Models of Randomness

Part I: a sketchy survey

Fritz Obermeyer

Department of Mathematics
Carnegie-Mellon University

2008:04:08

Outline

Motivational outline

Want to prove equivalence between randomized algorithms.

Need a programming language with random monad; and a domain-theoretic model of this language.

This talk surveys some attempts at models.

Later Part II tries to find a random monad in a type-as-ambiguity framework (closures).

Bayesian networks

Consider entirely first-order programs, with no looping/recursion, e.g.,

sample x from unif$(0, 1)$ in
sample y from unif$(0, \text{x})$ in
sample z from unif$(\text{x}, 0)$ in
if $\text{y} + \text{z} < 1/2$ then 0 else 1

(notice we can forget x after sampling y,z)
This is a Bayesian network (see picture).
All Bayesian networks are so expressible.

Markov chains

Consider iterative loops, e.g.

$\mathsf{x} \;\leftarrow\; \mathsf{sample\ x0\ from\ normal}(0,1) \in \mathsf{x0}.$
$\mathsf{while}\ \ \ldots\ \ :$
$\quad\quad \mathsf{sample\ n\ from\ normal}(0,1)\ \mathsf{in}$
$\quad\quad \mathsf{let\ x}' = 1\ +\ \mathsf{x}/2\ \mathsf{in}$
$\quad\quad \mathsf{x}\ \leftarrow\ \mathsf{x}'\ +\ \mathsf{n}$

This is a Markov chain (see picture).
All Markov chains are so expressible.

Prototype for a probability theory

- set of states/points (maybe with structure)
- space of events/predicates
- morphisms between state-event spaces
- random monad = probability distributions
- extra structure: products, exponentials, ...

We start with finite sets,
generalize to probability measures, then
weaken the event language,
add structure among points, and
end with fully algebraic approaches.

Finite sets: states, morphisms, extra structure

Start with a finite set X of states.
No need for event spacee.

Morphisms are just functions,
product, exponenial are standard.
NO recursive types, NO infinite types

Finite sets: Random functor?

Like the powerset, Rand is functorial,
On objects: $\mathsf{Rand}((\mathsf{X},\mathbf{F})) = (\mathsf{X}',\mathbf{F}')$ where

- $\mathsf{X}' \;=$ probability measures over $(\mathsf{X},\mathbf{F})$
- $\mathbf{F}' \;=$ generated by (right?)

$$\{\ \{\ \mathsf{f} \in \mathsf{X}' \ \mid\ \mathsf{f}^{-1}\mathsf{B} \ \subseteq\ \mathsf{A}\ \}\ \mid\ \mathsf{A} \in \mathbf{F},\ \mathsf{B} \in \mathsf{G}\ \}$$

On arrows: for $\mathsf{h} : (\mathsf{X},\mathbf{F}) \to (\mathbf{Y},\mathsf{G})$,
$\mathsf{Rand}(\mathsf{h}) = \mathsf{h}' : (\mathsf{X}',\mathbf{F}') \to (\mathbf{Y}',\mathsf{G}')$ where for $\mathsf{p} : \mathsf{X}'$ a probability measure on $(\mathsf{X},\mathbf{F})$, $\mathsf{B} \in \mathsf{G}$,

$$\mathsf{Rand}(\mathsf{h})(\mathsf{p})(\mathsf{B}) = \mathsf{p}(\mathsf{h}^{-1}\mathsf{B})$$

But random functor doesn't land in finite sets.

Random monad (pieces)

The probability functor forms a monad with natural

$$\text{always} : \forall a.\ a \rightarrow \text{Rand } a$$
$$\text{mix} : \forall a.\ \text{Rand}(\text{Rand } a) \rightarrow \text{Rand } a$$

equivalently a Kleisli triple with 'always' and

$$\text{sample} : \forall a, b.\ \text{Rand } a \rightarrow (a \rightarrow \text{Rand } b) \rightarrow \text{Rand } b$$

In finite-sets, semantics is

$$[\text{always } x](y) = \delta_{x,y}$$
$$[\text{mix } p](x) = \int [p](q)\ q(x)\ dq$$ *oops: p infinite*
$$[\text{sample } p\ f](y) = \sum_x [p](x)\ [f](y)(x)$$

Random monad (properties)

Being a monad requires equations

$$\begin{array}{l}
\mathsf{sample\ x\ from\ p\ in\ always\ x} = \mathsf{p}, \\
\mathsf{sample\ x\ from\ always\ y\ in\ f\ x} = \mathsf{f\ y}, \\
\mathsf{sample\ y\ from\ (sample\ x\ from\ p\ in\ f\ x)\ in\ g\ y} \\
\qquad = \ \mathsf{sample\ x\ from\ p\ in} \\
\qquad\quad \mathsf{sample\ y\ from\ f\ x\ in} \\
\qquad\quad \mathsf{g\ y}
\end{array}$$

Being a computational monad (a la Moggi) requires also:

- 'always' is mono
- monad plays nicely with products and sums

Probability measures: states, events, morphisms

Start with a state set X (unstructured).
The event space is a sigma-algebra F of X,
a structure $\langle \mathsf{X},\ \mathbf{F} \subseteq \mathcal{P}(\Omega),\ \neg,\ \bigvee^{\leq\omega} \rangle$,
where $\perp = \bigvee\ \emptyset,\ \top = \bigvee\ \mathbf{F}$ are definable.

A morphism is a sigma-algebra hom, a measurable function $\mathsf{f}\colon\ \mathsf{X} \rightarrow \mathbf{Y}$, whose preimage induces a hom

$$\mathsf{f}'\ :\ \langle \mathbf{F}, \neg, \biguplus \rangle\ \leftarrow\ \langle \mathsf{G}, \neg, \biguplus \rangle$$

Measures: Random states

A random state is a probability measure, a hom

$$\langle \mathbf{F}, \emptyset, \mathsf{X}, \biguplus^{\omega} \rangle \;\longrightarrow\; \langle [0,1], 0, 1, \sum^{\omega} \rangle$$

i.e., functions p satisfying

$\mathsf{p}(\bot) = 0$
$\mathsf{p}(\top) = 1$
$\mathsf{p}(\biguplus_{\mathsf{i}}\ \mathsf{A_i}) = \sum_{\mathsf{i}}\ \mathsf{p}(\mathsf{A_i})$

and hence $\mathsf{p}(\neg\ \mathsf{A}) = 1 - \mathsf{p}(\mathsf{A})$

Question equivalent to additivity+continuity?

Measures: extra structure

Product sigma-algebras are generaged by rectangles
Exponentials have pointwise sigma-algebra (right?)

NO untyped/unityped model of lambda-calculus

Probability valuations

Relax event logic from sigma-algebra to topology;
abstract away points to frames/locales/CHAs.

Definition

A probability valuation on F is a monotone $\mathsf{p} : \mathbf{F} \to [0,1]$ satisfying

$$\mathsf{p}(\bot) = 0, \quad \mathsf{p}(\top) = 1$$
$$\mathsf{p}(\mathsf{A}) + \mathsf{p}(\mathsf{B}) = \mathsf{p}(\mathsf{A} \sqcap \mathsf{B}) + \mathsf{p}(\mathsf{A} \sqcup \mathsf{B})$$

we also assume continuity (some authors don't).
(analogous to countable additivity?)

...extension theorems e.g.

Theorem

(Jones) Every continuous valuation on a continuous dcpo

Directed-complete partial orders.

Start with a dcpo of states.
Events are Scott-open sets (a frame).
Morphisms are Scott-continuous functions
(preserving joins, inducing frame-homs).

Directed joins of random things are random things
so dcpo is closed under random monad.

dcpo is also closed under products, function spaces,
coinductive typess, ...

Continuous dcpos

...but dcpos don't have enough structure to be "domains".

Continuous domains have more.
CONT is closed under probability monad.
But NOT closed under function spaces.

Lattices I: states, randomness

Start with a lattice X (e.g. real line).
Let event space be the upper sets.
To each valuation p, define a cpdf

$$p'(x) = p(\text{upper } x)$$

Dually, each cpdf d extends to a unique valuation

$$d'(\text{upper } x) = d(x)$$

Try again: (no event space)
Morphisms are lattice homs.
Random states are cpdfs, but...

NO random monad

A space of random lattice elements need not itself be a lattice.

Hence Rand∘Rand need not exist.

Example

the square lattice $\bot \sqsubseteq \mathsf{tr}, \mathsf{fa} \sqsubseteq \top$,

$$\bot + \mathsf{tr} \mid \bot + \mathsf{fa} \sqsubseteq \bot + \top,\ \mathsf{tr} + \mathsf{fa}$$

no unique minimal upper bound

The max of two cdfs may lead to negative densities.

Abstract probability algebras

Start with a dcpo with $\perp$.
Generate initial "R-algebra" with binary mixing $x + y$ subject to monotonicity and

$x + x = x$	*idempotence*
$x + y = y + x$	*commutativity*
$(\omega + x) + (y + z) = (\omega + z) + (y + x)$	*associativity*

- equivalent to arbitrary real mixing
- equivalent to valuations

Compare with initial join-semilattice ("J-algebra")

$$x \mid x = x$$
$$x \mid y = y \mid x$$
$$x \mid (y \mid x) = (x \mid y) \mid z$$

Probability and lattices

Problem the join/meet of two random things may not be a random thing.

R-algebra models randomness.
J-algebra models parallelism.
JR-algebra with distributivity.

$$(x + y) \mid z = (x \mid z) + (y \mid z)$$

models parallelism with randomness,
allows random normal form,
sampling semantics
JR-algebra models... nothing nice,
NO random normal form
Next time: can JR-algebras be made to work?

Summary and prospects

(again)
We started with finite sets,
generalized to probability measures, then
weakened the event language,
added structure among points, and
ended with fully algebraic approaches.