Using MIZAR to prove generic algebraic algorithms correct

Christoph Schwarzweller
Wilhelm-Schickard-Institut für Informatik
Sand13, D-72076 Tübingen
schwarzw@informatik.uni-tuebingen.de

22. April 1997

Zusammenfassung

MIZAR is a system that allows for proving mathematical theorems in a rather natural style. It provides a special input language for mathematical knowledge.
We use MIZAR to prove the mathematical correctness of algebraic algorithms, in particular the algorithms of Brown and Henrici for addition and multiplication in fraction fields.
We also include an introduction into the MIZAR language so that MIZAR beginners should be able to do the first steps in using MIZAR.

Inhaltsverzeichnis

1 Introduction

Modern algebraic algorithms should be formulated generically in terms of abstract mathematical structures ([Sc96]). Therfore proving them correct requires detailed knowledge about these domains which is very hard to formulate for theorem provers in general.
On the other hand there exist systems like AXIOM in which such algebraic domains can be expressed, but they do not include proof assistance for mathematical theorems.
As a consequence there is no system that could serve as a prove tool in the context of algebraic algorithms.

MIZAR([Ru92]) is a system that — originally intended for support in writing mathematical papers — admits to express mathematical knowledge in a very natural style using a special input language. It also includes a large library of MIZAR articles and a checker that verifies articles written in the MIZAR language.

We want to use MIZAR to fill the gap just mentioned, namely we want to prove the correctness of two algebraic algorithms of Brown and Henrici. Therefore we wrote an extensive MIZAR article: First we need some basics about *divisibility in integral domains* which are not yet included in the MIZAR library, before we can introduce the concept *greatest common divisor*. After that we want to show that the algorithms of Henrici and Brown ([He56]) concerning *addition and multiplication in fraction fields* are correct by formulating and proving correctness conditions in MIZAR.

2 The Algorithms of Brown and Henrici

In this section we present the algorithms of Brown and Henrici for addition and multiplication in fraction fields. For that we use the programming language SUCHTHAT ([LS96]) which allows for generic computation in the field of computer algebra.

Let `I` be an integral domain, and let `Q` be the fraction field of `I`. Based on algorithms for the arithmetic operations in `I` one obtains algorithms for the arithmetic operations in `Q` ([Co??]). To be able to choose a unique representative from each equivalence class of `Q`, we assume that `I` is a *gcdDomain*

that is an integral domain in which for any two elements exists a greatest common divisor.

```
let I be gcdDomain;
let Q be QuotientField of I;
```

We also assume that we have algorithms that constuct a fraction out of elements of I and that decompose a fraction into numerator and denominator.

```
Algorithm: r1 := num(r)
Input: r ∈ Q.
Output: r ∈ I. ||

Algorithm: r2 := denom(r)
Input: r ∈ Q.
Output: r ∈ I such that r2 ≠ 0. ||

Algorithm: t := denom(r,s)
Input: r,s ∈ I such that gcd(r,s) = 1.
Output: t ∈ Q such that t = r/s, t is normalized. ||
```

The algorithms accept *normalized frations* as input that is fractions `t` with `gcd(num(t),denom(t)) = 1` and `denom(t)` is in normalform (see end of section four). To get a normalized fraction as ouput one can use the usual addition resp. multiplication followed by computing the greatest common divisor of the result. However the algorithms of Brown and Henrici we present here in general are much more efficient ([Co??]).

```
Algorithm: t <- operator +(r,s)
Input: r,s ∈ Q, r,s normalized.
Output: t ∈ Q such that t = r+s, t normalized.
(1) [r=0 or s=0?]
    if r = 0 then {t := s;return};
    if s = 0 then {t := r;return}.
(2) [obtain numerators and denominators.]
    r1 := num(r); r2 := denom(r);
    s1 := num(s); s2 := denom(s).
```

```
(3) [compute gcd(r2,s2).]
    d := gcd(r2,s2).
(4) [gcd(r2,s2) = 1.]
    if d = 1 then { t:= fract((r1*s2)+(r2*s1),r2*s2);return}.
(5) [compute t1' and t2'.]
    r2' := r2/d; s2' := s2/d;
    t1' := r1*s2' + s1*r2'.
(6) [ t1' = 0.]
    if t1' = 0 then { t := 0;return}.
(7) [ general case.]
    e := gcd(t1',d);
    if e = 1 then { t := fract(t1',t2');return};
    t := fract(t1'/e,t2'/e). ||
```

```
Algorithm: t <- operator *(r,s)
Input: r,s ∈ Q, r,s normalized.
Output: t ∈ Q such that t = r*s, t normalized.
(1) [r=0 or s=0?]
    if r = 0 or s = 0 then {t := 0;return};
(2) [obtain numerators and denominators.]
    r1 := num(r); r2 := denom(r);
    s1 := num(s); s2 := denom(s).
(3) [r and s ∈ I.]
    if r2 = 1 and s2 = 1 then {t := fract(r1*s1,1);return}.
(4) [ r or s ∈ I.]
    if r2 = 1 then {t := fract((r1*s1)/gcd(r1,s2),s2);return};
    if s2 = 1 then {t := fract((r1*s1)/gcd(s1,r2),r2);return}.
(5) [general case.]
    d := gcd(r1,s2); e := gcd(s1,r2);
    t := fract((r1/d)*(s1/e),(r2/e)*(s2/d)). ||
```

In the rest of the paper we show that the two algorithms of Brown and Henrici are correct: For every gcdDomain `I` and fractions `r,s` $\in$ `I` such that the input specification is fulfilled, the output `t` of the algorithms fulfills the output specification. All proofs are written in the MIZAR language and have been accepted by the MIZAR checker.

Section three gives an introduction to the MIZAR language. The examples are some basics of divisibility in integral domains we need later on. Section four introduces *amplesets* that is sets of representatives. We will use

amplesets to define the greatest common divisor as a unique function. In section five we introduce *gcdDomains* and prove properties about greatest common divsors that are crucial for the theorems of Brown and Henrici which we establish in section six. Section seven finally contains the correctness proofs, to be more precise we define MIZAR functions that mirror the input/output behaviour of the algorithms and prove that the values of these functions have the desired properties.

3 The MIZAR Language

In this section we give a short introduction to the MIZAR language ([Tr93]). Thereby we illustrate the concepts with parts of our MIZAR article about *gcdDomains*, i.e. integral domains with greatest common divisor.
We use STWEB ([Br89]), a literate programming tool which allows the extraction of the MIZAR files out of our LaTeX document.

Each MIZAR article consists of two main parts: the *environment* part and the *text-proper* part.

"GCD.MIZ" 6a ≡
```
environ
⟨env 6b⟩
begin
⟨txtpr 7b⟩
```
◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

The *environment* consists of several *directives* indicating which items of the MIZAR library can be referenced in text-proper.

⟨env 6b⟩ ≡
```
vocabulary ⟨vocabulary 7a⟩
notation ⟨notation 40a⟩
constructors ⟨constructors 40b⟩
definitions ⟨definitions 40c⟩
theorems ⟨theorems 40d⟩
clusters ⟨clusters 40e⟩
schemes ⟨schemes 40f⟩
```
◇

Macro referenced in scrap 6a.

After each keyword follows a list of MIZAR article names e.g.

⟨vocabulary 7a⟩ ≡
```
BOOLE,VECTSP_1,VECTSP_2,REAL_1,LINALG_1,SFAMILY,GCD;
```
◇

Macro referenced in scrap 6b.

The directive *vocabulary* adds symbols of the named files to the article's internal lexicon. If there are new symbols (introduced in text-proper) these have to be put in an extra vocabulary file like `GCD.VOC` in this case.
The directives *notations* and *constructors* request the conceptual framework of the article. In MIZAR it is possible to introduce synonyms if another name is more appropriate in the actual context. So *constructors* give the concepts to be used in text-proper, and *notations* give the synonyms to be used for these concepts.
definitions and *theorems* indicate which definitions and theorems may be cited in the article. *schemes* describe second order theorems that can be referenced in text-proper.

The *text-proper* includes the new mathematical knowledge that is new definitions and theorems as well as the proofs for those. We present the main features of the MIZAR language for writing mathematical texts.

Reservations delcare the (mathematical) type of identifiers:

⟨txtpr 7b⟩ ≡
```
reserve X,Y,Z for set;
reserve I for domRing;
reserve a,b,c,d for Element of the carrier of I;
```
◇

Macro referenced in scrap 6a.

After this reservation `I` stands for an integral domain and `a,b,c` and `d` are Elements of `I`. Using this identifiers we can define new concepts of integral domains e.g. divisibility:

"GCD.MIZ" 7c ≡
```
definition
let I be domRing;
let x,y be Element of the carrier of I;
  pred x divides y means :Def1:
  ex z being Element of the carrier of I st y = xz;
```

```
antonym x not_divides y;
end;

definition
let I be domRing;
let x be Element of the carrier of I;
  pred x is_unit means :Def2:
  x divides (1.I);
         :: (1.I) is the multiplicative identity of I
antonym x is_no_unit;
end;

definition
let I be domRing;
let x,y be Element of the carrier of I;
  pred x is_associated_to y means :Def3:
  x divides y & y divides x;
antonym x is_not_associated_to y;
end;
```
◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

In MIZAR it is possible to define predicates (like we did), functions and modes (like domRing). We stress again that new symbols like `divides` have to be explicitly introduced in an extra file called `GCD.VOC`.

Now using these new definitions mathematical theorems can be formulated in a very natural way:

"GCD.MIZ" 8 ≡

```
theorem
L1: for a,b,c being Element of the carrier of I holds
    (a divides a) &
    ((a divides b & b divides c) implies (a divides c))
proof
⟨proof L1 9a⟩
end;

⟨more div 40g⟩
```
◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

L1 is a label (or marker) that makes it possible to reference this theorem in later parts of text-proper. Note that L1 only is an internal label, so theorems without a label also can be cited once the article is included in the MIZAR library.
The proof of this theorem is rather easy, because the MIZAR language allows to state a proof close to textbook style:

⟨proof L1 9a⟩ ≡

```
let A,B,C be Element of the carrier of I;
M1: now assume H1: A divides B & B divides C;
consider D being Element of the carrier of I such that
H2: AD = B by H1,Def1;
consider E being Element of the carrier of I such that
H3: BE = C by H1,Def1;
H4: A(DE) = (AD)E  by VECTSP_1:def 16
         .= BE     by H2
         .= C      by H3;
thus (A divides B & B divides C) implies A divides C
     by H4,Def1;
end;  :: M1
M2: A(1.I) = A by VECTSP_2:1;
M3: A divides A by M2,Def1;
thus thesis by M1,M3;
```

◇

Macro referenced in scrap 8.

As another example we prove the well known

"GCD.MIZ" 9b ≡

```
theorem
L11: for a,b being Element of the carrier of I holds
     (a is_associated_to b iff (ex c st (c is_unit & ac = b)))
proof
```

⟨proof L11 10a⟩

```
end;
```

◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

In MIZAR, an *if and only if* has to be proved by two implications, so we get

⟨proof L11 10a⟩ ≡

```
K1: for a,b being Element of the carrier of I holds
    (a is_associated_to b) implies
    (ex c being Element of the carrier of I
                    st (c is_unit & ac = b))
proof
⟨proof L11a 11a⟩
end;

K2: for a,b being Element of the carrier of I holds
    (ex c being Element of the carrier of I
                    st (c is_unit & ac = b))
    implies (a is_associated_to b)
proof
⟨proofL11b 10b⟩
end;

thus thesis by K1,K2;
```

◇

Macro referenced in scrap 9b.

The proof of K2 is straightforward, it looks like

⟨proofL11b 10b⟩ ≡

```
let A,B be Element of the carrier of I;
M1: (ex c st (c is_unit & Ac = B)) implies
    A is_associated_to B
proof
M2: now
assume H1: (ex c st (c is_unit & Ac = B));
consider C being Element of the carrier of I such that
H2: C is_unit & AC = B by H1;
H3: C divides (1.I) by H2,Def2;
H4: ex d st Cd = (1.I) by H3,Def1;
consider D being Element of the carrier of I such that
H5: CD = (1.I) by H4;
H6: A = A(1.I) by VECTSP_2:1
     .= A(CD)  by H5
     .= (AC)D  by VECTSP_1:def 16
     .= BD     by H2;
H7: B divides A by H6,Def1;
H8: A divides B by H2,Def1;
H9: A is_associated_to B by H7,H8,Def3;
```

```
thus thesis by H9;
end;  :: M2
thus thesis by M2;
end;  :: M1
thus thesis by M1;
◇
```

Macro referenced in scrap 10a.

The proof of K1 starts as usual by applying the definitions:

⟨proof L11a 11a⟩ ≡

```
let A,B be Element of the carrier of I;
assume H0: A is_associated_to B;
H2: A divides B & B divides A by H0,Def3;
H3: ex c st B = Ac by H2,Def1;
H4: ex d st A = Bd by H2,Def1;
consider C being Element of the carrier of I such that
H5: B = AC by H3;
consider D being Element of the carrier of I such that
H6: A = BD by H4;

⟨cases 11b⟩
◇
```

Macro referenced in scrap 10a.

But then as indicated by cases we have to distinguish A = (0.I) and A <> (0.I). To do so the MIZAR language has a special construct:

⟨cases 11b⟩ ≡

```
M: now per cases;

case A: A <> (0.I);
⟨caseA 12a⟩

case B: A = (0.I);
⟨caseB 12b⟩

end;   ::cases
thus thesis by M;
◇
```

Macro referenced in scrap 11a.

In this case it is obvious for MIZAR, that A and B together cover all possible cases, but it may happen, that this must be proved before and referenced at level M.
Now the rest of the proof is easy:

⟨caseA 12a⟩ ≡
```
H7: A = BD     by H6
     .= (AC)D  by H5
     .= A(CD)  by VECTSP_1:def 16;
H8: CD = (1.I) by H7,L10,A;
H9: C divides (1.I) by H8,Def1;
H10: C is_unit by H9,Def2;
thus (ex c being Element of the carrier of I st
             (c is_unit & B = Ac)) by H10,H5;
◇
```
Macro referenced in scrap 11b.

⟨caseB 12b⟩ ≡
```
H1: B = AC     by H5
     .= (0.I)  by B,VECTSP_2:26;
H2: B = (0.I)       by H1
     .= (0.I)(1.I)  by VECTSP_2:1
     .= A(1.I)      by B;
H3: (1.I) is_unit
    proof
    M1: (1.I)(1.I) = (1.I) by VECTSP_2:1;
    M2: (1.I) divides (1.I) by M1,Def1;
    thus thesis by M2,Def2;
    end;
thus (ex c being Element of the carrier of I st
              (c is_unit & B = Ac)) by H2,H3;
◇
```
Macro referenced in scrap 11b.

Other properties about divisibility we need to prove to establish our main results are included in the appendix.

4 AmpleSets

An *ampleset* A for an integral domain I is a set $A \subseteq I$ which contains exactly one element from each class of associates ([Sc96] or [Co??]). The existence

of amplesets is due to the *Axiom of Choice*, which is included in MIZAR as a theorem.
We need amplesets to define the *greatest common divisor* as a unique function in the usual sense and not as a subset of the integral domain.

We start by defining classes of associates.
Note that in MIZAR defining a function requires a *correctness proof*. To see an example of such a proof refer to section five where we prove the correctness of the gcd-function.

"GCD.MIZ" 13a ≡
```
definition
let I be domRing;
let a be Element of the carrier of I;
  func Class a
      -> non empty Subset of the carrier of I means :Defh1:
  (for b being Element of the carrier of I holds
   b ∈ it iff b is_associated_to a);
```
⟨correctness Class 49⟩

```
definition
let I be domRing;
  func Classes I
        -> Subset-Family of the carrier of I means :Defh2:
  (for A being Subset of the carrier of I holds
   A ∈ it iff
  (ex a being Element of the carrier of I st A = Class a));
```
⟨correctness Classes 51a⟩
◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

Here are also some basic results concerning `Class` and `Classes` we need during the existence proof of amplesets. The proofs are included in the appendix.

"GCD.MIZ" 13b ≡
```
theorem
CL1: for a,b being Element of the carrier of I holds
     Class a ∩ Class b <> ∅ implies Class a = Class b
```
⟨proof CL1 51b⟩

```
theorem
CL2: for I being domRing holds Classes I is non empty
⟨proof CL2 52⟩

theorem
CL3: for X being Subset of the carrier of I holds
     X ∈ Classes I implies X is non empty
⟨proof CL3 53a⟩
◇
```

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

Now we are ready to define — and prove the existence of — amplesets for integral domains.

"GCD.MIZ" 14a ≡

```
definition
let I be domRing;
mode Am of I
   -> non empty Subset of the carrier of I means :Def8a:
 (for a being Element of the carrier of I
  ex z being Element of it
  st z is_associated_to a) &
 (for x,y being Element of it holds x <> y implies
  x is_not_associated_to y);
⟨existence Am 14b⟩
◇
```

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

To prove the existence we first have to establish three properties of `Classes I` which are presupposition for the application of the axiom of choice.

⟨existence Am 14b⟩ ≡

```
existence
proof
K: now let I be domRing;
set M = Classes I;
K1: M is non empty by CL2;
reconsider M as non empty set by K1;
K2: for X st X ∈ M holds X <> ∅
    ⟨proof K2 53b⟩
K3: for X,Y st X ∈ M & Y ∈ M & X <> Y holds X ∩ Y = ∅
```

```
    ⟨proof K3 53c⟩

  K4: ex Choice being set st
      for X st X ∈ M ex x being Any
      st Choice ∩ X = { x } by K2,K3,WELLORD2:27;
  consider AmpS' being set such that K5:
      for X st X ∈ M ex x being Any
      st AmpS' ∩ X = {x} by K4;
  K5a: AmpS' is non empty
       ⟨proof K5a 54a⟩
  reconsider AmpS' as non empty set by K5a;
  ⟨existence Am 2 15a⟩
 ◇
```

Macro referenced in scrap 14a.

Now `Amp'` is a set that contains exactly one element out of each class of asscoiates, but we need a set in which there are only these elements and no other ones. So we define:

⟨existence Am 2 15a⟩ ≡

```
  set AmpS =
   { x where x is Element of AmpS':
     ex X being non empty Subset of the carrier of I
     st X ∈ M & AmpS' ∩ X = { x } };
  ⟨existence Am 3 15b⟩
 ◇
```

Macro referenced in scrap 14b.

Now we can prove that `Amp` is a subset of `the carrier of I`.

⟨existence Am 3 15b⟩ ≡

```
  K6a: for X being Element of M holds
       ex z being Element of AmpS st AmpS ∩ X = {z}
       ⟨proof K6a 54b⟩
  K6: AmpS is non empty Subset of the carrier of I
      ⟨proof K6 55⟩
  reconsider AmpS as non empty Subset of the carrier of I by K6;
  ⟨existence Am 4 16a⟩
 ◇
```

Macro referenced in scrap 15a.

It remains to prove that `Amp` indeed has the desired properties, that is

⟨existence Am 4 16a⟩ ≡

```
K7: for a being Element of the carrier of I
     ex z being Element of AmpS
     st z is_associated_to a
     ⟨proof K7 16b⟩
K8: for x,y being Element of AmpS holds
     x <> y implies x is_not_associated_to y
     ⟨proof K8 17a⟩
thus ex s being non empty Subset of the carrier of I st
   (for a being Element of the carrier of I
    ex z being Element of s
    st z is_associated_to a) &
   (for x,y being Element of s holds x <> y implies
     x is_not_associated_to y)  by K7,K8;
end;  :: K
thus thesis by K;
end;  :: existence
end;
```

◇

Macro referenced in scrap 15b.

Of course both properties K7 and K8 hold because Amp is defined via the axiom of choice. To be more precise: The first property follows from the fact that for each X ∈ Classes I the intersection of X and Amp is non empty. The second property follows because the cardinality of this intersection is one.

⟨proof K7 16b⟩ ≡

```
proof
let a be Element of the carrier of I;
H0: Class a ∈ M by Defh2;
reconsider N = Class a as Element of M by H0;
consider z being Element of AmpS such that
H1: AmpS ∩ N = {z} by K6a;
H1a: z ∈ {z} by ENUMSET1:4;
H1b: z ∈ AmpS ∩ Class a by H1a,H1;
H2: z ∈ Class a by H1b,BOOLE:def 3;
H3: z is_associated_to a by H2,Defh1;
thus thesis by H3;
end;
```

◇

Macro referenced in scrap 16a.

⟨proof K8 17a⟩ ≡

```
proof
let x,y be Element of AmpS;
assume H0: x <> y;
assume H1: x is_associated_to y;
H2: x is_associated_to x by L2;
H3: x ∈ Class x by H2,Defh1;
H4: y is_associated_to x by H1,L2;
H5: y ∈ Class x by H4,Defh1;
H6: x ∈ AmpS ∩ Class x by H3,BOOLE:def 3;
H7: y ∈ AmpS ∩ Class x by H5,BOOLE:def 3;
H8: Class x ∈ M by Defh2;
consider z being Element of AmpS such that
H9: AmpS ∩ Class x = {z} by H8,K6a;
H10: x ∈ {z} by H6,H9;
H11: x = z by H10,ENUMSET1:3;
H12: y ∈ {z} by H7,H9;
H13: y = z by H12,ENUMSET1:3;
H14: x = y by H11,H13;
H15: contradiction by H0,H14;
thus thesis by H15;
end;
◇
```

Macro referenced in scrap 16a.

So we established the existence of amplesets for integral domains. As a matter of convienience we require that `(1.I)` is always an element of our amplesets whereas `(0.I)` is always an element of the set because the class of associates of `(0.I)` contains only one element.

"GCD.MIZ" 17b ≡

```
definition
let I be domRing;
mode AmpleSet of I
   -> non empty Subset of the carrier of I means :Def8:
  it is Am of I & (1.I) ∈ it;
⟨existence AmpleSet 56⟩

reserve Amp for AmpleSet of I;
◇
```

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

To show the existence one takes an ampleset A of I and exchanges the element associated to (1.I) for (1.I):

⟨definition of A' 18a⟩ ≡

```
let A be Am of I;
consider x being Element of A such that
H1: x is_associated_to (1.I) by Def8a;
set A' = { z where z is Element of A : z <> x } U {(1.I)};
```

◇

Macro referenced in scrap 56.

The rest of the proof consists of establishing the desired properties of A' and is very close to the proof just given. It is included in the appendix.

As we will see using an ampleset to define the gcd-function only suffices to show that for the output t of our algorithms holds gcd(num(t),denom(t)) = 1. To establish that t is normalized we need an additional property of our ampleset:

"GCD.MIZ" 18b ≡

```
definition
let I be domRing;
let Amp be AmpleSet of I;
  pred Amp is_multiplicative means :Def25:
  for x,y being Element of Amp holds xy ∈ Amp;
end;
```

◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

Here is a summary of some other properties of an ampleset we need later on:

"GCD.MIZ" 18c ≡

```
theorem
AMP: for Amp being AmpleSet of I holds
     ((1.I) ∈ Amp) &
     (for a being Element of the carrier of I
      ex z being Element of Amp
      st z is_associated_to a) &
```

```
      (for x,y being Element of Amp holds x <> y
       implies x is_not_associated_to y)
```
⟨proof AMP 59a⟩

```
theorem
AMP0: for Amp being AmpleSet of I holds
      (0.I) is Element of Amp
```
⟨proof AMP0 61a⟩

```
theorem
AMP1: for x,y being Element of Amp holds
      x is_associated_to y implies x = y
```
⟨proof AMP1 61b⟩

```
theorem
AMP5: for Amp being AmpleSet of I holds
      Amp is_multiplicative implies
       (for x,y being Element of Amp holds
       (y divides x & y <> (0.I)) implies x/y ∈ Amp)
```
⟨proof AMP5 59b⟩

◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

We end this section with the definition of a *normalform* modulo an ampleset: The normalform of an element x is the element of the ampleset that is associated to x.

"GCD.MIZ" 19 ≡

```
definition
let I be domRing;
let Amp be AmpleSet of I;
let x be Element of the carrier of I;
 func NF(x,Amp) -> Element of the carrier of I means :Def20:
 it ∈ Amp & it is_associated_to x;
```
⟨correctness NF 62a⟩

◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

As an easy consequence we get

"GCD.MIZ" 20a ≡

```
theorem
NF1: for Amp being AmpleSet of I holds
     NF((0.I),Amp) = (0.I) & NF((1.I),Amp) = (1.I)
⟨proof NF1 62b⟩

theorem
NF3: for Amp being AmpleSet of I
     for a being Element of the carrier of I holds
     a ∈ Amp iff a = NF(a,Amp)
⟨proof NF3 63a⟩
◇
```

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

We will use this normalform to define normalized fractions in section seven.

5 GCD-Domains

In this section we introduce *gcdDomians*, e.g. integral domains with greatest common divisors. Therefore we first define an attribute `gcd-like` for integral domains and then a `gcdDomain` to be an integral domain for which this attribute is fulfilled. Note the *cluster definition*: In MIZAR one have to prove that the objects one defines do exist.

This section also includes five theorems about the gcd-function which we need to prove the theorems of Brown and Henrici in the next section.

"GCD.MIZ" 20b ≡

```
definition
let I be domRing;
  attr I is gcd-like means :Def7:
  (for x,y being Element of the carrier of I
  ex z being Element of the carrier of I st
        z divides x & z divides y &
        (for zz being Element of the carrier of I
         st (zz divides x & zz divides y)
         holds (zz divides z)));
end;
```

```
definition
  cluster gcd-like domRing;
⟨existence gcdDomain 63b⟩
:: proved by showing that a Field is a gcd-like domRing
end;

definition
  mode gcdDomain is gcd-like domRing;
end;

reserve I for gcdDomain;
```
◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

Now that we have defined gcdDomains, we want to introduce a *gcd-function* in these structures. The problem is that the greatest common divsor is unique only up to associates. So we use the concept of ample sets to get a function in its usual sense. For that the `AmpleSet` has become an argument of the gcd-function because changing the ampleset may change the value the gcd-function returns.
Note that although `gcd` returns an element of `AmpleSet` the type of this element is `Element of the carrier of I`.

"GCD.MIZ" 21 ≡
```
definition
let I be gcdDomain;
let Amp be AmpleSet of I;
let x,y be Element of the carrier of I;
func gcd(x,y,Amp) -> Element of the carrier of I means :Def4:
        it ∈ Amp &
        it divides x & it divides y &
       (for z being Element of the carrier of I
        st (z divides x & z divides y)
        holds (z divides it));
⟨existence gcd 22a⟩
⟨uniqueness gcd 23a⟩
end;

⟨more gcd 65⟩
```
◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

The correctness proof that belongs to a function definition proceeds as follows: One have to prove *existence* e.g. that for every integral domain there is a function with the required properties and *uniqueness* e.g. that every two elements both fulfilling the properties of the definition are identical.

To show the existence we take an element u of I with the required gcd-properties. u exists because I is a gcdDomain.

⟨existence gcd 22a⟩ ≡

```
existence
 proof
 consider u being Element of the carrier of I such that
 H1: u divides x & u divides y &
    (for zz being Element of the carrier of I
              st (zz divides x & zz divides y)
              holds (zz divides u)) by Def7;
 ⟨existence gcd 2 22b⟩
◇
```

Macro referenced in scrap 21.

All that remains is to prove that the element z of Amp that is associated to u also has these properties:

⟨existence gcd 2 22b⟩ ≡

```
consider z being Element of Amp such that
H2: z is_associated_to u by AMP;
H3: z divides u by H2,Def3;
H4: z divides x & z divides y by H3,H1,L1;
H6: for zz being Element of the carrier of I
    st (zz divides x & zz divides y) holds (zz divides z)
    proof
    let zz be Element of the carrier of I;
    assume M1: zz divides x & zz divides y;
    M2: zz divides u by M1,H1;
    M3: u divides z by H2,Def3;
    M4: zz divides z by M2,M3,L1;
    thus thesis by M4;
    end;
thus thesis by H4,H6;
end;
◇
```

Macro referenced in scrap 22a.

Uniqueness follows from the fact that two elments of Amp that are associated to each other are indeed identical.

⟨uniqueness gcd 23a⟩ ≡

```
uniqueness
 proof
 K1: now
 let z1 be Element of the carrier of I such that
 H1: z1 ∈ Amp &
     z1 divides x & z1 divides y &
     (for z being Element of the carrier of I
      st (z divides x & z divides y)
      holds (z divides z1));
 let z2 be Element of the carrier of I such that
 H2: z2 ∈ Amp &
     z2 divides x & z2 divides y &
     (for z being Element of the carrier of I
      st (z divides x & z divides y)
      holds (z divides z2));
 H3: z1 is_associated_to z2
     proof
     M4: z2 divides z1 & z1 divides z2 by H1,H2;
     thus thesis by M4,Def3;
     end;
 thus z1 = z2 by H1,H2,H3,AMP;
 end;   :: now
 thus thesis by K1;
 end;
```

◇

Macro referenced in scrap 21.

The next step is to establish five basic properties about the gcd-function. We will need them later to prove the theorems of Henrici and Brown. Note that these properties hold for arbitrary amplesets.

"GCD.MIZ" 23b ≡

```
theorem
T0: for Amp being AmpleSet of I
    for a,b,c being Element of the carrier of I holds
    gcd(gcd(a,b,Amp),c,Amp) = gcd(a,gcd(b,c,Amp),Amp)
```

⟨proof T0 69a⟩

```
theorem
T1: for Amp being AmpleSet of I
    for a,b,c being Element of the carrier of I holds
    gcd(ac,bc,Amp) is_associated_to cgcd(a,b,Amp)
⟨proof T1 69b⟩

theorem
T2: for Amp being AmpleSet of I
    for a,b,c being Element of the carrier of I holds
    gcd(a,b,Amp) = (1.I) implies
    gcd(a,bc,Amp) = gcd(a,c,Amp)
⟨proof T2 25a⟩

theorem
T3: for Amp being AmpleSet of I
    for a,b,c being Element of the carrier of I holds
    (c = gcd(a,b,Amp) & c <> (0.I)) implies
    gcd(a/c,b/c,Amp) = (1.I)
⟨proof T3 72⟩

theorem
T4: for Amp being AmpleSet of I
    for a,b,c being Element of the carrier of I holds
    gcd(a+(bc),c,Amp) = gcd(a,c,Amp)
⟨proof T4 73⟩
◇
```

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

To do so we first had to prove several lemmas we do not want to list up here, but which are included in the appendix.

Here we only present the proof of theorem `T2` as an example. We point out that this proof is nothing more than a translation of a proof given in a textbook. Note that most steps invoke the application of a lemma. The technique to prove the other theorems is exactly the same and we put the code in the appendix.

The proof of `T2` proceeds as follows: The first part consists of showing that `gcd(a,bc,Amp)` is associated to `gcd(a,c,Amp)`.

⟨proof T2 25a⟩ ≡

```
proof
let Amp be AmpleSet of I;
let A,B,C be Element of the carrier of I;
assume H1: gcd(A,B,Amp) = (1.I);
H2: gcd(AC,BC,Amp) is_associated_to Cgcd(A,B,Amp)
    by T1;
H3: Cgcd(A,B,Amp) = C(1.I)  by H1
                  .= C      by VECTSP_2:1;
H4: gcd(AC,BC,Amp) is_associated_to C by H2,H3;
H5: C is_associated_to gcd(AC,BC,Amp) by H4,L2;
H6: gcd(A,C,Amp) is_associated_to
    gcd(A,gcd(AC,BC,Amp),Amp) by H5,L14;
H7a: gcd(A,gcd(AC,BC,Amp),Amp) =
     gcd(gcd(A,AC,Amp),BC,Amp) by T0;
H7: gcd(A,gcd(AC,BC,Amp),Amp) is_associated_to
    gcd(gcd(A,AC,Amp),BC,Amp) by H7a,L2;
H8: gcd(A,C,Amp) is_associated_to
    gcd(gcd(A,AC,Amp),BC,Amp) by H6,H7,L2;
H9: gcd(A,AC,Amp) is_associated_to A
    ⟨proof H9 74a⟩
H10: gcd(gcd(A,AC,Amp),BC,Amp) is_associated_to
     gcd(A,BC,Amp) by H9,L14;
H11: gcd(A,C,Amp) is_associated_to gcd(A,BC,Amp)
     by H8,H10,L2;
H12: gcd(A,BC,Amp) is_associated_to gcd(A,C,Amp)
     by H11,L2;
⟨proof T2 2 25b⟩
◇
```

Macro referenced in scrap 23b.

Now the identity of `gcd(a,bc,Amp)` and `gcd(a,c,Amp)` follows because they are associated to each other and both included in the set `Amp` of representatives.

⟨proof T2 2 25b⟩ ≡

```
H13: gcd(A,BC,Amp) is Element of Amp by Def4;
H14: gcd(A,C,Amp) is Element of Amp by Def4;
H15: gcd(A,BC,Amp) = gcd(A,C,Amp) by H12,H13,H14,AMP;
thus thesis by H15;
end;
◇
```

Macro referenced in scrap 25a.

6 The Theorems of Brown and Henrici

Now we are ready to prove the theorems of Brown and Henrici concerning addition and multiplication in fraction fields. The properties established in these two theorems are crucial for correctness of the algorithms.

"GCD.MIZ" 26 ≡

```
theorem HEN1:
for Amp being AmpleSet of I
for r1,r2,s1,s2 being Element of the carrier of I holds
 (gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I) &
  r2 <> (0.I) & s2 <> (0.I))
 implies
 gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
     r2(s2/gcd(r2,s2,Amp)),Amp) =
 gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
     gcd(r2,s2,Amp),Amp)
⟨proof HEN1 27⟩

theorem HEN2:
for Amp being AmpleSet of I
for r1,r2,s1,s2 being Element of the carrier of I holds
 (gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I) &
  r2 <> (0.I) & s2 <> (0.I))
 implies
 gcd((r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)),
     (r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp)),Amp) = (1.I)
⟨proof HEN2 28⟩
◇
```

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

The proof of `HEN1` takes advantage of the fact that

```
                    r2(s2/gcd(r2,s2,Amp))
= gcd(r2,s2,Amp)(r2/gcd(r2,s2,Amp))(s2/gcd(r2,s2,Amp))
```

so that two applications of theorem `T2` eliminate `(s2/gcd(r2,s2,Amp))` and `(r2/gcd(r2,s2,Amp))` leaving the desired `gcd(r2,s2,Amp)` as the second argument of the `gcd`.
So all we have to do is to show that the assumptions of `T2` holds. Here we present the first eliminaton only. The other one is similar and can be found in the appendix.

⟨proof HEN1 27⟩ ≡

```
proof
let Amp be AmpleSet of I;
let r1,r2,s1,s2 be Element of the carrier of I;
assume H1: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I)
           & r2 <> (0.I) & s2 <> (0.I);
consider d being Element of the carrier of I such that
H2: d = gcd(r2,s2,Amp);
H2a: d divides s2 by H2,Def4;
H2b: d divides r2 by H2,Def4;
K: d <> (0.I) by H2,H1,L12;
consider r being Element of the carrier of I such that
H4: r = r2/d;
consider s being Element of the carrier of I such that
H5: s = s2/d;

H6: gcd((r1s)+(s1r),s,Amp) = gcd(s1r,s,Amp) by T4;
H7: gcd(s,s1,Amp) = (1.I)
    ⟨proof H7 75a⟩
H8: gcd(s,s1r,Amp) = gcd(s,r,Amp) by H7,T2;
H9: gcd(r,s,Amp) = (1.I) by H4,H5,H2,K,T3;
H10:   gcd((r1s)+(s1r),s,Amp)
     = gcd(s1r,s,Amp)  by H6
    .= gcd(s,s1r,Amp)  by L13
    .= gcd(s,r,Amp)    by H8
    .= gcd(r,s,Amp)    by L13
    .= (1.I)           by H9;
    :: assumption of T2 holds
H11: r2s = s(dr)
     ⟨proof H11 75b⟩
    :: the equality mentioned above
H12:   gcd((r1s)+(s1r),r2s,Amp)
     = gcd((r1s)+(s1r),s(dr),Amp) by H11
    .= gcd((r1s)+(s1r),dr,Amp)    by H10,T2;
    :: the first elimination
⟨proof HEN1 2 74b⟩
◇
```

Macro referenced in scrap 26.

The proof of HEN2 also consists of two steps. In the first step one shows that

$$\begin{aligned}&\texttt{gcd(r2/gcd(s1,r2,Amp),r1/gcd(r1,s2,Amp),Amp)},\\&\texttt{gcd(s2/gcd(r1,s2,Amp),s1/gcd(s1,r2,Amp),Amp)},\\&\texttt{gcd(s2/gcd(r1,s2,Amp),r1/gcd(r1,s2,Amp),Amp)}\text{ and}\\&\texttt{gcd(r2/gcd(s1,r2,Amp),r1/gcd(r1,s2,Amp),Amp)}\end{aligned}$$

are all equal to (1.R). This is done by using theorem T3.

⟨proof HEN2 28⟩ ≡

```
proof
let Amp be AmpleSet of I;
let r1,r2,s1,s2 be Element of the carrier of I;
assume H1: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I)
           & r2 <> (0.I) & s2 <> (0.I);
consider d1 being Element of the carrier of I such that
H2: d1 = gcd(r1,s2,Amp);
consider d2 being Element of the carrier of I such that
H3: d2 = gcd(s1,r2,Amp);
H4: d1 <> (0.I) by H2,H1,L12;
H5: d2 <> (0.I) by H3,H1,L12;
consider r1' being Element of the carrier of I such that
H6: r1' = r1/d1;
consider s1' being Element of the carrier of I such that
H7: s1' = s1/d2;
consider r2' being Element of the carrier of I such that
H8: r2' = r2/d2;
consider s2' being Element of the carrier of I such that
H9: s2' = s2/d1;
H27: gcd(r2',r1',Amp) = (1.I)
     ⟨proof H27 76b⟩
H45: gcd(s1',s2',Amp) = (1.I)
     ⟨proof H45 77a⟩
M1: gcd(s2',r1',Amp) = gcd(r1',s2',Amp)      by L13
                     .= gcd(r1/d1,s2/d1,Amp) by H6,H9
                     .= (1.I)                by H4,H2,T3;
M2: gcd(s1',r2',Amp) = gcd(s1/d2,r2/d2,Amp)  by H7,H8
                     .= (1.I)                by H5,H3,T3;
⟨proof HEN2 2 29a⟩
◇
```

Macro referenced in scrap 26.

Now we can put together the desired result by using theorem T2:

⟨proof HEN2 2 29a⟩ ≡

```
M3: gcd(r1's1',r2',Amp) = gcd(r2',r1's1',Amp) by L13
                        .= gcd(r2',s1',Amp)    by H27,T2
                        .= gcd(s1',r2',Amp)    by L13
                        .= (1.I)               by M2;
M4:   gcd(r1's1',r2's2',Amp)
    = gcd(r1's1',s2',Amp)  by M3,T2
   .= gcd(s2',r1's1',Amp)  by L13
   .= gcd(s2',s1',Amp)     by M1,T2
   .= gcd(s1',s2',Amp)     by L13
   .= (1.I)                by H45;
M5: gcd((r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)),
        (r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp)),Amp) = (1.I)
    by M4,H6,H7,H8,H9,H2,H3;
thus thesis by M5;
end;
```

◇

Macro referenced in scrap 28.

7 Correctness of the Algorithms

In this section we formulate and prove correctness conditions for the algorithms of Brown and Henrici concerning addition and multiplication in fraction fields.

First we define two further predicates for elements of integral domains. It may be helpful to consider x as the numerator and y as the denominator of a fraction.

"GCD.MIZ" 29b ≡

```
definition
let I be gcdDomain;
let x,y be Element of the carrier of I;
 pred x canonical y means :Def10:
 ex Amp being AmpleSet of I st gcd(x,y,Amp) = (1.I);
end;

definition
let I be gcdDomain;
let Amp be AmpleSet of I;
let x,y be Element of the carrier of I;
  pred x,y are_normalized_wrt Amp means :Def27:
```

```
    gcd(x,y,Amp) = (1.I) & y ∈ Amp & y <> (0.I);
  end;
```

⟨more def 77b⟩
◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

Note that the predicate `canonical` is independent of the ampleset. This is due to the fact that our amplesets always contain `(1.I)`. As we already mentioned using non multiplicative amplesets allows proving that the output of the algorithms is again canonical but not that it is normalized.
The second predicate describes *normalized fractions*. Changing the ampleset may change the value of this predicate due to the second term of the definition. Here we stress again that the theorems we will prove hold for arbitrary (multiplicative) amplesets.

The next step is to define MIZAR functions that mirror the input/output behaviour of the two algorithms. For that we use the `if`-construct of the MIZAR language. We stress that the guards of this `if` need not be disjunct. Therefore one also have to prove *consistency* of the definition that is whenever two guards hold at the same time the corresponding values of the `if` have to be identical.
Of course it is easy to reformulate the guards to make them disjunct in the usual sense of programming languages but we prefer to presuppose as less as possible. Note that proving consistency can be seen as proving the possibility to evaluate the `if`-construct in parallel taking the first terminating branch as the returned value.

7.1 The Addition Algorithm

The definition of the *addition function* is a straightforward translation of the algorithm presented in section two. `add1` gives the numerator of the result, `add2` the denominator.

```
"GCD.MIZ" 30 ≡
      definition
      let I be gcdDomain;
      let Amp be AmpleSet of I;
      let r1,r2,s1,s2 be Element of the carrier of I;
```

```
assume A: r1 canonical r2 & s1 canonical s2 &
          r2 = NF(r2,Amp) & s2 = NF(s2,Amp);
 func add1(r1,r2,s1,s2,Amp)
         -> Element of the carrier of I means :Def11a:
 it = s1 if r1 = (0.I),
 it = r1 if s1 = (0.I),
 it = r1s2 + r2s1 if gcd(r2,s2,Amp) = (1.I),
 it = (0.I) if (r1(s2/gcd(r2,s2,Amp))) +
               (s1(r2/gcd(r2,s2,Amp))) = (0.I)
 otherwise it = ((r1(s2/gcd(r2,s2,Amp))) +
                  (s1(r2/gcd(r2,s2,Amp))))  /
                 gcd((r1(s2/gcd(r2,s2,Amp))) +
                     (s1(r2/gcd(r2,s2,Amp))),
                     gcd(r2,s2,Amp),Amp);
existence;
uniqueness;
⟨consistency add1 79⟩
end;

definition
let I be gcdDomain;
let Amp be AmpleSet of I;
let r1,r2,s1,s2 be Element of the carrier of I;
assume A: r1 canonical r2 & s1 canonical s2 &
          r2 = NF(r2,Amp) & s2 = NF(s2,Amp);
 func add2(r1,r2,s1,s2,Amp)
         -> Element of the carrier of I means :Def12a:
 it = s2 if r1 = (0.I),
 it = r2 if s1 = (0.I),
 it = r2s2 if gcd(r2,s2,Amp) = (1.I),
 it = (1.I) if (r1(s2/gcd(r2,s2,Amp))) +
               (s1(r2/gcd(r2,s2,Amp))) = (0.I)
 otherwise it = (r2(s2/gcd(r2,s2,Amp)))  /
                gcd((r1(s2/gcd(r2,s2,Amp))) +
                    (s1(r2/gcd(r2,s2,Amp))),
                    gcd(r2,s2,Amp),Amp);
existence;
uniqueness;
⟨consistency add2 81⟩
end;
```

◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

Now we formulate the two theorems that ensures the correctness of the addition algorithm. Due to the first one the output `t` is again normalized, due to the second one the algorithm indeed is an addition algorithm (that is the output `t` and the result of the usual addition are members of the same equivalence class of the fraction field).

"GCD.MIZ" 32 ≡

```
theorem
  for Amp being AmpleSet of I
  for r1,r2,s1,s2 being Element of the carrier of I holds
   (Amp is_multiplicative &
    r1,r2 are_normalized_wrt Amp &
    s1,s2 are_normalized_wrt Amp)
  implies
    add1(r1,r2,s1,s2,Amp),add2(r1,r2,s1,s2,Amp)
    are_normalized_wrt Amp
⟨proof ALG1 86b⟩

theorem
  for Amp being AmpleSet of I
  for r1,r2,s1,s2 being Element of the carrier of I holds
   (Amp is_multiplicative &
    r1,r2 are_normalized_wrt Amp &
    s1,s2 are_normalized_wrt Amp)
  implies
    add1(r1,r2,s1,s2,Amp)(r2s2) =
    add2(r1,r2,s1,s2,Amp)((r1s2)+(s1r2))
⟨proof ALG2 88⟩
◇
```

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

The proof of the theorems consists of proving the cases of the `if`-construct. The cases that are not trivial are handled by showing that theorem `HEN1` is applicable. This yields that the result is canonical. The other properties we need follow directly from the fact that we use multiplicative amplesets.

Here we give an example of a trivial case and of a case using `HEN1`. The other cases can be found in the appendix.

⟨example cases ALG1 33a⟩ ≡

```
case A: r1 = (0.I);
A1: add1(r1,r2,s1,s2,Amp) = s1 by A,H0,H3a,Def11a;
A2: add2(r1,r2,s1,s2,Amp) = s2 by A,H0,H3a,Def12a;
A3:   gcd(add1(r1,r2,s1,s2,Amp),add2(r1,r2,s1,s2,Amp),Amp)
    = gcd(s1,s2,Amp)  by A1,A2
   .= (1.I)           by H3;
thus thesis by A3,A2,H0b,Def27;

case C: gcd(r2,s2,Amp) = (1.I);
C1: add1(r1,r2,s1,s2,Amp) = (r1s2)+(r2s1) by C,H0,H3a,Def11a;
C2: add2(r1,r2,s1,s2,Amp) = r2s2 by C,H0,H3a,Def12a;
C3:   gcd(add1(r1,r2,s1,s2,Amp),add2(r1,r2,s1,s2,Amp),Amp)
    = gcd((r1s2)+(r2s1),r2s2,Amp) by C1,C2
   .= gcd((r1s2)+(s1r2),r2s2,Amp)
   .= gcd((r1(s2/(1.I)))+(s1r2),r2s2,Amp) by L7a
   .= gcd((r1(s2/(1.I)))+(s1(r2/(1.I))),r2s2,Amp) by L7a
   .= gcd((r1(s2/(1.I)))+(s1(r2/(1.I))),r2(s2/(1.I)),Amp)
      by L7a
   .= gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
          r2(s2/gcd(r2,s2,Amp)),Amp) by C
   .= gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
          gcd(r2,s2,Amp),Amp) by H0,H3,HEN1
   .= (1.I)  by C,GCD2;
reconsider r2,s2 as Element of Amp by H0b;
C4: r2s2 ∈ Amp by H0a,Def25;
C5: r2s2 <> (0.I) by H0,VECTSP_2:15;
thus thesis by C2,C3,C4,C5,Def27;
```

◇

Macro referenced in scrap 86b.

The proof of theorem ALG2 is nothing more than equational resoning. Again we give two cases as examples.

⟨example cases ALG2 33b⟩ ≡

```
case C: gcd(r2,s2,Amp) = (1.I);
C1: add1(r1,r2,s1,s2,Amp) = (r1s2)+(r2s1) by C,H0,Def11a;
C2: add2(r1,r2,s1,s2,Amp) = r2s2 by C,H0,Def12a;
C3:   add1(r1,r2,s1,s2,Amp)(r2s2)
    = ((r1s2)+(r2s1))(r2s2)  by C1
   .= (r2s2)((r1s2)+(r2s1))
   .= add2(r1,r2,s1,s2,Amp)((r1s2)+(r2s1)) by C2;
thus thesis by C3;
```

```
case D: (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
        = (0.I);
D11: add1(r1,r2,s1,s2,Amp) = (0.I) by D,H0,Def11a;
D12: add2(r1,r2,s1,s2,Amp) = (1.I) by D,H0,Def12a;
D13: (r1s2)+(s1r2) = (0.I)
     ⟨proof D13 98⟩
D14:   add1(r1,r2,s1,s2,Amp)(r2s2)
     = (0.I)(r2s2)           by D11
    .= (0.I)                 by VECTSP_2:26
    .= (1.I)(0.I)            by VECTSP_2:26
    .= (1.I)((r1s2)+(s1r2))  by D13
    .= add2(r1,r2,s1,s2,Amp)((r1s2)+(s1r2)) by D12;
thus thesis by D14;
◇
```

Macro referenced in scrap 88.

7.2 The Multiplication Algorithm

The method for proving the mulitplication algorithm correct is identical to the one used in the last subsection. First we define a *nominator* and a *denumerator function* mirroring the behaviour of the algorithm.

"GCD.MIZ" 34 ≡

```
definition
let I be gcdDomain;
let Amp be AmpleSet of I;
let r1,r2,s1,s2 be Element of the carrier of I;
assume r1 canonical r2 & s1 canonical s2 &
       r2 = NF(r2,Amp) & s2 = NF(s2,Amp);
  func mult1(r1,r2,s1,s2,Amp)
            -> Element of the carrier of I means :Def13:
  it = (0.I) if r1 = (0.I) or s1 = (0.I),
  it = r1s1 if r2 = (1.I) & s2 = (1.I),
  it = (r1s1)/gcd(r1,s2,Amp) if s2 <> (0.I) & r2 = (1.I),
  it = (r1s1)/gcd(s1,r2,Amp) if r2 <> (0.I) & s2 = (1.I)
  otherwise it = (r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp));
existence;
uniqueness;
⟨consistency mult1 92⟩
end;
```

```
definition
let I be gcdDomain;
let Amp be AmpleSet of I;
let r1,r2,s1,s2 be Element of the carrier of I;
assume AS: r1 canonical r2 & s1 canonical s2 &
           r2 = NF(r2,Amp) & s2 = NF(s2,Amp);
  func mult2(r1,r2,s1,s2,Amp)
           -> Element of the carrier of I means :Def14:
  it = (1.I) if r1 = (0.I) or s1 = (0.I),
  it = (1.I) if r2 = (1.I) & s2 = (1.I),
  it = s2/gcd(r1,s2,Amp) if s2 <> (0.I) & r2 = (1.I),
  it = r2/gcd(s1,r2,Amp) if r2 <> (0.I) & s2 = (1.I)
  otherwise it = (r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp));
existence;
uniqueness;
⟨consistency mult2 95⟩
end;
```
◇

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

The theorems we have to prove to establish the correctness of the multiplication algorithm are

"GCD.MIZ" 35 ≡
```
theorem
  for Amp being AmpleSet of I
  for r1,r2,s1,s2 being Element of the carrier of I holds
   (Amp is_multiplicative &
    r1,r2 are_normalized_wrt Amp &
    s1,s2 are_normalized_wrt Amp)
  implies
    mult1(r1,r2,s1,s2,Amp),mult2(r1,r2,s1,s2,Amp)
    are_normalized_wrt Amp
⟨proof ALG3 99⟩

theorem
  for Amp being AmpleSet of I
  for r1,r2,s1,s2 being Element of the carrier of I holds
   (Amp is_multiplicative &
    r1,r2 are_normalized_wrt Amp &
    s1,s2 are_normalized_wrt Amp)
  implies
```

```
    mult1(r1,r2,s1,s2,Amp)(r2s2) =
    mult2(r1,r2,s1,s2,Amp)(r1s1)
⟨proof ALG4 101⟩
◇
```

File defined by scraps 6a, 7c, 8, 9b, 13ab, 14a, 17b, 18bc, 19, 20ab, 21, 23b, 26, 29b, 30, 32, 34, 35.

Again the proof that the output of the algorithm is normalized consists of trivial cases and cases that use theorem HEN2. We give two examples here:

⟨example cases ALG3 36⟩ ≡

```
case B: r2 = (1.I) & s2 = (1.I);
B1: mult1(r1,r2,s1,s2,Amp) = r1s1 by B,H0,Def13;
B2: mult2(r1,r2,s1,s2,Amp) = (1.I) by B,H0,Def14;
B3: gcd(mult1(r1,r2,s1,s2,Amp),mult2(r1,r2,s1,s2,Amp),Amp)
    = gcd((r1s1),(1.I),Amp)  by B1,B2
   .= (1.I)                   by GCD2;
B4: (1.I) ∈ Amp by AMP;
B5: (1.I) <> (0.I) by VECTSP_1:def 21;
thus thesis by B2,B3,B4,B5,Def27;

case E: not(r1 = (0.I) or s1 = (0.I)) &
        not(r2 = (1.I) & s2 = (1.I)) &
        not(s2 <> (0.I) & r2 = (1.I)) &
        not(r2 <> (0.I) & s2 = (1.I));
E1: mult1(r1,r2,s1,s2,Amp) =
       (r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)) by E,H0,Def13;
E2: mult2(r1,r2,s1,s2,Amp) =
       (r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp)) by E,H0,Def14;
E3:   gcd(mult1(r1,r2,s1,s2,Amp),mult2(r1,r2,s1,s2,Amp),Amp)
    = gcd((r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)),
           mult2(r1,r2,s1,s2,Amp),Amp)  by E1
   .= gcd((r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)),
           (r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp)),Amp)
      by E2
   .= (1.I) by H0,H3,HEN2;
E10a: gcd(r1,s2,Amp) divides s2 by Def4;
E10b: gcd(r1,s2,Amp) <> (0.I) by H0,L12;
E10: s2/gcd(r1,s2,Amp) <> (0.I) by H0,E10a,E10b,L26;
E11a: gcd(r1,s2,Amp) ∈ Amp by Def4;
reconsider z1 = gcd(r1,s2,Amp) as Element of Amp by E11a;
reconsider s2 as Element of Amp by H3;
E11: s2/z1 ∈ Amp by AMP5,E10a,E10b,H0a;
E12a: gcd(s1,r2,Amp) divides r2 by Def4;
```

```
E12b: gcd(s1,r2,Amp) <> (0.I) by H0,L12;
E12: r2/gcd(s1,r2,Amp) <> (0.I) by H0,E12a,E12b,L26;
E13a: gcd(s1,r2,Amp) ∈ Amp by Def4;
reconsider z2 = gcd(s1,r2,Amp) as Element of Amp by E13a;
reconsider r2 as Element of Amp by H3;
E13: r2/z2 ∈ Amp by AMP5,E12a,E12b,H0a;
reconsider u = s2/z1 as Element of Amp by E11;
reconsider v = r2/z2 as Element of Amp by E13;
E14: vu ∈ Amp by Def25,H0a;
E15: vu <> (0.I) by E10,E12,VECTSP_2:15;
thus thesis by E3,E2,E14,E15,Def27;
◇
```

Macro referenced in scrap 99.

At last we also give two example cases out of the proof of theorem ALG4.

⟨example cases ALG4 37⟩ ≡

```
case B: r2 = (1.I) & s2 = (1.I);
B1: mult1(r1,r2,s1,s2,Amp) = r1s1 by B,H0,Def13;
B2: mult2(r1,r2,s1,s2,Amp) = (1.I) by B,H0,Def14;
B3:   mult1(r1,r2,s1,s2,Amp)(r2s2)
    = mult1(r1,r2,s1,s2,Amp)((1.I)(1.I))  by B
   .= mult1(r1,r2,s1,s2,Amp)(1.I)         by VECTSP_2:1
   .= mult1(r1,r2,s1,s2,Amp)              by VECTSP_2:1
   .= r1s1                                by B1
   .= (1.I)(r1s1)                         by VECTSP_2:1
   .= mult2(r1,r2,s1,s2,Amp)(r1s1)        by B2;
thus thesis by B3;

case D: r2 <> (0.I) & s2 = (1.I);
D1: mult1(r1,r2,s1,s2,Amp) = (r1s1)/gcd(s1,r2,Amp)
    by D,H0,Def13;
D2: mult2(r1,r2,s1,s2,Amp) = r2/gcd(s1,r2,Amp) by D,H0,Def14;
D3: gcd(s1,r2,Amp) divides s1 by Def4;
D4: gcd(s1,r2,Amp) divides s1r1 by D3,L6a;
D5: gcd(s1,r2,Amp) divides (s1r1)r2 by D4,L6a;
D6:   ((r1s1)/gcd(s1,r2,Amp))(r2s2)
    = ((r1s1)/gcd(s1,r2,Amp))(r2(1.I))  by D
   .= ((r1s1)/gcd(s1,r2,Amp))r2         by VECTSP_2:1
   .= ((r1s1)r2)/gcd(s1,r2,Amp)         by H1,D4,D5,L8;
D8: gcd(s1,r2,Amp) divides r2 by Def4;
D9: gcd(s1,r2,Amp) divides r2r1 by D8,L6a;
D10: gcd(s1,r2,Amp) divides (r2r1)s1 by D9,L6a;
```

```
D11:   (r2/gcd(s1,r2,Amp))(r1s1)
     = ((r2/gcd(s1,r2,Amp))r1)s1  by VECTSP_1:def 16
    .= ((r2r1)/gcd(s1,r2,Amp))s1  by H1,D8,D9,L8
    .= ((r2r1)s1)/gcd(s1,r2,Amp)  by H1,D9,D10,L8
    .= ((r1r2)s1)/gcd(s1,r2,Amp)
    .= (r1(r2s1))/gcd(s1,r2,Amp)  by VECTSP_1:def 16
    .= (r1(s1r2))/gcd(s1,r2,Amp)
    .= ((r1s1)r2)/gcd(s1,r2,Amp)  by VECTSP_1:def 16;
D12:   mult1(r1,r2,s1,s2,Amp)(r2s2)
     = ((r1s1)/gcd(s1,r2,Amp))(r2s2)  by D1
    .= ((r1s1)r2)/gcd(s1,r2,Amp)      by D6
    .= (r2/gcd(s1,r2,Amp))(r1s1)      by D11
    .= mult2(r1,r2,s1,s2,Amp)(r1s1)   by D2;
thus thesis by D12;
◇
```

Macro referenced in scrap 101.

8 Conclusion and Further Work

We have proved the correctness of the algorithms of Brown and Henrici concerning addition and multiplication in fraction fields using the MIZAR system. There are three points about this approach we want to mention here:

We have proved the correctness of the algorithms not for a special domain but in an generic algebraic setting.
The MIZAR checker has verified our article. So we get proof assistance that decreases the possibiltity of errors in algebraic proofs.
The MIZAR system also serves as a database for mathematical knowledge. So if one wants to prove other generic algorithms correct, one may take advantage of our MIZAR article by using some lemmas about integral domains or gcdDomains. That is once these lemmas have been proved new proofs can start on a higher level and become shorter.

We did not define fracton fields explicitly in our article but numerator and denominator functions for the algorithms. We plan to remove these functions by filling the gap concerning fration fields.

Another point is that we proved the correctness of the algorithms by hand: We defined functions mirroring the algorithms and formulated theo-

rems that ensure the desired properties of correctness. We think about (automatically) decomposing algorithms using *Floyd-Hoare-Logic* into algebraic theorems which once they have been proved in MIZAR would allow to conclude the correctness of the algorithms.

Acknowledgements We are grateful to Prof. Dr. R. Loos for initiating this research. Also we would like to thank Prof. Dr. D. Musser and Dr. S. Schupp for pointing our attention to MIZAR as well as Prof. Dr. A. Trybulec for many useful hints concerning the MIZAR language.

A References

Literatur

[Br89] Preston Briggs, NUWEB - A simple literate programming tool. May 1989, preston@cd.rice.edu

[Co??] George Collins, lecture manuscript.

[He56] P. Henrici, A subroutine for computations with rational numbers. J.ACM 3(1), 1956

[LS96] Rdiger Loos and Sibylle Schupp, SUCHTHAT User's guide. October 1996, schupp@cs.rpi.edu

[Ru92] Piotr Rudnicki, An overwiew of the MIZAR project. June 1992, piotr@cs.ualberta.ca

[Sc96] Sibylle Schupp, Generic programming - SUCHTHAT one can build an algebraic library. Dissertation, University of Tübingen, 1996

[Tr93] Andrzej Trybulec, Some features of the MIZAR language. July 1993, trybulec@math.uw.bialystok.pl

B Additional MIZAR Code

B.1 Environment

⟨notation 40a⟩ ≡

```
TARSKI,BOOLE,STRUCT_0,RLVECT_1,SETFAM_1,VECTSP_1, VECTSP_2;
```

◇

Macro referenced in scrap 6b.

⟨constructors 40b⟩ ≡

```
ALGSTR_1;
```

◇

Macro referenced in scrap 6b.

⟨definitions 40c⟩ ≡

```
TARSKI,BOOLE;
```

◇

Macro referenced in scrap 6b.

⟨theorems 40d⟩ ≡

```
TARSKI,BOOLE,WELLORD2,SUBSET_1,ENUMSET1,VECTSP_1,VECTSP_2;
```

◇

Macro referenced in scrap 6b.

⟨clusters 40e⟩ ≡

```
STRUCT_0,VECTSP_1,VECTSP_2;
```

◇

Macro referenced in scrap 6b.

⟨schemes 40f⟩ ≡

```
SETFAM_1,GROUP_2;
```

◇

Macro referenced in scrap 6b.

B.2 Divisibility in Integral Domains

⟨more div 40g⟩ ≡

```
theorem
IDOM1: for a,b,c being Element of the carrier of I holds
```

```
       (a <> (0.I)) implies (((ab) = (ac) implies b = c) &
                             ((ba) = (ca) implies b = c))
proof
let a,b,c be Element of the carrier of I;
assume H0: a <> (0.I);
K1: now assume H1: ab = ac;
H2: (0.I) = (ab) + (-(ab))  by VECTSP_2:1
         .= (ab) + (-(ac))  by H1
         .= (ab) + (a(-c))  by VECTSP_2:28
         .= a(b + (-c))     by VECTSP_2:1
         .= a(b - c)        by VECTSP_1:12;
H3: b - c = (0.I) by H2,H0,VECTSP_2:15;
H4: c = (0.I) + c       by VECTSP_2:1
     .= (b - c) + c     by H3
     .= (b + (-c)) + c  by VECTSP_1:12
     .= b + (c + (-c))  by VECTSP_2:1
     .= b + (0.I)       by VECTSP_2:1
     .= b               by VECTSP_2:1;
thus b = c by H4;
end;  :: K1
thus thesis by K1;
end;

definition
let I be domRing;
let x,y be Element of the carrier of I;
assume d1: y divides x;
assume d2: y <> (0.I);
  func x/y -> Element of the carrier of I means :Def5:
  ity = x;
existence
  proof
  H1: ex z being Element of the carrier of I
                       st x = yz by d1,Def1;
  thus thesis by H1;
  end;
uniqueness
  by d2,IDOM1;
end;

theorem
L1a: for a,b,c,d being Element of the carrier of I holds
    ((b divides a) & (d divides c)) implies (bd) divides (ac)
proof
```

```
let a,b,c,d be Element of the carrier of I;
assume H1: (b divides a) & (d divides c);
consider x being Element of the carrier of I such that
H2: bx = a by H1,Def1;
consider y being Element of the carrier of I such that
H3: dy = c by H1,Def1;
H4: (bd)(yx) = ((bd)y)x  by VECTSP_1:def 16
            .= (b(dy))x  by VECTSP_1:def 16
            .= (bc)x     by H3
            .= (cb)x
            .= c(bx)     by VECTSP_1:def 16
            .= ca        by H2
            .= ac;
thus thesis by H4,Def1;
end;

theorem
L2: for a,b,c being Element of the carrier of I holds
    (a is_associated_to a) &
    ((a is_associated_to b) implies (b is_associated_to a)) &
    ((a is_associated_to b & b is_associated_to c)
                  implies (a is_associated_to c))
proof
let A,B,C be Element of the carrier of I;
H1: A  (1.I) = A  by VECTSP_2:1;
H2: A divides A by H1,Def1;
H9: A is_associated_to A by H2,Def3;
M1: now
assume H3: A is_associated_to B;
H4: A divides B & B divides A by H3,Def3;
thus A is_associated_to B implies
     B is_associated_to A by H4,Def3;
end;  :: M1
M2: now
assume H5: A is_associated_to B & B is_associated_to C;
H6: A divides B & B divides A by H5,Def3;
H7: B divides C & C divides B by H5,Def3;
H8: A divides C & C divides A by H6,H7,L1;
thus ((A is_associated_to B) & (B is_associated_to C))
     implies (A is_associated_to C) by H8,Def3;
end;
thus thesis by H9,M1,M2;
end;
```

```
theorem
L3: for a,b,c being Element of the carrier of I holds
    (a divides b) implies (c  a divides c  b)
proof
let A,B,C be Element of the carrier of I;
assume H1: A divides B;
consider D being Element of the carrier of I such that
H2: AD = B by H1,Def1;
H3: (CA)D = C(AD)  by VECTSP_1:def 16
         .= CB     by H2;
H4: CA divides CB by H3,Def1;
thus thesis by H4;
end;

theorem
L6: for a,b being Element of the carrier of I holds
    (a divides (ab)) & (b divides (ab)) by Def1;

theorem
L6a: for a,b,c being Element of the carrier of I holds
     a divides b implies a divides (bc)
proof
let a,b,c be Element of the carrier of I;
assume H0: a divides b;
consider d being Element of the carrier of I such that
H1: ad = b by H0,Def1;
H2: a(dc) = (ad)c  by VECTSP_1:def 16
         .= bc     by H1;
H3: a divides (bc) by H2,Def1;
thus thesis by H3;
end;

theorem
L26: for a,b being Element of the carrier of I holds
     (b divides a &  b <> (0.I))
     implies (a/b = (0.I) iff a = (0.I))
proof
let a,b be Element of the carrier of I;
assume H0: b divides a & b <> (0.I);
K1: now assume H1: a/b = (0.I);
consider d being Element of the carrier of I such that
H2: d = a/b;
H2a: d = (0.I) by H1,H2;
H3: a = d  b      by H2,H0,Def5
```

```
     .= (0.I)  b  by H2a
     .= (0.I)      by VECTSP_2:26;
thus a/b = (0.I) implies a = (0.I) by H3;
end;  :: K1
K2: now assume H1: a = (0.I);
consider d being Element of the carrier of I such that
H2: d = a/b;
H3: (0.I) = a      by H1
         .= d  b  by H2,H0,Def5;
H4: d = (0.I) by H3,H0,VECTSP_2:15;
thus a = (0.I) implies a/b = (0.I) by H2,H4;
end;  :: K2
thus thesis by K1,K2;
end;

theorem
L7: for a being Element of the carrier of I holds
    (a <> (0.I)) implies (a/a = (1.I))
proof
let A be Element of the carrier of I;
assume H0: A <> (0.I);
consider A' being Element of the carrier of I such that
H1: A' = A/A;
H2: A divides A by L1;
H3: A'A = A by H0,H1,H2,Def5
     .= (1.I)A  by VECTSP_2:1;
H5: A' = (1.I) by H0,H3,IDOM1;
thus thesis by H1,H5;
end;

theorem
L7a: for a being Element of the carrier of I
     holds a/(1.I) = a
proof
let a be Element of the carrier of I;
consider A being Element of the carrier of I such that
H0: A = a/(1.I);
H1: (1.I) <> (0.I) by VECTSP_1: def 21;
H2: (1.I)a = a by VECTSP_2:1;
H3: (1.I) divides a by H2,Def1;
H4: A = A(1.I) by VECTSP_2:1
     .= a      by H0,H1,H3,Def5;
thus thesis by H4,H0;
end;
```

```
theorem
L8: for a,b,c being Element of the carrier of I holds
   (c <> (0.I)) implies
    (((c divides (ab) & c divides a) implies
                         ((ab)/c = (a/c)b)) &
     ((c divides (ab) & c divides b) implies
                        ((ab)/c = a(b/c))))
proof
let A,B,C be Element of the carrier of I;
assume H0: C <> (0.I);
K1: now
assume H1: (C divides (AB)) & (C divides A);
consider A1 being Element of the carrier of I such that
H2: A1 = (AB)/C;
H3: A1C = AB by H2,H1,H0,Def5;
consider A2 being Element of the carrier of I such that
H4: A2 = A/C;
H5: A2C = A by H4,H1,H0,Def5;
H6: A1C = AB      by H3
       .= (A2C)B  by H5
       .= A2(CB)  by VECTSP_1:def 16
       .= A2(BC)
       .= (A2B)C  by VECTSP_1:def 16;
H7: A1 = (A2B) by H0,H6,IDOM1;
H8: (AB)/C = (A/C)B by H7,H2,H4;
thus ((C divides (AB)) & (C divides A)) implies
              ((AB)/C = (A/C)B) by H8;
end;  :: K1
K2: now
assume H1: (C divides (AB)) & (C divides B);
consider A1 being Element of the carrier of I such that
H2: A1 = (AB)/C;
H3: A1  C = A  B by H2,H1,H0,Def5;
consider A2 being Element of the carrier of I such that
H4: A2 = B/C;
H5: A2  C = B by H4,H1,H0,Def5;
H6: A1C = AB      by H3
       .= A(A2C)  by H5
       .= (AA2)C  by VECTSP_1:def 16;
H7: A1 = (AA2) by H0,H6,IDOM1;
H8: (AB)/C = A(B/C) by H7,H2,H4;
thus ((C divides (AB)) & (C divides B)) implies
              ((AB)/C = A(B/C)) by H8;
```

```
end;  :: K2
thus thesis by K1,K2;
end;

theorem
L8a: for a,b,c being Element of the carrier of I holds
     (c <> (0.I) &
      c divides a & c divides b & c divides (a + b))
     implies (a/c) + (b/c) = (a + b)/c
proof
let a,b,c be Element of the carrier of I;
assume H0: c <> (0.I);
assume H1:  c divides a & c divides b & c divides (a + b);
consider d being Element of the carrier of I such that
H2: d = a/c;
consider e being Element of the carrier of I such that
H3: e = b/c;
H4: dc = a by H2,H1,H0,Def5;
H5: ec = b by H3,H1,H0,Def5;
H6: a + b = (dc) + (ec)  by H4,H5
         .= (d + e)c     by VECTSP_2:1;
H7: c divides c by L1;
H8: c divides (d + e)  c by H6,H1;
H9: (a + b)/c = ((d + e)c)/c  by H6
             .= (d + e)(c/c)  by H0,H7,H8,L8
             .= (d + e)(1.I)  by H0,L7
             .= d + e         by VECTSP_2:1;
thus thesis by H9,H2,H3;
end;

theorem
 for a,b,c being Element of the carrier of I holds
 (c <> (0.I) & c divides a & c divides b) implies
 ((a/c) = (b/c) iff a = b)
proof
let a,b,c be Element of the carrier of I;
assume H0: c <> (0.I);
assume H1: c divides a & c divides b;
K1: now assume H4: (a/c) = (b/c);
consider d being Element of the carrier of I such that
H5: d = (a/c);
H6: dc = a by H0,H1,H5,Def5;
consider e being Element of the carrier of I such that
H7: e = (b/c);
```

```
H8: ec = b by H0,H1,H7,Def5;
H9: d = e by H5,H7,H4;
H10: a = dc  by H6
      .= ec  by H9
      .= b   by H8;
thus ((a/c) = (b/c)) implies (a = b) by H10;
end;  :: K1
thus thesis by K1;
end;

theorem
L8c: for a,b,c,d being Element of the carrier of I holds
     (b <> (0.I) & d <> (0.I) & b divides a & d divides c)
     implies (a/b)(c/d) = (ac)/(bd)
proof
let a,b,c,d be Element of the carrier of I;
assume H0: b <> (0.I) & d <> (0.I) &
           b divides a & d divides c;
consider x being Element of the carrier of I such that
H1: x = a/b;
consider y being Element of the carrier of I such that
H2: y = c/d;
consider z being Element of the carrier of I such that
H3: z = (ac)/(bd);
H4: xb = a by H0,H1,Def5;
H5: yd = c by H0,H2,Def5;
H6: (bd) divides (a  c) by H0,L1a;
H7: (bd) <> (0.I) by H0,VECTSP_2:15;
H8: z(bd) = ac        by H3,H7,H6,Def5
         .= (xb)(yd)  by H4,H5
         .= x(b(yd))  by VECTSP_1:def 16
         .= x((by)d)  by VECTSP_1:def 16
         .= x((yb)d)
         .= x(y(bd))  by VECTSP_1:def 16
         .= (xy)(bd)  by VECTSP_1:def 16;
H9: z = (xy) by H8,H7,IDOM1;
thus thesis by H9,H1,H2,H3;
end;

theorem
L9: for a,b,c being Element of the carrier of I holds
    ((a <> (0.I)) & ((ab) divides (ac)))
    implies (b divides c)
proof
```

```
let A,B,C be Element of the carrier of I;
assume H1: (A <> (0.I)) & (AB) divides (AC);
consider D being Element of the carrier of I such that
H2: (AB)D = AC by H1,Def1;
H3: A  (BD) = AC by H2,VECTSP_1:def 16;
H9: (A(BD))/A = (A/A)(BD)
    proof
    M1: A divides (A(BD)) by L6;
    M2: A divides A by L1;
    thus thesis by M1,M2,H1,L8;
    end;
H10: (AC)/A = (A/A)C
     proof
     M1: A divides (AC) by L6;
     M2: A divides A by L1;
     thus thesis by M1,M2,H1,L8;
     end;
H11: BD = (1.I)(BD)  by VECTSP_2:1
       .= (A/A)(BD)  by L7,H1
       .= (A(BD))/A  by H9
       .= (AC)/A     by H3
       .= (A/A)C     by H10
       .= (1.I)C     by L7,H1
       .= C          by VECTSP_2:1;
thus thesis by H11,Def1;
end;

theorem
 for a being Element of the carrier of I holds
 a is_associated_to (0.I) implies a = (0.I)
proof
let A be Element of the carrier of I;
assume H0: A is_associated_to (0.I);
H1: (0.I) divides A by H0,Def3;
consider D being Element of the carrier of I such that
H2: (0.I)D = A by H1,Def1;
H3: A = (0.I) by H2,VECTSP_2:26;
thus thesis by H3;
end;

theorem
L10: for a,b,c being Element of the carrier of I holds
     ((a <> (0.I)) & (ab = a)) implies (b = (1.I))
proof
```

```
let A,B be Element of the carrier of I;
K1: now assume H1: (A <> (0.I)) & (AB = A);
consider A' being Element of the carrier of I such that
H2: A' = A/A;
consider B' being Element of the carrier of I such that
H3: B' = (AB)/A;
H6: A' = (1.I) by H2,L7,H1;
H7: (AB)/A = (A/A)  B
    proof
    M1: A divides (AB) by L6;
    thus thesis by H1,L8,M1;
    end;
H8: B' = (AB)/A  by H3
      .= (A/A)B  by H7
      .= A'B     by H2
      .= B       by H6,VECTSP_2:1;
H10: A' = B' by H1,H2,H3;
thus (AB = A) implies (B = (1.I)) by H6,H10,H8;
end;  :: K1
thus thesis by K1;
end;

theorem
L15: for a,b,c being Element of the carrier of I holds
     ((c <> (0.I)) & ((ca) is_associated_to (cb)))
     implies (a is_associated_to b)
proof
let A,B,C be Element of the carrier of I;
assume H0: (C <> (0.I)) & ((CA) is_associated_to (CB));
H1: (CA) divides (CB) by H0,Def3;
H2: A divides B by H1,H0,L9;
H3: (CB) divides (CA) by H0,Def3;
H4: B divides A by H3,H0,L9;
thus thesis by H2,H4,Def3;
end;
```

◇

Macro referenced in scrap 8.

B.3 AmpleSets

⟨correctness Class 49⟩ ≡

```
existence
  proof
```

```
    set M = { b where b is Element of the carrier of I:
                                  b is_associated_to a};
    K1: M is non empty Subset of the carrier of I
        proof
        K2: now let B be Any;
        K3: now assume L1: B ∈ M;
        L2: ex B' being Element of the carrier of I st
            B = B' & B' is_associated_to a by L1;
        L3: B ∈ (the carrier of I) by L2;
        thus (B ∈ M) implies B ∈ (the carrier of I) by L3;
        end;
        thus (B ∈ M) implies B ∈ (the carrier of I) by K3;
        end;
        L4: M c= (the carrier of I) by K2,TARSKI:def 3;
        L5: M is non empty
            proof
            H1: a is_associated_to a by L2;
            H2: a ∈ M by H1;
            thus thesis by H2;
            end;
        thus thesis by L4,L5;
        end;
    K4: now let A be Element of the carrier of I;
    H1: (A ∈ M) implies (A is_associated_to a)
        proof
        assume M1: A ∈ M;
        M2: ex A' being Element of the carrier of I st
            A = A' & A' is_associated_to a by M1;
        M3: A is_associated_to a by M2;
        thus thesis by M3;
        end;
    thus (A ∈ M) iff (A is_associated_to a) by H1;
    end;
    K5: for A being Element of the carrier of I holds
        (A ∈ M) iff (A is_associated_to a) by K4;
    thus thesis by K1,K5;
    end;
  uniqueness
    proof
    let M,N be non empty Subset of the carrier of I;
    assume H1: for A being Element of the carrier of I holds
              (A ∈ M iff A is_associated_to a);
    assume H2: for A being Element of the carrier of I holds
              (A ∈ N iff A is_associated_to a);
```

```
      H3: for a being Element of the carrier of I holds
          (a ∈ M iff a ∈ N)
          proof
          let A be Element of the carrier of I;
          K1: now assume M1: A ∈ M;
          M2: A is_associated_to a by H1,M1;
          M3: A ∈ N by M2,H2;
          thus (A ∈ M) implies (A ∈ N) by M3;
          end;
          K2: now assume M1: A ∈ N;
          M2: A is_associated_to a by H2,M1;
          M3: A ∈ M by M2,H1;
          thus (A ∈ N) implies (A ∈ M) by M3;
          end;
          thus thesis by K1,K2;
          end;
      H4: M = N by H3,SUBSET_1:8;
      thus thesis by H4;
      end;
    end;
```

◇

Macro referenced in scrap 13a.

⟨correctness Classes 51a⟩ ≡

```
    existence
      from SubFamEx;
    uniqueness
      proof
      let F1,F2 be Subset-Family of the carrier of I;
      assume A: for A being Subset of the carrier of I holds
       A ∈ F1 iff
       (ex a being Element of the carrier of I st A = Class a);
      assume B: for A being Subset of the carrier of I holds
       A ∈ F2 iff
       (ex a being Element of the carrier of I st A = Class a);
      thus thesis from SubFamComp(A,B);
      end;
    end;
```

◇

Macro referenced in scrap 13a.

⟨proof CL1 51b⟩ ≡

```
proof
let a,b be Element of the carrier of I;
assume H0: Class a ∩ Class b <> ∅;
H0a: Class a meets Class b by H0,BOOLE:119;
consider Z being Any such that
H1: Z ∈ Class a & Z ∈ Class b by H0a,BOOLE:def 5;
H2: Z ∈ Class a by H1;
H2a: Z is Element of the carrier of I by H2;
reconsider Z as Element of the carrier of I by H2a;
H3: Z ∈ Class b by H1;
H4: Z is_associated_to a by H2,Defh1;
H5: Z is_associated_to b by H3,Defh1;
H6: c ∈ Class a implies c ∈ Class b
    proof
    assume H7: c ∈ Class a;
    H8: c is_associated_to a by H7,Defh1;
    H9: a is_associated_to c by H8,L2;
    H10: Z is_associated_to c by H4,H9,L2;
    H11: b is_associated_to Z by H5,L2;
    H12: b is_associated_to c by H11,H10,L2;
    H13: c is_associated_to b by H12,L2;
    H14: c ∈ Class b by H13,Defh1;
    thus thesis by H14;
    end;
H15: c ∈ Class b implies c ∈ Class a
    proof
    assume H7: c ∈ Class b;
    H16: c is_associated_to b by H7,Defh1;
    H17: b is_associated_to c by H16,L2;
    H18: Z is_associated_to c by H5,H17,L2;
    H19: a is_associated_to Z by H4,L2;
    H20: a is_associated_to c by H19,H18,L2;
    H21: c is_associated_to a by H20,L2;
    H22: c ∈ Class a by H21,Defh1;
    thus thesis by H22;
    end;
H23: c ∈ Class a iff c ∈ Class b by H6,H15;
H24: Class a = Class b by H23,SUBSET_1:8;
thus thesis by H24;
end;
```

◇

Macro referenced in scrap 13b.

⟨proof CL2 52⟩ ≡

```
 proof
let I be domRing;
H3: Class (1.I) ∈ Classes I by Defh2;
thus thesis by H3;
end;
```
◇

Macro referenced in scrap 13b.

⟨proof CL3 53a⟩ ≡
```
 proof
let X be Subset of the carrier of I;
assume H0: X ∈ Classes I;
H1: ex a being Element of the carrier of I st
    X = Class a by H0,Defh2;
thus thesis by H1;
end;
```
◇

Macro referenced in scrap 13b.

⟨proof K2 53b⟩ ≡
```
proof
let X be Any such that H0: X ∈ M;
consider A being Element of the carrier of I such that
H1: X = Class A by H0,Defh2;
thus thesis by H1;
end;
```
◇

Macro referenced in scrap 14b.

⟨proof K3 53c⟩ ≡
```
proof
let X,Y be Any such that H0: X ∈ M & Y ∈ M & X <> Y;
assume H1: X ∩ Y <> ∅;
consider A being Element of the carrier of I such that
H2: X = Class A by H0,Defh2;
consider B being Element of the carrier of I such that
H3: Y = Class B by H0,Defh2;
H4: X = Y by H1,H2,H3,CL1;
H5: contradiction by H0,H4;
thus thesis by H5;
end;
```
◇

Macro referenced in scrap 14b.

⟨proof K5a 54a⟩ ≡

```
proof
M0: Class (1.I) ∈ M by Defh2;
consider x being Any such that
M1: AmpS' ∩ Class (1.I) = {x} by K5,M0;
M2: x ∈ {x} by ENUMSET1:4;
M3: x ∈ AmpS' ∩ Class (1.I) by M2,M1;
M4: x ∈ AmpS' by M3,BOOLE:def 3;
thus thesis by M4;
end;
```

◇

Macro referenced in scrap 14b.

⟨proof K6a 54b⟩ ≡

```
proof
let X be Element of M;
consider x being Any such that
H1: AmpS' ∩ X = {x} by K5;
H2a: X ∈ Classes I;
H2: X is non empty Subset of the carrier of I by H2a,CL3;
H3: x ∈ {x} by ENUMSET1:4;
H4: x ∈ AmpS' ∩ X by H3,H1;
H5: x ∈ AmpS' by H4,BOOLE:def 3;
H5a: x ∈ X by H4,BOOLE:def 3;
H6: ex X being non empty Subset of the carrier of I
    st X ∈ M & AmpS' ∩ X = {x} by H2,H1;
H7: x ∈ AmpS by H5,H6;
H8: AmpS ∩ X = {x}
    proof
    K: now let y be Any;
    M0: now assume M1: y ∈ {x};
    M2: y = x by M1,ENUMSET1:3;
    M3: x ∈ AmpS ∩ X by H5a,H7,BOOLE:def 3;
    M4: y ∈ AmpS ∩ X by M3,M2;
    thus y ∈ {x} implies y ∈ AmpS ∩ X by M4;
    end;  :: M0
    M5: now assume M6: y ∈ AmpS ∩ X;
    M7: y ∈ X by M6,BOOLE:def 3;
    M8: y ∈ AmpS by M6,BOOLE:def 3;
    consider zz being Element of AmpS' such that
    M9: y = zz &
```

```
        (ex X being non empty Subset of the carrier of I
         st X ∈ M & AmpS' ∩ X = {zz}) by M8;
      M10: y ∈ AmpS' by M9;
      M11: y ∈ AmpS' ∩ X by M7,M10,BOOLE:def 3;
      M12: y ∈ {x} by M11,H1;
      thus y ∈ AmpS ∩ X implies y ∈ {x} by M12;
      end;  :: M5
      thus y ∈ {x} iff y ∈ AmpS ∩ X by M0,M5;
      end;  :: K
      thus thesis by K,TARSKI:2;
      end;
    thus thesis by H7,H8;
    end;
```

◇

Macro referenced in scrap 15b.

⟨proof K6 55⟩ ≡

```
    proof
    H0: AmpS is non empty
      proof
      M0: Class (1.I) ∈ M by Defh2;
      consider x being Any such that
      M1: AmpS' ∩ Class (1.I) = {x} by K5,M0;
      M2: x ∈ {x} by ENUMSET1:4;
      M3: x ∈ AmpS' ∩ Class (1.I) by M2,M1;
      M4: x ∈ AmpS' by M3,BOOLE:def 3;
      M5: x ∈ AmpS by M4,M1,M0;
      thus thesis by M5;
      end;
    reconsider AmpS as non empty set by H0;
    H2: now let A be Any;
    H2a: now assume H3: A ∈ AmpS;
    H3a: A ∈ { x where x is Element of AmpS':
               ex X being non empty Subset of the carrier of I
               st X ∈ M & AmpS' ∩ X = { x }} by H3;
    consider x being Element of AmpS' such that
    H4: A = x &
      (ex X being non empty Subset of the carrier of I
       st X ∈ M & AmpS' ∩ X = {x}) by H3a;
    consider X being non empty Subset of the carrier of I
    such that H4a: X ∈ M & AmpS' ∩ X = {x} by H4;
    H5: x ∈ {x} by ENUMSET1:4;
    H6: x ∈ AmpS' ∩ X by H4a,H5;
    H7: x ∈ X by H6,BOOLE:def 3;
```

```
H8a: x ∈ the carrier of I by H7;
H8: A ∈ the carrier of I by H8a,H4;
thus (A ∈ AmpS) implies (A ∈ the carrier of I) by H8;
end;  :: H2a
thus (A ∈ AmpS) implies (A ∈ the carrier of I) by H2a;
end;
H10: AmpS c= the carrier of I by H2,TARSKI:def 3;
thus thesis by H10;
end;
```
◇

Macro referenced in scrap 15b.

⟨existence AmpleSet 56⟩ ≡

```
existence
proof
H0: now
⟨definition of A' 18a⟩
H2: (1.I) ∈ A'
    proof
    M1: (1.I) ∈ {(1.I)} by ENUMSET1:4;
    thus thesis by M1,BOOLE:def 2;
    end;
H2a: A' is non empty by H2;
reconsider A' as non empty set by H2a;
H3: for x being Element of A' holds x = (1.I) or x ∈ A
    proof
    let y be Element of A';
    M3: now per cases by BOOLE:def 2;

    case A: y ∈ {z where z is Element of A: z <> x};
    A1: ex zz being Element of A st y = zz & zz <> x by A;
    A2: y ∈ A by A1;
    thus y = (1.I) or y ∈ A by A2;

    case B: y ∈ {(1.I)};
    B1: y = (1.I) by B,ENUMSET1:3;
    thus y = (1.I) or y ∈ A by B1;

    end;  :: cases
    M4: y ∈ A' implies (y = (1.I) or y ∈ A) by M3;
    thus thesis by M4;
    end;
H4: A' is non empty Subset of the carrier of I
    proof
```

```
    M1: now let x be Any;
    M2: now
    assume M3: x ∈ A';
    M4: x ∈ the carrier of I
        proof
        M4a: now per cases by M3,H3;
        case A: x = (1.I);
                thus thesis by A;
        case B: x ∈ A;
                thus thesis by B;
        end;  :: cases
        thus thesis by M4a;
        end;
    thus x ∈ A' implies x ∈ the carrier of I by M4;
    end;  :: M2
    thus x ∈ A' implies x ∈ the carrier of I by M2;
    end;  :: M1
    M5: A' c= the carrier of I by M1,TARSKI:def 3;
    thus thesis by M5;
    end;
reconsider A' as non empty Subset of the carrier of I by H4;
H5: for a being Element of the carrier of I
    ex z being Element of A'
    st z is_associated_to a
    proof
    let a be Element of the carrier of I;
    M0: now per cases;

    case A: a is_associated_to (1.I);
    A1: (1.I) is_associated_to a by A,L2;
    thus ex z being Element of A' st z is_associated_to a
         by A1,H2;

    case B: a is_not_associated_to (1.I);
    consider z being Element of A such that
    B1: z is_associated_to a by Def8a;
    B3: z <> x
        proof
        assume M1: z = x;
        M2: z is_associated_to (1.I) by M1,H1;
        M3: a is_associated_to z by B1,L2;
        M4: a is_associated_to (1.I) by M3,M2,L2;
        thus thesis by M4,B;
        end;
```

```
    B4: z ∈ {zz where zz is Element of A : zz <> x}
        by B3;
    B5: z ∈ A' by B4,BOOLE:def 2;
    thus ex z being Element of A' st z is_associated_to a
         by B1,B5;
    end;  :: cases
    thus thesis by M0;
    end;
H6: for z,y being Element of A' holds
    z <> y implies z is_not_associated_to y
    proof
    let z,y be Element of A';
    assume M0: z <> y;
    M1: now per cases;

    case A: z = (1.I) & y = (1.I);
    thus thesis by A,M0;

    case B: z = (1.I) & y <> (1.I);
    B1: y ∈ A by B,H3;
    B2: not( y ∈ {(1.I)} ) by B,ENUMSET1:3;
    B4: y ∈ {zz where zz is Element of A: zz <> x}
        by B2,BOOLE:def 2;
    B5a: ex zz being Element of A st y = zz & zz <> x by B4;
    B5: y <> x by B5a;
    B6: x is_associated_to z by B,H1;
    assume B7: z is_associated_to y;
    B8: x is_associated_to y by B6,B7,L2;
    B9: for z1,z2 being Element of A holds z1 <> z2 implies
        z1 is_not_associated_to z2 by Def8a;
    B10: x is_not_associated_to y by B9,B5,B1;
    thus thesis by B10,B8;

    case C: z <> (1.I) & y = (1.I);
    C1: z ∈ A by C,H3;
    C2: not( z ∈ {(1.I)} ) by C,ENUMSET1:3;
    C4: z ∈ {zz where zz is Element of A: zz <> x}
        by C2,BOOLE:def 2;
    C5a: ex zz being Element of A st z = zz & zz <> x by C4;
    C5: z <> x by C5a;
    C6: x is_associated_to y by C,H1;
    C6a: y is_associated_to x by C6,L2;
    assume C7: z is_associated_to y;
    C8: z is_associated_to x by C6a,C7,L2;
```

```
        C9: for z1,z2 being Element of A holds z1 <> z2 implies
            z1 is_not_associated_to z2 by Def8a;
        C10: z is_not_associated_to x by C9,C5,C1;
        thus thesis by C10,C8;

        case D: z <> (1.I) & y <> (1.I);
        D1: z ∈ A by D,H3;
        D2: y ∈ A by D,H3;
        thus thesis by M0,D1,D2,Def8a;

        end;  :: cases
        thus thesis by M1;
        end;
    H7: A' is Am of I by H5,H6,Def8a;
    thus thesis by H2,H7;
    end;  :: H0
    thus thesis by H0;
    end;
    end;
```

◇

Macro referenced in scrap 17b.

⟨proof AMP 59a⟩ ≡

```
    proof
    let Amp be AmpleSet of I;
    H0: (1.I) ∈ Amp by Def8;
    H1: Amp is Am of I by Def8;
    H2: (for a being Element of the carrier of I
         ex z being Element of Amp
         st z is_associated_to a) &
        (for x,y being Element of Amp holds x <> y
         implies x is_not_associated_to y) by H1,Def8a;
    thus thesis by H0,H2;
    end;
```

◇

Macro referenced in scrap 18c.

⟨proof AMP5 59b⟩ ≡

```
    proof
    let Amp be AmpleSet of I;
    assume H0: Amp is_multiplicative;
    let x,y be Element of Amp;
    assume H1: y divides x & y <> (0.I);
```

```
M: now per cases;

case A: x <> (0.I);
consider d being Element of the carrier of I such that
H2: d = x/y;
H2a: x = yd by H2,H1,Def5;
consider d' being Element of Amp such that
H3: d' is_associated_to d by AMP;
H3a: d is_associated_to d' by H3,L2;
consider u being Element of the carrier of I such that
H4: u is_unit & du = d' by H3a,L11;
H5: ux = u(yd)  by H2a
      .= (yd)u
      .= y(du)  by VECTSP_1:def 16
      .= y(ud)
      .= yd'    by H4;
H5a: yd' ∈ Amp by H0,Def25;
H6: ux ∈ Amp by H5a,H5;
H7: x is_associated_to ux
    proof
    M1: x divides x by L1;
    M2: x divides ux by M1,L6a;
    M3: u divides (1.I) by H4,Def2;
    consider e being Element of the carrier of I such that
    M4: ue = (1.I) by M3,Def1;
    M5: (ux)e = e(ux)
              .= (eu)x  by VECTSP_1:def 16
              .= (1.I)x by M4
              .= x      by VECTSP_2:1;
    M6: ux divides x by M5,Def1;
    thus thesis by M2,M6,Def3;
    end;
H8: (1.I)x = x  by VECTSP_2:1
           .= ux by H7,H6,AMP1;
H9: u = (1.I) by H8,IDOM1,A;
H10: d' = du     by H4
       .= d(1.I) by H9
       .= d      by VECTSP_2:1;
thus thesis by H10,H2;

case B: x = (0.I);
consider d being Element of the carrier of I such that
M0: d = x/y;
M0a: x = yd by M0,H1,Def5;
```

```
M1: xy = (0.I)y  by B
      .= (0.I)   by VECTSP_2:26;
M1a: x = (0.I) by VECTSP_2:15,M1,H1;
M2: d = (0.I) by VECTSP_2:15,M1a,H1,M0a;
M3: (0.I) is Element of Amp by AMP0;
thus thesis by M0,M3,M2;

end;  :: cases
thus thesis by M;
end;
```
◇

Macro referenced in scrap 18c.

⟨proof AMP0 61a⟩ ≡

```
proof
let Amp be AmpleSet of I;
H0b: for a being Element of the carrier of I
     ex z being Element of Amp
     st z is_associated_to a by AMP;
consider A being Element of Amp such that
H0: A is_associated_to (0.I) by H0b;
H1: (0.I) divides A by H0,Def3;
consider D being Element of the carrier of I such that
H2: (0.I)D = A by H1,Def1;
H3: A = (0.I) by H2,VECTSP_2:26;
thus thesis by H3;
end;
```
◇

Macro referenced in scrap 18c.

⟨proof AMP1 61b⟩ ≡

```
proof
let x,y be Element of Amp;
assume H0: x is_associated_to y;
H1: now per cases;
case A: x = y;
thus x = y by A;
case B: x <> y;
B1: x is_not_associated_to y by B,AMP;
B2: contradiction by B1,H0;
thus x = y by B2;
end;  :: cases
thus thesis by H1;
```

```
end;
```
◇

Macro referenced in scrap 18c.

⟨correctness NF 62a⟩ ≡
```
existence
  proof
  K: now let Amp be AmpleSet of I;
  let x be Element of the carrier of I;
  consider z being Element of Amp such that
  H0: z is_associated_to x by AMP;
  thus ex zz being Element of the carrier of I st
    zz ∈ Amp & zz is_associated_to x by H0;
  end;  :: K
  thus thesis by K;
  end;
uniqueness
  proof
  let z1,z2 be Element of the carrier of I such that
  H0: z1 ∈ Amp & z1 is_associated_to x &
      z2 ∈ Amp & z2 is_associated_to x;
  H0a: z1 is Element of Amp &
       z2 is Element of Amp by H0;
  H1: x is_associated_to z2 by H0,L2;
  H2: z1 is_associated_to z2 by H0,H1,L2;
  H3: z1 = z2 by H0a,H2,AMP1;
  thus thesis by H3;
  end;
end;
```
◇

Macro referenced in scrap 19.

⟨proof NF1 62b⟩ ≡
```
proof
let Amp be AmpleSet of I;
H0: (1.I) is_associated_to (1.I) by L2;
H1: (1.I) ∈ Amp by Def8;
H2: NF((1.I),Amp) = (1.I) by H0,H1,Def20;
H3: (0.I) is_associated_to (0.I) by L2;
H4: (0.I) is Element of Amp by AMP0;
H5: NF((0.I),Amp) = (0.I) by H3,H4,Def20;
thus thesis by H2,H5;
end;
```
◇

Macro referenced in scrap 20a.

⟨proof NF3 63a⟩ ≡

```
proof
let Amp be AmpleSet of I;
let a be Element of the carrier of I;
K1: now assume H0: a ∈ Amp;
H1: a is_associated_to a by L2;
H2: a = NF(a,Amp) by H0,H1,Def20;
thus a ∈ Amp implies a = NF(a,Amp) by H2;
end;  :: K1
thus thesis by K1,Def20;
end;
```

◇

Macro referenced in scrap 20a.

B.4 GCD-Domains

⟨existence gcdDomain 63b⟩ ≡

```
existence
 proof
 consider F being strict Field;
 reconsider F as comRing;
 H1: F is domRing by VECTSP_2:13;
 reconsider F as domRing by H1;
 H2: now
 let x,y be Element of the carrier of F;
 H3: now per cases;

 case A: x <> (0.F) & y <> (0.F);
 A1: x = (1.F)  x by VECTSP_2:1;
 A2: (1.F) divides x by A1,Def1;
 A3: y = (1.F)  y by VECTSP_2:1;
 A4: (1.F) divides y by A3,Def1;
 A5: for zz being Element of the carrier of F
     st (zz divides x & zz divides y)
     holds (zz divides (1.F))
     proof
     let zz be Element of the carrier of F;
     assume M0: zz divides x & zz divides y;
     M1: now per cases;
     case AA: zz <> (0.F);
     consider zz' being Element of the carrier of F such that
```

```
    M11: zz  zz' = (1.F) by AA,VECTSP_1:def 20;
    thus zz divides (1.F) by M11,Def1;
    case AB: zz = (0.F);
    assume M12: zz divides x;
    consider d being Element of the carrier of F such that
    M13: zzd = x by M12,Def1;
    M14: x = zzd     by M13
          .= (0.F)d  by AB
          .= (0.F)   by VECTSP_2:26;
    M15: contradiction by M14,A;
    thus zz divides (1.F) by M15;
    end;  :: M1
    thus thesis by M0,M1;
    end;
thus ex z being Element of the carrier of F st
       z divides x &
       z divides y &
       (for zz being Element of the carrier of F
        st (zz divides x & zz divides y)
        holds (zz divides z)) by A2,A4,A5;

case B: x = (0.F);
B0: y divides y by L1;
B1: y  (0.F) = (0.F) by VECTSP_2:26;
B2: y divides (0.F) by B1,Def1;
B3: for z being Element of the carrier of F
    st (z divides (0.F) & z divides y)
    holds (z divides y);
thus ex z being Element of the carrier of F st
       z divides x &
       z divides y &
       (for zz being Element of the carrier of F
        st (zz divides x & zz divides y)
        holds (zz divides z)) by B,B0,B2,B3;

case C: y = (0.F);
C0: x divides x by L1;
C1: x(0.F) = (0.F) by VECTSP_2:26;
C2: x divides (0.F) by C1,Def1;
C3: for z being Element of the carrier of F
    st (z divides x & z divides (0.F))
    holds (z divides x);
thus ex z being Element of the carrier of F st
       z divides x &
```

```
            z divides y &
            (for zz being Element of the carrier of F
             st (zz divides x & zz divides y)
             holds (zz divides z)) by C,C0,C2,C3;

       end;  :: cases
       thus ex z being Element of the carrier of F st
              z divides x &
              z divides y &
              (for zz being Element of the carrier of F
               st (zz divides x & zz divides y)
               holds (zz divides z)) by H3;
       end;  :: H2
       H4: F is gcd-like by H2,Def7;
       thus thesis by H4;
       end;
```

◇

Macro referenced in scrap 20b.

⟨more gcd 65⟩ ≡

```
      theorem
      L0: for Amp being AmpleSet of I
          for a,b being Element of the carrier of I holds
          gcd(a,b,Amp) divides a & gcd(a,b,Amp) divides b by Def4;

      theorem
      L4: for Amp being AmpleSet of I
          for a,b,c being Element of the carrier of I holds
          c divides gcd(a,b,Amp) implies (c divides a & c divides b)
      proof
      let Amp be AmpleSet of I;
      let A,B,C be Element of the carrier of I;
      assume H1: C divides gcd(A,B,Amp);
      consider D being Element of the carrier of I such that
      H2: CD = gcd(A,B,Amp) by H1,Def1;
      H3: gcd(A,B,Amp) divides A by L0;
      consider E being Element of the carrier of I such that
      H4: gcd(A,B,Amp)E = A by H3,Def1;
      H5: C(DE) = (CD)E          by VECTSP_1:def 16
                .= gcd(A,B,Amp)E  by H2
                .= A              by H4;
      H6: C divides A by H5,Def1;
      H7: gcd(A,B,Amp) divides B by L0;
      consider E being Element of the carrier of I such that
```

```
H8: gcd(A,B,Amp)E = B by H7,Def1;
H9: C(DE) = (CD)E          by VECTSP_1:def 16
         .= gcd(A,B,Amp)E  by H2
         .= B              by H8;
H10: C divides B by H9,Def1;
thus thesis by H6,H10;
end;

theorem
L13: for Amp being AmpleSet of I
     for a,b being Element of the carrier of I
     holds gcd(a,b,Amp) = gcd(b,a,Amp)
proof
let Amp be AmpleSet of I;
let A,B be Element of the carrier of I;
consider D being Element of the carrier of I such that
H1: D = gcd(A,B,Amp);
H11: D ∈ Amp by Def4,H1;
H2: D divides B & D divides A by H1,L0;
H3: for z being Element of the carrier of I
    st (z divides B & z divides A)
    holds (z divides D) by H1,Def4;
H4: D = gcd(B,A,Amp) by H11,H2,H3,Def4;
thus gcd(A,B,Amp) = gcd(B,A,Amp) by H1,H4;
end;

theorem
GCD1: for Amp being AmpleSet of I
      for a being Element of the carrier of I holds
      gcd(a,(0.I),Amp) = NF(a,Amp) &
      gcd((0.I),a,Amp) = NF(a,Amp)
proof
let Amp be AmpleSet of I;
let A be Element of the carrier of I;
H0: NF(A,Amp)is_associated_to A by Def20;
H1: NF(A,Amp) divides A by H0,Def3;
H2: NF(A,Amp)(0.I) = (0.I) by VECTSP_2:26;
H3: NF(A,Amp) divides (0.I) by H2,Def1;
H4: for z being Element of the carrier of I
    st (z divides A & z divides (0.I))
    holds (z divides NF(A,Amp))
    proof
    let z be Element of the carrier of I;
    assume M0: z divides A & z divides (0.I);
```

```
    M1: A divides NF(A,Amp) by H0,Def3;
    thus thesis by M1,M0,L1;
    end;
H5: NF(A,Amp) ∈ Amp by Def20;
H6: gcd(A,(0.I),Amp) = NF(A,Amp) by H1,H3,H4,H5,Def4;
thus thesis by H6,L13;
end;

theorem
GCD0: for Amp being AmpleSet of I holds
      gcd((0.I),(0.I),Amp) = (0.I)
proof
let Amp being AmpleSet of I;
H2: gcd((0.I),(0.I),Amp) = NF((0.I),Amp) by GCD1;
H3: NF((0.I),Amp) = (0.I) by NF1;
thus thesis by H2,H3;
end;

theorem
GCD2: for Amp being AmpleSet of I
      for a being Element of the carrier of I holds
      gcd(a,(1.I),Amp) = (1.I) & gcd((1.I),a,Amp) = (1.I)
proof
let Amp be AmpleSet of I;
let A be Element of the carrier of I;
H0: (1.I) ∈ Amp by Def8;
H1: (1.I) divides (1.I) by L1;
H2: (1.I)A = A by VECTSP_2:1;
H3: (1.I) divides A by H2,Def1;
H4: for z being Element of the carrier of I
    st (z divides A & z divides (1.I))
    holds (z divides (1.I));
H5: gcd(A,(1.I),Amp) = (1.I) by H0,H1,H3,H4,Def4;
thus thesis by H5,L13;
end;

theorem
L12: for Amp being AmpleSet of I
     for a,b being Element of the carrier of I holds
     gcd(a,b,Amp) = (0.I) iff (a = (0.I) & b = (0.I))
proof
let Amp be AmpleSet of I;
let A,B be Element of the carrier of I;
H0: (A = (0.I) & B = (0.I)) implies (gcd(A,B,Amp) = (0.I))
```

```
    proof
    assume H0: A = (0.I) & B = (0.I);
    H3: gcd(A,B,Amp) = NF(A,Amp) by H0,GCD1;
    H4: NF(A,Amp) = (0.I) by H0,NF1;
    thus thesis by H4,H3;
    end;
K: now assume H1: gcd(A,B,Amp) = (0.I);
H2: (0.I) divides A & (0.I) divides B by H1,Def4;
consider D being Element of the carrier of I such that
H3: (0.I)D = A by H2,Def1;
H4: A = (0.I) by H3,VECTSP_2:26;
consider E being Element of the carrier of I such that
H5: (0.I)E = B by H2,Def1;
H6: B = (0.I) by H5,VECTSP_2:26;
thus (gcd(A,B,Amp) = (0.I)) implies (A = (0.I) & B = (0.I))
     by H4,H6;
end;
thus thesis by H0,K;
end;

theorem
L14: for Amp being AmpleSet of I
     for a,b,c being Element of the carrier of I holds
     (b is_associated_to c) implies
     ((gcd(a,b,Amp) is_associated_to gcd(a,c,Amp)) &
      (gcd(b,a,Amp) is_associated_to gcd(c,a,Amp)))
proof
let Amp be AmpleSet of I;
let A,B,C be Element of the carrier of I;
assume H1: B is_associated_to C;
H2: B divides C by H1,Def3;
H3: gcd(A,B,Amp) divides B by L0;
H4: gcd(A,B,Amp) divides C by H2,H3,L1;
H5: gcd(A,B,Amp) divides A by L0;
H6: gcd(A,B,Amp) divides gcd(A,C,Amp) by H4,H5,Def4;
H7: gcd(A,B,Amp) = gcd(B,A,Amp) by L13;
H8: gcd(A,C,Amp) = gcd(C,A,Amp) by L13;
H9: gcd(B,A,Amp) divides gcd(C,A,Amp) by H6,H7,H8;
H10: C divides B by H1,Def3;
H11: gcd(A,C,Amp) divides C by L0;
H12: gcd(A,C,Amp) divides B by H10,H11,L1;
H13: gcd(A,C,Amp) divides A by L0;
H14: gcd(A,C,Amp) divides gcd(A,B,Amp) by H13,H12,Def4;
H15: gcd(C,A,Amp) divides gcd(B,A,Amp) by H7,H8,H14;
```

```
H16: gcd(A,B,Amp) is_associated_to gcd(A,C,Amp)
     by H6,H14,Def3;
H17: gcd(B,A,Amp) is_associated_to gcd(C,A,Amp)
     by H9,H15,Def3;
thus thesis by H16,H17;
end;
```

◇

Macro referenced in scrap 21.

B.5 Proof of the Basic Properties

⟨proof T0 69a⟩ ≡

```
proof
let Amp be AmpleSet of I;
let A,B,C be Element of the carrier of I;
consider D being Element of the carrier of I such that
H1: D = gcd(gcd(A,B,Amp),C,Amp);
consider E being Element of the carrier of I such that
H2: E = gcd(A,gcd(B,C,Amp),Amp);
H3: D divides gcd(A,B,Amp) & D divides C by H1,L0;
H4: D divides A & D divides B & D divides C by L4,H3;
H5: D divides A & D divides gcd(B,C,Amp) by H4,Def4;
H6: D divides E by H2,H5,Def4;
H7: E divides gcd(B,C,Amp) & E divides A by H2,L0;
H8: E divides B & E divides C & E divides A by L4,H7;
H9: E divides C & E divides gcd(A,B,Amp) by H8,Def4;
H10: E divides D by H1,H9,Def4;
H11: D is_associated_to E by H6,H10,Def3;
H12: D is Element of Amp by H1,Def4;
H13: E is Element of Amp by H2,Def4;
H14: D = E by H11,H12,H13,AMP;
thus thesis by H1,H2,H14;
end;
```

◇

Macro referenced in scrap 23b.

⟨proof T1 69b⟩ ≡

```
proof
let Amp be AmpleSet of I;
let A,B,C be Element of the carrier of I;
M: now per cases;
```

```
case A: C <> (0.I);
consider D being Element of the carrier of I such that
H1: D = gcd(A,B,Amp);
K: now per cases;

case A1: D <> (0.I);
consider E being Element of the carrier of I such that
H2: E = gcd((AC),(BC),Amp);
H3: D divides A & D divides B by H1,L0;
H4: CD divides AC & CD divides BC by H3,L3;
H5: CD divides gcd((AC),(BC),Amp) by H4,Def4;
H6: CD divides E by H5,H2;
consider F being Element of the carrier of I such that
H7: E = (CD)F by H6,Def1;
H8: E divides AC by H2,L0;
H9: E divides BC by H2,L0;
H10: ((CD)F) divides AC by H8,H7;
H11: ((CD)F) divides BC by H9,H7;
H12: (DF) divides A & (DF) divides B
     proof
     consider G being Element of the carrier of I such that
     M1: ((CD)F)G = AC by H10,Def1;
     M2: (C(DF))G = ((CD)F)G  by VECTSP_1:def 16
                 .= CA        by M1;
     M3: (C(DF)) divides C  A by M2,Def1;
     M4: (DF) divides A by M3,L9,A;
     consider G being Element of the carrier of I such that
     M5: ((CD)F)G = BC by H11,Def1;
     M6: (C(DF))G = ((CD)F)G  by VECTSP_1:def 16
                 .= CB        by M5;
     M7: (C(DF)) divides CB by M6,Def1;
     M8: (DF) divides B by M7,L9,A;
     thus thesis by M4,M8;
     end;
H13: DF divides gcd(A,B,Amp) by H12,Def4;
H14: DF divides D by H13,H1;
H15: F divides (1.I)
     proof
     M1: D = D(1.I) by VECTSP_2:1;
     M2: DF divides D(1.I) by M1,H14;
     thus thesis by M2,L9,A1;
     end;
H16: F is_unit by H15,Def2;
H18: ex f being Element of the carrier of I
```

```
     st (f is_unit & (CD)f = E) by H7,H16;
H19: (CD) is_associated_to E by H18,L11;
H20: E is_associated_to (CD) by H19,L2;
thus gcd((AC),(BC),Amp) is_associated_to (Cgcd(A,B,Amp))
     by H20,H1,H2;

case A2: D = (0.I);
N1: gcd(A,B,Amp) = (0.I) by A2,H1;
N2: A = (0.I) & B = (0.I) by N1,L12;
N3: Cgcd(A,B,Amp) = (0.I) by N1,VECTSP_2:26;
N4:   gcd((AC),(BC),Amp)
    = gcd(((0.I)C),((0.I)C),Amp)  by N2
   .= gcd((0.I),((0.I)C),Amp)     by VECTSP_2:26
   .= gcd((0.I),(0.I),Amp)        by VECTSP_2:26
   .= (0.I)                       by GCD0
   .= Cgcd(A,B,Amp)               by N3;
N5:   gcd((AC),(BC),Amp)(1.I)
    = gcd((AC),(BC),Amp)          by VECTSP_2:1
   .= Cgcd(A,B,Amp)               by N4;
N6: gcd((AC),(BC),Amp) divides Cgcd(A,B,Amp) by N5,Def1;
N7:   (Cgcd(A,B,Amp))(1.I)
    = Cgcd(A,B,Amp)    by VECTSP_2:1
   .= gcd((AC),(BC),Amp) by N4;
N8: Cgcd(A,B,Amp) divides gcd((AC),(BC),Amp) by N7,Def1;
thus gcd((AC),(BC),Amp) is_associated_to (Cgcd(A,B,Amp))
     by Def3,N6,N8;

end;  ::cases K
thus gcd((AC),(BC),Amp) is_associated_to (Cgcd(A,B,Amp))
     by K;

case B: C = (0.I);
H1: AC = (0.I) by B,VECTSP_2:26;
H2: BC = (0.I) by B,VECTSP_2:26;
H3:   gcd((AC),(BC),Amp)
    = gcd((0.I),(0.I),Amp)  by H1,H2
   .= (0.I)                 by GCD0
   .= (0.I)gcd(A,B,Amp)     by VECTSP_2:26
   .= Cgcd(A,B,Amp)         by B;
H4:   gcd((AC),(BC),Amp)(1.I)
    = gcd((AC),(BC),Amp)  by VECTSP_2:1
    .= Cgcd(A,B,Amp)      by H3;
H5: gcd((AC),(BC),Amp) divides (Cgcd(A,B,Amp)) by H4,Def1;
H6:   (Cgcd(A,B,Amp))(1.I)
```

```
      = Cgcd(A,B,Amp)       by VECTSP_2:1
     .= gcd((AC),(BC),Amp)  by H3;
   H7: (Cgcd(A,B,Amp)) divides gcd((AC),(BC),Amp) by H6,Def1;
   thus gcd((AC),(BC),Amp) is_associated_to (Cgcd(A,B,Amp))
        by H5,H7,Def3;

   end; ::cases M
   thus thesis by M;
   end;
```
◇

Macro referenced in scrap 23b.

⟨proof T3 72⟩ ≡

```
   proof
   let Amp be AmpleSet of I;
   let A,B,C be Element of the carrier of I;
   assume H0: ((C = gcd(A,B,Amp)) & (C <> (0.I)));
   consider A1 being Element of the carrier of I such that
   H1: A1 = A/C;
   consider B1 being Element of the carrier of I such that
   H2: B1 = B/C;
   M1: C divides A by Def4,H0;
   H3: A1C = A by H1,Def5,M1,H0;
   M2: C divides B by Def4,H0;
   H4: B1C = B by H2,Def5,M2,H0;
   H5: gcd(A,B,Amp) = gcd((A1C),(B1C),Amp) by H3,H4;
   H6: gcd((A1C),(B1C),Amp) is_associated_to (Cgcd(A1,B1,Amp))
       by T1;
   H7: C is_associated_to (Cgcd(A1,B1,Amp)) by H0,H5,H6;
   M3: (C(1.I)) is_associated_to (Cgcd(A1,B1,Amp))
        by H7,VECTSP_2:1;
   H8: (1.I) is_associated_to gcd(A1,B1,Amp) by M3,L15,H0;
   H9: gcd(A1,B1,Amp) is_associated_to (1.I) by H8,L2;
   H10: gcd(A1,B1,Amp) is Element of Amp by Def4;
   H11: (1.I) is Element of Amp by Def8;
   H12: gcd(A1,B1,Amp) = (1.I) by H9,H10,H11,AMP;
   H13: ((C = gcd(A,B,Amp)) & (C <> (0.I))) implies
        (gcd(A1,B1,Amp) = (1.I)) by H12;
   H14: ((C = gcd(A,B,Amp)) & (C <> (0.I))) implies
        (gcd((A/C),(B/C),Amp) = (1.I)) by H1,H2,H13;
   thus thesis by H0,H14;
   end;
```
◇

Macro referenced in scrap 23b.

⟨proof T4 73⟩ ≡

```
proof
let Amp be AmpleSet of I;
let A,B,C be Element of the carrier of I;
consider D being Element of the carrier of I such that
H1: D = gcd(A,C,Amp);
H2: D divides A & D divides C by H1,Def4;
H2a: D divides C by H2;
H2b: D is Element of Amp by H1,Def4;
consider E being Element of the carrier of I such that
H3: DE = A by H2,Def1;
consider F being Element of the carrier of I such that
H4: DF = C by H2,Def1;
H5: D divides (A + (BC))
    proof
    M1: D(E+(FB)) = (DE) + (D(FB))  by VECTSP_2:1
                .= (DE) + ((DF)B)  by VECTSP_1:def 16
                .= A + (CB)        by H3,H4
                .= A + (BC);
    thus thesis by M1,Def1;
    end;
H6: for z being Element of the carrier of I
    st (z divides (A + (BC)) & (z divides C))
    holds (z divides D)
    proof
    let Z be Element of the carrier of I;
    assume M1: Z divides (A + (BC)) & (Z divides C);
    M1a: (Z divides C) by M1;
    consider X being Element of the carrier of I such that
    M2: ZX = C by M1,Def1;
    consider Y being Element of the carrier of I such that
    M3: ZY = A + (BC) by M1,Def1;
    M4: Z(Y+(-(BX))) = (ZY) + (Z(-(BX)))    by VECTSP_2:1
                    .= (ZY) + (Z(-(XB)))
                    .= (ZY) + (-(Z(XB)))    by VECTSP_2:28
                    .= (ZY) + (-((ZX)B))    by VECTSP_1:def 16
                    .= (A + (BC)) + (-(CB)) by M2,M3
                    .= A + ((BC) + (-(CB))) by VECTSP_2:1
                    .= A + ((BC) + (-(BC)))
                    .= A + (0.I)            by VECTSP_2:1
                    .= A                    by VECTSP_2:1;
    M5: Z divides A by M4,Def1;
```

```
    M6: Z divides D by M1a,M5,H1,Def4;
    thus thesis by M6;
    end;
  H7: D = gcd((A + (B  C)),C,Amp) by H2a,H2b,H5,H6,Def4;
  thus thesis by H1,H7;
  end;
```
◇

Macro referenced in scrap 23b.

⟨proof H9 74a⟩ ≡

```
  proof
  M1: A = A(1.I) by VECTSP_2:1;
  M7: A is_associated_to A by L2;
  M2: A is_associated_to (A(1.I)) by M1,M7;
  M3: gcd(A,(AC),Amp) is_associated_to
      gcd((A(1.I)),(AC),Amp) by M2,L14;
  M4: gcd((A(1.I)),(AC),Amp)
      is_associated_to (Agcd((1.I),C,Amp)) by T1;
  M5: Agcd((1.I),C,Amp) = A(1.I)  by GCD2
                        .= A      by VECTSP_2:1;
  M6: gcd((A(1.I)),(AC),Amp) is_associated_to A by M5,M4;
  thus thesis by M6,M3,L2;
  end;
```
◇

Macro referenced in scrap 25a.

B.6 Proof of the Theorems

⟨proof HEN1 2 74b⟩ ≡

```
  H13: gcd((r1s)+(s1r),r,Amp) = gcd(r1s,r,Amp) by T4;
  H14: gcd(r,r1,Amp) = (1.I)
       ⟨proof H14 76a⟩
  H15: gcd(r,r1s,Amp) = gcd(r,s,Amp) by H14,T2;
  H16: gcd(r,s,Amp) = (1.I) by H4,H5,H2,K,T3;
  H17:   gcd((r1s)+(s1r),r,Amp)
       = gcd(r1s,r,Amp)     by H13
      .= gcd(r,r1s,Amp)     by L13
      .= gcd(r,s,Amp)          by H15
      .= (1.I)               by H16;
  H18:   gcd((r1s)+(s1r),dr,Amp)
       = gcd((r1s)+(s1r),d,Amp) by H17,T2;
  H19:   gcd((r1s)+(s1r),r2s,Amp)
       = gcd((r1s)+(s1r),d,Amp)
```

```
      by H12,H18;
H20: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         r2(s2/gcd(r2,s2,Amp)),Amp) =
      gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         gcd(r2,s2,Amp),Amp)
      by H19,H4,H5,H2;
thus thesis by H20;
end;
```

◇

Macro referenced in scrap 27.

⟨proof H7 75a⟩ ≡

```
proof
M1: gcd(s,s1,Amp) divides s1 by Def4;
M2: gcd(s,s1,Amp) divides s by Def4;
consider e being Element of the carrier of I such that
M3: gcd(s,s1,Amp)e = s by M2,Def1;
M4:   gcd(s,s1,Amp)(ed)
   = (gcd(s,s1,Amp)e)d  by VECTSP_1:def 16
  .= sd                 by M3
  .= s2                    by K,H2a,H5,Def5;
M5: gcd(s,s1,Amp) divides s2 by M4,Def1;
M6: gcd(s,s1,Amp) divides gcd(s1,s2,Amp) by M1,M5,Def4;
M7: gcd(s,s1,Amp) divides (1.I) by M6,H1;
M8: (1.I)gcd(s,s1,Amp) = gcd(s,s1,Amp) by VECTSP_2:1;
M9: (1.I) divides gcd(s,s1,Amp) by M8,Def1;
M10: gcd(s,s1,Amp) is_associated_to (1.I) by M7,M9,Def3;
M11: gcd(s,s1,Amp) is Element of Amp by Def4;
M12: (1.I) is Element of Amp by Def8;
thus thesis by M10,M11,M12,AMP;
end;
```

◇

Macro referenced in scrap 27.

⟨proof H11 75b⟩ ≡

```
proof
H0: d divides d by L1;
H0a: d divides (dr2) by L6;
H1: r2s = ((1.I)r2)s  by VECTSP_2:1
       .= ((d/d)r2)s  by K,L7
       .= ((dr2)/d)s  by K,H0,H0a,L8
       .= (d(r2/d))s  by K,H2b,H0a,L8
       .= (dr)s       by H4
```

```
        .= s(dr);
    thus thesis by H1;
    end;
```
◇

Macro referenced in scrap 27.

⟨proof H14 76a⟩ ≡

```
    proof
    M1: gcd(r,r1,Amp) divides r1 by Def4;
    M2: gcd(r,r1,Amp) divides r by Def4;
    consider e being Element of the carrier of I such that
    M3: gcd(r,r1,Amp)e = r by M2,Def1;
    M4:   gcd(r,r1,Amp)(ed)
        = (gcd(r,r1,Amp)e)d  by VECTSP_1:def 16
       .= rd                 by M3
       .= r2                 by K,H2b,H4,Def5;
    M5: gcd(r,r1,Amp) divides r2 by M4,Def1;
    M6: gcd(r,r1,Amp) divides gcd(r1,r2,Amp) by M1,M5,Def4;
    M7: gcd(r,r1,Amp) divides (1.I) by M6,H1;
    M8: (1.I)gcd(r,r1,Amp) = gcd(r,r1,Amp) by VECTSP_2:1;
    M9: (1.I) divides gcd(r,r1,Amp) by M8,Def1;
    M10: gcd(r,r1,Amp) is_associated_to (1.I) by M7,M9,Def3;
    M11: gcd(r,r1,Amp) is Element of Amp by Def4;
    M12: (1.I) is Element of Amp by Def8;
    thus thesis by M10,M11,M12,AMP;
    end;
```
◇

Macro referenced in scrap 74b.

⟨proof H27 76b⟩ ≡

```
    proof
    H10: d1 divides r1 by H2,Def4;
    H11: r1'd1 = r1 by H4,H6,H10,Def5;
    H12: r1' divides r1 by H11,Def1;
    H13: d2 divides r2 by H3,Def4;
    H14: r2'd2 = r2 by H5,H8,H13,Def5;
    H15: r2' divides r2 by H14,Def1;
    H16: gcd(r1',r2',Amp) divides r1' by Def4;
    H17: gcd(r1',r2',Amp) divides r2' by Def4;
    H18: gcd(r1',r2',Amp) divides r1 by H16,H12,L1;
    H19: gcd(r1',r2',Amp) divides r2 by H17,H15,L1;
    H20: gcd(r1',r2',Amp) divides gcd(r1,r2,Amp) by H18,H19,Def4;
    H21: gcd(r1',r2',Amp) divides (1.I) by H20,H1;
```

```
H22: (1.I)gcd(r1',r2',Amp) = gcd(r1',r2',Amp) by VECTSP_2:1;
H23: (1.I) divides gcd(r1',r2',Amp) by H22,Def1;
H24: gcd(r1',r2',Amp) is_associated_to (1.I) by H21,H23,Def3;
H25: gcd(r1',r2',Amp) is Element of Amp by Def4;
H26: (1.I) is Element of Amp by Def8;
H27a: gcd(r1',r2',Amp) = (1.I) by H24,H25,H26,AMP;
thus thesis by H27a,L13;
end;
```
◇

Macro referenced in scrap 28.

⟨proof H45 77a⟩ ≡

```
proof
H28: d1 divides s2 by H2,Def4;
H29: s2'd1 = s2 by H4,H9,H28,Def5;
H30: s2' divides s2 by H29,Def1;
H31: d2 divides s1 by H3,Def4;
H32: s1'd2 = s1 by H5,H7,H31,Def5;
H33: s1' divides s1 by H32,Def1;
H34: gcd(s1',s2',Amp) divides s1' by Def4;
H35: gcd(s1',s2',Amp) divides s2' by Def4;
H36: gcd(s1',s2',Amp) divides s1 by H34,H33,L1;
H37: gcd(s1',s2',Amp) divides s2 by H35,H30,L1;
H38: gcd(s1',s2',Amp) divides gcd(s1,s2,Amp) by H37,H36,Def4;
H39: gcd(s1',s2',Amp) divides (1.I) by H38,H1;
H40: (1.I)gcd(s1',s2',Amp) = gcd(s1',s2',Amp) by VECTSP_2:1;
H41: (1.I) divides gcd(s1',s2',Amp) by H40,Def1;
H42: gcd(s1',s2',Amp) is_associated_to (1.I) by H39,H41,Def3;
H43: gcd(s1',s2',Amp) is Element of Amp by Def4;
H44: (1.I) is Element of Amp by Def8;
thus thesis by H42,H43,H44,AMP;
end;
```
◇

Macro referenced in scrap 28.

B.7 Proofs of Correctness

⟨more def 77b⟩ ≡

```
definition
let I be gcdDomain;
let Amp be AmpleSet of I;
let x,y be Element of the carrier of I;
  pred x,y are_canonical_wrt Amp means :Def10a:
```

```
  gcd(x,y,Amp) = (1.I);
end;

theorem
CAN: for Amp,Amp' being AmpleSet of I
     for x,y being Element of the carrier of I holds
     x,y are_canonical_wrt Amp iff x,y are_canonical_wrt Amp'
proof
let Amp,Amp' being AmpleSet of I;
let x,y be Element of the carrier of I;
K1: now assume H0: x,y are_canonical_wrt Amp;
H1: gcd(x,y,Amp) = (1.I) by H0,Def10a;
H2: for z being Element of the carrier of I
    st (z divides x & z divides y)
    holds (z divides (1.I)) by H1,Def4;
H3: (1.I)x = x by VECTSP_2:1;
H4: (1.I)y = y by VECTSP_2:1;
H5: (1.I) divides x by H3,Def1;
H6: (1.I) divides y by H4,Def1;
H7: (1.I) ∈ Amp' by Def8;
H8: gcd(x,y,Amp') = (1.I) by H2,H5,H6,H7,Def4;
thus x,y are_canonical_wrt Amp' by H8,Def10a;
end; :: K1
K2: now assume H0: x,y are_canonical_wrt Amp';
H1: gcd(x,y,Amp') = (1.I) by H0,Def10a;
H2: for z being Element of the carrier of I
    st (z divides x & z divides y)
    holds (z divides (1.I)) by H1,Def4;
H3: (1.I)x = x by VECTSP_2:1;
H4: (1.I)y = y by VECTSP_2:1;
H5: (1.I) divides x by H3,Def1;
H6: (1.I) divides y by H4,Def1;
H7: (1.I) ∈ Amp by Def8;
H8: gcd(x,y,Amp) = (1.I) by H2,H5,H6,H7,Def4;
thus x,y are_canonical_wrt Amp by H8,Def10a;
end; :: K2
thus thesis by K1,K2;
end;

theorem
CAN1: for Amp being AmpleSet of I
      for x,y being Element of the carrier of I holds
      x canonical y implies gcd(x,y,Amp) = (1.I)
proof
```

```
let Amp be AmpleSet of I;
let x,y be Element of the carrier of I;
assume H0: x canonical y;
consider Amp' being AmpleSet of I such that
H1: gcd(x,y,Amp') = (1.I) by H0,Def10;
H2: x,y are_canonical_wrt Amp' by H1,Def10a;
H3: x,y are_canonical_wrt Amp by H2,CAN;
H4: gcd(x,y,Amp) = (1.I) by H3,Def10a;
thus thesis by H4;
end;
```

◇

Macro referenced in scrap 29b.

⟨consistency add1 79⟩ ≡

```
consistency
proof
V1: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I) by A,CAN1;
K02: r1 = (0.I) & gcd(r2,s2,Amp) = (1.I) implies
     for z being Element of the carrier of I
     holds z = s1 iff z = (r1s2) + (r2s1)
     proof
     assume H0: r1 = (0.I) & gcd(r2,s2,Amp) = (1.I);
     let z be Element of the carrier of I;
     H1: r2 = NF(r2,Amp)        by A
           .= gcd((0.I),r2,Amp) by GCD1
           .= gcd(r1,r2,Amp)    by H0
           .= (1.I)             by V1;
     H2: (r1s2) + (r2s1) = (0.I) + (r2s1) by H0,VECTSP_2:26
                         .= r2s1          by VECTSP_2:1
                         .= (1.I)s1       by H1
                         .= s1            by VECTSP_2:1;
     H3: z = s1 iff z = (r1s2) + (r2s1) by H2;
     thus thesis by H3;
     end;
K03: r1 = (0.I) & (r1(s2/gcd(r2,s2,Amp))) +
                  (s1(r2/gcd(r2,s2,Amp))) = (0.I)
     implies for z being Element of the carrier of I
     holds z = s1 iff z = (0.I)
     proof
     assume H0: r1 = (0.I) & (r1(s2/gcd(r2,s2,Amp))) +
                             (s1(r2/gcd(r2,s2,Amp))) = (0.I);
     let z be Element of the carrier of I;
     H1: r2 = NF(r2,Amp)        by A
           .= gcd((0.I),r2,Amp) by GCD1
```

```
        .= gcd(r1,r2,Amp)    by H0
        .= (1.I)             by V1;
    H2: gcd(r2,s2,Amp) = gcd((1.I),s2,Amp)  by H1
                       .= (1.I)             by GCD2;
    H2a: (1.I) <> (0.I) by VECTSP_1: def 21;
    H3:   (0.I)
        = (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
          by H0
       .= ((0.I)(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
          by H0
       .= (0.I)+(s1(r2/gcd(r2,s2,Amp))) by VECTSP_2:26
       .= s1(r2/gcd(r2,s2,Amp)) by VECTSP_2:1
       .= s1((1.I)/gcd(r2,s2,Amp)) by H1
       .= s1((1.I)/(1.I)) by H2
       .= s1(1.I) by L7,H2a
       .= s1 by VECTSP_2:1;
    H4: z = s1 iff z = (0.I) by H3;
    thus thesis by H4;
    end;
K12: s1 = (0.I) & gcd(r2,s2,Amp) = (1.I) implies
    for z being Element of the carrier of I
    holds z = r1 iff z = (r1s2) + (r2s1)
    proof
    assume H0: s1 = (0.I) & gcd(r2,s2,Amp) = (1.I);
    let z be Element of the carrier of I;
    H1: s2 = NF(s2,Amp)        by A
          .= gcd((0.I),s2,Amp) by GCD1
          .= gcd(s1,s2,Amp)    by H0
          .= (1.I)             by V1;
    H2: (r1s2) + (r2s1) = (r1s2) + (0.I) by H0,VECTSP_2:26
                        .= r1s2          by VECTSP_2:1
                        .= r1(1.I)       by H1
                        .= r1            by VECTSP_2:1;
    H3: z = r1 iff z = (r1s2) + (r2s1) by H2;
    thus thesis by H3;
    end;
K13: s1 = (0.I) & (r1(s2/gcd(r2,s2,Amp))) +
                 (s1(r2/gcd(r2,s2,Amp))) = (0.I)
    implies for z being Element of the carrier of I
    holds z = r1 iff z = (0.I)
    proof
    assume H0: s1 = (0.I) & (r1(s2/gcd(r2,s2,Amp))) +
                            (s1(r2/gcd(r2,s2,Amp))) = (0.I);
    let z be Element of the carrier of I;
```

```
      H1: s2 = NF(s2,Amp)        by A
            .= gcd((0.I),s2,Amp) by GCD1
            .= gcd(s1,s2,Amp)    by H0
            .= (1.I)             by V1;
      H2: gcd(r2,s2,Amp) = gcd(r2,(1.I),Amp)  by H1
                         .= (1.I)             by GCD2;
      H2a: (1.I) <> (0.I) by VECTSP_1: def 21;
      H3:   (0.I)
          = (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
            by H0
         .= (r1(s2/gcd(r2,s2,Amp)))+((0.I)(r2/gcd(r2,s2,Amp)))
            by H0
         .= (r1(s2/gcd(r2,s2,Amp)))+(0.I) by VECTSP_2:26
         .= r1(s2/gcd(r2,s2,Amp)) by VECTSP_2:1
         .= r1((1.I)/gcd(r2,s2,Amp)) by H1
         .= r1((1.I)/(1.I)) by H2
         .= r1(1.I) by L7,H2a
         .= r1 by VECTSP_2:1;
      H4: z = r1 iff z = (0.I) by H3;
      thus thesis by H4;
      end;
    K23: gcd(r2,s2,Amp) = (1.I) &
      (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))) = (0.I)
      implies for z being Element of the carrier of I
      holds z = (r1s2) + (r2s1) iff z = (0.I)
      proof
      assume H0: gcd(r2,s2,Amp) = (1.I) &
      (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))) = (0.I);
      let z be Element of the carrier of I;
      H1:   (0.I)
          = (r1(s2/gcd(r2,s2,Amp))) + (s1(r2/gcd(r2,s2,Amp)))
            by H0
         .= (r1(s2/(1.I))) + (s1(r2/gcd(r2,s2,Amp))) by H0
         .= (r1(s2/(1.I))) + (s1(r2/(1.I))) by H0
         .= (r1s2) + (s1(r2/(1.I))) by L7a
         .= (r1s2) + (s1r2) by L7a;
      H2: z = (r1s2) + (r2s1) iff z = (0.I) by H1;
      thus thesis by H2;
      end;
    thus thesis by K02,K03,K12,K13,K23;
    end;
```

◇

Macro referenced in scrap 30.

⟨consistency add2 81⟩ ≡

```
consistency
proof
V1: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I) by A,CAN1;
K01: r1 = (0.I) & s1 = (0.I) implies
     for z being Element of the carrier of I
     holds z = s2 iff z = r2
     proof
     assume H0: r1 = (0.I) & s1 = (0.I);
     let z be Element of the carrier of I;
     H1: r2 = NF(r2,Amp)        by A
           .= gcd((0.I),r2,Amp) by GCD1
           .= gcd(r1,r2,Amp)    by H0
           .= (1.I)             by V1;
     H2: s2 = NF(s2,Amp)        by A
           .= gcd((0.I),s2,Amp) by GCD1
           .= gcd(s1,s2,Amp)    by H0
           .= (1.I)             by V1;
     H3: z = s2 iff z = r2 by H1,H2;
     thus thesis by H3;
     end;
K02: r1 = (0.I) & gcd(r2,s2,Amp) = (1.I) implies
     for z being Element of the carrier of I
     holds z = s2 iff z = r2s2
     proof
     assume H0: r1 = (0.I) & gcd(r2,s2,Amp) = (1.I);
     let z be Element of the carrier of I;
     H1: r2 = NF(r2,Amp)        by A
           .= gcd((0.I),r2,Amp) by GCD1
           .= gcd(r1,r2,Amp)    by H0
           .= (1.I)             by V1;
     H2: r2s2 = (1.I)s2 by H1
              .= s2         by VECTSP_2:1;
     H3: z = s2 iff z = r2s2 by H2;
     thus thesis by H3;
     end;
K03: r1 = (0.I) & (r1(s2/gcd(r2,s2,Amp))) +
                  (s1(r2/gcd(r2,s2,Amp))) = (0.I)
     implies  for z being Element of the carrier of I
     holds z = s2 iff z = (1.I)
     proof
     assume H0: r1 = (0.I) & (r1(s2/gcd(r2,s2,Amp))) +
                             (s1(r2/gcd(r2,s2,Amp))) = (0.I);
     let z be Element of the carrier of I;
```

```
      H1: r2 = NF(r2,Amp)        by A
            .= gcd((0.I),r2,Amp) by GCD1
            .= gcd(r1,r2,Amp)    by H0
            .= (1.I)             by V1;
      H2: gcd(r2,s2,Amp) = gcd((1.I),s2,Amp)  by H1
                         .= (1.I)             by GCD2;
      H2a: (1.I) <> (0.I) by VECTSP_1: def 21;
      H3:   (0.I)
          = (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
            by H0
         .= ((0.I)(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
            by H0
         .= (0.I)+(s1(r2/gcd(r2,s2,Amp))) by VECTSP_2:26
         .= s1(r2/gcd(r2,s2,Amp)) by VECTSP_2:1
         .= s1((1.I)/gcd(r2,s2,Amp)) by H1
         .= s1((1.I)/(1.I)) by H2
         .= s1(1.I) by L7,H2a
         .= s1 by VECTSP_2:1;
      H4: s2 = NF(s2,Amp)        by A
            .= gcd((0.I),s2,Amp) by GCD1
            .= gcd(s1,s2,Amp)    by H3
            .= (1.I)             by V1;
      thus thesis by H4;
      end;
K12:  s1 = (0.I) & gcd(r2,s2,Amp) = (1.I) implies
      for z being Element of the carrier of I
      holds z = r2 iff z = r2s2
      proof
      assume H0: s1 = (0.I) & gcd(r2,s2,Amp) = (1.I);
      let z be Element of the carrier of I;
      H1: s2 = NF(s2,Amp)        by A
            .= gcd((0.I),s2,Amp) by GCD1
            .= gcd(s1,s2,Amp)    by H0
            .= (1.I)             by V1;
      H2: r2s2 = r2(1.I) by H1
              .= r2    by VECTSP_2:1;
      H3: z = r2 iff z = r2  s2 by H2;
      thus thesis by H3;
      end;
K13:  s1 = (0.I) & (r1(s2/gcd(r2,s2,Amp))) +
                   (s1(r2/gcd(r2,s2,Amp))) = (0.I)
      implies for z being Element of the carrier of I
      holds z = r2 iff z = (1.I)
      proof
```

```
      assume H0: s1 = (0.I) & (r1(s2/gcd(r2,s2,Amp))) +
                              (s1(r2/gcd(r2,s2,Amp))) = (0.I);
      let z be Element of the carrier of I;
      H1: s2 = NF(s2,Amp)        by A
            .= gcd((0.I),s2,Amp) by GCD1
            .= gcd(s1,s2,Amp)    by H0
            .= (1.I)             by V1;
      H2: gcd(r2,s2,Amp) = gcd(r2,(1.I),Amp)  by H1
                         .= (1.I)             by GCD2;
      H2a: (1.I) <> (0.I) by VECTSP_1: def 21;
      H3:   (0.I)
          = (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
            by H0
         .= (r1(s2/gcd(r2,s2,Amp)))+((0.I)(r2/gcd(r2,s2,Amp)))
            by H0
         .= (r1(s2/gcd(r2,s2,Amp)))+(0.I) by VECTSP_2:26
         .= r1(s2/gcd(r2,s2,Amp)) by VECTSP_2:1
         .= r1((1.I)/gcd(r2,s2,Amp)) by H1
         .= r1((1.I)/(1.I)) by H2
         .= r1(1.I) by L7,H2a
         .= r1 by VECTSP_2:1;
      H4: r2 = NF(r2,Amp)        by A
            .= gcd((0.I),r2,Amp) by GCD1
            .= gcd(r1,r2,Amp)    by H3
            .= (1.I)             by V1;
      H5: z = r2 iff z = (1.I) by H4;
      thus thesis by H5;
      end;
K23:  gcd(r2,s2,Amp) = (1.I) &
      (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))) = (0.I)
      implies for z being Element of the carrier of I
      holds z = r2s2 iff z = (1.I)
      proof
      assume H0: gcd(r2,s2,Amp) = (1.I) &
      (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))) = (0.I);
      let z be Element of the carrier of I;
      H1:   (0.I)
          = (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
            by H0
         .= (r1(s2/(1.I)))+(s1(r2/gcd(r2,s2,Amp))) by H0
         .= (r1(s2/(1.I)))+(s1(r2/(1.I))) by H0
         .= (r1s2)+(s1(r2/(1.I))) by L7a
         .= (r1s2)+(s1r2) by L7a;
      H2: r1s2 = (0.I) - (s1r2) by H1,VECTSP_2:22
```

```
          .= -(s1r2)        by VECTSP_2:23;
      H2a: s1r2 = (0.I) - (r1s2) by H1,VECTSP_2:22
           .= -(r1s2)        by VECTSP_2:23;
      H3: gcd(r2,r1,Amp) = (1.I) by V1,L13;
      H4: gcd(r2,r1s2,Amp) = gcd(r2,s2,Amp) by T2,H3
                         .= (1.I)          by H0;
      H5: (1.I) = gcd(r2,-(s1r2),Amp) by H4,H2
           .= gcd((1.I)r2,-(s1r2),Amp) by VECTSP_2:1
           .= gcd((1.I)r2,(-s1)r2,Amp) by VECTSP_2:28;
      H6: gcd((1.I)r2,(-s1)r2,Amp) is_associated_to
          r2gcd((1.I),(-s1),Amp) by T1;
      H7: r2gcd((1.I),(-s1),Amp) = r2(1.I) by GCD2
                              .= r2      by VECTSP_2:1;
      H8: (1.I) is_associated_to r2gcd((1.I),(-s1),Amp)
          by H5,H6;
      H9: (1.I) is_associated_to r2 by H8,H7;
      H10: (1.I) ∈ Amp by AMP;
      H11: r2 = NF(r2,Amp)  by A
             .= (1.I) by H9,H10,Def20;
      H12: gcd(s2,s1,Amp) = (1.I) by V1,L13;
      H13: gcd(s2,s1r2,Amp) = gcd(s2,r2,Amp) by T2,H12
                            .= (1.I)          by H0,L13;
      H14: (1.I) = gcd(s2,-(r1s2),Amp) by H13,H2a
            .= gcd((1.I)s2,-(r1s2),Amp) by VECTSP_2:1
            .= gcd((1.I)s2,(-r1)s2,Amp) by VECTSP_2:28;
      H15: gcd((1.I)s2,(-r1)s2,Amp) is_associated_to
           s2gcd((1.I),(-r1),Amp) by T1;
      H16: s2gcd((1.I),(-r1),Amp) = s2(1.I) by GCD2
                                .= s2      by VECTSP_2:1;
      H17: (1.I) is_associated_to s2gcd((1.I),(-r1),Amp)
           by H14,H15;
      H18: (1.I) is_associated_to s2 by H17,H16;
      H19: s2 = NF(s2,Amp)  by A
             .= (1.I)       by H18,H10,Def20;
      H20: r2s2 = (1.I) by H11,H19,VECTSP_2:1;
      H21: z = r2s2 iff z = (1.I) by H20;
      thus thesis by H21;
      end;
    thus thesis by K01,K02,K03,K12,K13,K23;
    end;
   ◇
```

Macro referenced in scrap 30.

⟨proof D30 85⟩ ≡

```
proof
M1: z2 divides gcd(r2,s2,Amp) by Def4;
M2: gcd(r2,s2,Amp) divides r2 by Def4;
M3: z2 divides r2 by M1,M2,L1;
thus thesis by M3,L6a;
end;
```
◇

Macro referenced in scrap 86b.

⟨proof D32b 86a⟩ ≡

```
proof
consider zz being Element of the carrier of I such that
M1: zz = z1/z2;
M3: r2s2 <> (0.I) by H0,VECTSP_2:15;
M4a: gcd(r2,s2,Amp) divides r2s2 by D28a,L6a;
M4: z1 = (r2s2)/gcd(r2,s2,Amp) by D28a,D28b,M4a,L8;
M6: z1 <> (0.I) by M4,M3,M4a,D28b,L26;
M7: zz <> (0.I) by M1,M6,D30,D31,L26;
thus thesis by M7,M1;
end;
```
◇

Macro referenced in scrap 86b.

⟨proof ALG1 86b⟩ ≡

```
proof
let Amp be AmpleSet of I;
let r1,r2,s1,s2 be Element of the carrier of I;
assume H0a: Amp is_multiplicative &
            r1,r2 are_normalized_wrt Amp &
            s1,s2 are_normalized_wrt Amp;
H0b: r2 ∈ Amp & s2 ∈ Amp &
    r2 <> (0.I) & s2 <> (0.I) by H0a,Def27;
H0: r2 = NF(r2,Amp) & s2 = NF(s2,Amp) &
    r2 <> (0.I) & s2 <> (0.I) by H0b,NF3;
H3: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I)
    by H0a,Def27;
H3a: r1 canonical r2 & s1 canonical s2 by H3,Def10;
M: now per cases;

case B: s1 = (0.I);
B1: add1(r1,r2,s1,s2,Amp) = r1 by B,H0,H3a,Def11a;
B2: add2(r1,r2,s1,s2,Amp) = r2 by B,H0,H3a,Def12a;
B3:   gcd(add1(r1,r2,s1,s2,Amp),add2(r1,r2,s1,s2,Amp),Amp)
```

```
    = gcd(r1,r2,Amp)   by B1,B2
   .= (1.I)            by H3;
thus thesis by B3,B2,H0b,Def27;

case D1: (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
         = (0.I);
D11: add1(r1,r2,s1,s2,Amp) = (0.I) by D1,H0,H3a,Def11a;
D12: add2(r1,r2,s1,s2,Amp) = (1.I) by D1,H0,H3a,Def12a;
D13:   gcd(add1(r1,r2,s1,s2,Amp),add2(r1,r2,s1,s2,Amp),Amp)
     = gcd((0.I),(1.I),Amp)  by D11,D12
    .= (1.I)                 by GCD2;
D14: (1.I) ∈ Amp by AMP;
D15: (1.I) <> (0.I) by VECTSP_1:def 21;
thus thesis by D12,D13,D14,D15,Def27;

case D2: r1 <> (0.I) & s1 <> (0.I) & gcd(r2,s2,Amp) <> (1.I)
         & (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
            <> (0.I);
D21: add1(r1,r2,s1,s2,Amp) =
     ((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))))  /
     gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         gcd(r2,s2,Amp),Amp)
     by D2,H0,H3a,Def11a;
D22: add2(r1,r2,s1,s2,Amp) =
     (r2(s2/gcd(r2,s2,Amp)))  /
     gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         gcd(r2,s2,Amp),Amp)
     by D2,H0,H3a,Def12a;
D25:  gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
           (r2(s2/gcd(r2,s2,Amp))),Amp)
    = gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
           gcd(r2,s2,Amp),Amp)  by H0,H3,HEN1;
D26a: gcd(r2,s2,Amp) <> (0.I) by H0,L12;
D26: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         gcd(r2,s2,Amp),Amp)
     <> (0.I) by D26a,L12;
D27: gcd(
       ((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))))  /
        gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
            gcd(r2,s2,Amp),Amp),
       ((r2(s2/gcd(r2,s2,Amp))))  /
        gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
            gcd(r2,s2,Amp),Amp),
       Amp)
```

```
    = (1.I) by D26,D25,T3;
  reconsider r2,s2 as Element of Amp by H0b;
  D28a: gcd(r2,s2,Amp) divides s2 by Def4;
  D28b: gcd(r2,s2,Amp) <> (0.I) by H0,L12;
  D28c: gcd(r2,s2,Amp) ∈ Amp by Def4;
  D28d: s2/gcd(r2,s2,Amp) ∈ Amp by D28a,D28b,D28c,H0a,AMP5;
  reconsider z3 = s2/gcd(r2,s2,Amp) as Element of Amp by D28d;
  D28: r2z3 ∈ Amp by H0a,Def25;
  reconsider z1 = r2(s2/gcd(r2,s2,Amp)) as Element of Amp
       by D28;
  reconsider z2 =
       gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
           gcd(r2,s2,Amp),Amp) as Element of Amp by Def4;
  D30: z2 divides z1
      ⟨proof D30 85⟩
  D31: z2 <> (0.I) by D28b,L12;
  D32a: z1 / z2 ∈ Amp by D30,D31,AMP5,H0a;
  D32b: z1 / z2  <> (0.I)
       ⟨proof D32b 86a⟩
  D32: (r2(s2/gcd(r2,s2,Amp)))  /
       gcd((r1(s2/gcd(r2,s2,Amp))) + (s1  (r2/gcd(r2,s2,Amp))),
           gcd(r2,s2,Amp),Amp) ∈ Amp by D32a;
  thus thesis by D21,D22,D27,D32,D32b,Def27;

  ⟨example cases ALG1 33a⟩

  end;  ::cases M
  thus thesis by M;
  end;
 ◇
```

Macro referenced in scrap 32.

⟨proof ALG2 88⟩ ≡

```
  proof
  let Amp be AmpleSet of I;
  let r1,r2,s1,s2 be Element of the carrier of I;
  assume H0a: Amp is_multiplicative &
              r1,r2 are_normalized_wrt Amp &
              s1,s2 are_normalized_wrt Amp;
  H0b: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I) &
       s2 ∈ Amp & r2 ∈ Amp by H0a,Def27;
  H0: r1 canonical r2 & s1 canonical s2 &
      r2 <> (0.I) & s2 <> (0.I) &
      r2 = NF(r2,Amp) & s2 = NF(s2,Amp)
```

```
    by Def27,H0a,H0b,Def10,NF3;
M: now per cases;

case A: r1 = (0.I);
A1: add1(r1,r2,s1,s2,Amp) = s1 by A,H0,Def11a;
A2: add2(r1,r2,s1,s2,Amp) = s2 by A,H0,Def12a;
A3:   add2(r1,r2,s1,s2,Amp)((r1s2)+(s1r2))
    = s2((r1s2)+(s1r2))            by A2
   .= s2(((0.I)s2)+(s1r2))         by A
   .= s2((0.I)+(s1r2))             by VECTSP_2:26
   .= s2(s1r2)                     by VECTSP_2:1
   .= (s2s1)r2                     by VECTSP_1:def 16
   .= (s1s2)r2
   .= s1(s2r2)                     by VECTSP_1:def 16
   .= s1(r2s2)
   .= add1(r1,r2,s1,s2,Amp)(r2s2)  by A1;
thus thesis by A3;

case B: s1 = (0.I);
B1: add1(r1,r2,s1,s2,Amp) = r1 by B,H0,Def11a;
B2: add2(r1,r2,s1,s2,Amp) = r2 by B,H0,Def12a;
B3:   add2(r1,r2,s1,s2,Amp)((r1s2)+(s1r2))
    = r2((r1s2)+(s1r2))            by B2
   .= r2((r1s2)+((0.I)r2))         by B
   .= r2((r1s2)+(0.I))             by VECTSP_2:26
   .= r2(r1s2)                     by VECTSP_2:1
   .= (r2r1)s2                     by VECTSP_1:def 16
   .= (r1r2)s2
   .= r1(r2s2)                     by VECTSP_1:def 16
   .= add1(r1,r2,s1,s2,Amp)(r2s2)  by B1;
thus thesis by B3;

case D2: r1 <> (0.I) & s1 <> (0.I) & gcd(r2,s2,Amp) <> (1.I) &
        (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
        <> (0.I);
D21: add1(r1,r2,s1,s2,Amp) =
     ((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))) /
      gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
          gcd(r2,s2,Amp),Amp)
     by D2,H0,Def11a;
D22: add2(r1,r2,s1,s2,Amp) =
     (r2(s2/gcd(r2,s2,Amp))) /
     gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         gcd(r2,s2,Amp),Amp)
```

```
     by D2,H0,Def12a;
D23: gcd(r2,s2,Amp) <> (0.I) by H0,L12;
D24: gcd(r2,s2,Amp) divides s2 by L0;
D25: gcd(r2,s2,Amp) divides r2 by L0;
D26: gcd(r2,s2,Amp) divides r1s2 by D24,L6a;
D27: gcd(r2,s2,Amp) divides s1r2 by D25,L6a;
D28: gcd(r2,s2,Amp) divides (r1s2)  r2 by D26,L6a;
D29: gcd(r2,s2,Amp) divides (s1r2)r2 by D27,L6a;
D30: gcd(r2,s2,Amp) divides ((r1s2)r2)s2 by D28,L6a;
D31: gcd(r2,s2,Amp) divides ((s1r2)r2)s2 by D29,L6a;
D32:   ((r1(s2/gcd(r2,s2,Amp))) +
       (s1(r2/gcd(r2,s2,Amp))))(r2s2)
     = ((r1(s2/gcd(r2,s2,Amp)))(r2s2)) +
       ((s1(r2/gcd(r2,s2,Amp)))(r2s2)) by VECTSP_2:1
    .= (((r1s2)/gcd(r2,s2,Amp))(r2s2)) +
       ((s1(r2/gcd(r2,s2,Amp)))(r2s2)) by D23,D24,D26,L8
    .= (((r1s2)/gcd(r2,s2,Amp))(r2s2)) +
       (((s1r2)/gcd(r2,s2,Amp))(r2s2)) by D23,D25,D27,L8
    .= ((((r1s2)/gcd(r2,s2,Amp))r2)s2) +
       (((s1r2)/gcd(r2,s2,Amp))(r2s2)) by VECTSP_1:def 16
    .= ((((r1s2)/gcd(r2,s2,Amp))r2)s2) +
       ((((s1r2)/gcd(r2,s2,Amp))r2)s2) by VECTSP_1:def 16
    .= ((((r1s2)r2)/gcd(r2,s2,Amp))s2) +
       ((((s1r2)/gcd(r2,s2,Amp))r2)s2) by D23,D26,D28,L8
    .= ((((r1s2)r2)/gcd(r2,s2,Amp))s2) +
       ((((s1r2)r2)/gcd(r2,s2,Amp))s2) by D23,D27,D29,L8
    .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
       ((((s1r2)r2)/gcd(r2,s2,Amp))s2) by D23,D28,D30,L8
    .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
       ((((s1r2)r2)s2)/gcd(r2,s2,Amp)) by D23,D29,D31,L8;
D33a: gcd(r2,s2,Amp) divides (r2s2) by D25,L6a;
D33: gcd(r2,s2,Amp) divides (r2s2)r1 by D33a,L6a;
D34: gcd(r2,s2,Amp) divides (r2s2)s1 by D33a,L6a;
D35: gcd(r2,s2,Amp) divides ((r2s2)r1)s2 by D33,L6a;
D36: gcd(r2,s2,Amp) divides ((r2s2)s1)r2 by D34,L6a;
D37:   (r2(s2/gcd(r2,s2,Amp)))((r1s2)+(s1r2))
     = ((r2(s2/gcd(r2,s2,Amp)))(r1s2))+
       ((r2(s2/gcd(r2,s2,Amp)))(s1r2)) by VECTSP_2:1
    .= (((r2(s2/gcd(r2,s2,Amp)))r1)s2) +
       ((r2(s2/gcd(r2,s2,Amp)))(s1r2)) by VECTSP_1:def 16
    .= (((r2(s2/gcd(r2,s2,Amp)))r1)s2) +
       (((r2(s2/gcd(r2,s2,Amp)))s1)r2) by VECTSP_1:def 16
    .= ((((r2s2)/gcd(r2,s2,Amp))r1)s2) +
       (((r2(s2/gcd(r2,s2,Amp)))s1)r2) by D23,D24,D33a,L8
```

```
      .= ((((r2s2)/gcd(r2,s2,Amp))r1)s2) +
         ((((r2s2)/gcd(r2,s2,Amp))s1)r2) by D23,D24,D33a,L8
      .= ((((r2s2)r1)/gcd(r2,s2,Amp))s2) +
         ((((r2s2)/gcd(r2,s2,Amp))s1)r2) by D23,D33a,D33,L8
      .= ((((r2s2)r1)/gcd(r2,s2,Amp))s2) +
         ((((r2s2)s1)/gcd(r2,s2,Amp))r2) by D23,D33a,D34,L8
      .= ((((r2s2)r1)s2)/gcd(r2,s2,Amp)) +
         ((((r2s2)s1)/gcd(r2,s2,Amp))r2) by D23,D33,D35,L8
      .= ((((r2s2)r1)s2)/gcd(r2,s2,Amp)) +
         ((((r2s2)s1)r2)/gcd(r2,s2,Amp)) by D23,D34,D36,L8
      .= (((r1(r2s2))s2)/gcd(r2,s2,Amp)) +
         ((((r2s2)s1)r2)/gcd(r2,s2,Amp))
      .= (((r1(s2r2))s2)/gcd(r2,s2,Amp)) +
         ((((r2s2)s1)r2)/gcd(r2,s2,Amp))
      .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
         ((((r2s2)s1)r2)/gcd(r2,s2,Amp)) by VECTSP_1:def 16
      .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
         (((s1(r2s2))r2)/gcd(r2,s2,Amp))
      .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
         ((s1((r2s2)r2))/gcd(r2,s2,Amp)) by VECTSP_1:def 16
      .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
         ((s1(r2(r2s2)))/gcd(r2,s2,Amp))
      .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
         ((s1((r2r2)s2))/gcd(r2,s2,Amp)) by VECTSP_1:def 16
      .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
         (((s1(r2r2))s2)/gcd(r2,s2,Amp)) by VECTSP_1:def 16
      .= ((((r1s2)r2)s2)/gcd(r2,s2,Amp)) +
         ((((s1r2)r2)s2)/gcd(r2,s2,Amp)) by VECTSP_1:def 16;
D38:   ((r1(s2/gcd(r2,s2,Amp))) +
       (s1(r2/gcd(r2,s2,Amp))))(r2  s2)
     = (r2(s2/gcd(r2,s2,Amp)))((r1s2)+(s1r2))
     by D32,D37;
D39: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         gcd(r2,s2,Amp),Amp)
     <> (0.I) by L12,D23;
D40: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         gcd(r2,s2,Amp),Amp) divides
     ((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))))
     by Def4;
D41: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
         gcd(r2,s2,Amp),Amp) divides
     ((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))))(r2s2)
     by D40,L6a;
D42: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
```

```
          gcd(r2,s2,Amp),Amp)
     divides gcd(r2,s2,Amp) by Def4;
D43: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
          gcd(r2,s2,Amp),Amp)
     divides r2 by D25,D42,L1;
D44: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
          gcd(r2,s2,Amp),Amp)
     divides (r2(s2/gcd(r2,s2,Amp))) by D43,L6a;
D45: gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
          gcd(r2,s2,Amp),Amp)
     divides
     (r2(s2/gcd(r2,s2,Amp)))((r1s2)+(s1r2)) by D44,L6a;
D46:    add1(r1,r2,s1,s2,Amp)(r2s2)
     =  (((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))) /
        gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
            gcd(r2,s2,Amp),Amp)) (r2s2) by D21
    .=  (((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))))
         (r2s2)) /
        gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
            gcd(r2,s2,Amp),Amp) by D39,D40,D41,L8
    .=  ((r2(s2/gcd(r2,s2,Amp)))((r1s2)+(s1r2))) /
        gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
            gcd(r2,s2,Amp),Amp) by D38
     .= ((r2(s2/gcd(r2,s2,Amp))) /
        gcd((r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp))),
            gcd(r2,s2,Amp),Amp)) ((r1s2)+(s1r2))
        by D39,D44,D45,L8
     .= add2(r1,r2,s1,s2,Amp)((r1s2)+(s1r2)) by D22;
thus thesis by D46;

⟨example cases ALG2 33b⟩

end;  :: cases M
thus thesis by M;
end;
```

◇

Macro referenced in scrap 32.

⟨consistency mult1 92⟩ ≡

```
consistency
proof
K01: (r1 = (0.I) or s1 = (0.I)) & (r2 = (1.I) & s2 = (1.I))
     implies for z being Element of the carrier of I
     holds z = (0.I) iff z = r1s1
```

```
      proof
      assume H0: (r1 = (0.I) or s1 = (0.I)) &
                 (r2 = (1.I) & s2 = (1.I));
      H1: r1 = (0.I) or s1 = (0.I) by H0;
      M: now per cases by H1;
      case A: r1 = (0.I);
      A1: r1s1 = (0.I)s1 by A
              .= (0.I)   by VECTSP_2:26;
      thus thesis by A1;
      case B: s1 = (0.I);
      B1: r1s1 = r1(0.I) by B
              .= (0.I)   by VECTSP_2:26;
      thus thesis by B1;
      end;  :: cases
      thus thesis by M;
      end;
K02: ((r1 = (0.I) or s1 = (0.I)) & (s2 <> (0.I) & r2 = (1.I)))
      implies for z being Element of the carrier of I
      holds z = (0.I) iff z = (r1s1)/gcd(r1,s2,Amp)
      proof
      assume H0: (r1 = (0.I) or s1 = (0.I)) &
                  s2 <> (0.I) & r2 = (1.I);
      let z be Element of the carrier of I;
      H1: r1 = (0.I) or s1 = (0.I) by H0;
      H2: r1s1 = (0.I)
          proof
          M: now per cases by H1;
          case A: r1 = (0.I);
          A1: r1s1 = (0.I)s1 by A
                  .= (0.I)   by VECTSP_2:26;
          thus thesis by A1;
          case B: s1 = (0.I);
          B1: r1s1 = r1(0.I) by B
                  .= (0.I)   by VECTSP_2:26;
          thus thesis by B1;
          end;  :: cases
          thus thesis by M;
          end;
      H3: gcd(r1,s2,Amp) divides r1 by Def4;
      H4: gcd(r1,s2,Amp) divides r1s1 by H3,L6a;
      H5: gcd(r1,s2,Amp) <> (0.I) by H0,L12;
      consider d being Element of the carrier of I such that
      H6: d = (r1s1)/gcd(r1,s2,Amp);
      H7: dgcd(r1,s2,Amp) = r1s1   by H6,H4,H5,Def5
```

```
                    .= (0.I)  by H2;
     H8:   (r1s1)/gcd(r1,s2,Amp) = d by H6
        .= (0.I)      by H7,H5,VECTSP_2:15;
     thus thesis by H8;
     end;
K03: ((r1 = (0.I) or s1 = (0.I)) & (r2 <> (0.I) & s2 = (1.I)))
     implies for z being Element of the carrier of I
     holds z = (0.I) iff z = (r1s1)/gcd(s1,r2,Amp)
     proof
     assume H0: (r1 = (0.I) or s1 = (0.I)) &
               r2 <> (0.I) & s2 = (1.I);
     let z be Element of the carrier of I;
     H1: r1 = (0.I) or s1 = (0.I) by H0;
     H2: r1s1 = (0.I)
         proof
         M: now per cases by H1;
         case A: r1 = (0.I);
         A1: r1s1 = (0.I)s1  by A
                 .= (0.I)    by VECTSP_2:26;
         thus thesis by A1;
         case B: s1 = (0.I);
         B1: r1s1 = r1(0.I)  by B
                 .= (0.I)    by VECTSP_2:26;
         thus thesis by B1;
         end;  :: cases
         thus thesis by M;
         end;
     H3: gcd(s1,r2,Amp) divides s1 by Def4;
     H4: gcd(s1,r2,Amp) divides r1s1 by H3,L6a;
     H5: gcd(s1,r2,Amp) <> (0.I) by H0,L12;
     consider d being Element of the carrier of I such that
     H6: d = (r1s1)/gcd(s1,r2,Amp);
     H7: dgcd(s1,r2,Amp) = r1s1   by H6,H4,H5,Def5
                        .= (0.I)  by H2;
     H8:   (r1s1)/gcd(s1,r2,Amp) = d by H6
        .= (0.I)      by H7,H5,VECTSP_2:15;
     thus thesis by H8;
     end;
K12: ((r2 = (1.I) & s2 = (1.I)) & (s2 <> (0.I) & r2 = (1.I)))
     implies for z being Element of the carrier of I
     holds z = r1s1 iff z = (r1s1)/gcd(r1,s2,Amp)
     proof
     assume H0: r2 = (1.I) & s2 = (1.I) &
               s2 <> (0.I) & r2 = (1.I);
```

```
      H1: gcd(r1,s2,Amp) = gcd(r1,(1.I),Amp) by H0
                        .= (1.I)             by GCD2;
      H2: (r1s1)/gcd(r1,s2,Amp) = (r1s1)/(1.I) by H1
                                .= r1s1        by L7a;
      thus thesis by H2;
      end;
K13:  ((r2 = (1.I) & s2 = (1.I)) & (r2 <> (0.I) & s2 = (1.I)))
      implies for z being Element of the carrier of I
      holds z = r1s1 iff z = (r1s1)/gcd(s1,r2,Amp)
      proof
      assume H0: r2 = (1.I) & s2 = (1.I) &
                 r2 <> (0.I) & s2 = (1.I);
      H1: gcd(s1,r2,Amp) = gcd(s1,(1.I),Amp) by H0
                        .= (1.I)             by GCD2;
      H2: (r1s1)/gcd(s1,r2,Amp) = (r1s1)/(1.I) by H1
                                .= r1s1        by L7a;
      thus thesis by H2;
      end;
K23:  ((s2 <> (0.I) & r2 = (1.I)) & (r2 <> (0.I) & s2 = (1.I)))
      implies for z being Element of the carrier of I
      holds z = (r1s1)/gcd(r1,s2,Amp) iff
            z = (r1s1)/gcd(s1,r2,Amp)
      proof
      assume H0: s2 <> (0.I) & r2 = (1.I) &
                 r2 <> (0.I) & s2 = (1.I);
      H1: gcd(r1,s2,Amp) = gcd(r1,(1.I),Amp) by H0
                        .= (1.I)             by GCD2;
      H2: gcd(s1,r2,Amp) = gcd(s1,(1.I),Amp) by H0
                        .= (1.I)             by GCD2;
      H3:   (r1s1)/gcd(r1,s2,Amp) = (r1s1)/(1.I) by H1
         .= (r1s1)/gcd(s1,r2,Amp) by H2;
      thus thesis by H3;
      end;
thus thesis by K01,K02,K03,K12,K13,K23;
end;
```

◇

Macro referenced in scrap 34.

⟨consistency mult2 95⟩ ≡

```
consistency
proof
V1: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I)
    by AS,CAN1;
K02: (r1 = (0.I) or s1 = (0.I)) & (s2 <> (0.I) & r2 = (1.I))
```

```
      implies for z being Element of the carrier of I
      holds z = (1.I) iff z = s2/gcd(r1,s2,Amp)
      proof
      assume H0: (r1 = (0.I) or s1 = (0.I)) &
                 s2 <> (0.I) & r2 = (1.I);
      H1: r1 = (0.I) or s1 = (0.I) by H0;
      M: now per cases by H1;
      case A: r1 = (0.I);
      A1: gcd(r1,s2,Amp) = gcd((0.I),s2,Amp) by A
                       .= NF(s2,Amp)        by GCD1
                       .= s2                by AS;
      A2: s2/gcd(r1,s2,Amp) = s2/s2  by A1
                            .= (1.I)  by L7,H0;
      thus thesis by A2;
      case B: s1 = (0.I);
      B1: (1.I) = gcd(s1,s2,Amp)     by V1
               .= gcd((0.I),s2,Amp)  by B
               .= NF(s2,Amp)         by GCD1
               .= s2                 by AS;
      B2: gcd(r1,s2,Amp) = gcd(r1,(1.I),Amp)  by B1
                       .= (1.I)               by GCD2;
      B2a: (1.I) <> (0.I) by VECTSP_1: def 21;
      B3: s2/gcd(r1,s2,Amp) = (1.I)/(1.I) by B1,B2
                            .= (1.I) by L7,B2a;
      thus thesis by B3;
      end;  :: cases
      thus thesis by M;
      end;
K03:  (r1 = (0.I) or s1 = (0.I)) & (r2 <> (0.I) & s2 = (1.I))
      implies for z being Element of the carrier of I
      holds z = (1.I) iff z = r2/gcd(s1,r2,Amp)
      proof
      assume H0: (r1 = (0.I) or s1 = (0.I)) &
                 r2 <> (0.I) & s2 = (1.I);
      H1: r1 = (0.I) or s1 = (0.I) by H0;
      M: now per cases by H1;
      case A: s1 = (0.I);
      A1: gcd(s1,r2,Amp) = gcd((0.I),r2,Amp) by A
                       .= NF(r2,Amp)        by GCD1
                       .= r2                by AS;
      A2: r2/gcd(s1,r2,Amp) = r2/r2  by A1
                            .= (1.I)  by L7,H0;
      thus thesis by A2;
      case B: r1 = (0.I);
```

```
      B1: (1.I) = gcd(r1,r2,Amp)     by V1
                .= gcd((0.I),r2,Amp)  by B
                .= NF(r2,Amp)         by GCD1
                .= r2                 by AS;
      B2: gcd(s1,r2,Amp) = gcd(s1,(1.I),Amp)  by B1
                         .= (1.I)             by GCD2;
      B2a: (1.I) <> (0.I) by VECTSP_1: def 21;
      B3: r2/gcd(s1,r2,Amp) = (1.I)/(1.I) by B1,B2
                            .= (1.I) by L7,B2a;
      thus thesis by B3;
      end;  :: cases
      thus thesis by M;
      end;
K12:  (r2 = (1.I) & s2 = (1.I)) & (s2 <> (0.I) & r2 = (1.I))
      implies for z being Element of the carrier of I
      holds z = (1.I) iff z = s2/gcd(r1,s2,Amp)
      proof
      assume H0: r2 = (1.I) & s2 = (1.I) &
                 s2 <> (0.I) & r2 = (1.I);
      H1a: (1.I) <> (0.I) by VECTSP_1: def 21;
      H1: gcd(r1,s2,Amp) = gcd(r1,(1.I),Amp) by H0
                         .= (1.I) by GCD2;
      H2: s2/gcd(r1,s2,Amp) = (1.I)/(1.I) by H0,H1
                            .= (1.I) by H1a,L7;
      thus thesis by H2;
      end;
K13:  (r2 = (1.I) & s2 = (1.I)) & (r2 <> (0.I) & s2 = (1.I))
      implies for z being Element of the carrier of I
      holds z = (1.I) iff z = r2/gcd(s1,r2,Amp)
      proof
      assume H0: r2 = (1.I) & s2 = (1.I) &
                 r2 <> (0.I) & s2 = (1.I);
      H1a: (1.I) <> (0.I) by VECTSP_1: def 21;
      H1: gcd(s1,r2,Amp) = gcd(s1,(1.I),Amp) by H0
                         .= (1.I) by GCD2;
      H2: r2/gcd(s1,r2,Amp) = (1.I)/(1.I) by H0,H1
                            .= (1.I) by H1a,L7;
      thus thesis by H2;
      end;
K23:  (s2 <> (0.I) & r2 = (1.I)) & (r2 <> (0.I) & s2 = (1.I))
      implies for z being Element of the carrier of I
      holds z = s2/gcd(r1,s2,Amp) iff z = r2/gcd(s1,r2,Amp)
      proof
      assume H0: s2 <> (0.I) & r2 = (1.I) &
```

```
                r2 <> (0.I) & s2 = (1.I);
      H1: gcd(r1,s2,Amp) = gcd(r1,(1.I),Amp) by H0
                        .= (1.I)  by GCD2;
      H2: gcd(s1,r2,Amp) = gcd(s1,(1.I),Amp) by H0
                        .= (1.I)  by GCD2;
      H3:   s2/gcd(r1,s2,Amp) = (1.I)/(1.I) by H0,H1
         .= r2/gcd(s1,r2,Amp) by H0,H2;
      thus thesis by H3;
      end;
   thus thesis by K02,K03,K12,K13,K23;
   end;
```

◇

Macro referenced in scrap 34.

⟨proof D13 98⟩ ≡

```
   proof
   H1: gcd(r2,s2,Amp) <> (0.I) by H0,L12;
   H2: gcd(r2,s2,Amp) divides s2 by L0;
   H3: gcd(r2,s2,Amp) divides r2 by L0;
   H4: gcd(r2,s2,Amp) divides r1  s2 by H2,L6a;
   H5: gcd(r2,s2,Amp) divides s1  r2 by H3,L6a;
   H6:   (0.I)
       = (r1(s2/gcd(r2,s2,Amp)))+(s1(r2/gcd(r2,s2,Amp)))
         by D
      .= (r1s2)/gcd(r2,s2,Amp)+(s1(r2/gcd(r2,s2,Amp)))
         by H1,H2,H4,L8
      .= (r1s2)/gcd(r2,s2,Amp)+(s1r2)/gcd(r2,s2,Amp)
         by H1,H3,H5,L8;
   consider e being Element of the carrier of I such that
   H7: gcd(r2,s2,Amp)e = r2 by H3,Def1;
   consider f being Element of the carrier of I such that
   H8: gcd(r2,s2,Amp)f = s2 by H2,Def1;
   H9:    gcd(r2,s2,Amp)((e  s1)+(fr1))
        = (gcd(r2,s2,Amp)(es1))+(gcd(r2,s2,Amp)(f  r1))
          by VECTSP_2:1
       .= ((gcd(r2,s2,Amp)  e)s1)+(gcd(r2,s2,Amp)(fr1))
          by VECTSP_1:def 16
       .= ((gcd(r2,s2,Amp)e)s1)+((gcd(r2,s2,Amp)f)r1)
          by VECTSP_1:def 16
       .= (r2s1)+(s2r1) by H7,H8
       .= (s2r1)+(r2s1)
       .= (r1s2)+(s1r2);
   H10: gcd(r2,s2,Amp) divides (r1s2)+(s1r2) by H9,Def1;
   H11:   (0.I)
```

```
        = (r1s2)/gcd(r2,s2,Amp)+(s1r2)/gcd(r2,s2,Amp) by H6
       .= ((r1s2)+(s1r2))/gcd(r2,s2,Amp) by H1,H4,H5,H10,L8a;
H12:   (0.I)
        = (0.I)gcd(r2,s2,Amp)  by VECTSP_2:26
       .= ((r1s2)+(s1r2)) by H11,H1,H10,Def5;
thus thesis by H12;
end;
```
◇

Macro referenced in scrap 33b.

⟨proof ALG3 99⟩ ≡

```
proof
let Amp be AmpleSet of I;
let r1,r2,s1,s2 be Element of the carrier of I;
assume H0a: Amp is_multiplicative &
            r1,r2 are_normalized_wrt Amp &
            s1,s2 are_normalized_wrt Amp;
H3: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I) &
    s2 ∈ Amp & r2 ∈ Amp by H0a,Def27;
H0: r1 canonical r2 & s1 canonical s2 &
    r2 <> (0.I) & s2 <> (0.I) &
    r2 = NF(r2,Amp) & s2 = NF(s2,Amp)
    by Def27,H0a,H3,Def10,NF3;
H2: gcd(r1,s2,Amp) <> (0.I) & gcd(s1,r2,Amp) <> (0.I)
    by H0,L12;
M: now per cases;

case A: r1 = (0.I) or s1 = (0.I);
A1: mult1(r1,r2,s1,s2,Amp) = (0.I) by A,H0,Def13;
A2: mult2(r1,r2,s1,s2,Amp) = (1.I) by A,H0,Def14;
A3:   gcd(mult1(r1,r2,s1,s2,Amp),mult2(r1,r2,s1,s2,Amp),Amp)
    = gcd((0.I),(1.I),Amp)  by A1,A2
   .= (1.I)                 by GCD2;
A4: (1.I) ∈ Amp by AMP;
A5: (1.I) <> (0.I) by VECTSP_1:def 21;
thus thesis by A2,A3,A4,A5,Def27;

case C: s2 <> (0.I) & r2 = (1.I);
C1: mult1(r1,r2,s1,s2,Amp) = (r1s1)/gcd(r1,s2,Amp)
    by C,H0,Def13;
C2: mult2(r1,r2,s1,s2,Amp) = s2/gcd(r1,s2,Amp) by C,H0,Def14;
C3: gcd(s1,r2,Amp) = (1.I) by C,GCD2;
C4: r2/gcd(s1,r2,Amp) = (1.I)
      proof
```

```
      M1: (1.I) <> (0.I) by VECTSP_1: def 21;
      M2: r2/gcd(s1,r2,Amp) = (1.I)/(1.I)  by C,C3
                          .= (1.I)         by M1,L7;
      thus thesis by M2;
      end;
C5: gcd(r1,s2,Amp) divides r1 by Def4;
C6: gcd(r1,s2,Amp) divides r1s1 by C5,L6a;
C7:   (r1s1)/gcd(r1,s2,Amp)
    = (r1/gcd(r1,s2,Amp))s1  by H2,C5,C6,L8
   .= (r1/gcd(r1,s2,Amp))(s1/(1.I))  by L7a
   .= (r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp))  by C3;
C8:   s2/gcd(r1,s2,Amp)
    = (s2/gcd(r1,s2,Amp))(1.I)                 by VECTSP_2:1
   .= (s2/gcd(r1,s2,Amp))(r2/gcd(s1,r2,Amp))  by C4;
C9:   gcd(mult1(r1,r2,s1,s2,Amp),mult2(r1,r2,s1,s2,Amp),Amp)
    = gcd((r1s1)/gcd(r1,s2,Amp),s2/gcd(r1,s2,Amp),Amp)
      by C1,C2
   .= gcd((r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)),
          (s2/gcd(r1,s2,Amp))(r2/gcd(s1,r2,Amp)),Amp)
      by C7,C8
   .= (1.I) by H0,H3,HEN2;
C10a: gcd(r1,s2,Amp) divides s2 by Def4;
C10b: gcd(r1,s2,Amp) <> (0.I) by H0,L12;
C10: s2/gcd(r1,s2,Amp) <> (0.I) by H0,C10a,C10b,L26;
C11a: gcd(r1,s2,Amp) ∈ Amp by Def4;
reconsider zz = gcd(r1,s2,Amp) as Element of Amp by C11a;
reconsider s2 as Element of Amp by H3;
C11: s2/zz ∈ Amp by AMP5,C10a,C10b,H0a;
thus thesis by C2,C9,C10,C11,Def27;

case D: r2 <> (0.I) & s2 = (1.I);
D1: mult1(r1,r2,s1,s2,Amp) = (r1s1)/gcd(s1,r2,Amp)
    by D,H0,Def13;
D2: mult2(r1,r2,s1,s2,Amp) = r2/gcd(s1,r2,Amp) by D,H0,Def14;
D3: gcd(r1,s2,Amp) = (1.I) by D,GCD2;
D4: s2/gcd(r1,s2,Amp) = (1.I)
      proof
      M1: (1.I) <> (0.I) by VECTSP_1: def 21;
      M2: s2/gcd(r1,s2,Amp) = (1.I)/(1.I)  by D,D3
                          .= (1.I)         by M1,L7;
      thus thesis by M2;
      end;
D5: gcd(s1,r2,Amp) divides s1 by Def4;
D6: gcd(s1,r2,Amp) divides (r1  s1) by D5,L6a;
```

```
D7:   (r1s1)/gcd(s1,r2,Amp)
    = r1(s1/gcd(s1,r2,Amp))  by H2,D5,D6,L8
   .= (r1/(1.I))(s1/gcd(s1,r2,Amp))  by L7a
   .= (r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp))  by D3;
D8:    r2/gcd(s1,r2,Amp)
    = (r2/gcd(s1,r2,Amp))(1.I)  by VECTSP_2:1
   .= (r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp))  by D4;
D9:   gcd(mult1(r1,r2,s1,s2,Amp),mult2(r1,r2,s1,s2,Amp),Amp)
    = gcd((r1s1)/gcd(s1,r2,Amp),r2/gcd(s1,r2,Amp),Amp)
      by D1,D2
   .= gcd((r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)),
          (s2/gcd(r1,s2,Amp))(r2/gcd(s1,r2,Amp)),Amp)
      by D7,D8
   .= (1.I) by H0,H3,HEN2;
D10a: gcd(s1,r2,Amp) divides r2 by Def4;
D10b: gcd(s1,r2,Amp) <> (0.I) by H0,L12;
D10: r2/gcd(s1,r2,Amp) <> (0.I) by H0,D10a,D10b,L26;
D11a: gcd(s1,r2,Amp) ∈ Amp by Def4;
reconsider zz = gcd(s1,r2,Amp) as Element of Amp by D11a;
reconsider r2 as Element of Amp by H3;
D11: r2/zz ∈ Amp by AMP5,D10a,D10b,H0a;
thus thesis by D2,D9,D10,D11,Def27;

⟨example cases ALG3 36⟩

end; :: cases
thus thesis by M;
end;
```

◇

Macro referenced in scrap 35.

⟨proof ALG4 101⟩ ≡

```
proof
let Amp be AmpleSet of I;
let r1,r2,s1,s2 be Element of the carrier of I;
assume H0a: Amp is_multiplicative &
            r1,r2 are_normalized_wrt Amp &
            s1,s2 are_normalized_wrt Amp;
H0b: gcd(r1,r2,Amp) = (1.I) & gcd(s1,s2,Amp) = (1.I) &
     s2 ∈ Amp & r2 ∈ Amp by H0a,Def27;
H0: r1 canonical r2 & s1 canonical s2 &
    r2 <> (0.I) & s2 <> (0.I) &
    r2 = NF(r2,Amp) & s2 = NF(s2,Amp)
    by Def27,H0a,H0b,Def10,NF3;
```

```
H1: gcd(r1,s2,Amp) <> (0.I) & gcd(s1,r2,Amp) <> (0.I)
    by H0,L12;
M: now per cases;

case A: r1 = (0.I) or s1 = (0.I);
A1: mult1(r1,r2,s1,s2,Amp) = (0.I) by A,H0,Def13;
A3: mult1(r1,r2,s1,s2,Amp)(r2s2) = (0.I) by A1,VECTSP_2:26;
K: now per cases by A;
 case A1: r1 = (0.I);
 A4:   mult2(r1,r2,s1,s2,Amp)(r1s1)
     = mult2(r1,r2,s1,s2,Amp)((0.I)s1)  by A1
    .= mult2(r1,r2,s1,s2,Amp)(0.I)      by VECTSP_2:26
    .= (0.I)                            by VECTSP_2:26;
 thus thesis by A4,A3;
 case A2: s1 = (0.I);
 A5:   mult2(r1,r2,s1,s2,Amp)(r1s1)
     = mult2(r1,r2,s1,s2,Amp)(r1(0.I))  by A2
    .= mult2(r1,r2,s1,s2,Amp)(0.I)      by VECTSP_2:26
    .= (0.I)                            by VECTSP_2:26;
 thus thesis by A5,A3;
 end; :: case K
thus thesis by K;

case C: s2 <> (0.I) & r2 = (1.I);
C1: mult1(r1,r2,s1,s2,Amp) = (r1s1)/gcd(r1,s2,Amp)
    by C,H0,Def13;
C2: mult2(r1,r2,s1,s2,Amp) = s2/gcd(r1,s2,Amp) by C,H0,Def14;
C3: gcd(r1,s2,Amp) divides r1 by Def4;
C4: gcd(r1,s2,Amp) divides r1s1 by C3,L6a;
C5: gcd(r1,s2,Amp) divides (r1s1)s2 by C4,L6a;
C6:   ((r1s1)/gcd(r1,s2,Amp))(r2s2)
    = ((r1s1)/gcd(r1,s2,Amp))((1.I)s2)  by C
   .= ((r1s1)/gcd(r1,s2,Amp))s2         by VECTSP_2:1
   .= ((r1s1)s2)/gcd(r1,s2,Amp)         by H1,C4,C5,L8;
C8: gcd(r1,s2,Amp) divides s2 by Def4;
C9: gcd(r1,s2,Amp) divides s2r1 by C8,L6a;
C10: gcd(r1,s2,Amp) divides (s2r1)s1 by C9,L6a;
C11:   (s2/gcd(r1,s2,Amp))(r1s1)
     = ((s2/gcd(r1,s2,Amp))r1)s1    by VECTSP_1:def 16
    .= ((s2  r1)/gcd(r1,s2,Amp))s1  by H1,C8,C9,L8
    .= ((s2r1)s1)/gcd(r1,s2,Amp)    by H1,C9,C10,L8
    .= ((r1s2)s1)/gcd(r1,s2,Amp)
    .= (r1(s2s1))/gcd(r1,s2,Amp)    by VECTSP_1:def 16
    .= (r1(s1s2))/gcd(r1,s2,Amp)
```

```
    .= ((r1s1)s2)/gcd(r1,s2,Amp)    by VECTSP_1:def 16;
C12:   mult1(r1,r2,s1,s2,Amp)(r2s2)
     = ((r1s1)/gcd(r1,s2,Amp))(r2s2)  by C1
    .= ((r1s1)s2)/gcd(r1,s2,Amp)      by C6
    .= (s2/gcd(r1,s2,Amp))(r1s1)      by C11
    .= mult2(r1,r2,s1,s2,Amp)(r1s1)   by C2;
thus thesis by C12;

case E: not(r1 = (0.I) or s1 = (0.I)) &
        not(r2 = (1.I) & s2 = (1.I)) &
        not(s2 <> (0.I) & r2 = (1.I)) &
        not(r2 <> (0.I) & s2 = (1.I));
E1: mult1(r1,r2,s1,s2,Amp) =
    (r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)) by E,H0,Def13;
E2: mult2(r1,r2,s1,s2,Amp) =
    (r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp)) by E,H0,Def14;
E3: gcd(r1,s2,Amp) divides r1 by Def4;
E4: gcd(s1,r2,Amp) divides s1 by Def4;
E5: (gcd(r1,s2,Amp)gcd(s1,r2,Amp)) divides r1s1 by E3,E4,L1a;
E6: (gcd(r1,s2,Amp)gcd(s1,r2,Amp)) divides (r1s1)r2
    by E5,L6a;
E7: (gcd(r1,s2,Amp)gcd(s1,r2,Amp)) divides ((r1s1)r2)s2
    by E6,L6a;
E8: (gcd(r1,s2,Amp)gcd(s1,r2,Amp)) <> (0.I) by H1,VECTSP_2:15;
E9:   ((r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)))(r2  s2)
    = ((r1s1)/(gcd(r1,s2,Amp)gcd(s1,r2,Amp)))(r2  s2)
      by H1,E3,E4,L8c
   .= (((r1s1)/(gcd(r1,s2,Amp)gcd(s1,r2,Amp)))r2)s2
      by VECTSP_1:def 16
   .= (((r1s1)r2)/(gcd(r1,s2,Amp)gcd(s1,r2,Amp)))s2
      by E8,E5,E6,L8
   .= (((r1s1)r2)s2)/(gcd(r1,s2,Amp)gcd(s1,r2,Amp))
      by E8,E6,E7,L8;
E10: gcd(s1,r2,Amp) divides r2 by Def4;
E11: gcd(r1,s2,Amp) divides s2 by Def4;
E12: (gcd(s1,r2,Amp)gcd(r1,s2,Amp)) divides r2s2
     by E10,E11,L1a;
E13: (gcd(s1,r2,Amp)gcd(r1,s2,Amp)) divides (r2s2)r1
     by E12,L6a;
E14: (gcd(s1,r2,Amp)gcd(r1,s2,Amp)) divides ((r2s2)r1)s1
     by E13,L6a;
E15:   ((r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp)))(r1s1)
     = ((r2s2)/(gcd(s1,r2,Amp)gcd(r1,s2,Amp)))(r1s1)
       by H1,E10,E11,L8c
```

```
      .= (((r2s2)/(gcd(s1,r2,Amp)gcd(r1,s2,Amp)))r1)s1
         by VECTSP_1:def 16
      .= (((r2s2)r1)/(gcd(s1,r2,Amp)gcd(r1,s2,Amp)))s1
         by E8,E12,E13,L8
      .= (((r2s2)r1)s1)/(gcd(s1,r2,Amp)gcd(r1,s2,Amp))
         by E8,E13,E14,L8
      .= ((r1(r2s2))s1)/(gcd(s1,r2,Amp)gcd(r1,s2,Amp))
      .= (r1((r2s2)s1))/(gcd(s1,r2,Amp)gcd(r1,s2,Amp))
         by VECTSP_1:def 16
      .= (r1(s1(r2s2)))/(gcd(s1,r2,Amp)gcd(r1,s2,Amp))
      .= (r1((s1r2)s2))/(gcd(s1,r2,Amp)gcd(r1,s2,Amp))
         by VECTSP_1:def 16
      .= ((r1(s1r2))s2)/(gcd(s1,r2,Amp)gcd(r1,s2,Amp))
         by VECTSP_1:def 16
      .= (((r1s1)r2) s2)/(gcd(s1,r2,Amp)gcd(r1,s2,Amp))
         by VECTSP_1:def 16;
E16:  mult1(r1,r2,s1,s2,Amp)(r2s2)
     = ((r1/gcd(r1,s2,Amp))(s1/gcd(s1,r2,Amp)))(r2  s2)
         by E1
      .= (((r1s1)r2)s2)/(gcd(r1,s2,Amp)gcd(s1,r2,Amp))
         by E9
      .= ((r2/gcd(s1,r2,Amp))(s2/gcd(r1,s2,Amp)))(r1s1)
         by E15
      .= mult2(r1,r2,s1,s2,Amp)(r1s1)
         by E2;
thus thesis by E16;

⟨example cases ALG4 37⟩

end;  ::cases
thus thesis by M;
end;
```

◇

Macro referenced in scrap 35.

C Indices

C.1 Files

C.2 Macros

⟨proof AMP0 61a⟩ Referenced in scrap 18c.
⟨proof AMP1 61b⟩ Referenced in scrap 18c.
⟨proof AMP5 59b⟩ Referenced in scrap 18c.
⟨proof AMP 59a⟩ Referenced in scrap 18c.
⟨proof CL1 51b⟩ Referenced in scrap 13b.
⟨proof CL2 52⟩ Referenced in scrap 13b.
⟨proof CL3 53a⟩ Referenced in scrap 13b.
⟨proof D13 98⟩ Referenced in scrap 33b.
⟨proof D30 85⟩ Referenced in scrap 86b.
⟨proof D32b 86a⟩ Referenced in scrap 86b.
⟨proof H11 75b⟩ Referenced in scrap 27.
⟨proof H14 76a⟩ Referenced in scrap 74b.
⟨proof H27 76b⟩ Referenced in scrap 28.
⟨proof H45 77a⟩ Referenced in scrap 28.
⟨proof H7 75a⟩ Referenced in scrap 27.
⟨proof H9 74a⟩ Referenced in scrap 25a.
⟨proof HEN1 2 74b⟩ Referenced in scrap 27.
⟨proof HEN1 27⟩ Referenced in scrap 26.
⟨proof HEN2 2 29a⟩ Referenced in scrap 28.
⟨proof HEN2 28⟩ Referenced in scrap 26.
⟨proof K2 53b⟩ Referenced in scrap 14b.
⟨proof K3 53c⟩ Referenced in scrap 14b.
⟨proof K5a 54a⟩ Referenced in scrap 14b.
⟨proof K6a 54b⟩ Referenced in scrap 15b.
⟨proof K6 55⟩ Referenced in scrap 15b.
⟨proof K7 16b⟩ Referenced in scrap 16a.
⟨proof K8 17a⟩ Referenced in scrap 16a.
⟨proof L11a 11a⟩ Referenced in scrap 10a.
⟨proof L11 10a⟩ Referenced in scrap 9b.
⟨proof L1 9a⟩ Referenced in scrap 8.
⟨proof NF1 62b⟩ Referenced in scrap 20a.
⟨proof NF3 63a⟩ Referenced in scrap 20a.
⟨proof T0 69a⟩ Referenced in scrap 23b.
⟨proof T1 69b⟩ Referenced in scrap 23b.
⟨proof T2 2 25b⟩ Referenced in scrap 25a.
⟨proof T2 25a⟩ Referenced in scrap 23b.
⟨proof T3 72⟩ Referenced in scrap 23b.
⟨proof T4 73⟩ Referenced in scrap 23b.
⟨proofL11b 10b⟩ Referenced in scrap 10a.
⟨schemes 40f⟩ Referenced in scrap 6b.
⟨theorems 40d⟩ Referenced in scrap 6b.
⟨txtpr 7b⟩ Referenced in scrap 6a.
⟨uniqueness gcd 23a⟩ Referenced in scrap 21.
⟨vocabulary 7a⟩ Referenced in scrap 6b.

C.3 Some Keywords