Probabilistic Termination by Monadic Affine Sized Typing

Ugo Dal Lago   Charles Grellois

FOCUS Team – INRIA & University of Bologna

ESOP 2017

Motivations

- Probabilistic programming languages are more and more pervasive in computer science: modeling uncertainty, robotics, cryptography, machine learning, AI...
- Quantitative notion of termination: almost-sure termination (AST)
- AST has been studied for imperative programs in the last years...
- ...but what about the probabilistic functional languages?

We introduce a monadic, affine sized type system sound for AST.

Sized Types: the Deterministic Case

Simply-typed λ-calculus is strongly normalizing (SN).

No longer true with the letrec construction. . .

Sized types: a decidable extension of the simple type system ensuring SN for λ-terms with letrec.

See notably:

- Hughes-Pareto-Sabry 1996, *Proving the correctness of reactive systems using sized types*,
- Barthe-Frade-Giménez-Pinto-Uustalu 2004, *Type-based termination of recursive definitions*.

Sized Types: the Deterministic Case

Sizes: $\mathfrak{s}, \mathfrak{r} \quad ::= \quad \mathfrak{i} \mid \infty \mid \widehat{\mathfrak{s}}$

$+$ size comparison underlying subtyping. Notably $\widehat{\infty} \equiv \infty$.

Idea: k successors = at most k constructors.

- $\mathsf{Nat}^{\widehat{\mathfrak{i}}}$ is 0,
- $\mathsf{Nat}^{\widehat{\widehat{\mathfrak{i}}}}$ is 0 or S 0,
- ...
- Nat^{∞} is any natural number. Often denoted simply Nat.

The same for lists,...

Sized Types: the Deterministic Case

Sizes: $\qquad \mathfrak{s}, \mathfrak{r} \quad ::= \quad \mathfrak{i} \mid \infty \mid \widehat{\mathfrak{s}}$

+ size comparison underlying subtyping. Notably $\widehat{\infty} \equiv \infty$.

Fixpoint rule:

$$\frac{\Gamma, f \,:\, \mathsf{Nat}^{\mathfrak{i}} \to \sigma \vdash M \,:\, \mathsf{Nat}^{\widehat{\mathfrak{i}}} \to \sigma[\mathfrak{i}/\widehat{\mathfrak{i}}] \qquad \mathfrak{i} \text{ pos } \sigma}{\Gamma \vdash \mathsf{letrec}\ f \ = \ M \,:\, \mathsf{Nat}^{\mathfrak{s}} \to \sigma[\mathfrak{i}/\mathfrak{s}]}$$

"To define the action of f on size $n+1$,
we only call recursively f on size at most n"

Sized Types: the Deterministic Case

Sizes: $\mathfrak{s}, \mathfrak{r} \quad ::= \quad \mathfrak{i} \mid \infty \mid \widehat{\mathfrak{s}}$

+ size comparison underlying subtyping. Notably $\widehat{\infty} \equiv \infty$.

Fixpoint rule:

$$\frac{\Gamma, f \,:\, \mathsf{Nat}^{\mathfrak{i}} \to \sigma \vdash M \,:\, \mathsf{Nat}^{\widehat{\mathfrak{i}}} \to \sigma[\mathfrak{i}/\widehat{\mathfrak{i}}] \qquad \mathfrak{i} \text{ pos } \sigma}{\Gamma \vdash \mathsf{letrec}\ f \ = \ M \,:\, \mathsf{Nat}^{\mathfrak{s}} \to \sigma[\mathfrak{i}/\mathfrak{s}]}$$

Typable $\Longrightarrow$ SN. Proof using reducibility candidates.

Decidable type inference.

Sized Types: Example in the Deterministic Case

From Barthe et al. (op. cit.):

$$
\begin{array}{ll}
\text{plus} \equiv (\mathsf{letrec} & plus_{:\text{Nat}^{\iota} \to \text{Nat} \to \text{Nat}} = \\
 & \lambda x_{:\text{Nat}^{\hat{\iota}}}.\ \lambda y_{:\text{Nat}}.\ \mathsf{case}\ x\ \mathsf{of}\ \{\mathsf{o} \Rightarrow y \\
 & \qquad\qquad\qquad\qquad \mid \mathsf{s} \Rightarrow \lambda x'_{:\text{Nat}^{\iota}}.\ \mathsf{s}\ \underbrace{(plus\ x'\ y)}_{:\text{Nat}} \\
 & \qquad\qquad\qquad\qquad \} \\
) : & \text{Nat}^{s} \to \text{Nat} \to \text{Nat}
\end{array}
$$

The case rule ensures that the size of x' is lesser than the one of x.
Size decreases during recursive calls $\Rightarrow$ SN.

A Probabilistic λ-calculus

$$M, N, \ldots \quad ::= \quad V \mid V\ V \mid \mathsf{let}\ x = M\ \mathsf{in}\ N \mid M \oplus_p N \mid \mathsf{case}\ V\ \mathsf{of}\ \{\,\mathsf{S} \to W \mid 0 \to Z\,\}$$

$$V, W, Z, \ldots \quad ::= \quad x \mid 0 \mid \mathsf{S}\ V \mid \lambda x.M \mid \mathsf{letrec}\ f = V$$

- Formulation equivalent to λ-calculus with $\oplus_p$, but constrained for technical reasons (A-normal form)
- Restriction to base type Nat for simplicity, but can be extended to general inductive datatypes (as in sized types)

A Probabilistic λ-calculus: Operational Semantics

$$\frac{}{\text{let } x = V \text{ in } M \ \rightarrow_v \ \left\{ (M[x/V])^1 \right\}}$$

$$\frac{}{(\lambda x.M)\ V \ \rightarrow_v \ \left\{ (M[x/V])^1 \right\}}$$

$$\frac{}{(\text{letrec } f = V)\ \left(c\ \overrightarrow{W}\right) \ \rightarrow_v \ \left\{ \left(V[f/(\text{letrec } f = V)]\ \left(c\ \overrightarrow{W}\right)\right)^1 \right\}}$$

(Call-by-value calculus)

A Probabilistic λ-calculus: Operational Semantics

$$\frac{}{\mathsf{case}\ \mathsf{S}\ V\ \mathsf{of}\ \{\,\mathsf{S} \to W \mid 0 \to Z\,\} \quad \to_v \quad \left\{\, (W\ V)^1 \,\right\}}$$

$$\frac{}{\mathsf{case}\ 0\ \mathsf{of}\ \{\,\mathsf{S} \to W \mid 0 \to Z\,\} \quad \to_v \quad \left\{\, (Z)^1 \,\right\}}$$

A Probabilistic λ-calculus: Operational Semantics

$$\frac{}{M \oplus_p N \ \rightarrow_v \ \left\{ M^p, N^{1-p} \right\}}$$

$$\frac{M \ \rightarrow_v \ \left\{ L_i^{p_i} \ \middle| \ i \in I \right\}}{\text{let } x = M \text{ in } N \ \rightarrow_v \ \left\{ (\text{let } x = L_i \text{ in } N)^{p_i} \ \middle| \ i \in I \right\}}$$

A Probabilistic λ-calculus: Operational Semantics

$$\frac{\mathscr{D} \overset{VD}{=} \left\{ M_j^{p_j} \mid j \in J \right\} + \mathscr{D}_V \qquad \forall j \in J, \quad M_j \to_v \mathscr{E}_j}{\mathscr{D} \to_v \left(\sum_{j \in J} p_j \cdot \mathscr{E}_j \right) + \mathscr{D}_V}$$

For $\mathscr{D}$ a distribution of terms:

$$[\![\mathscr{D}]\!] = \sup_{n \in \mathbb{N}} \left(\left\{ \mathscr{E}_n \mid \mathscr{D} \Rrightarrow_v^n \mathscr{E}_n \right\} \right)$$

where $\Rrightarrow_v^n$ is $\to_v^n$ followed by projection on values.

We let $[\![M]\!] = [\![\{ M^1 \}]\!]$.

M is AST iff $\sum [\![M]\!] = 1$.

Random Walks as Probabilistic Terms

- Biased random walk:

$$M_{bias} \; = \; \Big(\text{letrec } f \; = \; \lambda x.\text{case } x \text{ of } \Big\{ \mathsf{S} \to \lambda y.f(y) \oplus_{\frac{2}{3}} (f(\mathsf{S}\,\mathsf{S}\,y))) \;\; \Big| \;\; 0 \to 0 \Big\}\Big) \;\; \underline{n}$$

- Unbiased random walk:

$$M_{unb} \; = \; \Big(\text{letrec } f \; = \; \lambda x.\text{case } x \text{ of } \Big\{ \mathsf{S} \to \lambda y.f(y) \oplus_{\frac{1}{2}} (f(\mathsf{S}\,\mathsf{S}\,y))) \;\; \Big| \;\; 0 \to 0 \Big\}\Big) \;\; \underline{n}$$

$$\sum [\![\, M_{bias} \,]\!] \; = \; \sum [\![\, M_{unb} \,]\!] \; = \; 1$$

Capture this in a sized type system?

Another Term

We also want to capture terms as:

$$M_{nat} \ = \ \left(\text{letrec } f \ = \ \lambda x.x \oplus_{\frac{1}{2}} \mathsf{S} \ (f \ x)\right) \ 0$$

of semantics

$$[\![\, M_{nat} \,]\!] \ = \ \left\{(0)^{\frac{1}{2}}, (\mathsf{S} \ 0)^{\frac{1}{4}}, (\mathsf{S} \ \mathsf{S} \ 0)^{\frac{1}{8}}, \ldots\right\}$$

summing to 1.

(This is the geometric distribution.)

Beyond SN Terms, Towards Distribution Types

First idea: extend the sized type system with:

$$\text{Choice} \quad \frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash M \oplus_p N : \sigma}$$

and "unify" types of M and N by subtyping.

Kind of product interpretation of $\oplus$: we can't capture more than SN...

Beyond SN Terms, Towards Distribution Types

First idea: extend the sized type system with:

$$\text{Choice} \quad \frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash M \oplus_p N : \sigma}$$

and "unify" types of M and N by subtyping.

We get at best

$$f \ :\ \mathsf{Nat}^{\widehat{\widehat{i}}} \to \mathsf{Nat}^{\infty} \ \vdash\ \lambda y. f(y) \oplus_{\frac{1}{2}} (f(\mathsf{S}\,\mathsf{S}\,y))) \ :\ \mathsf{Nat}^{\widehat{i}} \to \mathsf{Nat}^{\infty}$$

and can't use a variation of the letrec rule on that.

Beyond SN Terms, Towards Distribution Types

We will use distribution types, built as follows:

$$\text{Choice} \quad \frac{\Gamma \,|\, \Theta \vdash M : \mu \qquad \Gamma \,|\, \Psi \vdash N : \nu \qquad \{\!|\, \mu \,|\!\} = \{\!|\, \nu \,|\!\}}{\Gamma \,|\, \Theta \oplus_p \Psi \vdash M \oplus_p N : \mu \oplus_p \nu}$$

Now

$$f : \left\{ \left(\mathsf{Nat}^{\mathfrak{i}} \to \mathsf{Nat}^{\infty}\right)^{\frac{1}{2}},\ \left(\mathsf{Nat}^{\widehat{\widehat{\mathfrak{i}}}} \to \mathsf{Nat}^{\infty}\right)^{\frac{1}{2}} \right\}$$

$$\vdash$$

$$\lambda y. f(y) \oplus_{\frac{1}{2}} (f(\mathsf{S}\,\mathsf{S}\,y))) \quad : \quad \mathsf{Nat}^{\widehat{\mathfrak{i}}} \to \mathsf{Nat}^{\infty}$$

Designing the Fixpoint Rule

$$f \ : \ \left\{ \left(\mathsf{Nat}^{\mathfrak{i}} \rightarrow \mathsf{Nat}^{\infty}\right)^{\frac{1}{2}},\ \left(\mathsf{Nat}^{\widehat{\widehat{\mathfrak{i}}}} \rightarrow \mathsf{Nat}^{\infty}\right)^{\frac{1}{2}} \right\}$$

$$\vdash$$

$$\lambda y.f(y) \oplus_{\frac{1}{2}} (f(\mathsf{S}\,\mathsf{S}\,y))) \ : \ \color{red}{\mathsf{Nat}^{\widehat{\mathfrak{i}}}} \rightarrow \mathsf{Nat}^{\infty}$$

induces a random walk on $\mathbb{N}$:

- on $n+1$, move to n with probability $\frac{1}{2}$, on $n+2$ with probability $\frac{1}{2}$,
- on 0, loop.

The type system ensures that there is no recursive call from size 0.

Random walk AST (= reaches 0 with proba 1) $\Rightarrow$ termination.

Designing the Fixpoint Rule

$$\frac{\begin{array}{c}\{\!|\, \Gamma \,|\!\} \;=\; \mathsf{Nat} \\ \mathfrak{i} \notin \Gamma \text{ and } \mathfrak{i} \text{ positive in } \nu \\ \left\{ (\mathsf{Nat}^{\mathfrak{s}_j} \to \nu[\mathfrak{i}/\mathfrak{s}_j])^{p_j} \;\middle|\; j \in J \right\} \text{ induces an AST sized walk} \\ \Gamma \,|\, f \;:\; \left\{ (\mathsf{Nat}^{\mathfrak{s}_j} \to \nu[\mathfrak{i}/\mathfrak{s}_j])^{p_j} \;\middle|\; j \in J \right\} \vdash V \;:\; \mathsf{Nat}^{\widehat{\mathfrak{i}}} \to \nu[\mathfrak{i}/\widehat{\mathfrak{i}}]\end{array}}{\Gamma \,|\, \emptyset \vdash \mathsf{letrec}\ f \;=\; V \;:\; \mathsf{Nat}^{\mathfrak{r}} \to \nu[\mathfrak{i}/\mathfrak{r}]}\ \text{LetRec}$$

Sized walk: AST is checked by an external PTIME procedure.

Generalized Random Walks and the Necessity of Affinity

A crucial feature: our type system is affine.

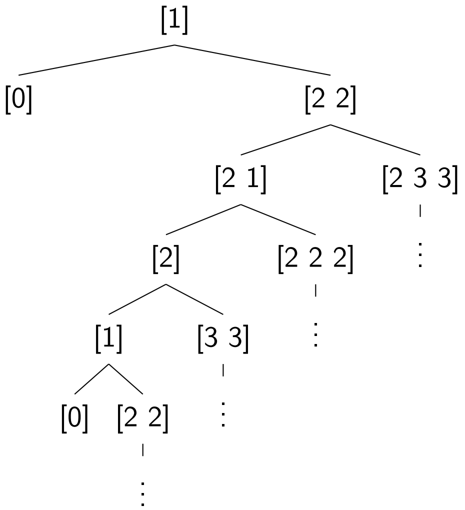
Higher-order symbols occur at most once. Consider:

$$M_{naff} \quad = \quad \mathsf{letrec}\ f \ = \ \lambda x.\mathsf{case}\ x\ \mathsf{of}\ \Big\{ \mathsf{S} \to \lambda y.f(y) \oplus_{\frac{2}{3}} (f(\mathsf{S}\,\mathsf{S}\,y)\,;\, f(\mathsf{S}\,\mathsf{S}\,y)) \ \Big| \ 0 \to 0 \Big\}$$

The induced sized walk is AST.

Generalized Random Walks and the Necessity of Affinity

Tree of recursive calls, starting from 1:



Leftmost edges have probability $\frac{2}{3}$; rightmost ones $\frac{1}{3}$.

This random process is not AST.

Problem: modelisation by sized walk only makes sense for affine programs.

Key Property I: Subject Reduction

Main idea: reduction of

$$\emptyset \mid \emptyset \vdash 0 \oplus 0 \; : \; \left\{ \left(\mathsf{Nat}^{\widehat{\mathfrak{s}}}\right)^{\frac{1}{2}}, \left(\mathsf{Nat}^{\widehat{\widehat{\mathfrak{r}}}}\right)^{\frac{1}{2}} \right\}$$

is to

$$\left\{ \left(0 \; : \; \mathsf{Nat}^{\widehat{\mathfrak{s}}}\right)^{\frac{1}{2}}, \left(0 \; : \; \mathsf{Nat}^{\widehat{\widehat{\mathfrak{r}}}}\right)^{\frac{1}{2}} \right\}$$

1. Same expectation type: $\frac{1}{2} \cdot \mathsf{Nat}^{\widehat{\mathfrak{s}}} + \frac{1}{2} \cdot \mathsf{Nat}^{\widehat{\widehat{\mathfrak{r}}}}$
2. Splitting of $[\![\, 0 \oplus 0 \,]\!]$ in a typed representation $\rightarrow$ notion of pseudo-representation

Key Property I: Subject Reduction

Theorem

Let $M \in \Lambda_{\oplus}$ be such that $\emptyset \mid \emptyset \vdash M : \mu$. Then there exists a closed typed distribution $\left\{ (W_j : \sigma_j)^{p'_j} \;\middle|\; j \in J \right\}$ such that

- $\mathbb{E}\left((W_j : \sigma_j)^{p'_j} \right) \preccurlyeq \mu$,
- *and that* $\left[(W_j)^{p'_j} \;\middle|\; j \in J \right]$ *is a pseudo-representation of* $[\![M]\!]$.

By the soundness theorem of next slide, this inequality is in fact an equality.

Key Property II: Typing Soundness

Theorem (Typing soundness)

If $\Gamma \mid \Theta \vdash M : \mu$, then M is AST.

Proof by reducibility, using set of candidates parametrized by probabilities.

Reducibility, the Probabilistic Case

Usual reducibility proof:

M closed of type $\sigma \quad \Rightarrow \quad M \in Red_\sigma \quad \Rightarrow \quad M$ is SN

In our setting:

Reducibility, the Probabilistic Case

Usual reducibility proof:

$$M \text{ closed of type } \sigma \quad \Rightarrow \quad M \in Red_\sigma \quad \Rightarrow \quad M \text{ is SN}$$

In our setting:

$$M \in TRed_\sigma^p \quad \Rightarrow \quad \sum [\![M]\!] \geq p$$

Reducibility, the Probabilistic Case

Usual reducibility proof:

$$M \text{ closed of type } \sigma \quad \Rightarrow \quad M \in Red_\sigma \quad \Rightarrow \quad M \text{ is SN}$$

In our setting:

$$M \text{ closed of type } \sigma \quad \Rightarrow \quad \forall p < 1,\ M \in TRed_\sigma^p \quad \Rightarrow \quad \forall p < 1,\ \sum [\![\, M \,]\!] \geq p$$

p increases with the number of fixpoint unfoldings we do, and we prove that M is in $TRed_\sigma^p$ iff its n-unfolding is.

Reducibility, the Probabilistic Case

Usual reducibility proof:

$$M \text{ closed of type } \sigma \quad \Rightarrow \quad M \in Red_\sigma \quad \Rightarrow \quad M \text{ is SN}$$

In our setting:

$$M \text{ closed of type } \sigma \quad \Rightarrow \quad M \in TRed_\sigma^1 \quad \Rightarrow \quad \sum [\![M]\!] = 1 \text{ i.e. } M \text{ AST}$$

by a continuity lemma.

Conclusion

Main features of the type system:

- Affine type system with distributions of types
- Sized walks induced by the letrec rule and solved by an external PTIME procedure
- Subject reduction + soundness for AST

Next steps:

- type inference (decidable again??)
- extensions with refinement types, non-affine terms

Thank you for your attention!

Conclusion

Main features of the type system:

- Affine type system with distributions of types
- Sized walks induced by the letrec rule and solved by an external PTIME procedure
- Subject reduction + soundness for AST

Next steps:

- type inference (decidable again??)
- extensions with refinement types, non-affine terms

Thank you for your attention!

Reducibility, the Probabilistic Case – Open Terms

Usual case: $\overrightarrow{x} : \overrightarrow{\sigma} \vdash M : \tau \quad \Rightarrow \quad \forall \overrightarrow{V} \in \overrightarrow{VRed_{\sigma}}, \;\; M[\overrightarrow{x} / \overrightarrow{V}] \in Red_{\tau}$

Reducibility, the Probabilistic Case – Open Terms

Usual case: $\overrightarrow{x} : \overrightarrow{\sigma} \vdash M : \tau \quad \Rightarrow \quad \forall \overrightarrow{V} \in \overrightarrow{VRed_{\sigma}}, \;\; M[\overrightarrow{x} / \overrightarrow{V}] \in Red_{\tau}$

In our setting: if $\Gamma \mid y : \{\tau_j^{p_j}\}_{j \in J} \vdash M : \mu$ then

- $\forall (q_i)_i \in [0,1]^n, \;\; \forall \overrightarrow{V} \in \prod_{i=1}^{n} \; \mathsf{VRed}_{\sigma_i}^{q_i},$
- $\forall \left(q'_j\right)_j \in [0,1]^J, \;\; \forall W \in \bigcap_{j \in J} \; \mathsf{VRed}_{\tau_j}^{q'_j},$
- we have $M[\overrightarrow{x}, y / \overrightarrow{V}, W] \in \mathsf{TRed}_{\mu}^{\alpha}$

where $\alpha \;\; = \;\; (\prod_{i=1}^{n} q_i) \left(\left(\sum_{j \in J} p_j q'_j \right) + 1 - \left(\sum_{j \in J} p_j \right) \right).$