From Hard Work to Trickery: A Systematic Approach to Probabilistic Rewriting.

Flavien BREUVART, Ugo Dal Lago

Focus, Inria team, Bologna

Crecogi: August 28 2016

Probabilistic Rewriting

Motivations

Randomized Algorithmics

Efficiency analysis

Cryptography

Security

Machine Learning

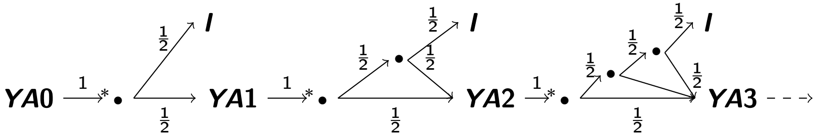
Modeling

Probabilistic λ-calculus (Weak head-reduction)

$$M, N \; := \; x \mid \lambda x.M \mid M \; N \mid M \oplus N$$

$$(\lambda x.M)\; N \xrightarrow{1} M[N/x]$$

$$\begin{tikzcd} & M \\ M \oplus N \arrow[ur, "\frac{1}{2}"] \arrow[dr, "\frac{1}{2}"'] & \\ & N \end{tikzcd}$$



$\boldsymbol{A} := \lambda ux.x(\lambda y.(u(\boldsymbol{S}x) \oplus y)\boldsymbol{I})$

$\mathrm{Prob}(\boldsymbol{YA0} \Downarrow) < 1$

The Quest for a Semantics

Denotational Semantics: a 20 years old challenge

Finding subclass of domains that:

- is a Cartesian close
- have probabilistic powerdomains

Real issue:
higher order probabilities

Operational Semantics: a hidden challenge

Rewriting theory with:

- probabilistic behaviors
- systematic proof-schemes

Real issue:
proba forces topological arguments

Unconventional solution: Probabilistic coherent spaces

[EhrhardPaganiTasson2014]

Unconventional solution ???

Our objective

What is a Systematic Proof Schema?
The Example of Infinite Rewriting

Question:

How to relate small-step and multi-step?

At the beguiling: Topology

Limits for Cantor topology of sequential small-step reductions.

Now-day: Coinduction

$$\frac{M \to^* f(L_1,\dots,L_k) \qquad \forall i \leq k,\ L_i \to^\omega N_i}{M \to^\omega f(N_1,\dots,N_k)}$$

Coinduction Schema

For any relation $\rightsquigarrow$ over terms, if for all $M \rightsquigarrow f(N_1,...,N_k)$, there is $L_1,...,L_k$ such that $M \to^* f(L_1,...,L_k)$ and $L_i \rightsquigarrow N_i$, then $\rightsquigarrow \subseteq \to^\omega$.

What About Probabilistic Rewriting

Probabilities and Non-determinism does not mix well

For now, let's forget about non determinism. This means:

Fixing a strategy **Big step rather than multistep**

Probabilities are inherently topological

$[0,1]$ is, before all, a topological space...

Most rewriting theory's tools are continuous

Bisimulations, encoding, typing, modeling...

Can we treat those tools without referring to topology?

What About Probabilistic Rewriting

Probabilities and Non-determinism does not mix well

For now, let's forget about non determinism. This means:

Fixing a strategy **Big step rather than multistep**

Probabilities are inherently topological

$[0,1]$ is, before all, a topological space...

Most rewriting theory's tools are continuous

Bisimulations, encoding, typing, modeling...

Can we treat those tools without referring to topology?

Yes! but there is a price to pay: a dynamic target

Probabilistic Rewriting System

Randomized function

$f : U \twoheadrightarrow V$ denotes a function $f : U \to \mathscr{D}(V)$

$\mathscr{D}(V) = \{d \in V \to [0,1] \mid \Sigma_{v \in V} d(v) \leq 1\}$

Definition of (Abstract) Probabilistic Rewriting System

Terms	**Normal forms**	**Small step**
Λ	$\Lambda = \Lambda_V \uplus \Lambda_R$	$\mathtt{reduction} : \Lambda_R \twoheadrightarrow \Lambda$

Remark: We only consider deterministic systems (with a strategy)

Coalgebraic approach of the big step reduction [Hasuo]

The evaluation $\mathtt{eval} : \Lambda \twoheadrightarrow \Lambda_V$ corresponds to the final arrow
in the $(\_ \times \mathbb{N})$-coalgebra category over set and randomized functions.

Theorem: Randomized Encoding

Λ, Π	probabilistic rewriting systems.
$\texttt{encoding} : \Lambda \twoheadrightarrow \Pi$	a randomized function preserving NF and reducibles.

$$
\begin{tikzcd}
\Lambda_R \arrow[r, two heads, "\text{encoding}"] \arrow[d, two heads, "\text{reduction}"] & \Pi_R \arrow[d, two heads, "\text{reduction}"'] \\
\Lambda \arrow[r, two heads, "\text{encoding}"] & \Pi
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
\Lambda \arrow[r, two heads, "\text{encoding}"] \arrow[d, two heads, "\text{eval}"] & \Pi \arrow[d, two heads, "\text{eval}"'] \\
\Lambda_V \arrow[r, two heads, "\text{encoding}"] & \Pi_V
\end{tikzcd}
$$

Dynamic is Essential

$$
\begin{tikzcd}
\Lambda_R \arrow[dr, "\text{interpret}"] \arrow[dd, two heads, "\text{reduction}"] & \\
& \Pi \\
\Lambda \arrow[ur, "\text{interpret}"'] &
\end{tikzcd}
\qquad \not\Rightarrow \qquad
\begin{tikzcd}
\Lambda \arrow[dr, "\text{interpret}"] \arrow[dd, two heads, "\text{eval}"] & \\
& \Pi \\
\Lambda_V \arrow[ur, "\text{interpret}"'] &
\end{tikzcd}
$$

Only true if $||\texttt{interpret}(M)|| \leq ||\texttt{eval}(M)||$

Require a nontrivial realisability proof.

Example: Performing Choices First

Example

In any (binary) probabilistic rewriting system, the probabilistic choices can be chosen uniformly over $\mathtt{Bool}^\omega$ at the beginning.

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\Lambda_R \arrow[r, two heads, "\mathtt{Unif}(\{_\}\times\mathtt{Bool}^\omega)"] \arrow[d, two heads, "\text{reduction}"] & \Lambda_R \times \mathtt{Bool}^\omega \arrow[d, two heads, "\text{reduction}"'] \\
\Lambda \arrow[r, two heads, "\mathtt{Unif}(\{_\}\times\mathtt{Bool}^\omega)"] & \Lambda \times \mathtt{Bool}^\omega
\end{tikzcd}
\quad\Longrightarrow\quad
\begin{tikzcd}[column sep=huge, row sep=huge]
\Lambda \arrow[r, two heads, "\mathtt{Unif}(\{_\}\times\mathtt{Bool}^\omega)"] \arrow[d, two heads, "\text{eval}"] & \Lambda \times \mathtt{Bool}^\omega \arrow[d, two heads, "\text{eval}"'] \\
\Lambda_V \arrow[r, two heads, "\mathtt{Unif}(\{_\}\times\mathtt{Bool}^\omega)"] & \Lambda_V \times \mathtt{Bool}^\omega
\end{tikzcd}$$

Example: Performing Choices First

Example

In any (binary) probabilistic rewriting system, the probabilistic choices can be chosen uniformly over $\texttt{Bool}^\omega$ at the beginning.

$$\begin{tikzcd}
\Lambda_R \arrow[r, two heads, "\texttt{Unif}(\{_\}\times\texttt{Bool}^\omega)"] \arrow[d, two heads, "\text{reduction}"] & \Lambda_R \times \texttt{Bool}^\omega \arrow[d, two heads, "\text{reduction}"] \\
\Lambda \arrow[r, two heads, "\texttt{Unif}(\{_\}\times\texttt{Bool}^\omega)"] & \Lambda \times \texttt{Bool}^\omega
\end{tikzcd}
\qquad
\begin{tikzcd}
\Lambda \arrow[r, two heads, "\texttt{Unif}(\{_\}\times\texttt{Bool}^\omega)"] \arrow[d, two heads, "\text{eval}"] & \Lambda \times \texttt{Bool}^\omega \arrow[d, two heads, "\text{eval}"] \\
\Lambda_V \arrow[r, two heads, "\texttt{Unif}(\{_\}\times\texttt{Bool}^\omega)"] & \Lambda_V \times \texttt{Bool}^\omega
\end{tikzcd}$$

Our theorem is Valid

for continuous probabilities!

Probabilistic Intersection Types

From Probabilistic Coherence Spaces To Probabilistic Intersection Types

Standard translation:

- compacts points $\rightsquigarrow$ intersection types
- prime algebraic points $\rightsquigarrow$ linear types

$$\frac{\pi}{\vdash M : p{\cdot}\alpha}$$

probabilistic bound or weight

The adequation (reformulation)

$$\mathrm{Prob}(M\Downarrow) \;=\; \sum W(M)$$

where $W(M) = \left[p \;\middle|\; \dfrac{\pi}{\vdash M : p{\cdot}*}\right]$

Underlying function

$$||\mathtt{eval}(M)|| \;=\; ||\mathtt{deriv}(M)|| \qquad \mathtt{deriv} : \begin{pmatrix} \Lambda & \to & \mathscr{D}(\Pi) \\ M & \mapsto & \left\{\dfrac{\pi}{\vdash M : p{\cdot}*} \mapsto p\right\} \end{pmatrix}$$

Probabilistic Intersection Types

From Probabilistic Coherence Spaces To Probabilistic Intersection Types

Standard translation:

- compacts points $\rightsquigarrow$ inters
- prime algebraic points $\rightsquigarrow$

$$\frac{\pi}{\vdash M : p{\cdot}\alpha}$$

p IS NOT the probability

for M to be of type α

Rather, p IS the probability

for π to be a proof of $\vdash M : \alpha$

The ade

Pro

where $W(M) = \left[p \;\middle|\; \frac{\pi}{\vdash M : p{\cdot}*} \right]$

Underlying function

$$||\texttt{eval}(M)|| \;=\; ||\texttt{deriv}(M)|| \qquad \texttt{deriv} : \begin{pmatrix} \Lambda & \rightarrow & \mathscr{D}(\Pi) \\ M & \mapsto & \left\{ \frac{\pi}{\vdash M : p{\cdot}*} \mapsto p \right\} \end{pmatrix}$$

Sketching the Proof of intersection types

Cut Elimination

$$\frac{\pi}{\vdash M : p{\cdot}*} \rightsquigarrow \frac{\pi'}{\vdash M' : q{\cdot}*}$$

such that: $\rightsquigarrow$ is

- normalizing
- deterministic

"Poliadic λ-calculus"

Small-step distrib.

$$\begin{tikzcd} \Lambda_R \arrow[r, two heads, "\text{deriv}"] \arrow[d, two heads, "\text{reduction}"] & \Pi_R \arrow[d, "\text{red.}"] \\ \Lambda \arrow[r, two heads, "\text{deriv}"] & \Pi \end{tikzcd}$$

Value determinism

$\forall$ normal form V,

Unicity of derivation

$$\vdash V : 1{\cdot}*$$

$$\frac{}{\vdash \lambda x.M : 1{\cdot}*}$$

Big-step distribution

$$\begin{tikzcd} \Lambda \arrow[r, two heads, "\text{deriv}"] \arrow[d, two heads, "\text{eval}"] & \Pi \arrow[d, "\text{eval}"] \\ \Lambda_v \arrow[r, "\text{deriv}"] & \Pi_V \end{tikzcd}$$

Conclusion

$$\begin{aligned} ||\texttt{eval}(M)|| &= ||\texttt{deriv}(\texttt{eval}(M))|| \\ &= ||\texttt{eval}(\texttt{deriv}(M))|| \\ &= ||\texttt{deriv}(M)|| \end{aligned}$$

And Then?
Introduce non-determinism

Convex set of distributions

A randomized simulation is a function

$$f : U \to \mathscr{C}(\mathscr{D}(V))$$

targeting convex sets of distributions.

- Our Theorem holds for randomized simulation,
- A randomized encoding is a functional randomized simulation,
- A probabilistic bisimulation a derandomized randomized simulation.

Maybe a direction to treat real rewriting issues

Probabilistic confluence, Powerful bisimulations...