Denotational Semantics for Probabilistic Programs

Hongseok Yang

KAIST

1 Introduction

In this lecture note, we describe the denotational semantics of a first-order programming language with discrete random choices. We focus mostly on spelling out all the details, which we covered in our lectures but did not appear in lecture slides.

2 Syntax and Typing Rules of a First-order Programming Language with Discrete Random Choices

$$
\begin{array}{llr}
\text{Types} & \tau ::= \texttt{bool} & \text{Boolean Type} \\
& \quad \mid \texttt{rational} & \text{Rational Type} \\
& \quad \mid \texttt{dist}[\texttt{bool}] & \text{Boolean-Distribution Type} \\
& \quad \mid \texttt{dist}[\texttt{rational}] & \text{Rational-Distribution Type}
\end{array}
$$

$$
\begin{array}{llr}
\text{Expressions} & e ::= c & \text{Constants} \\
& \quad \mid x & \text{Variables} \\
& \quad \mid (p\ e_1\ \ldots\ e_n) & \text{Function Applications} \\
& \quad \mid (\texttt{let}\ [x\ e_1] e_2) & \text{Let Expressions} \\
& \quad \mid (\texttt{if}\ e_0\ e_1\ e_2) & \text{Conditional Expressions}
\end{array}
$$

$$
\begin{array}{llr}
\text{Constants} & c ::= 1.2 \mid \ldots & \text{Rational Numbers} \\
& \quad \mid \texttt{true} \mid \texttt{false} & \text{Booleans}
\end{array}
$$

$$
\begin{array}{llr}
\text{Primitive Operations} & p ::= \texttt{and} \mid \texttt{or} \mid \ldots & \text{Boolean Op.} \\
& \quad \mid + \mid * \mid \ldots & \text{Arithematic Op.} \\
& \quad \mid \texttt{flip} \mid \texttt{poisson} \mid \ldots & \text{Distribution Constr.} \\
& \quad \mid \texttt{sample}_{\texttt{bool}} \mid \texttt{sample}_{\texttt{rational}} & \text{Sampling Op.}
\end{array}
$$

Typing Contexts	$\Gamma ::= x_1 : \tau_1, \ldots, x_n : \tau_n$
Typed Expressions	$\Gamma \vdash e : \tau$

Rules for typed expressions (or typing judgments):

$$\frac{\text{TypeConst}(c) = \tau}{\Gamma \vdash c : \tau} \qquad \frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau}$$

$$\frac{\text{TypePrimOp}(p) = (\tau_1, \ldots, \tau_n) \to \tau \qquad \Gamma \vdash e_i : \tau_i \text{ for all } i}{\Gamma \vdash (p\ e_1\ \ldots\ e_n) : \tau}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma, x : \tau_1 \vdash e_2 : \tau}{\Gamma \vdash (\texttt{let}\ [x\ e_1]\ e_2) : \tau}$$

$$\frac{\Gamma \vdash e_0 : \texttt{bool} \qquad \Gamma \vdash e_1 : \tau \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash (\texttt{if}\ e_0\ e_1\ e_2) : \tau}$$

where

$\text{TypeConst}(\texttt{true}) = \texttt{bool}$
$\text{TypeConst}(\texttt{false}) = \texttt{bool}$
$\text{TypeConst}(1.2) = \texttt{rational}$
$\text{TypeConst}(-2) = \texttt{rational}$
$\ldots$

$\text{TypePrimOp}(\texttt{and}) = (\texttt{bool}, \texttt{bool}) \to \texttt{bool}$
$\text{TypePrimOp}(\texttt{or}) = (\texttt{bool}, \texttt{bool}) \to \texttt{bool}$
$\text{TypePrimOp}(+) = (\texttt{rational}, \texttt{rational}) \to \texttt{rational}$
$\text{TypePrimOp}(*) = (\texttt{rational}, \texttt{rational}) \to \texttt{rational}$
$\text{TypePrimOp}(\texttt{flip}) = (\texttt{rational}) \to \texttt{dist}[\texttt{bool}]$
$\text{TypePrimOp}(\texttt{poisson}) = (\texttt{rational}) \to \texttt{dist}[\texttt{rational}]$
$\text{TypePrimOp}(\texttt{sample}_{\texttt{bool}}) = (\texttt{dist}[\texttt{bool}]) \to \texttt{bool}$
$\text{TypePrimOp}(\texttt{sample}_{\texttt{rational}}) = (\texttt{dist}[\texttt{rational}]) \to \texttt{rational}$
$\ldots$

3 Semantics of Expressions

Three notations: for all booleans b, and all $a \in A$ and $c \in C$,

$$[b] \in \{0, 1\}$$
$$[b] = \begin{cases} 1 \text{ if } b = \mathrm{tt} \\ 0 \text{ if } b = \mathrm{ff} \end{cases}$$

$$\delta_a \in \mathrm{DiscProb}(A)$$
$$\delta_a(a') = \begin{cases} 1 \text{ if } a = a' \\ 0 \text{ otherwise} \end{cases}$$

$$\mathrm{bind} : \mathrm{DiscProb}(A) \times (A \to \mathrm{DiscProb}(C)) \to \mathrm{DiscProb}(C)$$
$$\mathrm{bind}(p, f)(c) = \sum_{a \in A} (p(a) \times f(a)(c)) = \mathbb{E}_{p(a)}[f(a)(c)]$$

Instead of $\mathrm{bind}(p, f)$, we often write

$$p \text{ bind } f.$$

Also, we use the λ expression to define mathematical functions in our definition of semantics.

The semantics of expressions is defined by induciton on the size of typing derivations.

$$\begin{aligned}
[\![\Gamma \vdash c : \tau]\!]\eta(v) &= \delta_c(v) \\
[\![\Gamma \vdash x : \tau]\!]\eta(v) &= \delta_{\eta(x)}(v) \\
[\![\Gamma \vdash (p\ e_1\ \ldots\ e_n) : \tau]\!]\eta &= ([\![\Gamma \vdash e_1 : \tau_1]\!]\eta) \text{ bind } \lambda v_1. \\
&\quad ([\![\Gamma \vdash e_2 : \tau_2]\!]\eta) \text{ bind } \lambda v_2. \\
&\quad \ldots \\
&\quad ([\![\Gamma \vdash e_n : \tau_n]\!]\eta) \text{ bind } \lambda v_n. \\
&\quad \mathrm{mean}(p)(v_1\ \ldots\ v_n) \\
[\![\Gamma \vdash (\texttt{let}\ [x\ e_1]\ e_2) : \tau]\!]\eta &= ([\![\Gamma \vdash e_1 : \tau_1]\!]\eta) \text{ bind } \lambda v_1. \\
&\quad [\![\Gamma \vdash e_2 : \tau_2]\!](\eta[x \mapsto v_1]) \\
[\![\Gamma \vdash (\texttt{if}\ e_0\ e_1\ e_2) : \tau]\!]\eta &= [\![\Gamma \vdash e_0 : \texttt{bool}]\!]\eta \text{ bind } \lambda b. \\
&\quad [b] \cdot [\![\Gamma \vdash e_1 : \tau]\!]\eta + [\neg b] \cdot [\![\Gamma \vdash e_2 : \tau]\!]\eta
\end{aligned}$$

Here we assume the definition of mean, which we will explain now. For a primitive operation p such that $\mathrm{TypePrimOp}(p) = (\tau_1, \ldots, \tau_n) \to \tau$, the $\mathrm{mean}(p)$ is a function of the following type:

$$\mathrm{mean}(p) : [\![\tau_1]\!] \times \ldots \times [\![\tau_n]\!] \to \mathrm{DiscProb}([\![\tau]\!])$$

It is defined as follows:

$$
\begin{aligned}
\mathrm{mean}(\texttt{and})(b_1, b_2)(b) &= \delta_{b_1 \wedge b_2}(b) \\
\mathrm{mean}(\texttt{or})(b_1, b_2)(b) &= \delta_{b_1 \vee b_2}(b) \\
\mathrm{mean}(+)(r_1, r_2)(r) &= \delta_{r_1 + r_2}(r) \\
\mathrm{mean}(*)(r_1, r_2)(r) &= \delta_{r_1 * r_2}(r) \\
\mathrm{mean}(\texttt{flip})(r)(p) &= \begin{cases} \delta_{[\mathrm{tt} \mapsto r; \mathrm{ff} \mapsto (1-r)]}(p) & \text{if } r \in [0,1] \\ \delta_{[\mathrm{tt} \mapsto 1; \mathrm{ff} \mapsto 0]}(p) & \text{if } r \notin [0,1] \end{cases} \\
\mathrm{mean}(\texttt{poisson})(r)(p) &= \begin{cases} \delta_{\mathrm{Poisson}(r)}(p) & \text{if } r > 0 \\ \delta_{\mathrm{Poisson}(1)}(p) & \text{otherwise} \end{cases} \\
\mathrm{mean}(\texttt{sample}_{\texttt{bool}})(p)(v) &= p(v) \\
\mathrm{mean}(\texttt{sample}_{\texttt{rational}})(p)(v) &= p(v)
\end{aligned}
$$