Isabelle/HOL による高階確率的プログラム検証

平田 路和　南出 靖彦　佐藤 哲也

確率的プログラミング言語とは，機械学習や統計モデリングにおいて確率モデルの記述，推論を行うために使われる言語である．本研究では高階確率的プログラム検証のためのフレームワーク PPV の中核部分，及び PPV の意味論的基礎となる準ボレル空間 (quasi-Borel spaces) を Isabelle/HOL で実装し，例としてモンテカルロ法の検証を行なった．実装する上で，先行研究では省略されていた可積分性に関する公理と算術演算に関する公理を追加した．モンテカルロ法の検証では，可積分性に関する議論を含んだ検証を行うことができた．

1　はじめに

確率的プログラミング言語とは，確率分布を記述し，分布からのサンプリングや分布の期待値の計算を自動で実行する機能をもったプログラミング言語で，統計分野や機械学習で活用されている [5] [15] [10] [14].

確率的プログラミング言語の意味論は，確率分布のモナド（Giry モナド）を用いて可測関数の形で自然に与えることができるが，関数型に対応する可測空間が一般には存在しないことから，高階な言語に拡張することが難しかった．この問題に，Heunen らは，扱う空間を制限した準ボレル空間 (quasi-Borel spaces) を導入することで，高階確率的プログラミング言語に自然に意味を与えることができることを示した [6]．さらに，佐藤らは，この準ボレル空間による意味論をベースに，連続分布からのサンプリングおよび条件付分布をサポートする高階確率的プログラミング言語 HPProg を設計し，そのプログラム論理 PL，UPL，RPL を与えた [12]．この検証フレームワークを PPV(Probabilistic Programming Verification framework) という．

これまで，確率的プログラムに関する証明支援系 Isabelle/HOL による検証は，一階の関数型確率的プログラムから確率密度関数へのコンパイルを検証した [3] や，暗号理論における安全性を示すための [9] などが挙げられる．これらは離散的な空間に限った場合や，測度論に基づいた検証となっているため，高階関数と連続分布をサポートする言語についての検証はできない．

本研究の貢献は

- Isabelle/HOL による準ボレル空間の形式化．準ボレル空間を構成し，準ボレル空間の圏と可測空間の圏の間の随伴，準ボレル空間がなす圏の確率モナドが可換な強モナドであることを Isabelle/HOL で証明した．
- 条件付き推論を含まない，言語 HPProg，論理体系 PL・UPL の shallow embedding による形式化．本研究ではさらに，先行研究では扱われていなかった可積分性の議論もできるように，PL・UPL 内の述語の定義を拡張してある．
- 証明支援系による，可積分性の議論も含んだモンテカルロ法 ([12] の例) の厳密な検証.

である．

本研究は準ボレル空間に基づいた形式化であるた

Formal Verification of Higher-Order Probabilistic Program in Isabelle/HOL

This is an unrefereed paper. Copyrights belong to the Author(s).

Hirata Michikazu, Minamide Yasuhiko, Sato Tetsuya, 東京工業大学情報理工学院, School of Computing, Tokyo Institute of Technology.

め，先行研究ではできなかった高階関数と連続分布を含んだ検証ができる．準ボレル空間の形式化，及び準ボレル空間に基づいた検証は，Isabelle/HOL 以外の証明支援系を含めても私たちの知る限りまだない．

2 Isabelle/HOL によるモンテカルロ法の検証

2.1 プログラム検証フレームワーク: PPV

PPV は，Aguirre らによる関数型プログラム論理 [1] をもとにした，高階関数型確率的プログラム検証のためのフレームワークで，言語 HPProg，論理体系 PL，UPL を含んでいる †1．HPProg は，サンプリングや observe といった機能をもった関数型言語である．型 T に対して，型 $P[T]$ は T 上の確率測度の型である．PL は $\Gamma \mid \Psi \vdash_{\mathrm{PL}} \phi$ という判定式をもつ PPV の基礎を成す論理体系である．Γ は型環境，Ψ は仮定の集合，ϕ は結論を表し，判定式は「環境 Γ で，Ψ が成り立つならば ϕ が成り立つ」という意味である．UPL(unary program logic) は，プログラムの unary な性質を検証するため論理体系で，$\Gamma \mid \Psi \vdash_{\mathrm{UPL}} e : T \mid \phi$ という形の判定式をもつ．PL との違いは，プログラム e と e の型 T が判定式に含まれている．また，結論 ϕ にはプログラム変数と呼ばれるプログラムを表す変数 $\mathbf{r}$ が含まれている．判定式は,「環境 Γ で e は型 T をもつ」かつ「環境 Γ で Ψ が成り立つならば $\phi[e/\mathbf{r}]$ が成り立つ」という意味である．

HPProg の表示的意味論では準ボレル空間を用いる．準ボレル空間とは，高階関数と連続分布を同時に扱うような言語の表示的意味論を与えるために [6] で導入された空間であり，測度論をもとに構成されている．ここで，HPProg プログラムを素朴に可測関数として解釈しようとすると，$\lambda(f,x) : (\mathtt{real} \Rightarrow \mathtt{real}) \times \mathtt{real}.f(x)$ のようなプログラムを解釈できないことが知られている †2．準ボレル空間を用いることでこのような問題を回避することができる．

2.2 モンテカルロ法の検証

本研究で実装した PPV を使い，[12] の例にある，モンテカルロ法により平均値の近似計算を行うプログラム (図 1) の検証を行なった (図 2(2))†3．

プログラム *montecarlo* は型 $\mathtt{nat} \Rightarrow P[\mathtt{real}]$ をもつ．*montecarlo* は，引数 n をもらって，$\mathtt{d}$ から n 個のサンプルを逐次的に生成し，その $\mathtt{h}$ による観測値の平均の分布を返す関数である．

本研究では，先行研究の例では省略されていたプログラムの可積分性も PL の上で示した (図 2(1))．$\mu : P[T]$ と $f : T \Rightarrow P[\mathtt{real}]$ に対して，述語 *integrable* μ f は期待値 $\mathbb{E}_{x\sim\mu}[f\ x]$ が有限値で存在する (f が測度 μ の下で可積分である) ことを意味している．可積分性は，判定式 (図 2(2)) を示す際に用いる期待値の線形性を使うために必要となる．

図 2(2) の判定式は同分布上の大数の弱法則を表している．

定理 2.1 (大数の弱法則)**.** $(X_n)_{n=1,2,\cdots}$ が共通の平均 μ をもつ互いに独立な実数値確率変数で，$\sup_n \mathbb{V}[X_n] < \infty$ とする．このとき，任意の $\varepsilon > 0$ に対して，

$$\lim_{n\to\infty} \Pr\left[\left|\frac{X_1 + \ldots X_n}{n} - \mu\right| \geq \varepsilon\right] = 0$$

が成り立つ．

この判定式は，$\mathtt{h}$ の定義域が実数であっても成立するため，測度論による解釈では意味が取れなかった連続分布と高階関数を扱うようなプログラムの検証となっている．

2.3 形式化のコスト

本研究では，標準ボレル空間，及び準ボレル空間の形式化におよそ 3ヶ月，PPV とモンテカルロ法の形式化におよそ 3ヶ月の合計 6ヶ月の期間を要した．Isabelle/HOL で記述したおおよそのコード量を以下の表に記す．

表からわかるように，標準ボレル空間と準ボレル空

---

†1 PPV には論理体系 RPL も含まれているが，本研究で形式化をしていないので，ここでは説明を省略する．

†2 Aumann の研究 [2] による．ここで述べられたプログラムの自然な解釈である関数適用 $\mathbf{Meas}(\mathbb{R},\mathbb{R}) \times \mathbb{R} \to \mathbb{R}$ $((f,x) \mapsto fx)$ は可測でないことが証明されている．また，可測空間と可測写像の圏 **Meas** はカルテシアン閉でなく，特に冪対象 $\mathbb{R}^{\mathbb{R}}$ が存在しない．

†3 読み易くするため，プログラムなどの表現方法を実際の実装と少し変えている．

$$
\begin{aligned}
&\mathit{montecarlo}\ \mathtt{n} \equiv \mathtt{if}\ (\mathtt{n} = 0)\ \mathtt{then\ return\ 0}\\
&\qquad\qquad\qquad\qquad\ \mathtt{else\ do}\ \{\\
&\qquad\qquad\qquad\qquad\ \mathtt{m} \leftarrow \mathit{montecarlo}\ (\mathtt{n} - 1);\\
&\qquad\qquad\qquad\qquad\ \mathtt{x} \leftarrow \mathtt{d};\\
&\qquad\qquad\qquad\qquad\ \mathtt{return}\ ((1/\mathtt{n}) * (\mathtt{h}(\mathtt{x}) + \mathtt{m} * (\mathtt{n} - 1)))\ \}
\end{aligned}
$$

図 1　モンテカルロ法により h の期待値を近似計算するプログラム

$$
\begin{aligned}
&\Gamma \equiv \varepsilon : \mathtt{real}, \mu : \mathtt{real}, \sigma : \mathtt{real}, \mathtt{d} : P[\mathtt{X}], \mathtt{h} : \mathtt{X} \Rightarrow \mathtt{real}\\
&\Psi \equiv \{\sigma^2 = \mathbb{V}_{\mathtt{x}\sim\mathtt{d}}[\mathtt{h\ x}], \mu = \mathbb{E}_{\mathtt{x}\sim\mathtt{d}}[\mathtt{h\ x}], \varepsilon > 0, \mathit{integrable}\ \mathtt{d\ h}, \mathit{integrable}\ \mathtt{d}\ \mathtt{h}^2\}
\end{aligned}
$$

$$
\Gamma \mid \Psi \vdash_{\mathrm{PL}} \forall \mathtt{n} : \mathtt{nat}.\mathit{integrable}\ (\mathit{montecarlo}\ \mathtt{n})\ (\lambda \mathtt{x}.\mathtt{x}) \wedge \mathit{integrable}\ (\mathit{montecarlo}\ \mathtt{n})\ (\lambda \mathtt{x}.\mathtt{x}^2) \tag{1}
$$

$$
\Gamma \mid \Psi \vdash_{\mathrm{UPL}} \mathit{montecarlo} : \mathtt{nat} \Rightarrow P[\mathtt{real}] \mid \forall \mathtt{n} : \mathtt{nat}.\mathtt{n} > 0 \rightarrow \Pr_{\mathtt{y}\sim\mathtt{r\ n}}[|\mathtt{y} - \mu| \geq \varepsilon] \leq \sigma^2/\mathtt{n}\varepsilon^2 \tag{2}
$$

図 2　本研究で示したモンテカルロ法の検証

```
definition hp_montecarlo ::
   "('b × 'a qbs_prob_space) × ('a ⇒ real) ⇒ nat ⇒ real qbs_prob_space" where
"hp_montecarlo ≡
 hp_rec_nat (hp_return ℝ_Q (hp_const 0))
    (hp_lambda (hp_lambda (hp_bind var1
                           (hp_lambda (hp_bind var5
                                       (hp_lambda
        (hp_return ℝ_Q ((hp_app var5 var1 +_t var2 *_t hp_real var4) /_t hp_real (hp_suc var4)))))))))"
definition "Φmon ≡ {hp_const 0 <_PL var5, var4 =_PL hp_expect var2 var1, var3^t2 =_PL hp_var var2 var1,
                    hp_integrable var2 var1, hp_integrable var2 (var1 *_t var1)}"
```

図 3　Isabelle/HOL でのプログラム *montecarlo* と仮定 Ψ の定義

```
lemma montecalro_integrable:
",ℝ_Q,,ℝ_Q,,ℝ_Q,,monadP_qbs X,,exp_qbs X ℝ_Q | Φmon
     ⊢_PL ∀_PL n ∈_PL ℕ_Q. hp_integrable (hp_app hp_montecarlo (hp_const n)) hp_id
                    ∧_PL hp_integrable (hp_app hp_montecarlo (hp_const n)) (hp_id *_t hp_id)"
lemma montecarlo_judgement:
 ",ℝ_Q,,ℝ_Q,,ℝ_Q,,monadP_qbs X,,exp_qbs X ℝ_Q | Φmon
    ⊢_UPL hp_montecarlo ;; exp_qbs ℕ_Q (monadP_qbs ℝ_Q)
          | λr. ∀_PL n ∈_PL ℕ_Q. hp_const 0 <_PL hp_const n
                    ⟶_PL hp_prob (hp_app r (hp_const n)) {y. var5 ≤_PL |hp_const y -_t var4|_t}_t
                              ≤_PL (var3^t2 /_t hp_real (hp_const n)) *_t (hp_const 1 /_t var5^t2)"
```

図 4　モンテカルロ法を検証した Isabelle/HOL のコード

間の形式化が大部分を占めている．また表中では分けていないが，準ボレル空間全体のうち，準ボレル空間上の確率測度・確率モナドの形式化に，半分以上の量を要した．

Isabelle/HOL で実装した HPProg でのモンテカルロ法のプログラムと本研究で示した PL，UPL 判定式は図 3，図 4 のようになった．ラムダ式は，変数の束縛を自然数の番号で対応づける De Bruijn index で表されている．図 1，図 2 の変数とは $\mathtt{var5} = \varepsilon$, $\mathtt{var4} = \mu$ $\mathtt{var3} = \sigma$, $\mathtt{var2} = \mathtt{d}$, $\mathtt{var1} = \mathtt{h}$ と対応づけられる．証明は，先行研究 [12] の Section 2，Appendix 1 に示されている方針に沿って行なった．

標準ボレル空間	2600 行
準ボレル空間	8800 行
PPV	2400 行
モンテカルロ法 (図 2(1))	140 行
モンテカルロ法 (図 2(2))	300 行
合計	14240 行

3 Isabelle/HOL における測度論，確率論

Isabelle/HOL[11] とは証明支援系の一つで，数学や計算機科学に関する定義や定理を厳密に書くことをサポートするソフトウェアである．Isabelle/HOL は高階論理 (Higher-Order Logic, HOL) に基づいており，特徴として，

- 強い論理体系で，排中律や選択公理を使うことができる．
- 多相型，Haskell style の型クラスをサポートしている．
- Sledgehammer と呼ばれる built-in ツールを使うことにより，自動証明器による証明探索ができる．

等が挙げられる．

準ボレル空間は測度論をもとに構成されている．ここでは，この先用いる概念や記法を確認する．Isabelle/HOL のライブラリ [7] には，型`'a` 上の測度空間を値にもつ型`'a measure` が定義されている．測度空間 μ :: `'a measure` の構成要素である集合，σ-加法族，測度をそれぞれ

$$\texttt{space}\ \mu :: \texttt{'a set}$$
$$\texttt{sets}\ \mu :: \texttt{'a set set}$$
$$\texttt{emeasure}\ \mu :: \texttt{'a set} \Rightarrow \texttt{ennreal}$$

で取り出すことができる．型 `ennreal` は非負拡張実数を表す型である．2 つの測度空間の積測度は `X` $\bigotimes_M$ `Y` :: (`'a` × `'b`) `measure`，`X` から `Y` への可測写像全体の集合は `X` $\to_M$ `Y` :: (`'a` ⇒ `'b`) `set`，ルベーグ積分は $\int$ `x.f x` $\partial\mu$ :: `real` [†4] という記法が使われている．

位相空間 $(X, \mathcal{O}_X)$ を可測空間として参照する場合，可測集合全体は位相 $\mathcal{O}_X$ から生成されるボレル集合体 $\mathcal{B}[\mathcal{O}_X]$ である．Isabelle/HOL では定数 `borel` :: `'a` :: `topological_space measure` が定義されている．`borel` は，位相空間型クラスのインスタンスとなっている型に定義されている定数で，σ-加法族は位相から生成されるボレル集合体となっている [†5]．本論文の実装では，可測空間としての実数，及び自然数 (離散空間) をそれぞれ

definition `"real_borel ≡ borel :: real measure"`
definition `"nat_borel ≡ borel :: nat measure"`

という定数で表す．

(X, Σ_X) から (Y, Σ_Y) への可測写像 f と X 上の測度 μ について，$U \in \Sigma_Y$ に対して

$$f_*\mu(U) = \mu(f^{-1}(U))$$

と定めると，$f_*\mu$ は (Y, Σ_Y) 上の測度となる．$f_*\mu$ を測度 μ の可測写像 f による像測度という [†6]．

圏 **Meas** は対象が可測空間で，射が可測写像であるような圏である．**Meas** には Giry モナド [4] と呼ばれる確率分布のモナド $(G, \eta_G, \gg\!=_G)$ が定義される．可測空間 X に対して $G(X)$ は X 上の確率測度全体がなす空間である．Isabelle/HOL のライブラリでは，測度空間 μ :: `'a measure` に対して

$$\texttt{prob_algebra}\ \mu :: \texttt{'a measure measure}$$
$$\texttt{return}\ \mu :: \texttt{'a} \Rightarrow \texttt{'a measure}$$
$$\texttt{bind} :: \texttt{'a measure} \Rightarrow (\texttt{'a} \Rightarrow \texttt{'b measure}) \Rightarrow \texttt{'b measure}$$

が定義されている．

4 準ボレル空間の形式化

4.1 標準ボレル空間

準ボレル空間の前に，本研究で重要になる可測空間のクラスである標準ボレル空間を Isabelle/HOL で形式化する．$(X, \mathcal{O}_X)$ を完備で可分な距離づけ可能であるような位相空間 (ポーランド空間) とする．標準ボレル空間とは，可測空間 $(X, \mathcal{B}[\mathcal{O}_X])$ のことである．標準ボレル空間は，可算空間であるか $\mathbb{R}$ と同型になることが知られている (Kuratowski の定理)．ここでの標準ボレル空間は，証明で使いやすい形をした

†4 返り値を実数にそろえるため，可積分でない時は 0 を返すよう定義されている．

†5 測度は零測度が与えられている．

†6 f が確率変数の時は分布ともいう．

以下の同値な定義を採用する．

定義 4.1. X を可測空間とする．2 つの可測写像

$$X \xrightarrow{f} \mathbb{R} \xrightarrow{g} X$$

で，$g \circ f = id_X$ をみたすようなものが存在する時 X は標準ボレル空間であるという．

Isabelle/HOL では以下のように定義できる．

```
definition standard_borel_R_retract
    :: "'a measure ⇒ bool" where
"standard_borel_R_retract X ≡
    ∃f ∈ X →M real_borel.
    ∃g ∈ real_borel →M X.
     ∀x ∈ space X.  (g ∘ f) x = x"
```

$\mathbb{N} \times \mathbb{R}$ や $\mathbb{R} \times \mathbb{R}$ は標準ボレル空間である．

```
lemma nat_real_standard_borel:
 "standard_borel_R_retract
     (nat_borel ⨂M real_borel)"
lemma real_real_standard_borel:
 "standard_borel_R_retract
       (real_borel ⨂M real_borel)"
```

$\mathbb{R} \times \mathbb{R}$ が標準ボレル空間であることは，以下の概略に沿って証明した．

2 つの可測写像 $\alpha : (0,1) \times (0,1) \to (0,1)$ と $\beta : (0,1) \to (0,1) \times (0,1)$ で $\beta \circ \alpha = id_{(0,1)\times(0,1)}$ をみたすものをつくればよい．α と β を構成できたら，$\mathbb{R}$ と $(0,1)$ の間の同型を用いて，2 つの可測写像 $\mathbb{R} \times \mathbb{R} \xrightarrow{f} \mathbb{R} \xrightarrow{g} \mathbb{R} \times \mathbb{R}$ で $g \circ f = id$ となるような f と g を構成することができる．

$r \in (0,1)$ の小数点以下を 2 進数で展開した時の小数第 n 位を r_n で表すことにする †7．α は $(r, r') \in (0,1) \times (0,1)$ を $0.r_1 r'_1 r_2 r'_2 \ldots$ に写し，β は $r \in (0,1)$ を $(0.r_1 r_3 \ldots, 0.r_2 r_4 \ldots)$ に写すような写像であるとする †8．これらの写像は，非減少な単関数の極限として表すことができるので可測となる．また，定義から $\beta \circ \alpha = id_{(0,1)\times(0,1)}$ となる．

4.2 準ボレル空間

確率論では，標本空間とよばれる可測空間 (Ω, Σ_Ω) を考え，ランダムな事象は $A \in \Sigma_\Omega$ として表現される．ランダムな事象の観測は，確率変数と呼ばれる可測関数 $f : \Omega \to X$ を通じて行う．このように，通常の確率論では可測空間を定義し，その後に確率変数が定義される．準ボレル空間では，標本空間を $\mathbb{R}$ に限定した “確率変数” を定義するところから始まる．

定義 4.2 (Heunen et al, [6])**.** 集合 X と $M_X \subseteq \mathbb{R} \to X$ について，(X, M_X) が準ボレル空間 (quasi-Borel spaces) であるとは以下の 3 つの規則

1. $\alpha \in M_X$ で $f : \mathbb{R} \to \mathbb{R}$ が可測関数のとき，$\alpha \circ f \in M_X$．
2. $\alpha : \mathbb{R} \to X$ が定数関数なら $\alpha \in M_X$．
3. $\{S_i\}_{i\in\mathbb{N}}$ を各 S_i が可測集合であるような $\mathbb{R}$ の可算分割とし，$\{\alpha_i\}_{i\in\mathbb{N}} \subseteq M_X$ とすると，$\sum_i 1_{S_i} \alpha_i \in M_X$．

をみたすときをいう．

本論文を通して，準ボレル空間 (X, M_X) を単に X と表すことがある．

準ボレル空間は Isabelle/HOL 上で図 5 のように定義することができる．型の定義には，**typedef** コマンドを使う．**typedef** はすでにある型の，非空な部分集合を表す型を定義するコマンドである．**typedef** は 2 つの定数 `Abs_`*newtype* と `Rep_`*newtype* を生成する．直観的に，`Abs_`*newtype* は元の型から新しい型への関数，`Rep_`*newtype* は新しい型から元の型への関数である．準ボレル空間の構成要素である集合・関数の集合を取り出す関数は `Rep_quasi_borel` を使って定義できる (図 5)．ライブラリの測度空間も **typedef** コマンドで定義されている [7].

(X, Σ_X) を可測空間とし，M_{Σ_X} を $\mathbb{R}$ から X への可測写像全体の集合とすると (X, M_{Σ_X}) は準ボレル空間となる．$\mathbb{R}$ などの可測空間を準ボレル空間として参照するときは，このように定義される準ボレル空間を表すものとする．

定義 4.3 (Heunen et al, [6])**.** X と Y を準ボレル空間とし $f : X \to Y$ を写像とする．すべての $\alpha \in M_X$ について $f \circ \alpha \in M_Y$ となるとき，f は X から Y への準ボレル写像とよばれる．

Isabelle/HOL では図 6 のように定義できる．

恒等写像は準ボレル写像で，準ボレル写像の合成もまた準ボレル写像となるので，準ボレル空間を対象とし，準ボレル写像を射とする圏 **QBS** を構成できる．圏 **QBS** はカルテシアン閉で，可算直和ももつ

†7 ただし，無限に 1 が続かないような方法で表す．

†8 β は $0.1010\ldots$ を $(1,0)$ に写すため，値域が $(0,1) \times (0,1)$ となっていない．実際の証明では f と g を構成するときに工夫をする必要がある．

```
definition qbs_closed1 :: "(real ⇒ 'a) set ⇒ bool"
 where "qbs_closed1 Mx ≡ (∀a ∈ Mx. ∀f ∈ real_borel →M real_borel. a ∘ f ∈ Mx)"

definition qbs_closed2 :: "['a set, (real ⇒ 'a) set] ⇒ bool"
 where "qbs_closed2 X Mx ≡ (∀x ∈ X. (λr. x) ∈ Mx)"

definition qbs_closed3 :: "(real ⇒ 'a) set ⇒ bool"
 where "qbs_closed3 Mx ≡ (∀P::real ⇒ nat. ∀Fi::nat ⇒ real ⇒ 'a.
                          (∀i. P -‘ {i} ∈ sets real_borel)
                           ⟶ (∀i. Fi i ∈ Mx)
                           ⟶ (λr. Fi (P r) r) ∈ Mx)"
typedef 'a quasi_borel =
 "{(X::'a set, Mx). Mx ⊆ UNIV → X ∧ qbs_closed1 Mx ∧ qbs_closed2 X Mx ∧ qbs_closed3 Mx}"

definition qbs_space :: "'a quasi_borel ⇒ 'a set" where
  "qbs_space X ≡ fst (Rep_quasi_borel X)"
definition qbs_Mx :: "'a quasi_borel ⇒ (real ⇒ 'a) set" where
  "qbs_Mx X ≡ snd (Rep_quasi_borel X)"
```

図 5　準ボレル空間の定義

```
definition qbs_morphism :: "['a quasi_borel, 'b quasi_borel] ⇒ ('a ⇒ 'b) set"
  (infixr "→Q" 60) where
  "qbs_morphism X Y ≡ {f ∈ qbs_space X → qbs_space Y. ∀α ∈ qbs_Mx X. f ∘ α ∈ qbs_Mx Y}"
```

図 6　準ボレル写像の定義

(Heunen et al, [6], Propositions 16–18).

4.3　可測空間との関係

可測写像 $f : X \to Y$ は (X, M_{Σ_X}) から (Y, M_{Σ_Y}) への準ボレル写像となる．したがって，$R(X) = (X, M_{\Sigma_X})$, $R(f) = f$ とすることで函手 $R : \mathbf{Meas} \to \mathbf{QBS}$ が得られる．

次に，準ボレル空間から可測空間を構成する．

補題 4.4 (Heunen et al, [6])**.** (X, M_X) を準ボレル空間とする．このとき

$$\Sigma_{M_X} = \{U \mid \forall \alpha \in M_X . \alpha^{-1}(U) \in \Sigma_{\mathbb{R}}\}$$

とすると，(X, Σ_{M_X}) は可測空間となる．

f が X から Y への準ボレル写像であるとき，f は (X, Σ_{M_X}) から (Y, Σ_{M_Y}) への可測写像となるので，函手 $L : \mathbf{QBS} \to \mathbf{Meas}$ を $L(X) = (X, \Sigma_{M_X})$, $L(f) = f$ と定めることができる．

函手 L と R は **QBS** と **Meas** の間の随伴をなす．

命題 4.5 (Heunen et al, [6]. Prop 15)**.** (Y, Σ_Y) を可測空間とする．このとき，以下が成り立つ．

1. X を準ボレル空間とすると，
   $$f \in \mathbf{Meas}(L(X), Y) \Leftrightarrow f \in \mathbf{QBS}(X, R(Y))$$
2. X を標準ボレル空間とすると
   $$f \in \mathbf{Meas}(X, Y) \Leftrightarrow f \in \mathbf{QBS}(R(X), R(Y))$$

特に X が標準ボレル空間なら，$L(R(X)) = X$ である．また，標準ボレル空間に対して，R は可算直積と可算直和を保存する．

4.4　確率モナド

QBS 上の確率モナドは，**Meas** 上の Giry モナドから性質を継承している．ここでは，準ボレル空間上の確率測度，及び **QBS** 上の確率モナドを形式化する．

定義 4.6 (Heunen et al, [6])**.** 準ボレル空間 X 上の確率測度とは，同値類 $[\alpha, \mu]_\sim$ のことである．ただし $\alpha \in M_X$，$\mu \in G(\mathbb{R})$ で，同値関係は，

$$(\alpha, \mu) \sim (\alpha', \mu') \overset{\text{def}}{\Leftrightarrow} \alpha_* \mu = \alpha'_* \mu'$$

で定義される．

同値関係 $\sim$ は，

($\star$) すべての準ボレル写像 $f : X \to \mathbb{R}$ について
$$\int (f \circ \alpha)\, d\mu = \int (f \circ \alpha')\, d\mu'$$
という条件と同値である．したがって，準ボレル写像 $f : X \to \mathbb{R}$ と X 上の確率測度 $[\alpha, \mu]_\sim$ に対して積

```
type_synonym 'a qbs_prob_t = "'a quasi_borel * (real ⇒ 'a) * real measure"

definition qbs_prob_t_integral :: "['a qbs_prob_t, 'a ⇒ real] ⇒ real" where
"qbs_prob_t_integral p f ≡
  (if f ∈ (fst p) →Q ℝQ
   then (∫ x. f (fst (snd p) x) ∂ (snd (snd p)))
   else 0)"


quotient_type 'a qbs_prob_space = "'a qbs_prob_t" / qbs_prob_eq
  morphisms rep_qbs_prob_space qbs_prob_space

lift_definition qbs_prob_integral :: "['a qbs_prob_space, 'a ⇒ real] ⇒ real"
is qbs_prob_t_integral
  by(auto simp add: qbs_prob_t_integral_def qbs_prob_eq_def)
```

図 7　準ボレル空間上の確率測度と積分

```
definition monadP_qbs_Px :: "'a quasi_borel ⇒ 'a qbs_prob_space set" where
"monadP_qbs_Px  X ≡ {s :: 'a qbs_prob_space. qbs_prob_space_qbs s = X}"

definition monadP_qbs_MPx :: "'a quasi_borel ⇒ (real ⇒ 'a qbs_prob_space) set" where
"monadP_qbs_MPx X ≡ {β. ∃α∈ qbs_Mx X. ∃g ∈ real_borel →M prob_algebra real_borel.
                        ∀r::real. β r = qbs_prob_space (X,α,g r)}"

definition monadP_qbs :: "'a quasi_borel ⇒ 'a qbs_prob_space quasi_borel" where
"monadP_qbs X ≡ Abs_quasi_borel (monadP_qbs_Px X, monadP_qbs_MPx X)"
```

図 8　確率モナド P

分を

$$\int f\, d[\alpha,\mu]_\sim \stackrel{\text{def}}{=} \int (f\circ\alpha)\, d\mu$$

と定めることができる．

Isabelle/HOL での実装において，準ボレル空間上の確率測度は，**quotient_type** コマンド [8] を使って定義する (図 7)．**quotient_type** コマンドは，元となる型から同値関係で商をとった新しい型を生成する．型 `'a` 上の準ボレル空間，実数から `'a` への関数，実数上の測度の組の型 `'a qbs_prob_t` を，定義 4.6 の同値関係 (図 7 の `qbs_prob_eq`) で割ることで準ボレル空間上の確率測度の型を定義することができる †9．準ボレル空間上の確率測度による積分は，`'a qbs_prob_t` 上で定義した関数を `lift_definition` で商型上の関数に持ち上げることで定義することができる．このとき，関数が well-defined であることを表す *respectfulness theorem* の証明を与える必要がある．

QBS 上に確率測度のモナド $(P,\eta,\gg\!=)$ を構成する．

命題 4.7 (Heunen et al, [6])**.** X を準ボレル空間とする．$P(X)$ を X 上の確率測度全体の集合とし，

$$M_{P(X)} \stackrel{\text{def}}{=} \{\beta \mid \exists\alpha\in M_X.\exists g\in \mathbf{Meas}(\mathbb{R},G(\mathbb{R})).\ \forall r\in\mathbb{R}.\beta(r)=[\alpha,g(r)]_\sim\}$$

とする．このとき，$(P(X),M_{P(X)})$ は準ボレル空間となる．

証明は，$\mathbb{N}\times\mathbb{R}$ が標準ボレル空間であること (lemma `nat_real_standard_borel`) を使う．Isabelle/HOL での定義は図 8 のようになる．

命題 4.8 (Heunen et al, [6])**.** X を準ボレル空間，μ を $\mathbb{R}$ 上の任意の確率測度として，$\eta_X : X\to P(X)$ を

$$\eta_X(x) \stackrel{\text{def}}{=} [\lambda r.x,\mu]_\sim$$

とする．さらに，$[\alpha,\mu]_\sim\in P(X)$，$f:X\to P(Y)$ を準ボレル写像とする．このとき，f が準ボレル写像であることから，ある $\beta\in M_Y$ と $g\in\mathbf{Meas}(\mathbb{R},G(\mathbb{R}))$

†9　本当は，(X,α,μ) という組全体ではなく，「$\alpha\in M_X$ かつ μ が確率測度であるような組」の中で商を取る必要がある．実際の実装では半同値関係で割ることでこの要求をみたすよう定義されている．

があって，$(f \circ \alpha)(r) = [\beta, g(r)]_\sim$ となる．この β と g を使って

$$[\alpha, \mu]_\sim \gg\!= f \stackrel{\text{def}}{=} [\beta, \mu \gg\!=_G g]_\sim$$

と定義する．

これらの演算子は well-defined である．

補題 4.9 (Heunen et al, [6])．$(P, \eta, \gg\!=)$ は **QBS** 上の可換な強モナドである．

この補題を示すために，Giry モナドがモナドであることと $\mathbb{R} \times \mathbb{R}$ が標準ボレル空間であること (lemma `real_real_standard_borel`) を使う．

関数 l を $l([\alpha, \mu]_\sim) = \alpha_* \mu$ とすると，l は $L(P(X))$ から $G(L(X))$ への単射な可測写像となる．とくに X が標準ボレル空間のとき，l は全単射である．

5 PPV の形式化

この節では，PPV で使われる言語 HPProg，PL，UPL を導入する．実装に合わせるため，型や項の定義を少し変えている．

PPV には準ボレル空間に基づいた健全な意味論が与えられている．PPV の実装はこの意味論に沿って行なう．実装の方法は，対象の論理体系を証明支援系の論理で直接記述する shallow embbedding による．Shallow embedding による形式化は，型付け規則や導出規則を補題として示す形で追加していくため，定数や公理を後から追加するのが容易である．また，型付けや導出規則が HOL に対して健全であることを確認することができる．

5.1 言語: HPProg

PPV で使われる関数型確率的プログラミング言語 HPProg の型は

$$T ::= \texttt{unit} \mid \texttt{nat} \mid \texttt{bool} \mid \texttt{real} \mid \texttt{preal} \mid T \times T \mid T \Rightarrow T \mid P[T]$$

で定義される．ここで `real` および `preal` はそれぞれ実数 $\mathbb{R}$，非負拡張実数 $[0, \infty]$ を表す型，$P[T]$ は T 上の確率測度の型である．

HPProg の項は

$$e ::= x \mid c \mid f \mid e\ e \mid \lambda x.e \mid \langle e, e \rangle \mid \pi_i(e) \mid rec_nat\ e\ e \mid return\ e \mid bind\ e\ e \mid Bernoulli(e) \mid Gauss(e, e)$$

で定義される．項には標準的な方法で型をつける．型付け規則の例をいくつか記しておく．

$$\frac{\Gamma \vdash e : T}{\Gamma \vdash return\ e : P[T]}$$

$$\frac{\Gamma \vdash e : P[T] \qquad \Gamma \vdash f : T \Rightarrow P[T']}{\Gamma \vdash bind\ e\ f : P[T']}$$

$$\frac{\Gamma \vdash e : \texttt{real}}{\Gamma \vdash Bernoulli(e) : P[\texttt{bool}]}$$

$$\frac{\Gamma \vdash e : \texttt{real} \qquad \Gamma \vdash e' : \texttt{real}}{\Gamma \vdash Gauss(e, e') : P[\texttt{real}]}$$

意味論では，HPProg の型 T は **QBS** の対象 $[\![T]\!]$ として解釈され，言語 HPProg の型付けされた項 $\Gamma \vdash e : T$ は **QBS** の射 $[\![\Gamma]\!] \to [\![T]\!]$ として解釈される．**QBS** はカルテシアン閉であるから，関数抽象や関数適用は自然に解釈できる．実装における型判定は以下のように定義できる．

```
definition
 "hpprog_typing Γ e T ≡ e ∈ Γ →_Q T"
```

各型付け規則は補題として示すことになる．たとえば実数の定数は以下のように定義，型付けを行うことができる．

```
definition hp_const :: "'a ⇒ 'env ⇒ 'a" where
"hp_const k ≡ (λenv. k)"

lemma hpt_realc:
  "Γ ⊢_t (hp_const (r :: real)) ;; ℝ_Q"
  using qbs_morphism_const[of r "ℝ_Q" Γ]
  by(simp add: hpprog_typing_def hp_const_def)
```

5.2 論理: PL

論理式で扱う項は，HPProg の項に期待値を表す項 $\mathbb{E}_{x \sim t}[t\ x]$ を追加したものである．論理式は以下で定義される．

$$\phi ::= (t = t) \mid (t < t) \mid \top \mid \bot \mid \phi \wedge \phi \mid \phi \to \phi \mid \forall x : T.\phi \mid \exists x : T.\phi$$

PL の判定式は

$$\Gamma \mid \Psi \vdash_{\mathrm{PL}} \phi$$

という形をしている．Γ は型環境，Ψ は仮定の集合，ϕ は結論である．PL の導出規則には以下のようなものがある．

$$\frac{\phi \in \Psi}{\Gamma \mid \Psi \vdash_{\mathrm{PL}} \phi}\ \mathrm{AX}$$

$$\frac{\Gamma \mid \Psi \vdash_{\mathrm{PL}} \psi \to \phi \qquad \Gamma \mid \Psi \vdash_{\mathrm{PL}} \psi}{\Gamma \mid \Psi \vdash_{\mathrm{PL}} \phi}\ \to_E$$

$$\frac{\Gamma \mid \Psi \vdash_{\mathrm{PL}} \phi[t/x] \qquad \Gamma \mid \Psi \vdash_{\mathrm{PL}} t = u}{\Gamma \mid \Psi \vdash_{\mathrm{PL}} \phi[u/x]}\ \mathrm{SUBST}$$

また，数の計算に関する規則や期待値の線形性と

いった等式を公理として追加する．先行研究では，非負関数の期待値だけを考えることで可積分性の議論を省略していた．先行研究のモンテカルロ法の検証例では一般の関数も扱うので，本研究では可積分性の議論も行う．

$$\frac{\Gamma \mid \Psi \vdash_{\mathrm{PL}} \mathit{integrable}\ \mu\ f \qquad \Gamma \mid \Psi \vdash_{\mathrm{PL}} \mathit{integrable}\ \mu\ g}{\Gamma \mid \Psi \vdash_{\mathrm{PL}} \mathbb{E}_{x\sim\mu}[f\ x] + \mathbb{E}_{x\sim\mu}[g\ x] = \mathbb{E}_{x\sim\mu}[f\ x + g\ x]}$$

$$\frac{\Gamma \vdash e : P[Y] \qquad \Gamma, y : Y \vdash e' : Z \qquad \Gamma \vdash e'' : Z \Rightarrow \texttt{real}}{\Gamma \mid \Psi \vdash_{\mathrm{PL}} \mathbb{E}_{x\sim \mathit{bind}\ e\ \lambda y.\mathit{return}(e')}[e''] = \mathbb{E}_{y\sim e}[e''[e'/x]]}$$

PL 判定式 $\Gamma \mid \Psi \vdash_{\mathrm{PL}} \phi$ の意味は，「任意の $x \in \Gamma$ について，$\Psi\ x$ が成り立つならば，$\phi\ x$ も成り立つ」と定義される †10．Isabelle/HOL では以下のように定義できる．

```
definition "hp_conjall Ψ ≡ (λenv. ∀φ∈Ψ. φ env)"
definition "pl_der Γ Ψ φ ≡
  (∀x∈ qbs_space Γ. hp_conjall Ψ x ⟶ φ x)"
```

各推論規則は，補題として示す．

```
lemma pl_ax:
  assumes "φ ∈ Ψ"
  shows "Γ ∣ Ψ ⊢PL φ"

lemma pl_impE:
  assumes "Γ ∣ Ψ ⊢PL ψ ⟶PL φ"
      and "Γ ∣ Ψ ⊢PL ψ"
    shows "Γ ∣ Ψ ⊢PL φ"

lemma pl_expect_add:
  assumes "Γ ∣ Ψ ⊢PL hp_integrable t e1"
      and "Γ ∣ Ψ ⊢PL hp_integrable t e2"
    shows "Γ ∣ Ψ ⊢PL hp_expect t (e1 +t e2)
        =PL hp_expect t e1 +t hp_expect t e2"
```

論理式での代入 $\phi[t/x]$ は，Isabelle/HOL での関数適用を使い ϕ `t` と表すことができ，代入規則は以下の形になる．

```
lemma pl_subst':
 assumes "φ = (λt. λk. φ' k (t k))"
         "Γ ∣ Ψ ⊢PL t =PL u"
     and "Γ ∣ Ψ ⊢PL φ t"
   shows "Γ ∣ Ψ ⊢PL φ u"
```

規則の一つ目の仮定で，論理式中の項の意味の取り方を制限している．代入を Isabelle/HOL の関数適用として表すことで，判定式にルールを適用する時のインスタンス化がうまくいきやすくなり，証明の負担軽減になる．

5.3 プログラム論理: UPL

UPL は

$$\Gamma \mid \Psi \vdash_{\mathrm{UPL}} e : T \mid \phi$$

という形の判定式をもつ．PL との違いは，判定式に HPProg 項 e とその型 T が含まれている．また，結論の論理式 ϕ にはプログラム変数と呼ばれる型 T の項 $\mathbf{r}$ が含まれている．UPL の導出規則には以下のようなものがある．

$$\frac{\Gamma \mid \Psi \vdash_{\mathrm{UPL}} e : T \mid \varphi_1 \quad \Gamma \mid \Psi \vdash_{\mathrm{PL}} \varphi_1[e/\mathbf{r}] \to \varphi_2[e/\mathbf{r}]}{\Gamma \mid \Psi \vdash_{\mathrm{UPL}} e : T \mid \varphi_2}$$

$$\frac{\Gamma \mid \Psi \vdash_{\mathrm{UPL}} f : \tau \Rightarrow \sigma \mid \forall x.\phi' \to \phi[\mathbf{r}\ x/\mathbf{r}] \quad \Gamma \mid \Psi \vdash_{\mathrm{UPL}} e : \tau \mid \phi'}{\Gamma \mid \Psi \vdash_{\mathrm{UPL}} f\ e : \sigma \mid \phi[e/x]}$$

$\Gamma \mid \Psi \vdash_{\mathrm{UPL}} e : T \mid \varphi$ の意味は「$\Gamma \vdash e : T$」かつ「すべての $x \in \Gamma$ について，$\Psi\ x$ が成り立つなら $\varphi[e/\mathbf{r}]\ x$ が成り立つ」である．Isabelle/HOL では

```
definition "upl_der Γ Ψ e T φ ≡
 ((Γ ⊢t e ;; T) ∧ (∃φ'. φ = (λt k. φ' k (t k)))
  ∧ (∀k∈qbs_space Γ. hp_conjall Ψ k ⟶ φ e k))"
```

と定義できる．定義の「$\exists\varphi'.\varphi = (\lambda t\ k.\varphi'\ k\ (t\ k))$」は，プログラム e の意味の取り方を限定するために必要な条件である．プログラム変数 $\mathbf{r}$ は関数抽象の束縛変数として表す．Isabelle/HOL の実装における UPL 判定式の例を示す．

```
lemma
 "Γ ∣ Ψ ⊢UPL hp_const 1 ;; ℕQ
            ∣ λr. hp_const 1 ≤PL r +t r"
```

UPL でも同様に，各導出規則は補題として示す．

```
lemma upl_app:
  assumes "φ =
        (λt r env. φ' (t env) env (r env))"
         "Γ ∣ Ψ ⊢UPL f ;; exp_qbs T1 T2 ∣
 (λr. ∀PLx∈PLT1. ψ (hp_const x) ⟶PL
     φ (hp_const x) (hp_app r (hp_const x)))"
      and "Γ ∣ Ψ ⊢UPL e ;; T1 ∣ ψ"
    shows "Γ ∣ Ψ ⊢UPL
                 hp_app f e ;; T2 ∣ φ e"

lemma upl_sub:
  assumes "ψ = (λr env. ψ' env (r env))"
          "Γ ∣ Ψ ⊢UPL e ;; T ∣ φ"
      and "Γ ∣ Ψ ⊢PL φ e ⟶PL ψ e"
    shows "Γ ∣ Ψ ⊢UPL e ;; T ∣ ψ"
```

UPL は PL に対して完全である．つまり以下の定理が成り立つ．

定理 5.1 (Sato et al, [12]. Theorem 6.1)**.** 以下は同

†10 先行研究 [12] では，準ボレル空間上の述語の圏 **Pred**(**QBS**) を使って意味を与えていたが，本研究では実装が容易な，圏を介さない定義を採用する．

値である．

- $\Gamma \mid \Psi \vdash_{\mathrm{PL}} \phi[e/\mathbf{r}]$ が導出できる．
- $\Gamma \mid \Psi \vdash_{\mathrm{UPL}} e : T \mid \phi$ が導出できる．

PL や UPL が意味論に基づいて定義されているので，完全性は定義を展開することで直ちに証明される．

```
lemma pl_upl_complete:
  assumes "Γ ⊢_t t ;; T"
      and "φ = (λt k. φ' k (t k))"
    shows "(Γ | Ψ ⊢_PL φ t) ⟷
             (Γ | Ψ ⊢_UPL t ;; T | φ)"
  using assms
  by(auto simp add: pl_der_def upl_der_def)
```

6 まとめ

本研究では，準ボレル空間，及び意味論に基づいた PPV を Isabelle/HOL で形式化し，先行研究の例が証明支援系の上でも証明することができることを示した．

現在の PPV の実装は，プログラムが De Bruijn index で表されており，可読性が低くプログラムを書くことも容易ではないため，プログラムの表現方法を改良していきたい．

また，条件付き推論を含むプログラムの検証ができるように，準ボレル空間上の測度を一般の測度に拡張していく．準ボレル空間の圏 **QBS** 上には σ 有限測度モナドが構成できる [13]．その証明は確率分布モナド P の場合と大きく異なるため，形式化には相応の手間がかかると予想される．

参 考 文 献

[1] Aguirre, A., Barthe, G., Gaboardi, M., Garg, D., and Strub, P.-Y.: A Relational Logic for Higher-Order Programs, *Proc. ACM Program. Lang.*, Vol. 1, No. ICFP(2017).

[2] Aumann, R. J.: Borel structures for function spaces, *Illinois Journal of Mathematics*, Vol. 5, No. 4(1961), pp. 614 – 630.

[3] Eberl, M., Hölzl, J., and Nipkow, T.: A Verified Compiler for Probability Density Functions, *Programming Languages and Systems*, Vitek, J.(ed.), Berlin, Heidelberg, Springer Berlin Heidelberg, 2015, pp. 80–104.

[4] Giry, M.: A categorical approach to probability theory, *Categorical Aspects of Topology and Analysis*, Banaschewski, B.(ed.), Berlin, Heidelberg, Springer Berlin Heidelberg, 1982, pp. 68–85.

[5] Goodman, N. D. et al.: Church: a language for generative models, *UAI 2008, Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence*, AUAI Press, 2008, pp. 220–229.

[6] Heunen, C., Kammar, O., Staton, S., and Yang, H.: A Convenient Category for Higher-Order Probability Theory, *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '17, IEEE Press, 2017.

[7] Hölzl, J. and Heller, A.: Three Chapters of Measure Theory in Isabelle/HOL, *Interactive Theorem Proving*, van Eekelen, M., Geuvers, H., Schmaltz, J., and Wiedijk, F.(eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, 2011, pp. 135–151.

[8] Kaliszyk, C. and Urban, C.: Quotients Revisited for Isabelle/HOL, *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, New York, NY, USA, Association for Computing Machinery, 2011, pp. 1639–1644.

[9] Lochbihler, A.: Probabilistic Functions and Cryptographic Oracles in Higher Order Logic, *Programming Languages and Systems*, Thiemann, P.(ed.), Berlin, Heidelberg, Springer Berlin Heidelberg, 2016, pp. 503–531.

[10] Narayanan, P., Carette, J., Romano, W., Shan, C., and Zinkov, R.: Probabilistic inference by program transformation in Hakaru (system description), *International Symposium on Functional and Logic Programming - 13th International Symposium, FLOPS 2016, Kochi, Japan, March 4-6, 2016, Proceedings*, Springer, 2016, pp. 62–79.

[11] Nipkow, T., Wenzel, M., and Paulson, L. C.: *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, Springer-Verlag, Berlin, Heidelberg, 2002.

[12] Sato, T., Aguirre, A., Barthe, G., Gaboardi, M., Garg, D., and Hsu, J.: Formal verification of higher-order probabilistic programs: reasoning about approximation, convergence, Bayesian inference, and optimization, *Proceedings of the ACM on Programming Languages*, Vol. 3, No. POPL(2019), pp. 1–30.

[13] Ścibior, A., Kammar, O., Vákár, M., Staton, S., Yang, H., Cai, Y., Ostermann, K., Moss, S. K., Heunen, C., and Ghahramani, Z.: Denotational Validation of Higher-Order Bayesian Inference, *Proc. ACM Program. Lang.*, Vol. 2, No. POPL(2017).

[14] Stan Development Team: Stan Modeling Language Users Guide and Reference Manual, Version 2.27.0, 2021.

[15] Wood, F., van de Meent, J. W., and Mansinghka, V.: A New Approach to Probabilistic Programming Inference, *Proceedings of the 17th International conference on Artificial Intelligence and Statistics*, 2014, pp. 1024–1032.