumann 61 & Halmos] ev is not measurable no matter which σ-algebra is used for $R \to_m R$.

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

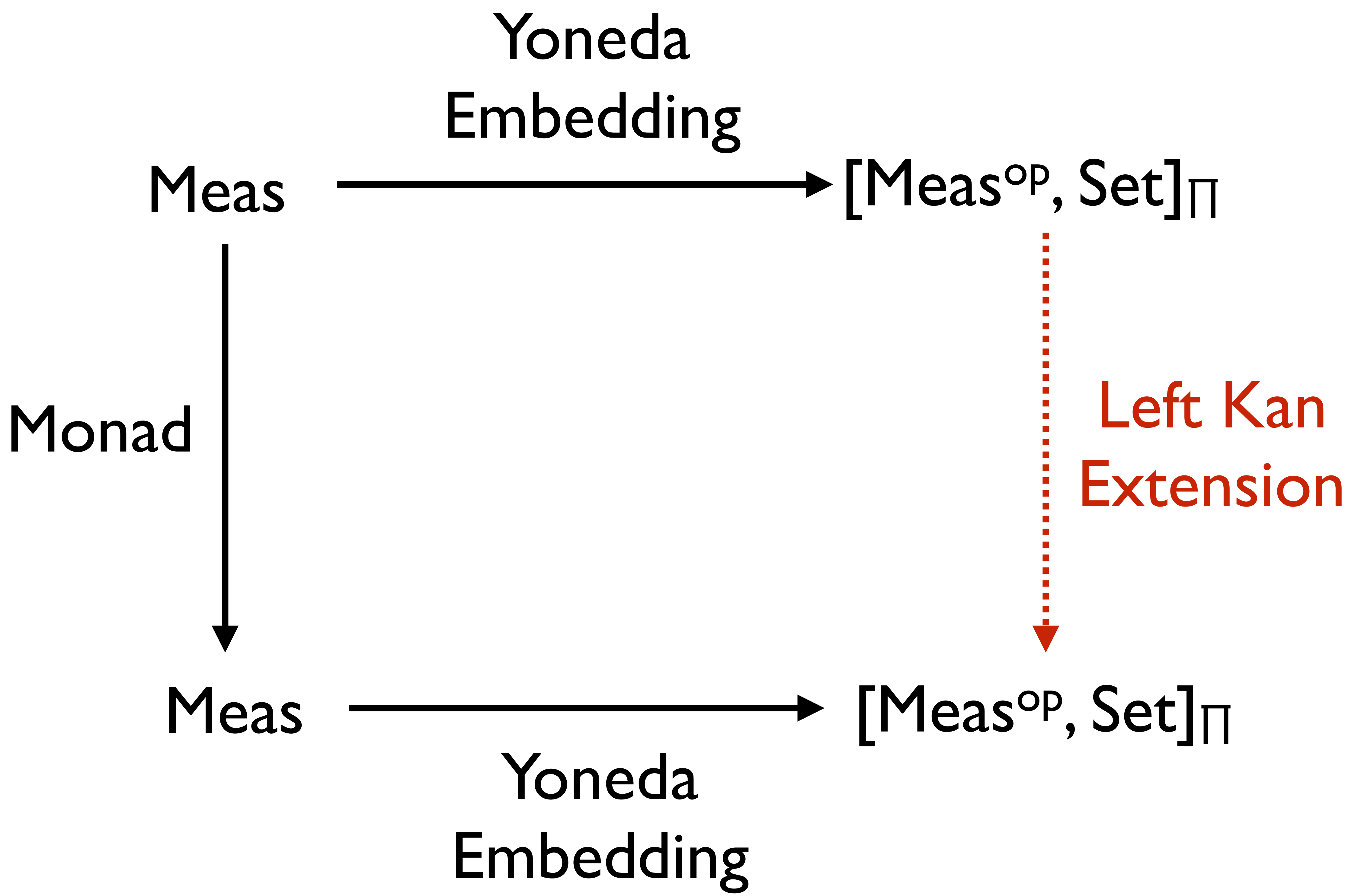
Use category theory to extend measure theory.



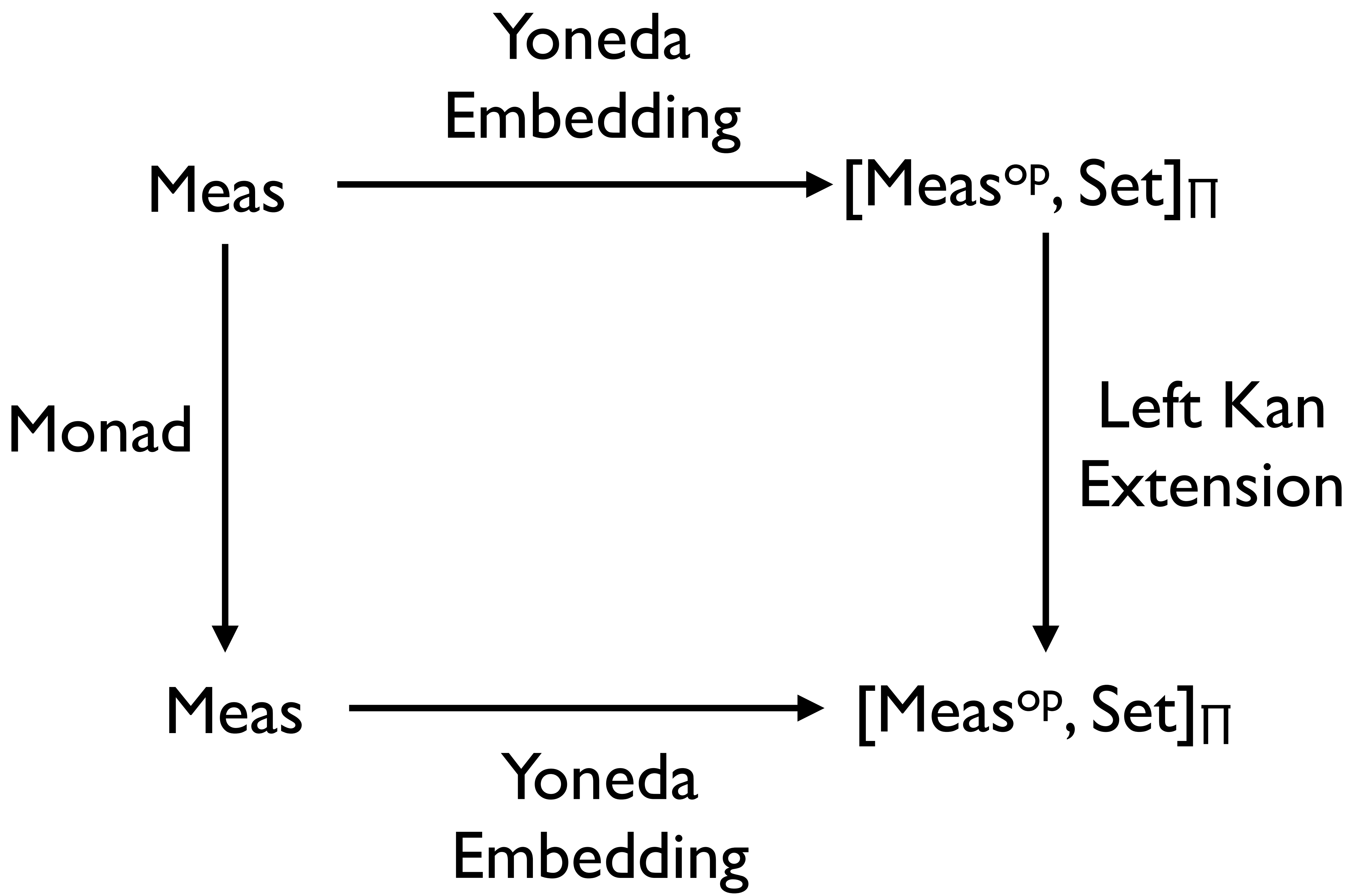
Use category theory to extend measure theory.



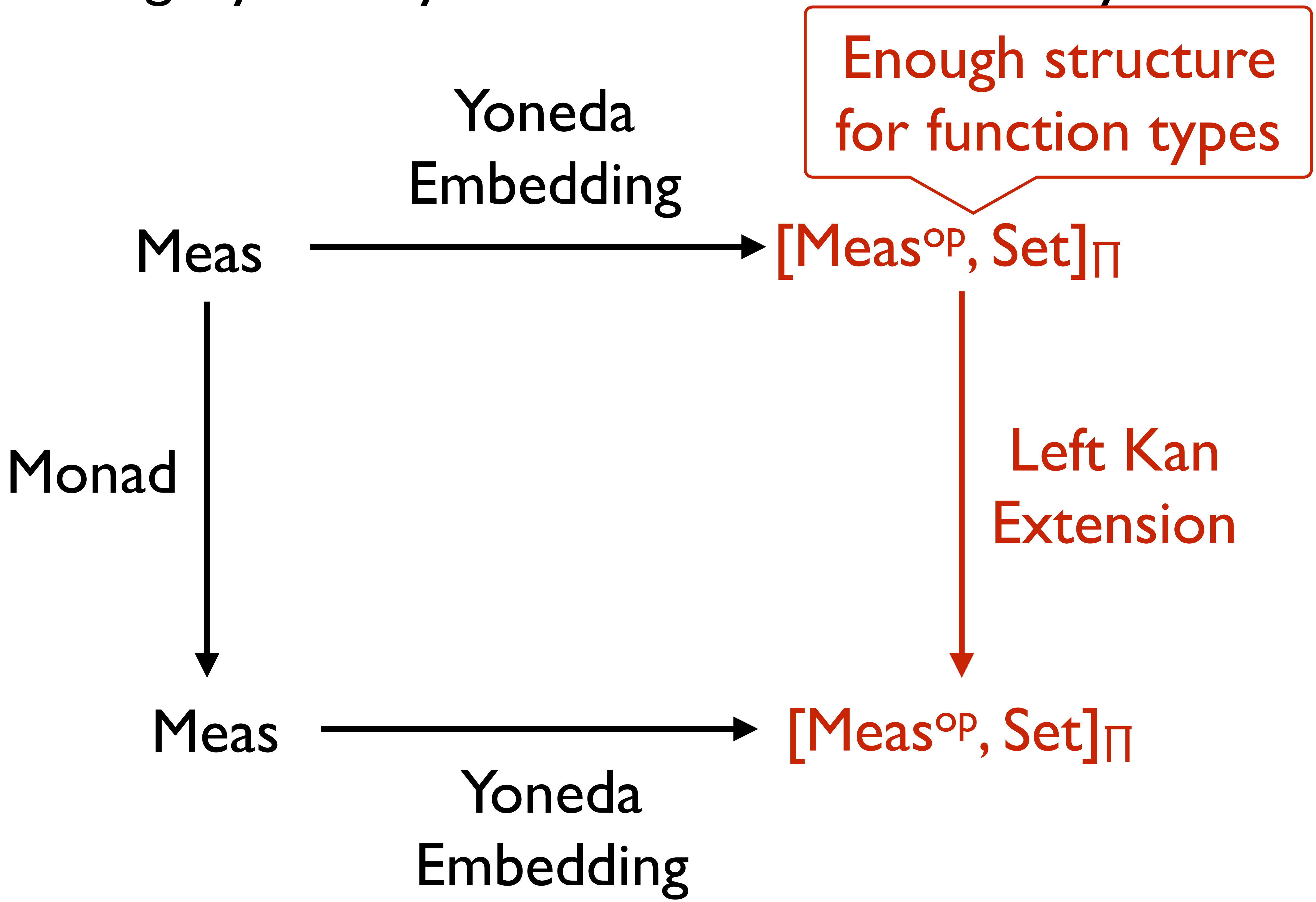
Use category theory to extend measure theory.



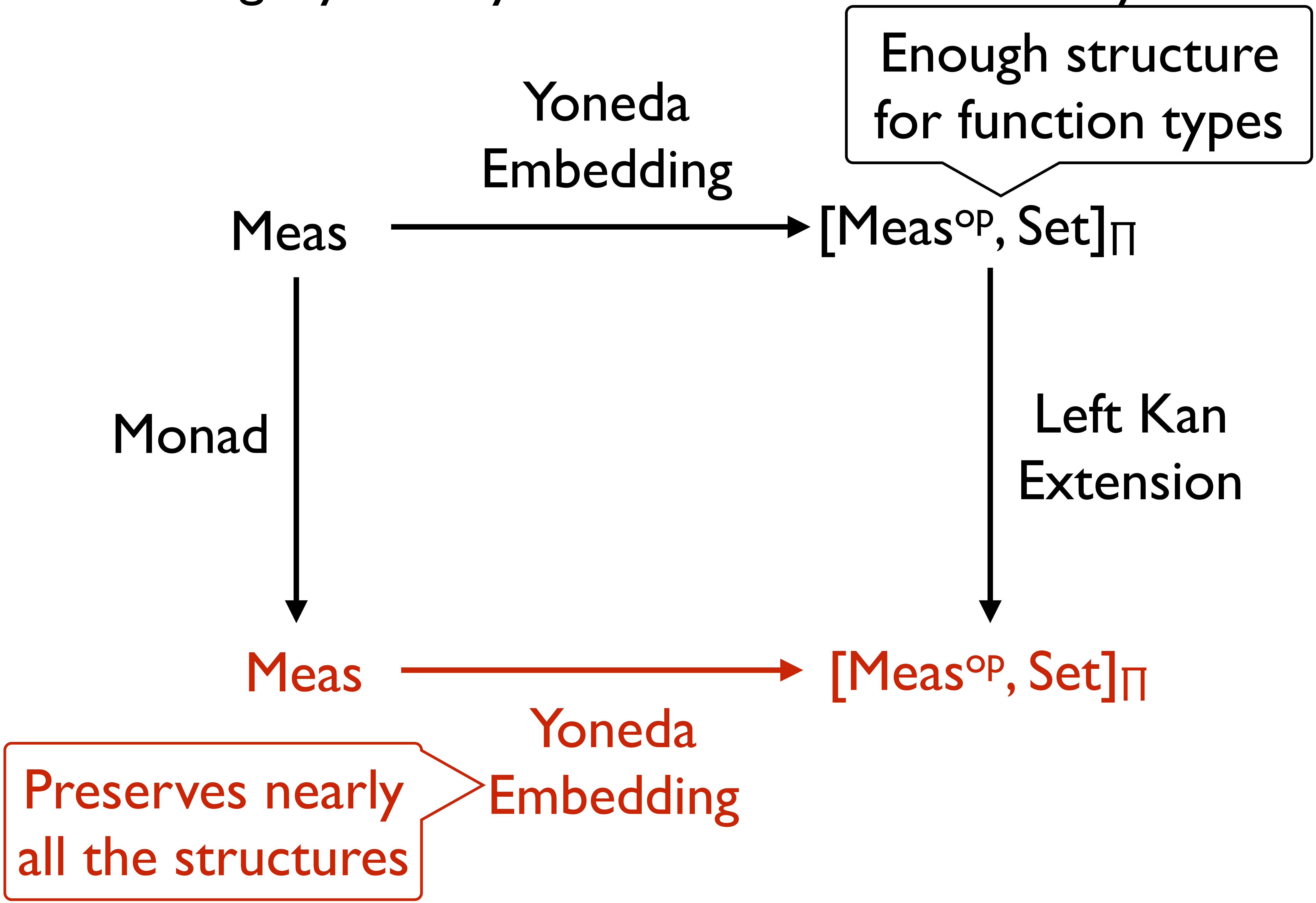
Use category theory to extend measure theory.



Use category theory to extend measure theory.



Use category theory to extend measure theory.



[Question] Are all definable functions from R to R in a high-order probabilistic PL measurable?

Our semantics says that the answer is yes for a core call-by-value language.

The monad $\underline{M}([\![R \to R]\!])$ at $[\![R \to R]\!]$ consists of:

equivalence classes of measurable functions $f : \Omega \times R \to R$ for probability spaces Ω.

The function f is what probabilists call a measurable stochastic process.

The extended monad $\underline{M}$ describes computations with dynamically allocated read-only variables.

$$\underline{M}(F)(w) =$$
$$\{\ [(a, f)]_{\sim}\ \mid\ \exists v.\ a \in F(v)\ \wedge\ f : w \to_m \mathrm{Prob}(R_{\geq 0} \times v)\ \}$$

The extended monad $\underline{M}$ describes computations with dynamically allocated read-only variables.

$$\underline{M}(F)(w) =$$
$$\{ \, [(a, f)]_{\sim} \;\; | \;\; \exists v. \;\; a \in F(v) \;\wedge\; f : w \to_{m} \mathrm{Prob}(R_{\geq 0} \times v) \, \}$$

F is the type of a value.

The extended monad $\underline{M}$ describes computations with dynamically allocated read-only variables.

$$\underline{M}(F)(w) =$$
$$\{\ [(a, f)]_{\sim}\ \mid\ \exists v.\ a \in F(v)\ \wedge\ f : w \rightarrow_m \mathrm{Prob}(R_{\geq 0} \times v)\ \}$$

F is the type of a value.

w represents a space of all random vars so far.

The extended monad $\underline{M}$ describes computations with dynamically allocated read-only variables.

$$\underline{M}(F)(w) =$$
$$\{\ [(a, f)]_{\sim}\ \mid\ \exists v.\ a \in F(v) \wedge f : w \to_m \mathrm{Prob}(R_{\geq 0} \times v)\ \}$$

F is the type of a value.
w represents a space of all random vars so far.
v extends w with new random variables according to f.

Further details

- Can be found in our archive paper:

  http://arxiv.org/pdf/1601.04943.pdf