A Machine Proof of the Proposition "$\mathbf{Ideal} \subseteq \bigcup_i \mathbf{PrimeIdeal}_i \Rightarrow \mathbf{Ideal} \subseteq \mathbf{PrimeIdeal}_i$"

陳 凌鈞
Chen Lingjun
小林 英恒
Hidetsune KOBAYASHI
村尾 裕一
Hirokazu MURAO
日本大学理工学部
電気通信大学電気通信学部
鈴木 秀男
Hideo SUZUKI
職業能力開発総合大東京校

1 はじめに

数式処理および計算機代数の研究は発展し，昨今では，数値・数式融合計算を初めとして他方面の分野との協調に関する研究が盛んである．そうした試みのひとつとして，自動証明システムとの連携に関する研究も盛んになってきている．その多くは，数式処理における計算の正当性を保障したり，証明システムで必要となる数式の計算に数式処理システムを利用しようというものである．そのためには当然，代数の基本概念や基本事項を形式化し自動証明システムで記述することが必要となる．一方我々は，環論は証明の自動化もそれほど難しくないであろうという認識に立ち，自動証明や証明支援のソフトウェアシステムを用いて，環論の抽象的な理論そのものを計算機上で展開する研究を数年前から始めている．本稿では，その経過報告の一部として，我々の試みの初期の段階で扱った表題の命題の形式化と計算機上での証明の方法について概説する．

抽象代数の形式化の試みは，初歩的なものからある程度進んだものまで色々な報告があるが，多くは断片的なものである．F. Kammuller と L. C. Paulson は，Isabelle [8, 7] を用いて，群とその基本的な性質を形式化し， Sylow の定理の証明に成功している．また C. Ballarin はやはり Isabelle を用いて，環と一変数多項式の演算を形式化し，代数的符合理論の定理の証明を行っている [2]．当然，これらの仕事には，環・体・イデアルおよび環準同型などの形式化も含まれている．勿論，数学の形式化のプロジェクトとして有名な Mizar [6] でも，抽象代数の形式化は行われている．例えば，P. Rudnicki らは既に左/右イデアル，有限生成イデアル，イデアルの演算，Noether 環などの形式化まで行なっている [9].

本稿では，我々のプロジェクトの初期段階における同様の試み，即ち，可換代数における基本概念の定義や基本事項の証明の形式化において，我々が実際に Isabelle を用いて行った方法を報告する．証明の大部分は自動化されておらず，変形手順を逐一記述している．形式化の対象は，M.F. Atiyah と I.G. MacDonald による可換環論の教科書 [1] の内容を題材としている．これまでに第一章の内容である環とイデアルの基本的な性質の Isabelle での記述が完了している．本稿では，それらの基本概念と Isabelle の簡単な紹介を行い，その後で形式化に関する詳細な議論を行う．最後に，素イデアルに関するやや複雑な性質である，表題に示した命題の証明を例として示す.

2 環論における基本概念

ここでは，環論の抽象的な議論で必要となる基本概念を導入しておく．但し，環は単位元を含む可換環とし，また，群はアーベル群とする．

環 R では，加法と乗法の二項演算を定義する．環は加法に関して群を構成し，また，乗法に関しては，単位元を含み，交換則，結合則および分配則を満たすとする．環 R のイデアル I は加法に関する部分群で，$R \times I \subseteq I$ とする．**商環** R/I は，R の加法と乗法を自然に継承する．

環 R の部分集合によって生成されるイデアルは，その部分集合を含む最小のイデアルである．イデアルの和 $I_1 + I_2$ および積 $I_1 I_2$ はそれぞれ，$I_1 \cup I_2$ および集合 $\{ xy \mid x \in I_1 \& y \in I_2 \}$ によって生成されるイデアルと定義する．

環 R の要素 x について，$xy = 1$ (1 は環 R の単位元を表す) を満す $y \in R$ が存在するとき，x は**単元**であるという．**体**とは，$1 \neq 0$ であり，0 でない要素は全て単元であるような環のこと．R の任意の元 x について，**単項イデアル** (x) とは $\{ ax \mid a \in R \}$．イデアル I (但し $I \neq R$) について，$xy \in I \Rightarrow x \in I \mid y \in I$ が成り立つならば I は**素イデアル** (prime) である．イデアル m (但し $m \neq R$) を含む任意のイデアル I が必ず m か R に等しいとき，m は**極大イデアル** (maximal) である．**Jacobson 根基** (Jacobson radical) とは，全ての極大イデアルの共通集合である．

3 Isabelle/HOL とは

Isabelle は対話型の汎用証明システムであり，複数の論理をサポートしている．そのひとつである Isabelle/HOL は，Church の理論 [4] に基づく HOL システム [5] を統合したもので，高階論理 (higher order logic) を実装する．我々はこの Isabelle/HOL を用いて数学概念を形式化することとした．

Isabelle での証明は普通は backward proof である．つまり，利用者は先ず証明すべきゴールを宣言し，そのゴールに対し tactics(resulution, assumption, elimnation-resolution, destruct-resolution 等) や tactical(tactics の組み合わせ) や simplification を適用し，次に証明すべきサブゴール群へと変形する．各サブゴールに対し同様の操作を繰り返し，サブゴールが無くなった時点で証明は完了する．Isabelle には，証明の自動化のために他に induction, classical tableau reasoning 等が用意されている．

4 環の形式化

形式化は，教科書にならって群と部分群の定義から始め，環およびイデアルへと発展させる．群や環は Isabelle の `RECORD` を用いて定義する．Isabelle の `RECORD` は tuple を一般化したもので，継承の機能を有する．
群(アーベル群) の構造と定義：

```
record 'a groupSig =
  carrier :: "'a set"
  add     :: "['a, 'a] ⇒ 'a"
  inverse :: "'a ⇒ 'a"
  zero    :: "'a"
```

```
"AGroup G ==
  add G ∈ carrier G → carrier G → carrier G &
  inverse G ∈ carrier G → carrier G  &  zero G ∈ carrier G &
  (∀x ∈ carrier G. ∀y ∈ carrier G.  ∀z ∈ carrier G.
    (add G (inverse G x) x = zero G)  &  (add G (zero G) x = x) &
    (add G x y = add G y x)  &  (add G (add G x y) z = add G x (add G y z)) )"
```

但し，$A \to B$ は A から B への関数の集合を表す．

群の基本的な性質も証明しておく必要がある．たとえば，零の唯一性は次のように記述する．

```
"[| AGroup G; a ∈ carrier G; x ∈ carrier G; add G x a = a |]⟹ x = zero G";
```

但し，x ∈ carrier G は x ∈ G を意味し，x : carrier G と入力してもよい．つぎに 部分群H は群 G の carrier の部分集合として定義する．

```
"Subgroup G H == H ⊆ carrier G  &  zero G ∈ H  &
                 ( ∀x ∈H. ∀y ∈H.  (add G x y ∈ H ) & (inverse G x ∈ H) )"
```

環 は，加法演算を継承して群の拡張として定義する．その構造 (record) と定義は次とおり．

```
record 'a ringSig = 'a groupSig +
  mul :: "['a, 'a] ⇒ 'a"
  one :: "'a"
```

```
"Ring R ==
  AGroup R &
  one R ∈ carrier R & mul R ∈ carrier R → carrier R → carrier R
  (∀x ∈ carrier R. ∀y ∈ carrier R.  ∀z ∈ carrier R.
    (mul R (mul R x y) z = mul R x (mul R y z)) &
    (mul R (add R x y) z = add R (mul R x z) (mul R y z)) &
    (mul R x (add R y z) = add R (mul R x y) (mul R x z)) &
    (mul R (one R) x = x)  &  (mul R x (one R) = x) &
    (mul R x y = mul R y x) )"
```

この定義より，環が加法に関して群となる ("Ring R ⟹ AGroup R") ことが Isabelle で証明可能であり，その結果，加法群に関するすべての定理が環に対しても有効となる．また，定義より基本的な性質も導出される．たとえば，

```
"[| Ring R; x ∈ carrier R |]⟹ mul R x (zero R) = zero R";
```

イデアルの性質の証明において n 個の 要素の和 と 積 が必要となるが，これらは次のように再帰的に定義する．

```
primrec
  nsum0_0   "nsum0 R f 0 = f 0"
  nsum0_suc "nsum0 R f (Suc n) = add R (f (Suc n)) (nsum0 R (f n))"
```

```
primrec
  nmul0_0   "nmul0 R f 0 = f 0"
  nmul0_suc "nmul0 R f (Suc n) = mul R (f (Suc n)) (nmul0 R (f n))"
```

あるひとつの要素について，複数回加えたあわせた和 (スカラー倍) とかけあわせた積 (べき乗) を各々 nscale と npow で表し，次のとおり定義する．

```
primrec
  nscale_0   "nscale R x 0 = zero R"
  nscale_suc "nscale R x (Suc n) = add R x (nscale R x n)"
```

```
primrec
  npow_0   "npow R x 0 = one R"
  npow_suc "npow R x (Suc n) = mul R x (npow R x n)"
```

$(-x)^n = x^n \mid -x^n$ や $(xy)^n = x^n y^n$ などの初等的な性質は帰納法により証明が可能．さらにこれらを用いて $(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i y^{n-i}$ という有用な定理も証明される．

```
"!!n. [| Ring R; x ∈ carrier R; y ∈ carrier R |] ⟹
    npow R (add R x y) n =
      nsum0 R (%i. nscale R (mul R (npow R x (n-i)) (npow R y i))
                            (n choose i)) n ";
```

環 R_1 から R_2 への 準同型写像 f：

```
"RingHomo R1 R2 f ==
  Ring R1  &  Ring R2  &  (f ∈ carrier R1 → carrier R2) &
  (∀x ∈ carrier R1. ∀y ∈ carrier R1.
    (f (add R1 x y) = add R2 (f x) (f y)) &
    (f (mul R1 x y) = mul R2 (f x) (f y))) &
  (f (one R1) = one R2 )"
```

以上より，環準同型は零は零へと，また逆元 (inverse) は逆元へと写すことが Isabelle で証明可能．

零因子 の集合：

```
"ZeroDivisor R ==
  { x. Ring R  &  x ∈ carrier R &
       (∃y ∈ carrier R. (y ≠ zero R)  &  (mul R x y = zero R)) } "
```

整域：

```
"IntegralDomain R ==
  Ring R  &  (one R ≠ zero R)  &  (∀x ∈ ZeroDivisor R. x = zero R) "
```

nilpotent 要素 (???) は上のべき乗の定義を用いて，次のとおり記述：

```
"Nilpotent R x == x ∈ carrier R  &  (∃n. npow R x n = zero R) "
```

R の 単元 全体：

```
"Unit R ==
  { x. Ring R  &  x ∈ carrier R  &  (∃y ∈ carrier R. mul R x y = one R) } "
```

体：

```
"Field R ==
  Ring R  &  (one R ≠ zero R) &
    (∀x ∈ carrier R. (x ≠ zero R) ⟶ (x ∈ Unit R)) "
```

5 イデアルの形式化

イデアル I は第 2 節で説明したとおり．形式化すると：

```
"Ideal R I ==
  Ring R  &  Subgroup R I  &  (∀x ∈ carrier R. ∀y ∈ I. mul R x y ∈ I) "
```

単項イデアル

```
"PrincipalIdeal x R == mul R x ‘ carrier R "
```

『‘』は Isabelle では，関数を集合に適用した像を意味する．素イデアル

```
"PrimeIdeal R I ==
  Ideal R I  &  (I ≠ carrier R) &
  (∀x ∈ carrier R. ∀y ∈ carrier R. (mul R x y ∈ I) ⟶ (x ∈ I | y ∈ I)) "
```

極大イデアル

```
"MaximalIdeal R I ==
  Ideal R I  &  (I ⊂ carrier R) &
  ¬(∃A. (Ideal R A) & (I ⊂ A & A ⊂ carrier R)) "
```

以降では，商環を扱うが，その carrier の集合の要素である coset を定義する．

```
"coset G H x == { z.  ∃y ∈ H. z = add G x y }"
```

coset に関する性質をいくつか用意する．

- `"[| AGroup G; Subgroup G H; x ∈ carrier G; coset G H x = H |]⟹ x ∈ H ";`
- `"[| AGroup G; Subgroup G H; x ∈ H |]⟹ coset G H x = H ";`
- `"[| AGroup G; Subgroup G H |]⟹ coset G H (zero G) = H ";`

coset の演算は次のとおり．

和
```
"cosetAdd G A B == { z.  ∃x ∈ A. ∃y ∈ B. z = add G x y }"
```

逆元
```
"cosetInverse G A == { y.  ∃x ∈ A. y = inverse G x }"
```

積
```
"cosetMul R H A B ==
  { z.  ∃x ∈ A. ∃y ∈ B. ∃z0 ∈ H. z = add R (mul R x y) z0 }"
```

これらを用いて 商環 を次の record として定義する．

```
"QRing R H == (|
  carrier = cosetset R H,  add = cosetAdd R,
  inverse = cosetInverse R,  zero = coset R H (zero R),
  mul = cosetMul R H,  one = coset R H (one R) |)"
```

この構造が環であること

```
"Ideal R I ⟹ Ring (QRing R I)"
```

及び，環 R の要素を対応する coset に写す写像 cosetMap が環準同型であることが証明できる．

```
"Ideal R I ⟹ RingHomo R (QRing R I) (cosetMap R I)";
```

さらに，cosetMap に関する性質をいくつか証明することにより，以下の事実も証明可能．

```
"PrimeIdeal R I ⟹ IntegralDomain (QRing R I) ";
"[| Ideal R I; IntegralDomain (QRing R I) |]⟹ PrimeIdeal R I ";
"MaximalIdeal R I ⟹ Field (QRing R I)";
"[| Ideal R I; Field (QRing R I) |]⟹ MaximalIdeal R I";
```

Zorn の補題 (Isabelle では証明済み) を用いれば，極大イデアルが存在することを証明可能．

```
"[| Ring R; ¬(ZeroRing R) |]⟹ ∃I. MaximalIdeal R I";
```

更に

```
"[| Ring R; Ideal R I; I ≠ carrier R |]⟹ ∃J. I ⊆ J & MaximalIdeal R J ";
"[| Ring R; x ∈ carrier R; x ∉ (Unit R) |]⟹ ∃I. (x ∈ I & MaximalIdeal R I) ";
```

べき零根基(nilradical) は，nilpotent 要素全体の集合：

```
"Nilradical R == { x.  Nilpotent R x } "
```

べき零根基がイデアルであることは，相当な手間を要するが Isabelle でも証明可能．

```
"Ring R ⟹ Ideal R (Nilradical R) ";
```

べき零根基がすべての素イデアルの共通集合であるという事実

```
"[| Ring R; ¬(ZeroRing R) |]⟹ Nilradical R = ⋂ { I.  PrimeIdeal R I } ";
```

も，かなりの手間を要するが証明が可能．証明は次の 2 つの段階からなる．

1. べき零根基の任意の要素 x はあらゆる素イデアルに含まれる．何故なら，x はべき零根基の要素なので，ある正数 n に対し $x^n = 0$ となり，このべき乗はすべての素イデアルに含まれるが，素イデアルの性質からべき乗でなく x そのものが素イデアルに含まれなければならない．

2. べき零根基に含まれない要素 x に対しては，x を含まない素イデアル I が必ず存在する．イデアルの集合「"{ I. Ideal R I & (∀n. (npow R x n) ∉ I) }" 」を考えると，Zorn の補題よりこの集合の極大要素であるイデアル M が存在するが，その M が素イデアルであることが証明できる．勿論，x = npow R x 1 ∉ M.

Jacobson radical に関する有用な性質 (前掲の教科書の Proposition 1.9) も証明済み．

```
"[| Ring R; x ∈ carrier R |]⟹
  (x ∈⋂ { I.  MaximalIdeal R I })
    = (∀y ∈ carrier R. (add R (one R) (inverse R (mul R x y))) ∈ Unit R)";
```

6 イデアルの演算

2 つのイデアルの共通集合もイデアルである.

```
"[| Ideal R I₁; Ideal R I₂ |]⟹ Ideal R (I₁ ∩ I₂) ";
```

複数個に一般化すると

```
"[| IndSet ≠ {}; ⋀i. i ∈ IndSet ⟹ Ideal R (f i) |]
    ⟹ Ideal R (⋂i∈IndSet. (f i)) ";
```

集合 A で生成されるイデアルとは，A を含むイデアルの共通集合である.

```
"IdealGenerated R A == ⋂ { I.  Ideal R I  &  A ⊆ I } "
```

```
"[| Ring R; A ⊆ carrier R |]⟹ Ideal R (IdealGenerated R A)";
```

2 つのイデアルの和と積を次のとおり定義する.

```
"IdealSum R I₁ I₂ == IdealGenerated R (I₁ ∪ I₂)"
```

```
"IdealProd R I₁ I₂ == IdealGenerated R { z.  ∃x ∈ I₁. ∃y ∈ I₂. z = mul R x y }"
```

これらがイデアルになることは，Isabelle でも割りと簡単に示すことができる．特に，和の要素は各イデアルの要素の和である.

```
"[| Ideal R I₁; Ideal R I₂ |]
    ⟹ IdealSum R I₁ I₂ = { z.  ∃x ∈ I₁. ∃y ∈ I₂. z = add R x y }";
```

この和の演算は可換で，かつ，結合則を満たす.

```
"IdealSum R I₁ I₂ = IdealSum R I₂ I₁ ";
```

```
"[| Ideal R I₁; Ideal R I₂; Ideal R I₃ |]
    ⟹ IdealSum R (IdealSum R I₁ I₂) I₃ = IdealSum R I₁ (IdealSum R I₂ I₃)";
```

積についても，可換で結合則が成り立ち，和と合わせた時に分配則が成り立つことも Isabelle で容易に示すことが可能．modular 則は

```
"[| Ideal R I₁; Ideal R I₂; Ideal R I₃; I₂ ⊆ I₁ |]⟹
    I₁ ∩ (IdealSum R I₂ I₃) = IdealSum R (I₁ ∩ I₂) (I₁ ∩ I₃)";
```

イデアルが互いに素 (coprime) とは

```
"IdealCoprime R I₁ I₂ == (IdealSum R I₁ I₂ = carrier R ) "
```

互いに素な 2 つのイデアルの共通集合は積に等しい.

```
"[| Ideal R I₁; Ideal R I₂; IdealCoprime R I₁ I₂ |]⟹ I₁ ∩ I₂ = IdealProd R I₁ I₂ ";
```

7 素イデアルの和集合に関する証明

これまでの準備に基づき，いよいよ表題の命題を形式化と証明を考察する．即ち

> **Proposition 1** *Let $P_0, \cdots, P_n$ be prime ideals and I be an ideal contained in $\bigcup_{i=0}^{n} P_i$, then $I \subseteq P_i$ for some i.*

主要部分の対偶をとり，Isabelle で記述すると次のようになる．

```
"⋀(n :: nat). Ideal R I ⟹
    ∀P. (∀i≤n. (PrimeIdeal R (P i)) & ¬(I ⊆ (P i)) ⟶ (¬(I ⊆ ⋃{..n} P)) )";
```

証明は n に関しる帰納法による．まず $n = 0$ の場合は自明で Isabelle でも自動的に証明が可能．次に，上の命題が n の場合に成り立てば $n+1$ の場合にも成り立つことを示す．証明の要点は，$\bigcup_{i=0}^{n+1} P_i$ の要素ではない I の要素をいかに見つけるかである．そのような要素 $z \in I$ $(z \notin \bigcup_{i=0}^{n+1} P_i)$ が見つかれば証明は済んだに等しい．集合関連の簡単な事実も証明して変形規則として適用した結果，帰納法で証明すべき事実は次のように表される．

```
"⋀na P.
  [| Ideal R I;
    ∀f. (∀i. i ≤ na ⟶ PrimeIdeal R (f i) & ¬(I ⊆ (f i)))
        ⟶ (∃x∈ I. ∀i. i ≤ na ⟶ x ∉ (f i));
    ∀i. i ≤ (Suc na) ⟶ PrimeIdeal R (P i) & ¬(I ⊆ (P i)) |]
                ⟹ ∃x ∈ I. ∀i. i ≤ (Suc na) ⟶ x ∉ (P i)"
```

前提条件の 2 つの条件から，simplification により直接，次の条件を満たす要素 x の存在が得られる．

```
"x ∈ I; ∀i. i ≤ na ⟶ x ∉ (P i); "
```

この x が ∉ P(na+1) ならば，この x を z として完了．そうでなければ

$$S^i(j) = \begin{cases} j & \text{if } j < i \\ j+1 & \text{otherwise} \end{cases}$$

という関数を用意すれば，

```
"∀i ≤na. ∃x ∈ I. ∀ j ≤ na. x ∉ (P (S^i j)) "
```

という事実が証明される．これより，次が示される．

```
"∀i. i ≤ na ⟶ (∃x ∈ I. ∀j. j ≤ (Suc na) ⟶ j ≠ i ⟶ x ∉ (P j)"
```

選択公理より，自然数から I への関数が存在し，次を得る．

```
"∃y. ∀ (i :: nat) ≤ na. (y i) ∈ I & (∀j ≤ (Suc na). j ≠ i ⟶ (y i) ∉ (P j) )"
```

もしいずれかの y_i (=y i) が $\notin P_i$ $(i \leq$ na$)$ ならば，その y_i を z として完了．さもなくば，$z =$ ‘x $+ \prod_{i=0}^{na} y_i$ とする．$z \in I$ でかつ $z \notin \bigcup_{i=0}^{n+1} P_i$ であることは，Isabelle でも簡単に示すことができる．この証明で用いた induction，induction に関連する汎用的な事実，及び，証明全体に関する詳細については [3] を参照．

8 おわりに

本稿では，表題に掲げた事実と証明の形式化の方法の概略を述べた．その方法は Isabelle を用いて実装されている．今後，readability を向上させるために，できるだけ早く Isabelle/Isar [10] へと移行し，併せて，ドキュメントを同梱するように書き直す予定である．そうすることにより，ドキュメント生成や theory の依存関係のグラフ表示などのツールの利用が可能となる．また，さらにより多くの命題の形式化と証明の実装を進めると共に，数学的事実のより良い構造化と証明の自動化の可能性の検討を進める予定である．

参 考 文 献

[1] M. F. Atiyah and I. Macdonald. *Introduction to Commutative Algebra*. Addison-Wesley, 1969.

[2] C. Ballarin. *Computer Algebra and Theorem Proving*. PhD thesis, University of Cambridge, 1999. available from `http://iaks-www.ira.uka.de/iaks-calmet/ballarin/publications.html`.

[3] Chen Lingjun, H. Kobayashi, H. Murao, and H. Suzuki. Notes on formalizing induction on the number of sets. In S. Colton and V. Sorge, editors, *Second Workshop on the Role of Automated Deduction in Mathematics: RADM. In conjunction with CADE-18*, pages 11–23, 2002.

[4] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[5] M. J. C. Gordon and T. F. Melham, editors. *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, 1993. see also `http://www.dcs.gla.ac.uk/ tfm/fmt/hol.html`.

[6] Mizar project. `http://www.mizar.org`.

[7] L. C. Paulson. The Isabelle reference manual. `http://isabelle.in.tum.de/doc/ref.pdf`.

[8] L. C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5:363–397, 1989.

[9] P. Rudnicki, C. Schwarzweller, and A. Trybulec. Commutative algebra in the Mizar system. *Journal of Symbolic Computation*, 32(1/2):143–169, 2001.

[10] M. Wenzel. Isabelle/Isar reference manual. `http://isabelle.in.tum.de/doc/isar-ref.pdf`.