少人数クラス

中村　薫

2022 年 10 月 1 日

目 次

1　Coq による q-類似の形式化

1.1　Coq

Coq とは, 定理証明支援系の 1 つであり, 数学的な証明が正しいかどうか判定するプログラムである. 人間がチェックすることが難しい複雑な証明でも正しさが保証され, また証明付きプログラミングにも応用される. 例えば, 命題 P, Q について, $P \Longrightarrow Q$ かつ P であれば, Q が成り立つということは, Coq では

```
From mathcomp Require Import ssreflect.

Theorem modus_ponens (P Q : Prop) : (P → Q) ∧ P → Q.
Proof.
  move⇒ [] pq p.
  by apply pq.
Qed.
```

と表現できる.
Coq による証明は, Curry-Howard 同型と呼ばれる,

$$\text{命題} \leftrightarrow \text{型}$$
$$\text{証明} \leftrightarrow \text{型に要素が存在する}$$

という対応関係に基づいている. また, 各論理演算子について, 以下のように対応している

$$P \text{ ならば } Q \quad P \to Q$$
$$P \text{ かつ } Q \quad P \times Q$$
$$P \text{ または } Q \quad P + Q$$

この同型をもとに上記の証明をもう一度考えてみると, $P \to Q$ と P という型に要素が存在することから, Q という型の要素を構成すればよいということである.
まず, 前提の要素それぞれに pq, p と名前をつける. これがプログラム中の `move⇒ [] pq p` のことである. ここで, $P \to Q$ という型は, 入力する値の型が P, 出力する値の型が Q であるような関数の型であるため, P の要素 p に pq を適用することで, Q の要素を構成することができる. この関数適用がプログラム中の `apply pq` のことである.

1.2 q-類似

q-類似とは, $q \to 1$ とすると通常の数学に一致するような拡張のことである. 例えば, 自然数 n の q-類似 $[n]$ は

$$[n] = 1 + q + q^2 + \cdots q^{n-1}$$

であり, $(x-a)^n$ の q-類似 $(x-a)_q^n$ は

$$(x-a)_q^n := \begin{cases} 1 & (n = 0) \\ (x-a)(x-qa)\cdots(x-q^{n-1}a) & (n \geq 1) \end{cases}$$

である. 本章では, $D_q(x-a)_q^n = [n](x-a)_q^{n-1}$(ここで, D_q は微分の q-類似) の形式化を目標とする.

1.3 形式化

様々な q-類似を考えるにあたって, まずは微分の q-類似から始める.

```
Definition dq (f : R → R) x := f (q * x) - f x.
Definition Dq f := dq f // dq id.
```

次に, 自然数の q-類似を定義する.

```
Definition qnat n : R := (q ^ n - 1) / (q - 1).
```

この `qnat` に対して, $(x^n)' = nx^{n-1}$ の q-類似が成り立つ.

```
Lemma qderiv_of_pow n x :
  x ≠ 0 → Dq (fun x ⇒ x ^ n) x = qnat n * x ^ (n - 1).
Proof.
  move⇒ Hx.
  rewrite /Dq /dq /qnat.
  rewrite -{4}(mul1r x) -mulrBl expfzMl.
    rewrite -GRing_add_div.
    rewrite [in x ^ n](_ : n = (n -1) +1) //.
      rewrite expfzDr // expr1z.
      rewrite mulrA -mulNr !red_frac_r //.
      rewrite GRing_add_div //.
      rewrite -{2}[x ^ (n - 1)]mul1r.
      rewrite -mulrBl mulrC mulrA.
      by rewrite [in (q - 1)^-1 * (q ^ n - 1)] mulrC.
    by rewrite subrK.
  by apply mulf_neq0.
Qed.
```

2　HoTT

3　coqgen プロジェクト

参考文献

[1] The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*