Continuous Probabilistic Programming I: Semantics

Steven Holtzen
s.holtzen@northeastern.edu

September 25, 2023

- We give a syntax for a tiny continuous PPL called TinyCont:

```

// expression terms
e ::=
  | x                // identifiers
  | ℝ               // real values
  | true | false
  | unif             // uniform distribution on the interval
  | return e
  | x ← e; e
  | e + e | e < e
```

Typing rules Types τ have the following inductive description:

$$\tau ::= \mathbb{R} \mid \mathbb{B}$$

Terms have the following typing judgments:

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau} \qquad \Gamma \vdash \mathbb{R} : \mathbb{R} \qquad \texttt{unif} : \texttt{Dist}(\mathbb{R})$$

$$\frac{\Gamma \vdash \texttt{e} : \tau}{\Gamma \vdash \texttt{return e} : \texttt{Dist}(\tau)} \qquad \frac{\Gamma \vdash \texttt{e}_1 : \texttt{Dist}(\tau) \qquad \Gamma \cup [x \mapsto \tau] \vdash \texttt{e}_2 : \tau'}{\Gamma \vdash \texttt{let x = e}_1 \texttt{ in e}_2 : \tau'}$$

An example program:

```
x ← unif;
return x + 1
```

Listing 1: Simple program.

1 *Denotational semantics*

- Denotation of types. Base types have two possible interpretations: measurable spaces (i.e. pairs (Ω, Σ) where Ω is a set and Σ is a σ-algebra on Ω:) and pure values. We denote the measurable space interpretation $[\![\cdot]\!]_M$:

$$[\![\mathbb{B}]\!]_M = (\{\texttt{true}, \texttt{false}\}, 2^{\{\texttt{true},\texttt{false}\}})$$
$$[\![\mathbb{R}]\!]_M = (\{\mathbb{R}, \mathcal{B}(\mathbb{R})\})$$

where $\mathcal{B}$ are the standard Borel-sets on $\mathbb{R}$. We say $v \in [\![\tau]\!]$ if v is an element of the σ-algebra of $[\![\tau]\!]$.

- Pure denotation of types is denoted $[\![\tau]\!]_p$ and is traditional:

$$[\![\mathbb{B}]\!]_p = \{\texttt{true}, \texttt{false}\}$$
$$[\![\mathbb{R}]\!]_p = \mathbb{R}.$$

- Denotational of distribution-typed terms has the form:

$$[\![\Gamma \vdash \texttt{e} : \texttt{Dist}(\tau)]\!] : [\![\tau]\!]_M \to [0,1] \tag{1}$$

- Denotation of pure terms has the a standard base type
- Inductive definition:

Some notation:
- Dirac delta on value v: δ_v
- Lebesgue measure on a set A: $\mathbb{L}(A)$

$$[\![v]\!] = v$$
$$[\![\texttt{return } v]\!] = \delta_v$$
$$[\![\texttt{unif}]\!]\,(A) = \mathbb{L}(A \cap [0,1])$$
$$[\![x \leftarrow \texttt{e}_1;\texttt{e}_2]\!]\,(A) = \int [\![\texttt{e}_1]\!]\,(v)\,[\![\texttt{e}_2[v/x]]\!]\,(A)\;\mathrm{d}v$$
$$[\![\texttt{e}_1 + \texttt{e}_2]\!] = [\![\texttt{e}_1]\!] + [\![\texttt{e}_2]\!]$$
$$[\![\texttt{e}_1 < \texttt{e}_2]\!] = \begin{cases} \texttt{true} & \text{if } [\![\texttt{e}_1]\!] < [\![\texttt{e}_2]\!] \\ \texttt{false} & \text{otherwise.} \end{cases}$$

2 *Operational semantics*

- Following Culpepper and Cobb [2017]. An *entropy space* is a measurable space $(\mathbb{S}, \Sigma_{\mathbb{S}})$ with meaure $\mu_{\mathbb{S}} : \Sigma_{\mathbb{S}} \to [0,1]$ and functions:

$$\pi_U : \mathbb{S} \to [0,1] \tag{2}$$
$$\pi_R, \pi_L : \mathbb{S} \to \mathbb{S} \tag{3}$$

such that the following equations hold:

1. $\mu_{\mathbb{S}}(\mathbb{S}) = 1$, and so for any $r \in \mathbb{R}^+$ it is the case that:

$$\int k\mu_{\mathbb{S}}(\mathrm{d}\sigma) = k. \tag{4}$$

2. For any measurable function $f : [0,1] \to \mathbb{R}^+$,

$$\int f(\pi_U(\sigma))\;\mu_{\mathbb{S}}(\mathrm{d}\sigma) = \int_{[0,1]} f(x)\mathbb{L}(\mathrm{d}x) \tag{5}$$

3. For any measurable $g : \mathbb{S} \times \mathbb{S} \to \mathbb{R}^+$:

$$\int g(\pi_L(\sigma), \pi_R(\sigma))\;\mu_{\mathbb{S}}(\mathrm{d}\sigma) = \int\int g(\sigma_1, \sigma_2)\mu_{\mathbb{S}}(\mathrm{d}\sigma_1)\mu_{\mathbb{S}}(\mathrm{d}\sigma_2) \tag{6}$$

- We give a Hilbert-cube style semantics in the style of Wand et al. [2018]. The relation has the form:

$$\sigma \vdash \mathtt{e} \Downarrow v \tag{7}$$

- Inductive description:

$$\sigma \vdash v \Downarrow v \qquad \frac{\pi_U(\sigma) = r}{\sigma \vdash \mathtt{unif} \Downarrow r} \qquad \mathtt{return}\ v \Downarrow v$$

$$\frac{\pi_L(\sigma) \vdash \mathtt{e}_1 \Downarrow v_1 \qquad \pi_R(\sigma) \vdash \mathtt{e}_2[v_1/x] \Downarrow v_2}{\sigma \vdash x \leftarrow \mathtt{e}_2; \mathtt{e}_2 \Downarrow v_2}$$

Notation:

- $\pi_L(\sigma)$: left project of σ. Similarly π_R is right.
- S: the entropy space $[0,1]^{\mathbb{N}}$

- For a term of $\mathtt{e}$ of type $\mathtt{Dist}(\tau)$, This defines an *evaluation function* $\mathrm{ev} : \Sigma \to \mathtt{e} \to [\![\tau]\!]_p$ that maps each (well-typed) term to the value that it steps to.

Theorem 1 (Adequacy). *If* $\Gamma \vdash \mathtt{e} : \mathtt{Dist}(\tau)$*, then for any* $A \in [\![\tau]\!]_M$ *it is the case that:* $\int [\mathrm{ev}(\sigma, \mathtt{e}) \in A]\ \mu_{\mathrm{S}}(\mathrm{d}\sigma) = [\![\mathtt{e}]\!]\,(A)$.

Proof. Structural induction on monadic terms. Base cases:

- `return v`. Case analysis:
  - Assume $v \in A$. Then:

$$\begin{aligned}\int [\mathrm{ev}(\sigma, \mathtt{return\ v}) \in A]\ \mu_{\mathrm{S}}(\mathrm{d}\sigma) &= \int 1\ \mathrm{d}\sigma \\ &= 1 && \text{By (4)} \\ &= [\![\mathtt{return\ v}]\!]\,(A).\end{aligned}$$

  Similar for the case $v \notin A$.

- `unif`. Then:

$$\begin{aligned}\int [\mathrm{ev}(\sigma, \mathtt{unif}) \in A]\ \mu_{\mathrm{S}}(\mathrm{d}\sigma) &= \int [\pi_U(\sigma) \in A]\ \mu_{\mathrm{S}}(\mathrm{d}\sigma) \\ &= \int_{[0,1]} [x \in A]\ \mathbb{L}(\mathrm{d}x) && \text{By (5)} \\ &= [\![\mathtt{unif}]\!]\,(A).\end{aligned}$$

□

References

Ryan Culpepper and Andrew Cobb. Contextual equivalence for probabilistic programs with continuous random variables and scoring. In *European Symposium on Programming*, pages 368–392. Springer, 2017.

Mitchell Wand, Ryan Culpepper, Theophilos Giannakopoulos, and Andrew Cobb. Contextual equivalence for a probabilistic language with continuous random variables and recursion. *Proceedings of the ACM on Programming Languages*, 2(ICFP):1–30, 2018.