Models of Randomness

Part II: random types-as-closures

Fritz Obermeyer

Department of Mathematics
Carnegie-Mellon University

2008:04:15

Notation and concepts

SK $=$ combinatory algebra
SKR $=$ random extension ($R = \lambda x, y.x + y$)
(note $(x + y) + z \neq x + (y + z)$; really $\frac{x+y}{2}$)
SKJ $=$ parallel/ambiguous extension ($\mathbf{J} = \lambda x, y.x \mid y$)
(some authors write join $x \mid y$ instead $x \sqcup y$)
SKJR $=$ random$+$ambiguous extension

All four assume HP-complete observational equivalence:
$M \sqsubseteq M'$ iff $\forall \underline{N}.\ M\ \underline{N}$ conv $\implies M'\ \underline{N}$ conv
where $\underline{N}$ ranges over traces $N_1, \ldots, N_k$ of arguments.

Definable types-as-closures builds on 2007 talk,
`http://www.math.cmu.edu/~fho/notes/`

Goal: a probability monad

In a curry type system in applied λ-calculi
(where types are properties of untyped terms),
we seek an instance of Moggi's computational monad

$$\begin{array}{lll} \text{Rand} & : & \text{type} \to \text{type} \\ \text{always} & : & \forall\alpha.\ \alpha \to \text{Rand}\ \alpha \\ \text{sample} & : & \forall\alpha,\beta.\ \text{Rand}\ \alpha \to (\alpha \to \text{Rand}\ \beta) \to \text{Rand}\ \beta \end{array}$$

In which applied λ-calculus should we look?
extend SK to randomized SKR.
Is Rand compatible with types-as-closures?
extend SK to random lattice SKJR.
look for "small" definable subspaces.

Abstract probability algebras

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

$$
\begin{array}{ll}
x + x = x & \textit{idempotence} \\
x + y = y + x & \textit{commutativity} \\
(w + x) + (y + z) = (w + z) + (y + x) & \textit{associativity}
\end{array}
$$

- equivalent to arbitrary real mixing (in dcpo)
- equivalent to valuations

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

$$
\begin{array}{l}
x \mid x = x \\
x \mid y = y \mid x \\
x \mid (y \mid x) = (x \mid y) \mid z
\end{array}
$$

Extending SK to SKR

Start with untyped combinatory algebra SK.
Extend to R-algebra with right-distributivity

$$(\mathsf{f} + \mathsf{g})\mathsf{x} = (\mathsf{f}\ \ \mathsf{x}) + (\mathsf{g}\ \ \mathsf{x})$$

SKR is randomized Turing-complete.

Consider the curry type system

$$\frac{\mathsf{x}{:}\alpha \vdash \mathsf{M}{:}\beta}{\lambda\mathsf{x}.\mathsf{M}{:}\alpha \to \beta} \qquad \frac{\mathsf{M}{:}\alpha \to \beta \qquad \mathsf{N}{:}\alpha}{\mathsf{M}\ \ \mathsf{N}{:}\beta} \qquad \frac{\mathsf{M}, \mathsf{N}{:}\mathsf{Rand}\ \ \alpha}{\mathsf{M} + \mathsf{N}{:}\mathsf{Rand}\ \ \alpha}$$

and embed SK into SKR via always $= \lambda\mathsf{x}.\mathsf{x}{:}\forall\alpha.\alpha \to \mathsf{Rand}\ \ \alpha$.
How to sample, raising $\alpha \to \mathsf{Rand}\ \ \beta$ to $\mathsf{Rand}\ \ \alpha \to \mathsf{Rand}\ \ \beta$?

Sampling booleans

Introduce $\mathbf{K} = \lambda \mathsf{x}, \mathsf{y}.\mathsf{x}$, $\mathbf{F} = \lambda \mathsf{x}, \mathsf{y}.\mathsf{y}$ with typing

$$\frac{}{\mathbf{K} : \mathsf{bool}} \qquad \frac{}{\mathbf{F} : \mathsf{bool}} \qquad \frac{\mathsf{M} : \mathsf{bool} \qquad \mathsf{N}, \mathsf{N}' : \alpha}{\mathsf{M}\ \mathsf{N}\ \mathsf{N}' : \alpha}$$

We can sample with

$$\mathsf{sample}_{\mathsf{bool}} = \lambda \mathsf{p}, \mathsf{f}.\ \mathsf{p}\ (\mathsf{f}\ \mathbf{K})\ (\mathsf{f}\ \mathbf{F})$$

Letting, e.g.,

$\mathsf{amb} \ := \ \lambda \mathsf{x}.\ \mathsf{x}\ (\mathsf{x}\ \mathbf{K}\ \mathbf{F})\ (\mathsf{x}\ \mathbf{F}\ \mathbf{K})$ *ambiguity*
$\mathsf{R} \ := \ \mathbf{K} + \mathbf{F}$, *even coin toss*

note the difference between

$\mathsf{amb}\ \mathsf{R} \ = \ \mathbf{K} + \mathbf{F}$ *random*
$\mathsf{sample}_{\mathsf{bool}}\ \mathsf{x}\ \mathsf{from}\ \mathsf{R}\ \mathsf{in}\ \mathsf{amb}\ \mathsf{x}$ *...sugar for*
$\quad = \ \mathsf{sample}_{\mathsf{bool}}\ \mathsf{R}\ \lambda \mathsf{x}.\ \mathsf{amb}\ \mathsf{x}$
$\quad = \ \mathbf{K}$ *deterministic*

Sampling natural numbers

Let $\mathsf{zero} = \lambda\_, \mathsf{x.x}$, $\mathsf{succ} = \lambda \mathsf{n}, \mathsf{f}, \mathsf{x}.\ \mathsf{f}(\mathsf{n}\ \mathsf{f}\ \mathsf{x})$ with typing

$$\frac{}{\mathsf{zero} : \mathsf{nat}} \qquad \frac{}{\mathsf{succ} : \mathsf{nat} \to \mathsf{nat}} \qquad \frac{\mathsf{n} : \mathsf{nat} \qquad \mathsf{s} : \alpha \to \alpha \qquad \mathsf{z} : \alpha}{\mathsf{n}\ \mathsf{s}\ \mathsf{z} : \alpha}$$

and sample with

$$\mathsf{sample}_{\mathsf{nat}} = \lambda \mathsf{p}, \mathsf{f}.\ \mathsf{p}\ (\mathsf{f} \circ \mathsf{succ})\ (\mathsf{f}\ \mathsf{zero})$$

then lift $\mathsf{succ} : \mathsf{nat} \to \mathsf{nat}$ to $\mathsf{succ}' : \mathsf{Rand}\ \mathsf{nat} \to \mathsf{Rand}\ \mathsf{nat}$

$$\mathsf{succ}' = \lambda \mathsf{p}.\ \mathsf{sample}_{\mathsf{nat}}\ \mathsf{p}\ \mathsf{succ}$$

Moral Church terms are already monadic

Sampling partial terms

General method for defining $\texttt{sample}_\alpha$ p f:

- enumerate domain α
- apply f to each x: α
- case-branch on x based on p

Method doesn't yet work for partial terms,
e.g. infinite domains, functions.

Method works if directed joins are definable,
e.g. in lattice models with randomness...
...but first look at typing in lattice models.

Types-as-closures

Curry-types are properties of untyped terms.
Consider closures in combinatory algebra with join "SKJ",

$$\mathbf{I} \sqsubseteq \alpha = \alpha \circ \alpha \qquad \textit{extensive, idempotent}$$

and the fixedpoint property $\mathsf{x} : \alpha$ iff $\mathsf{x} = \alpha\ \mathsf{x}$.
Typechecking becomes an equational problem.
Typing can be enforced at the term level:
e.g. (bool x) will be "boolean" for any x.
But we get extra "garbage" inhabitants, e.g.

$$\mathrm{inhab}(\mathsf{bool}) = \{\bot, \mathbf{K}, \mathbf{F}, \top\}$$

To minimize garbage, look for closures with small ranges.

Defining closures

How can we define closures?

General idea: join represents ambiguity.
When expecting a bool x, squint
(join over all ways of looking at x).
Only booleans remain unblurred,
all else blurs to $\top$=complete ambiguity.

Specifically: argue about action on Bohm trees.

What types are definable?
unit, bool, nat, Prod, Sum, Exp, recursive,
polymorphic, dependent, subtypes, type-of-types

(see 2007 talk on definable closures for details)

A lattice with randomness

Seeking a random monad in a lattice model,
consider the free JR-algebra over SK, where
application right-distributes over both

$$(f \mid g)x = f\ x \ \mid \ g\ x$$
$$(f + g)x = f\ x \ + \ g\ x$$

Call R-terms mixtures, J-terms joins,
RJ-terms slurries, e.g. $w \mid (x + (y \mid z))$
Application right-distributes over slurries,
but RJ distributivity fails,
so slurries can be infinitely deep.

JR-Bohm-trees

JR-Bohm trees are defined by

$$\frac{\mathsf{x}\ \ \mathsf{var} \qquad \mathsf{M}_1, \ldots, \mathsf{M}_k\ \ \mathsf{BT}}{\mathsf{x}\ \ \mathsf{M}_1\ \ \ldots\ \ \mathsf{M}_k\ \ \mathsf{BT}} \qquad \frac{\mathsf{x}\ \ \mathsf{var} \qquad \mathsf{M}\ \ \mathsf{BT}}{\lambda \mathsf{x}.\mathsf{M}\ \ \mathsf{BT}}$$

$$\frac{\underline{\mathsf{M}}\ \ \mathsf{Set}(\mathsf{BT})}{\bigsqcup \underline{\mathsf{M}}\ \ \mathsf{BT}} \qquad \frac{\underline{\mathsf{M}}\ \ \mathsf{Prob}(\mathsf{BT})}{\int \underline{\mathsf{M}}\ \ \mathsf{BT}}$$

Theorem

Every SKJR-term is equivalent to a slurry of BTs.

Ambiguity vs. parallelism

Historical operational semantics for join is parallelism.
Operational semantics for randomness is sampling.
Combining these requires distributivity

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

but distributivity fails in SKJR, e.g.

$$\begin{aligned}
&(\mathbf{I} \mid \bot + \top)\ (\bot + \top) \\
&\qquad = \ \bot + \top \mid \bot + \top \ = \bot + \top \\
&((\mathbf{I} \mid \bot) + (\mathbf{I} \mid \top))\ (\bot + \top) \\
&\qquad = \ \mathbf{I}(\bot + \top) + \top(\bot + \top) \ = (\bot + \top) + \top
\end{aligned}$$

We can still sample, but never in parallel, inside joins.
So we'll require R-normal forms for "nice" inhabitants.
For example...

Random convergence/divergence

The smallest nontrivial closure is

$$\mathsf{div} \;:=\; \lambda \mathsf{x}.\ \mathsf{x} \mid \mathsf{x}\ \top \mid \mathsf{x}\ \top\ \top \mid \mathsf{x}\ \top\ \top\ \top \mid \ldots$$

Theorem

In SKJ, $\mathrm{inhab}(\mathsf{div}) = \{\bot, \top\}$*.*

Since $\{\bot, \top\}$ is the two-point lattice,
we get R-normal forms for div, and thus

Theorem

In SKJR, $\mathrm{inhab}(\mathsf{div}) = \{\bot@\mathsf{t} + \top@(1-\mathsf{t}) \ \mid\ \mathsf{t}\ \in\ [0,1]\}$*.*

Convergence in SKJR is probabilistic;
observational information ordering is defined by
$\mathsf{M} \sqsubseteq \mathsf{M}'$ iff $\forall$trace $\underline{\mathsf{N}}$. $\mathrm{conv}(\mathsf{M}\ \underline{\mathsf{N}}) \leq \mathrm{conv}(\mathsf{M}'\ \underline{\mathsf{N}})$.

The range property, spectra

(recall we want small ranges, few inhabitants)
In SK, all nontrivial ranges are infinite.

In SKJ, some nontrivial ranges are finite,
e.g., $|\text{rng}(\top)| = 1$, $|\text{rng}(\text{div})| = 2$
$|\text{rng}(\text{semi})| = 3, \ldots$ ΛPAUSE In

SKJR, all nontrivial ranges are infinite,
but some are finite dimensional,
e.g., $\dim(\text{rng}(\top)) = 0$, $\dim(\text{rng}(\text{div})) = 1$.

Question What is the spectrum of dimensions?

Known At least $\{0, 1, 2^{\aleph_0}\}$.

Are there less trivial types?

In SKJ we can define types boool and semi with

$$\mathrm{inhab}(\text{boool}) = \{\bot, \mathbf{K}, \mathbf{F}, \mathbf{J}, \top\}$$
$$\mathrm{inhab}(\text{semi}) = \{\bot, \mathbf{I}, \top\}$$

and disambiguate $\lambda \text{x.x}(\text{x } \mathbf{K} \ \top)(\text{x } \top \ \mathbf{F}) : \text{boool} \to \text{bool}$,

$$\mathrm{inhab}(\text{bool}) = \{\bot, \mathbf{K}, \mathbf{F}, \top\}$$

Similarly in SKJR we can define semi, bool with inhabitants slurries of $\{\bot, \mathbf{I}, \top\}$ and $\{\bot, \mathbf{K}, \mathbf{F}, \top\}$.

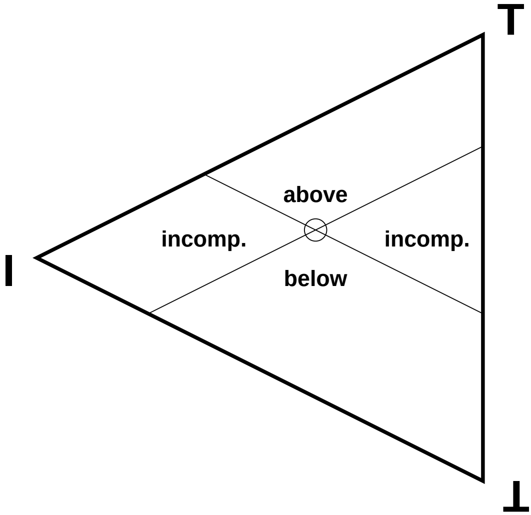
The $\text{Slurry}_\alpha, \text{sample}_\alpha, \text{always}_\alpha$ monad exists for many types: bool, nat, Prod, Sum, Exp, recursive. (not obvious)

But slurries aren't "nice": no R-normal forms, no sampling semantics.

Can we disambiguate to finite dimensional mixtures, without determinizing? (unknown)

Mixtures of semibooleans

The space of $\perp, \mathbf{I}, \top$-mixtures is 2-dimensional:



Slurries of semibooleans

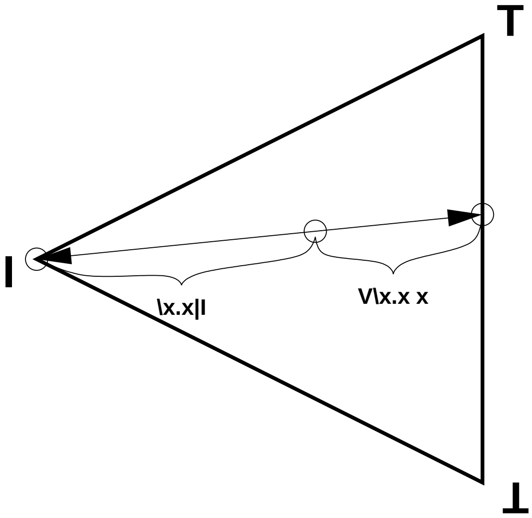
Theorem

$\perp, \mathbf{I}, \top$-slurries are completely characterized by their action on the unit interval $[\perp, \top]$.

We thus have JR-isomorphism with the
monotone convex functions $: [0,1] \to [0,1]$.
This space has dimension $2^{\aleph_0}$.
Can we raise to a finite-dimensional subspace?

A 1-dimensional subspace

Try raising with Vλx.x x and then Vλx.x $\mid$ $\mathbf{I}$:



A 1-dimensional subspace

Try raising with Vλx.x x and then Vλx.x $|$ $\mathbf{I}$:

Range is 1-dimensional,
almost has R-n.f. (binary join of mixtures)
but arguably useless...

The closure Vλx.x x is more useful
but has infinite dimensional range.

Mixtures of booleans

In SKJ, $\mathrm{inhab}(\mathsf{boool}) = \{\bot,\ \mathbf{K},\ \mathbf{F},\ \mathbf{J} = \mathbf{K} \mid \mathbf{F},\ \top\}$;
a "disambiguation" trick then raises $\mathbf{J}$ to $\top$
yielding $\mathrm{inhab}(\mathsf{bool}) = \{\bot, \mathbf{K}, \mathbf{F}, \top\}$.
In SKJR, disambiguation interferes with randomness,
and we're stuck with $\bot, \mathbf{K}, \mathbf{F}, \mathbf{J}, \top$-mixtures.

$\bot, \mathbf{K}, \mathbf{F}, \mathbf{J}, \top$-mixtures are 5-dimensional.
Cumulative distribution functions yield a model:

$$\{\ (\mathsf{b}, \mathsf{k}, \mathsf{f}, \mathsf{j}, \mathsf{t}) \in [0,1]^5 \ \mid\ \mathsf{k} \geq \mathsf{b},\ \ \mathsf{f} \geq \mathsf{b},\ \ \mathsf{j} + \mathsf{b} \geq \mathsf{k} + \mathsf{f},\ \ \mathsf{t} \geq \mathsf{j}\ \}$$

partially ordered componentwise.

Slurries of booleans

Theorem

$\bot, \mathbf{K}, \mathbf{F}, \top$-slurries are completely characterized by their action on two arguments in unit interval $[\bot, \top]$.

We thus have JR-isomorphism with the
monotone convex functions $: [0,1]^2 \to [0,1]$.
Can we raise to a finite-dimensional subspace?
Want to raise to a linear upper bound.

We can raise $\bot, \mathbf{K}, \mathbf{F}, \top$-slurries with λx.x x x.
and partially disambiguate with λx.x(x $\mathbf{K}$ $\top$)(x $\top$ $\mathbf{F}$).
???

Problems with SKJR

Problem Disambiguation fails:
even if we can raise slurries to mixtures,
random samples may be ambiguous.

Unambiguity can be checked equationally,
but not enforced at the term level.

Problem Lower-bounding fails:
without distributivity, we can't raise e.g.
$\perp + \top$ to $\mathbf{I} + \top$.

Adding parallelism allows lower-bounding,
but then requires sequentialization...