Regular Paper

可換環の局所化
Localization of Commutative Rings

渡瀬 泰成 *
Yasushige Watase*

* ywatase@ss.iij4u.or.jp

Proof checked by Mizar Version: 8.1.08 and MML Version: 5.53.1335
Received: November 28, 2018. Accepted: April 11, 2019.

Abstract

This article reports current progressing work on formalizing localization of a commutative ring. It consists mainly of a construction of ring of fraction. It is the generalized formalization of field of fraction of a integral domain [1]. Firstly it was formalized that a process of construction of a ring of fraction with respect to a multiplicative closed set. Secondly some properties and some interpretation of a ring of fraction in Zariski topolgical space.

1 はじめに

本稿では環の局所化の形式化について作成中のアーティクルについて報告する．FM 誌 26(4), 2018 にザリスキ位相の形式化を行ったアーティクル TOPZARI1 が掲載され，本稿はその続編となり，ちょうど参考文献 [2] の 1 章が TOPZARI1，3 章が本稿で扱うアーティクルに対応する．

2 環の局所化

環 R と R の部分集合で積閉集合 S が与えられると，分子を R の元，分母を S の元と思って分数の一般化が展開できる．R が整域の場合は，$\mathrm{R} \setminus 0 = \mathrm{R}^{\times}$ は積閉集合となり分数を定義して得られる分数環が商体となる．これは QUOFIELD（文献 [1]）として先行アーティクルとして形式化されている．

$\mathrm{R} \times \mathrm{R}^{\times}$ の元 $[a, s], [b, t]$ が等しい関係を $a \cdot t - b \cdot s = 0$ とすると，これは同値関係となり，商集合上に和・積が定義される．この商集合上にできた代数的対象は体の公理を満たし商体と呼ぶ．分数の表示は同値類の元を a/s と表記していることになる．

零因子を含む環の場合の分数環の構成を形式化を行う．

2.1 分数を定義する関係

環 R とその積閉な部分集合を S として，それらの直積集合 R × S に以下に定義した Frac_Equal という関係を定義する．有理数が等しいときの条件の一般化である．

Listing 1. TOPZARI2 - def 1

```
definition
  let R,S;
  let x,y be Element of [:[#]R,S:];
  pred x,y Frac_Equal R,S means
:: TOPZARI2:def 1

  ex x1,y1 be Element of [#]R,x2,y2,s1 be Element of S
  st x = [x1,x2] & y = [y1,y2] & (x1 * y2 − y1 * x2) * s1 = 0.R;
end;
```

関係 Frac_Equal は同値関係となることが証明される．

2.2 直積集合上の 2 項演算

つぎに $[\mathrm{R},\mathrm{S}] \times [\mathrm{R},\mathrm{S}]$ に和と積の 2 項演算を定義する．

R × S の算法 R × S に加法と乗法を定義する．加法は padd と表記して以下の対応を与える．

$$
\begin{array}{ccc}
[\mathrm{R},\mathrm{S}] \times [\mathrm{R},\mathrm{S}] & \stackrel{\text{padd}}{\to} & [\mathrm{R},\mathrm{S}] \\
\cup & & \cup \\
[x1, y1] \times [x2, y2] & \longmapsto & [x1 \cdot y2 + y1 \cdot x2, x2 \cdot y2]
\end{array}
$$

Listing 2. TOPZARI2 - def3

```
definition
  let R,S;
  let x,y be Element of [:[#]R,S:];
  func padd(x,y) −> Element of [:[#]R,S:] means
:: TOPZARI2:def 3

  ex x1,y1 be Element of [#]R,x2,y2 be Element of S
  st x = [x1,x2] & y = [y1,y2] & it = [x1*y2 + y1*x2, x2*y2];
end;
```

乗法は pmult と表記して以下の対応を与える．

$$
\begin{array}{ccc}
[\mathrm{R},\mathrm{S}] \times [\mathrm{R},\mathrm{S}] & \stackrel{\text{pmult}}{\longrightarrow} & [\mathrm{R},\mathrm{S}] \\
\cup & & \cup \\
[x1, y1] \times [x2, y2] & \longmapsto & [x1 \cdot x1, y1 \cdot y2]
\end{array}
$$

Listing 3. TOPZARI2 - def4

```
definition
  let R,S;
  let x,y be Element of [:[#]R,S:];
  func pmult(x,y) -> Element of [:[#]R,S:] means
:: TOPZARI2:def 4

  ex x1,y1 be Element of [#]R,x2,y2 be Element of S
  st x = [x1,x2] & y = [y1,y2] & it = [x1*y1, x2*y2];
end;
```

2.3　R × S の同値関係による商集合 ClassEqRel(R, S) の演算

加法 padd と乗法 pmult は ClassEqRel(R, S) 上に拡張され，それぞれ代表元によらず定義されことが証明される．算法の結合則を検証に於いて TOPZARI2:def 3, TOPZARI2:def 4 の定義では，関数の入れ子で演算の定義域に不整合が生じ，以下の様にそれぞれ fr_add と fr_mult と再定義にて解消した．

Listing 4. TOPZARI2 - def5

```
definition
  let R,S;
  func fr_add(R,S) -> BinOp of [:[#]R,S:] means
:: TOPZARI2:def 5

  for x,y holds it.(x,y) = padd(x,y);
end;
```

Listing 5. TOPZARI2 - def6

```
definition
  let R,S;
  func fr_mult(R,S) -> BinOp of [:[#]R,S:] means
:: TOPZARI2:def 6
  for v,w being Element of [:[#]R,S:] holds it.(v,w) = pmult(v,w);
end;
```

次に以下の 2 元を定義し，ClassEqRel(R, S) の演算として加法と乗法の単位元となることの証明を形式化した．

Listing 6. TOPZARI2 - def 9

```
definition
  let R,S;
  func 0.(R,S) -> Element of [:[#]R,S:] equals
:: TOPZARI2:def 9

  [0.R ,1.R ];
end;

definition
  let R,S;
  func 1.(R,S) -> Element of [:[#]R,S:] equals
```

```
:: TOPZARI2:def 10

  [1.R ,1.R ];
end;
```

以上の準備の下，ClassEqRel(R, S) に環構造が構成できこれを Fr_Ring(R, S) と定義した．短い表記としては S~R とした．この環は積閉集合 S に関する分数環という．特に $\mathfrak{p}$ を素イデアルとすると，$R \setminus \mathfrak{p}$ は積閉集合となる．この積閉集合で定義される分数環を $R_\mathfrak{p}$ と表記する．$R_\mathfrak{p}$ は局所環となり，正確な言い方ではないが，R の点 $\mathfrak{p}$ における局所化といい，本稿タイトルの所以となる．

Listing 7. TOPZARI2 - def 11

```
definition
  let R, S;
  func Fr_Ring(R,S) -> strict doubleLoopStr means
:: TOPZARI2:def 11

  the carrier of it = Class EqRel(R,S) & 1.it = Class(EqRel(R,S),1.(R,S)) &
  0.it = Class(EqRel(R,S),0.(R,S)) &
  (for x, y being Element of it ex a, b being Element of [:[#]R,S:] st
   x = Class(EqRel(R,S),a) & y = Class(EqRel(R,S),b) &
  (the addF of it).(x,y) = Class(EqRel(R,S),a+b) ) &
  for x, y being Element of it ex a, b being Element of [:[#]R,S:] st
  x = Class(EqRel(R,S),a) & y = Class(EqRel(R,S),b) &
  (the multF of it).(x,y) = Class(EqRel(R,S),a*b );
end;
```

Fr_Ring(R, S) が環の公理を満たすことをチェックして以下を得る．

Listing 8. TOPZARI2 - Th.27

```
theorem :: TOPZARI2:27
   S~R is Ring;
```

2.4 分数環の標準射

次の図式 R から S~R への標準的な環準同形を形式化した．

$$\begin{array}{ccc} R & \xrightarrow{\text{canHom}} & \mathrm{Fr\_Ring}(R,S) \\ \cup & & \cup \\ x & \longmapsto & [x, 1_{\mathrm{Fr\_Ring}(R,S)}] \end{array}$$

この射は単射となり，分数環の普遍性に現れる標準的な射となる．

Listing 9. TOPZARI2 - def 12

```
definition
let R,S;
func canHom(R,S) -> Function of R, S~R means
:: TOPZARI2:def 12
  for x being Element of R holds it.x = Class(EqRel(R,S),[x,1.R]);
end;
```

2.5 作成予定の形式化すべき定理

以上の『環の局所化』の形式化の定理，定義を実装したが，既に 2,000 行のコード化となっている．あまり多くを形式化するとアーティクル自体が長いものになり重要な事項を優先して全体をまとめる予定である．以下をコード化することとなろう．

- 分数環の普遍写像性（Universal Mapping Property）．
- $A \setminus \mathfrak{p}$ が積閉集合であること．
- $A_{\mathfrak{p}}$ は局所環．
- ザリスキ空間の開集合と局所化の関連．

3 考察と展望

参考とした可換代数の教科書 [2] では積閉集合の定義では零元の条件はない．他の多くの邦書では積閉集合の定義では零元含まないという条件を科す．形式化を展開する場合は後々のため定義の条件は最小限にすべきと考える．余計な条件は将来，繰り返して場合分けや条件チェックを証明に要求することになるからである．

局所化は，A-加群に対しても定義されるので，形式化の順番として，加群の局所化を定義して A 自身も A-加群なので特別なケースとして扱うこともできた．環上の加群はすでに 90 年代初頭より形式化されてきた．今後は [2] の 2 章の内容の様に完全列を利用した機能的な取り扱いも形式化が必要となると考える．

参考文献

[1] Schwarzweller C. The Field of Quotients Over an Integral Domain. Formalized Mathematics. 1998;7(**1**):69–79. Available from: `http://fm.mizar.org/1998-7/pdf7-1/quofield.pdf`.

[2] Atiyah MF, MacDonald IG. Introduction to Commutative Algebra. Addison Wesley Publishing Company; 1969.

[3] Watase Y. Zariski Topology. Formalized Mathematics. 2018;26(**4**). To be appeared.

Mizar article information

Works in Progress

TOPZARI2 Localization of Commutative Rings
by Yasushige Watase

Summary: This work is a continuation of the previous formalization of Zariski Topology [3]. The aim at this work is for futher formalization of the area of commutative algebra and algebraic geometry. The article consists mainly of a construction of ring of

fraction. It is the generalized formalization of field of fraction of a integral domain [1]. Firstly it was formalized that a process of construction of a ring of fraction with respect to a multiplicative closed set. Secondly some properties and some interpretation of a ring of fraction in Zariski topolgical space.

Listing 10. TOPZARI2 - abstract

```
environ
 vocabularies ARYTM_3, FUNCT_1, RELAT_1, RLVECT_1, VECTSP_1, CARD_3, ALGSTR_0,
      XBOOLE_0, TARSKI, FUNCT_2, STRUCT_0, SUBSET_1, SUPINF_2, NAT_1, MESFUNC1,
      GROUP_1, ARYTM_1, INT_2, QUOFIELD, MSSUBFAM, BINOP_1, LATTICES, NUMBERS,
      IDEAL_1, C0SP1, EQREL_1, ZFMISC_1, FUNCSDOM, WAYBEL20, CARD_FIL, RING_2,
      YELLOW_1, PARTFUN1, RELAT_2, MCART_1, CAT_1, TOPZARI2;

 notations
      TARSKI, XBOOLE_0, ZFMISC_1, BINOP_1, SUBSET_1, XTUPLE_0, SETFAM_1,
      DOMAIN_1, RELAT_1, RELAT_2, ORDINAL1, MEMBERED, WELLORD2, FUNCT_1,
      FUNCT_2, PARTFUN1, CARD_3, ARYTM_3, FUNCOP_1, PRALG_1,
      PBOOLE, STRUCT_0, ALGSTR_0, PRE_TOPC, TSEP_1, RLVECT_1,ORDERS_1,RINGCAT1,
      ORDERS_2, YELLOW_1, GROUP_1, VECTSP_1, IDEAL_1, RING_1,RING_2, RELSET_1,
      BINOM, GROUP_6, GCD_1, TEX_1, VECTSP_2, COMPLFLD, VECTSP10, QUOFIELD,
      C0SP1, FINSEQ_1, EQREL_1, POLYNOM3;

 constructors
      ARYTM_3, FUNCOP_1, PRALG_1, STRUCT_0, ALGSTR_0,
      PRE_TOPC, TSEP_1, TDLAT_3, TEX_4, RLVECT_1,ORDERS_1, ORDERS_2, RINGCAT1,
      YELLOW_1, GROUP_1, VECTSP_1, IDEAL_1, REALALG1, RING_1,RING_2, C0SP1,
      BINOM, RELSET_1, GCD_1, QUOFIELD;

 registrations XBOOLE_0, ORDINAL1, RELSET_1, XREAL_0, NAT_1, INT_1, MEMBERED,
      FINSEQ_1, STRUCT_0, VECTSP_1, ALGSTR_1, SUBSET_1, RELAT_1, FUNCT_2,
      ALGSTR_0, RLVECT_1, QUOFIELD, RINGCAT1, RING_1, C0SP1, FUNCT_1, PARTFUN1,
      RING_2, NAT_2, ENDALG, PBOOLE, MSSUBFAM, EQREL_1;

 requirements NUMERALS, SUBSET, BOOLE, REAL, ARITHM;

 definitions RELAT_1, FUNCT_2, VECTSP_1, GROUP_1, GROUP_6, IDEAL_1, INT_3,
      CARD_3, RING_2, ORDERS_1, XBOOLE_0, FUNCT_1, PBOOLE, FINSET_1, TARSKI,
      RING_1, PRE_TOPC, TDLAT_3, T_0TOPSP, RINGCAT1, REALALG1;

 equalities BINOP_1, XCMPLX_0, STRUCT_0, ALGSTR_0, FINSEQ_1,
      VECTSP10, WELLORD1, SUBSET_1, XBOOLE_0, FUNCOP_1;

 expansions STRUCT_0, GROUP_1, ALGSTR_0, RLVECT_1, VECTSP_1, IDEAL_1, XBOOLE_0,
      FUNCT_1, QUOFIELD, GROUP_6, TARSKI, GCD_1, ORDERS_1, RING_1, REALALG1,
      SUBSET_1, EQREL_1, BINOP_1, PBOOLE;

 theorems ALGSTR_0, GROUP_1, VECTSP_1, RLVECT_1, IDEAL_1, SUBSET_1, FUNCT_2,
      TARSKI, C0SP1, ZFMISC_1, RING_2, RELAT_1, XTUPLE_0, EQREL_1, BINOP_1,
      MCART_1;

 schemes FUNCT_2, EQREL_1, BINOP_1;

begin :: Preliminaries

reserve R,R1 for commutative Ring;
reserve A,B for non degenerated commutative Ring;

registration
  let R;
  cluster multiplicatively-closed for non empty Subset of R;
end;

reserve S for non empty multiplicatively-closed Subset of R;
reserve T for non empty multiplicatively-closed Subset of A;
reserve x,y,z,u,v,w,p,p1,q,q1 for Element of [:[#]R,S:];
reserve s,s1 for Element of S;
```

```
definition
  let R,S;
  let x,y be Element of [:[#]R,S:];
  pred x,y Frac_Equal R,S means
:: TOPZARI2:def 1

  ex x1,y1 be Element of [#]R,x2,y2,s1 be Element of S
  st x = [x1,x2] & y = [y1,y2] & (x1 * y2 - y1 * x2) * s1 = 0.R;
end;

theorem :: TOPZARI2:1
  0.R in S implies x,y Frac_Equal R,S;

definition let R,S;
  func EqRel(R,S) -> Equivalence_Relation of [:[#]R,S:] means
:: TOPZARI2:def 2

  [p,q] in it iff p,q Frac_Equal R,S;
end;

registration
  let R,S;
  cluster EqRel(R,S) -> non empty total symmetric transitive;
end;

registration
  let R;
  cluster [#]R -> non empty multiplicatively-closed;
end;

registration
  let R,S;
  cluster [:[#]R,S:] -> non empty;
end;

theorem :: TOPZARI2:2
  Class(EqRel(R,S),x) = Class(EqRel(R,S),y) iff x,y Frac_Equal R,S;

reserve x1,y1 for Element of [#]R;
reserve x2,y2 for Element of S;

definition
  let R,S;
  let x,y be Element of [:[#]R,S:];
  func padd(x,y) -> Element of [:[#]R,S:] means
:: TOPZARI2:def 3

  ex x1,y1 be Element of [#]R,x2,y2 be Element of S
  st x = [x1,x2] & y = [y1,y2] & it = [x1*y2 + y1*x2, x2*y2];
end;

definition
  let R,S;
  let x,y be Element of [:[#]R,S:];
  func pmult(x,y) -> Element of [:[#]R,S:] means
:: TOPZARI2:def 4

  ex x1,y1 be Element of [#]R,x2,y2 be Element of S
  st x = [x1,x2] & y = [y1,y2] & it = [x1*y1, x2*y2];
end;

definition
  let R,S;
  let x, y be Element of [:[#]R,S:];
  redefine func padd(x,y);
  commutativity;
end;

theorem :: TOPZARI2:3
```

```
pmult(x,y) = pmult(y,x);

definition
  let R,S;
  let x, y be Element of [:[#]R,S:];
  redefine func pmult(x,y);
  commutativity;
end;

theorem :: TOPZARI2:4
   pmult(padd(x,y),z), padd(pmult(x,z),pmult(y,z)) Frac_Equal R,S;

theorem :: TOPZARI2:5
  x,u Frac_Equal R,S & y,v Frac_Equal R,S implies
  pmult(x,y),pmult(u,v) Frac_Equal R,S;

theorem :: TOPZARI2:6
  x,u Frac_Equal R,S & y,v Frac_Equal R,S implies
  padd(x,y),padd(u,v) Frac_Equal R,S;

theorem :: TOPZARI2:7
  x,u Frac_Equal R,S & y,v Frac_Equal R,S & z,w Frac_Equal R,S implies
  padd(x,padd(y,z)),padd(u,padd(v,w)) Frac_Equal R,S;

definition
  let R,S;
  func fr_add(R,S) -> BinOp of [:[#]R,S:] means
:: TOPZARI2:def 5

  for x,y holds it.(x,y) = padd(x,y);
end;

definition
  let R,S;
  func fr_mult(R,S) -> BinOp of [:[#]R,S:] means
:: TOPZARI2:def 6

  for v,w being Element of [:[#]R,S:] holds it.(v,w) = pmult(v,w);
end;


definition
  let R,S,x,y;
  func x+y -> Element of [:[#]R,S:] equals
:: TOPZARI2:def 7
  (fr_add(R,S)).(x,y);
end;

definition
  let R,S,x,y;
  func x*y -> Element of [:[#]R,S:] equals
:: TOPZARI2:def 8
  (fr_mult(R,S)).(x,y);
end;

definition
  let R,S;
  func 0.(R,S) -> Element of [:[#]R,S:] equals
:: TOPZARI2:def 9

  [0.R ,1.R ];
end;

definition
  let R,S;
  func 1.(R,S) -> Element of [:[#]R,S:] equals
:: TOPZARI2:def 10

  [1.R ,1.R ];
```

```
end;

theorem :: TOPZARI2:8
   x + 0.(R,S) = x;

theorem :: TOPZARI2:9
   x * 1.(R,S) = x;

theorem :: TOPZARI2:10
   x * y = y * x;

theorem :: TOPZARI2:11
   x + y = y + x;

theorem :: TOPZARI2:12
  x+(y+z) = (x+y)+z;

theorem :: TOPZARI2:13
  x*(y*z) = (x*y)*z;

theorem :: TOPZARI2:14
  x,u Frac_Equal R,S & y,v Frac_Equal R,S implies
  x*y,u*v Frac_Equal R,S;

theorem :: TOPZARI2:15
  x,u Frac_Equal R,S & y,v Frac_Equal R,S implies
  x+y,u+v Frac_Equal R,S;

theorem :: TOPZARI2:16
   (x+y)*z, x*z + y*z Frac_Equal R,S;

begin :: Localization of Ring by S

definition
  let R, S;
::$N Locarization
  func Fr_Ring(R,S) -> strict doubleLoopStr means
:: TOPZARI2:def 11

  the carrier of it = Class EqRel(R,S) & 1.it = Class(EqRel(R,S),1.(R,S)) &
  0.it = Class(EqRel(R,S),0.(R,S)) &
  (for x, y being Element of it ex a, b being Element of [:[#]R,S:] st
   x = Class(EqRel(R,S),a) & y = Class(EqRel(R,S),b) &
  (the addF of it).(x,y) = Class(EqRel(R,S),a+b) ) &
  for x, y being Element of it ex a, b being Element of [:[#]R,S:] st
  x = Class(EqRel(R,S),a) & y = Class(EqRel(R,S),b) &
  (the multF of it).(x,y) = Class(EqRel(R,S),a*b );
end;

notation
  let R ,S;
  synonym S~R for Fr_Ring(R,S);
end;

reserve a, b, c for Element of [:[#]R,S:];
reserve r1, r2 for Element of [#]R;
reserve s1, s2 for Element of S;

registration
  let R ,S;
  cluster S~R -> non empty;
end;

theorem :: TOPZARI2:17
   0.R in S iff S~R is degenerated;

reserve x, y, z for Element of S~R;

theorem :: TOPZARI2:18
```

```
  ex a being Element of [:[#]R,S:] st x = Class(EqRel(R,S),a);

theorem :: TOPZARI2:19
  Class(EqRel(R,S),a) is Element of S~R;

theorem :: TOPZARI2:20
  Class(EqRel(R,S),[r1,s1]) is Element of S~R;

theorem :: TOPZARI2:21
  Class(EqRel(R,S),a) = Class(EqRel(R,S),[a`1,a`2]);

theorem :: TOPZARI2:22
  a`1 = a`2 implies Class(EqRel(R,S),a) = 1.(S~R);

theorem :: TOPZARI2:23
  x = Class(EqRel(R,S),a) & y = Class(EqRel(R,S),b) implies x+y =
  Class(EqRel(R,S),a+b);

theorem :: TOPZARI2:24
  x = Class(EqRel(R,S),a) & y = Class(EqRel(R,S),b) implies x*y =
  Class(EqRel(R,S),a*b);

theorem :: TOPZARI2:25
   Class(EqRel(R,S),a*b) = Class(EqRel(R,S),b*a);

theorem :: TOPZARI2:26
    x*y = y*x;

theorem :: TOPZARI2:27
   S~R is Ring;

registration
  let R,S;
  cluster S~R -> Abelian add-associative right_zeroed
    right_complementable associative well-unital distributive;
end;

registration
  let R,S;
  cluster S~R -> commutative;
end;

definition
let R,S;
func canHom(R,S) -> Function of R, S~R means
:: TOPZARI2:def 12

  for x being Element of R holds it.x = Class(EqRel(R,S),[x,1.R]);
end;

theorem :: TOPZARI2:28
   canHom(R,S) is RingHomomorphism;

reserve s for Element of S;
reserve g for Function of R, R1;

definition
let R ;
func Unit_Set(R) -> Subset of [#]R equals
:: TOPZARI2:def 13
{a where a is Element of R: a is Unit of R };
end;

registration
let R;
cluster Unit_Set(R) -> non empty;
end;

theorem :: TOPZARI2:29
```

a‘1 in S **implies** Class(EqRel(R,S),a) **is** Unit **of** S~R;