s-有限測度モナドを用いた確率的プログラミング言語のIsabelle/HOLによる形式化

平田 路和，南出 靖彦，佐藤 哲也 (東京工業大学)

確率的プログラミング言語

確率分布を記述するプログラミング言語

例 サイコロを２つ投げ，少なくとも1つの目が4であったとき の両方の目の合計値の分布 (出典: Sampson[1])

```
definition two_dice :: "nat qbs_measure" where
 "two_dice ≡ do {
              let die1 = die;
              let die2 = die;
              let twodice = die1 ⨂_Qmes die2;
              (x,y) ← condition twodice
                      (λ(x,y). x = 4 ∨ y = 4);
              return_qbs ℕ_Q (x + y)
           }"
```

```
corollary
 "𝒫(x in two_dice. x = 5)  = 2 / 11"
 "𝒫(x in two_dice. x = 6)  = 2 / 11"
 "𝒫(x in two_dice. x = 7)  = 2 / 11"
 "𝒫(x in two_dice. x = 8)  = 1 / 11"
 "𝒫(x in two_dice. x = 9)  = 2 / 11"
 "𝒫(x in two_dice. x = 10) = 2 / 11"
```

研究目的

高階関数・サンプリング・条件付き分布を扱うことのできる確率的プログラムの意味論基盤を証明支援系の上で構築する

関連研究

平田ら[2]

　準ボレル空間と確率モナドによる形式化(Isabelle/HOL)

　高階関数・サンプリングをサポート

Affeldtら[3]

　可測空間と s-有限カーネルによる形式化(Coq)

　サンプリング・条件付き分布をサポート

結果

先行研究では扱えなかった高階関数・サンプリング・条件付き分布のすべてを扱う確率的プログラムを扱うことができるようになった

先行研究よりプログラムの可読性・証明しやすさが向上した

実装手順

1. 測度論の概念の形式化(s-有限カーネル，ボレル同型定理)
2. 準ボレル空間ライブラリの拡張(証明自動化，s-有限測度モナド)
3. 確率的プログラムを記述，検証

準ボレル空間

準ボレル空間 (X, M_X)

X : 集合，　$M_X \subseteq \mathbb{R} \to X$: 「確率変数」の集合

定義 $f : X \to Y$が射 $\iff \forall \alpha \in M_X . f \circ \alpha \in M_Y$

高階プログラムの意味論には準ボレル空間が適している

評価関数 $ev : (\mathbb{R} \Rightarrow \mathbb{R}) \times \mathbb{R} \to \mathbb{R},\ ev(f, x) = f(x)$ について

ev は可測関数にはなり得ないが，準ボレル空間の射になる

事実 準ボレル空間X上のs-有限測度の空間$M(X)$ が定義される

$(M, \eta, \gg\!=)$が可換な強モナドとなるような$\eta, \gg\!=$ も定義される

標準ボレル空間($\mathbb{R}, \mathbb{N}, \Pi_{n\in\mathbb{N}}\mathbb{R}$等)上のs-有限測度は準ボレル空間上の測度として表すことができ，積分もできる.

※先行研究[2]は確率モナドを定義

- 確率モナド　$\Longrightarrow$ 確率測度，Giry モナド (確率カーネル)
- s-有限測度モナド $\Longrightarrow$ s-有限測度，s-有限カーネル

Isabelle/HOL での定義

準ボレル空間	`"X :: 'a quasi_borel"`
準ボレル空間の構造	`"qbs_space X :: 'a set"`
	`"qbs_Mx X :: (real ⇒ 'a) set"`
(2項)直積空間	`"X ⨂_Q Y :: ('a × 'b) quasi_borel"`
関数空間	`"X ⇒_Q Y :: ('a ⇒ 'b) quasi_borel"`
リスト空間	`"list_qbs X :: 'a list quasi_borel"` ※ $List[X] \cong \coprod_{n\in\mathbb{N}}\prod_{0\le k<n}X$ を用いて定義される
測度空間	`"monadM_qbs X :: 'a qbs_measure quasi_borel"`

`"x ∈ qbs_space X"` の証明自動化をMLで実装した

確率的プログラムの実装

Isabelle/HOL の項を確率的プログラムとして解釈

型 $T \Longrightarrow$ 準ボレル空間　　型判定 $\vdash t : T \Longrightarrow$ `"t ∈ qbs_space T"`

例: `"(+) ∈ ℝ_Q ⇒_Q ℝ_Q ⇒_Q ℝ_Q"` `"return_qbs X ∈ X ⇒_Q monadM_qbs X"`

```
lemma "(λs. do { x ← s; return_qbs ℝ_Q (x + 1) })
         ∈ monadM_qbs ℝ_Q ⇒_Q monadM_qbs ℝ_Q"
  by simp
```

プログラム例(Staton[4]):

- 各時刻における，自転車の通過台数(/時)がわかっている
- 自転車が通過する間隔, 1分を観測した
- 現在時刻は？

```
definition whattime :: "(real ⇒ real) ⇒ real qbs_measure" where
"whattime ≡ (λf. do {
                  let T = Uniform 0 24 in
                  query T (λt. let r = f t in
                                exponential_density r (1 / 60))
                })"
```

※時刻tにおける自転車が通過する間隔は指数分布$Ex(f(t))$に従う

Posterior ∝ Likelihood × Prior .

```
lemma whattime_qbs[qbs]: "whattime ∈ (ℝ_Q ⇒_Q ℝ_Q) ⇒_Q monadM_qbs ℝ_Q"
  by(simp add: whattime_def)
```

```
lemma
  assumes [qbs]: "f ∈ ℝ_Q ⇒_Q ℝ_Q" and [measurable]:"U ∈ sets borel"
      and "⋀r. f r ≥ 0"
  defines "N ≡ (∫t∈{0<..<24}. (f t * exp (- 1/ 60 * f t)) ∂lborel)"
  defines "N' ≡ (∫⁺t∈{0<..<24}. (f t * exp (- 1/ 60 * f t)) ∂lborel)"
  assumes "N' ≠ 0" and "N' ≠ ∞"
  shows "𝒫(t in whattime f. t ∈ U) =
       (∫t∈{0<..<24}∩U. (f t * exp (- 1/ 60 * f t)) ∂lborel) / N"
```

※指数分布の密度関数: exponential_density$(x \mid \lambda) = \lambda e^{-\lambda x}$ $(x > 0)$

プログラム例(佐藤ら[5]):

サンプルのリストから正規分布の平均を推定(分散既知)

```
primrec GaussLearn' :: "[real, real qbs_measure, real list]
                                   ⇒ real qbs_measure" where
  "GaussLearn' _ p [] = p"
| "GaussLearn' σ p (y#ls) = query (GaussLearn' σ p ls)
                                  (normal_density y σ)"
```

```
lemma "GaussLearn' ∈
       ℝ_Q ⇒_Q monadM_qbs ℝ_Q ⇒_Q list_qbs ℝ_Q ⇒_Q monadM_qbs ℝ_Q"
  by(simp add: GaussLearn'_def)
```

```
abbreviation "GaussLearn ≡ GaussLearn' σ"
```

```
lemma GaussLearn_Total:
  assumes [arith]: "ξ > 0" "n = length L"
  shows "GaussLearn (Gauss δ ξ) L =
  Gauss ((Total L*ξ^2+δ*σ^2)/(n*ξ^2+σ^2)) (sqrt ((ξ^2*σ^2)/(n*ξ^2+σ^2)))"
```

```
lemma GaussLearn_KL_divergence:
  fixes a b c d e K :: real
  assumes [arith]:"e > 0" "b > 0" "d > 0"
  shows "∃N. ∀L. length L > N ⟶ ¦Total L / length L¦ < K
 ⟶ KL_divergence (exp 1)
     (GaussLearn (Gauss a b) L) (GaussLearn (Gauss c d) L) < e"
```

展望

- 原始再帰的でない停止する関数がプログラムであることの導出
- 帰納的なデータ型(リスト等)に準ボレルの構造を与え，その上の定数が射であることの証明を自動化

参考文献

[1] Adrian Sampson. *Probabilistic programming*. http://adriansampson.net/doc/ppl.html Accessed: January 25. 2023.
[2] Michikazu Hirata, Yasuhiko Minamide, and Tetsuya Sato. *Program logic for higher-order probabilistic programs in Isabelle/HOL*. Functional and Logic Programming, pages 57–74, Cham, 2022. Springer International Publishing.
[3] Reynald Affeldt, Cyril Cohen, and Ayumu Saito. *Semantics of probabilistic programs using s-finite kernels in Coq*. In Proceedings of the 12th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2023, page 3–16, New York, NY, USA, 2023. Association for Computing Machinery.
[4] Sam Staton. *Probabilistic Programs as Measures*, page 43–74. Cambridge University Press, 2020
[5] Tetsuya Sato, Alejandro Aguirre, Gilles Barthe, Marco Gaboardi, Deepak Garg, and Justin Hsu. *Formal verification of higher-order probabilistic programs: reasoning about approximation, convergence, bayesian inference, and optimization*. Proceedings of the ACM on Programming Languages, 3(POPL):1–30, 2019.