

Formalizing Factorization on Euclidean Domains and Abstract Euclidean Algorithms*

Thaynara Arielly de Lima

Universidade Federal de Goiás, Brasil

thaynaradelima@ufg.br

Andréia Borges Avelar

Universidade de Brasília, Brasil

andreiaavelar@unb.br

André Luiz Galdino

Universidade Federal de Catalão, Brasil

andregaldino@ufcat.edu.br

Mauricio Ayala-Rincón

Universidade de Brasília, Brasil

ayala@unb.br

This paper discusses the extension of the PVS sub-theory for rings, part of the PVS algebra theory, with theorems related to the division algorithm for Euclidean rings and Unique Factorization Domains that are general structures where an analogous of the Fundamental Theorem of Arithmetic holds. First, we formalize the general abstract notions of divisibility, prime and irreducible elements in commutative rings, essential to dealing with unique factorization domains. Then, we formalize the landmark theorem establishing that every principal ideal domain is a unique factorization domain. Finally, we specify the theory of Euclidean domains and formally verify that the rings of integers, the Gaussian integers, and arbitrary fields are Euclidean domains. To highlight the benefits of such a general abstract discipline of formalization, we specify a Euclidean gcd algorithm for Euclidean domains and formalize its correctness. Also, we show how this correctness is inherited under adequate parameterizations for the structures of integers and Gaussian integers.

1 Introduction

The NASA PVS algebra library ([4]) was recently enriched with a series of theorems related to the theory of rings. The extension includes complete formalizations of the isomorphism theorems for rings, principal and prime and maximal ideals, and a general abstract version of the Chinese Remainder Theorem (CRT) which holds for abstract rings, including non-commutative rings. The benefit of formalizing algebraic results from this abstract theoretical perspective was made evident showing how, from the abstract version of CRT, the well-known numerical version of CRT for the ring of integers \mathbb{Z} was formalized [21].

In this work, we give another substantial step towards enriching the PVS abstract algebra library by formalizing properties about factorization in commutative rings regarding both unique factorization domains and Euclidean rings. Roughly, unique factorization domains are abstract structures for which a general version of the Fundamental Theorem of Arithmetic holds. On the other hand, Euclidean rings are equipped with a norm that allows defining a suitable generalization of Euclid's division lemma and consequently of notions such as greatest common divisor (gcd). The practicality of gcd is well-known in the ring \mathbb{Z} . Nevertheless, mathematicians know this notion is of general fundamental importance in abstract Euclidean domains for which in general, gcd should and may be defined in different manners.

Figure 1 highlights the subtheories subject of the extension to the PVS theory algebra discussed in this paper. The red ones are related to Euclidean rings and gcd algorithms for Euclidean domains,

*Project supported by FAPDF DE 00193.00001175/21-11 and CNPq Universal 409003/21-2 grants. Last author partially funded by CNPq grant 313290/21-0.

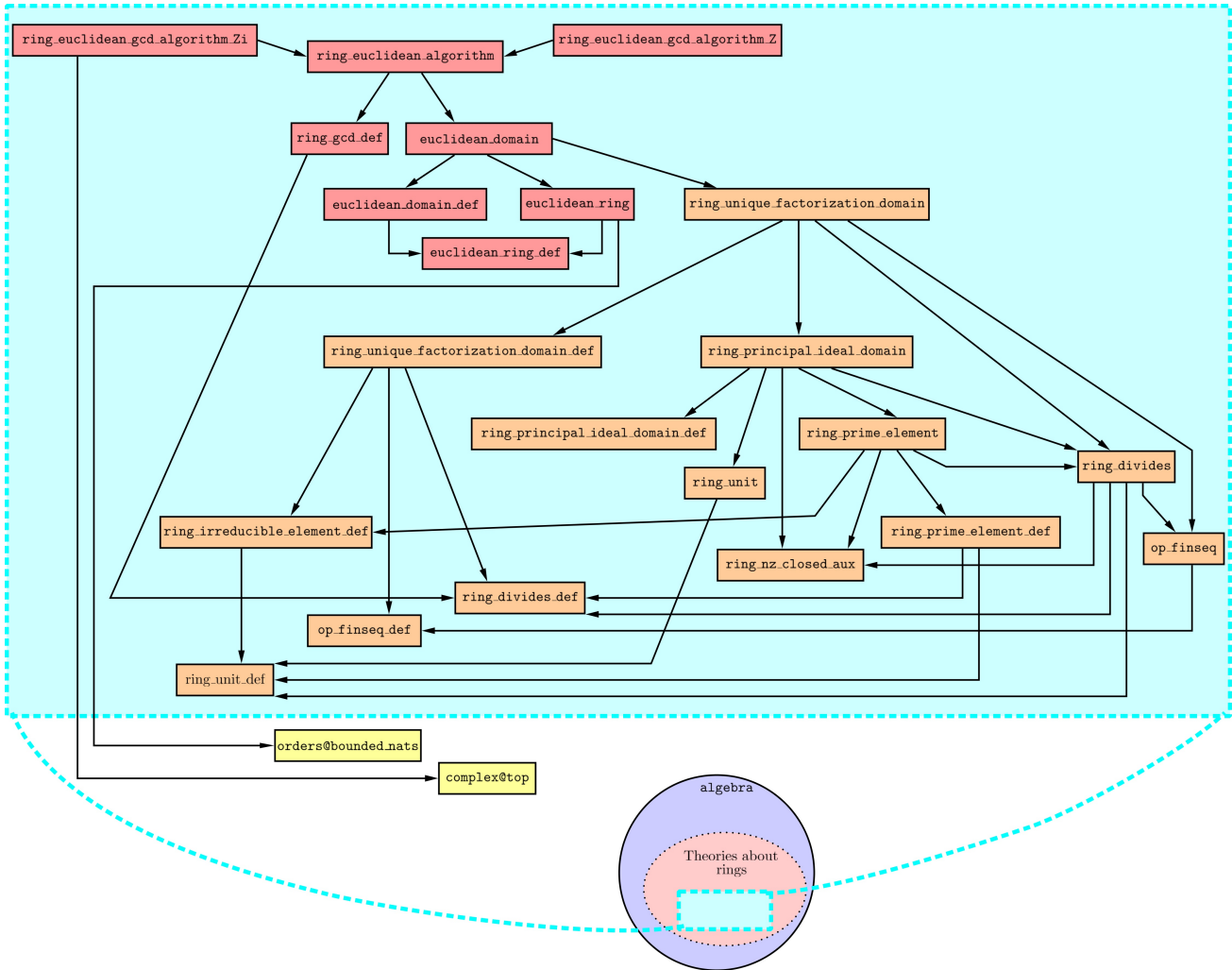


Figure 1: Ring theories expanding the PVS algebra library

and the orange ones are those related to unique factorization domains. The extension includes 210 new formulas enlarging the theory `algebra` from 1356 (cf [21]) to 1566 formalized lemmas.


The main motivation to formalize such structures is due to their potential theoretical and practical applications. Using the example of gcd, one can provide a general abstract version of the Euclidean algorithm to determine a gcd between two elements (Euclidean gcd algorithm) in a Euclidean domain. Since the ring of integers \mathbb{Z} , the Gaussian integers $\mathbb{Z}[i]$ (which are the subset of complex numbers whose real and imaginary parts are integer numbers) and rings of polynomials over integral domains are particular Euclidean domain structures, the Euclidean gcd algorithm can be applied over them, in a relatively straightforward manner, to compute gcds in different manners. Not only for the above mentioned structures, but for a variety of Euclidean domains.

Also, every element of a unique factorization domain can be factorized as a finite number of irreducible elements, and one can prove that Euclidean domains are unique factorization domains. These properties allow us to introduce modular arithmetic, and verify generic versions of Euler's Theorem and Fermat's Little Theorem for Euclidean domains, and promote factorization in Euclidean domains as a

convenient feature to develop efficient algorithms in symbolic computation [20], [11]. Thus, a formalization of the main results about unique factorization and Euclidean domains would allow the formal verification of more complex theories involving such structures in their scope.

The main contributions of this paper are listed below.

- We formalize the abstract notions of divisibility, prime and irreducible elements in commutative rings, which are essential to deal with unique factorization domains. In integral domains, prime elements are irreducible. The converse is not true in general. Among other properties, we formalize the theorem that establishes that in principal ideal domains (as it is well-known, it holds in \mathbb{Z}) irreducible elements are also prime.
- We specify unique factorization domains and formalize the theorem that every principal ideal domain is a unique factorization domain, which is a landmark result in abstract algebra.
- We specify the notion of Euclidean domain and formally verify that the rings \mathbb{Z} and $\mathbb{Z}[i]$, and any arbitrary field are Euclidean domains.
- We specify the general abstract notion of gcd for commutative rings, providing a general Euclidean gcd algorithm for Euclidean domains and formalize its correctness. Using this result, we parameterize the adequate norms and gcd relations for the rings \mathbb{Z} and $\mathbb{Z}[i]$; thus, obtaining in a straightforward manner the correctness of such instantiations of the abstract algorithm for these Euclidean domains. In this manner, we illustrate the benefits of maintaining the abstract general discipline of formalization for algebraic theories and the potential of such a discipline for application in concrete algebraic structures.

Organization of the paper. Section 2 presents a theoretical overview of unique factorization and Euclidean domains, pointing out the main concepts and results. Also, it comments on some differences between pen-and-paper proofs presented in Hungerford's textbook [17] and this formalization. Section 3 discusses the aspects of the formalization of the Euclidean gcd Algorithm for Euclidean Domains, as well as its application for two particular cases. Section 4 discusses related work and work in progress. Finally, Section 5 concludes and suggests future work. The formalizations were developed using the Prototype Verification System (PVS) and are available at [algebra](#) .

2 Formalization of Euclidean Domains

Notions such as prime element, division, and gcd between two elements and some landmark results, including the Fundamental Theorem of Arithmetic, Euclid's division lemma, and Euclidean Algorithm, are well established and widespread for the ring of integers. Such concepts and general versions of exciting results are extended for abstract algebraic structures ([17], [9], [12]) and are the scope of our formalization.

This section gives both a theoretical overview of the central notions and properties and discusses the PVS features used in their formalization. In addition, to highlight crucial differences between pen-and-paper vs formalized proofs, some analytical concepts and results are presented as enunciated in Chapter III of Hungerford's textbook [17].

2.1 Prime and irreducible elements on rings

The definitions of prime and irreducible elements rely on the general concept of divisibility on a ring. The specification of the notions of divisibility and associated elements are specified as the curried pred-

Specification 1: Divisibility and associated elements in the sub-theory `ring_divides_def` [↗](#)

```

R: VAR (ring?)
a, b: VAR T

divides?(R)(a: (R - {zero}), b: (R)): bool = EXISTS (x: (R)): a*x = b

associates?(R)(a,b:(R - {zero})): bool = divides?(R)(a,b) AND
                                         divides?(R)(b,a)

```

icates given in Specification 1. These predicates are abstracted for any ring structure given as their first argument, R .

In Hungerford’s textbook, the definition of divisibility relies on a commutative ring. It avoids the discrimination between an element’s left or right divisor, and since the main results demand a commutative ring in the hypothesis, it is a reasonable requirement. However, notice that commutativity is not a crucial property in such a notion since it only depends on the operation of multiplication in a ring. Because of that, we opted to generalize the definition and specify divisibility on non necessarily commutative rings as $(\text{divides?}(R)(a,b))$. Another interesting remark is related to the specification of $\text{associates?}(R)(a,b)$: Hungerford’s textbook omits that the type of the parameters a and b are non-zero elements. Of course, this is obvious since it is required in the definition of $\text{divides?}(R)(a,b)$. However, the lack of such a hypothesis is recurrent in several statements along the textbook that require it (for example, in Theorem 2.1).

In the sub-theory `ring_divides` [↗](#), we formalized the properties related to the divisibility stated in Theorem 2.1. Some of them involve the object “unit”.

In a ring $(R, +, *, \text{zero}, \text{one})$ with multiplication identity one , an element u is called a *unit* if u is left- and right-invertible; that is, if there exist elements $u_1^{-1}, u_2^{-1} \in R$ such that $u * u_1^{-1} = u_2^{-1} * u = \text{one}$.

Theorem 2.1 (Th.3.2, Hungerford [17]). *Let a, b and u be elements of a commutative ring R with identity.*

- (i) *a divides b (denoted as $a \mid b$) if and only if $(b) \subset (a)$, where (x) denotes the principal ideal generated by x .*
- (ii) *a and b are associates if and only if $(a) = (b)$.*
- (iii) *u is a unit if and only if $u \mid r$ for all $r \in R$.*
- (iv) *u is a unit if and only if $(u) = R$.*
- (v) *The relation “ a and b are associates” is an equivalence relation on R .*
- (vi) *If $a = br$, where $r \in R$ is a unit, then a and b are associates. If R is an integral domain, then the converse is true.*

Theorem 2.1 has a straightforward formalization due to the robustness of the formal framework previously developed for rings and principal ideals [21]. The formalization of the properties (i), (ii), and (iv) illustrates it clearly. In fact, by definition, (a) denotes the intersection of all ideals in a ring R containing the element a . The lemma `principal_ideal_charac` [↗](#) in theory `ring_principal_ideal` characterizes (a) as the set `one_gen(R)(a)` [↗](#) in the theory `ring_one_generator`. The last characterization depends on a sum, specified as `R_sigma`, over elements of a function in the ring R , defined over abstract types, as given in the theory `ring_basic_properties` [↗](#). The constructor `R_sigma` generalizes constructors in the `nasalib` built for specific theories as the theory of reals. Also, since R is a commutative

Specification 2: Definitions of irreducible and prime elements in the subtheories `ring_irreducible_element_def` [↗](#) and `ring_prime_element_def` [↗](#), respectively

```
R: VAR (ring_with_one?)
R_irreducible_element?(R)(x:(R)): bool = x/=zero AND
  (NOT unit?(R)(x)) AND
  (FORALL (a,b:(R)): x = a*b IMPLIES (unit?(R)(a) OR unit?(R)(b)))
%-----
R_prime_element?(R)(x:(R)): bool = x/=zero AND (NOT unit?(R)(x)) AND
  (FORALL (a,b:(R)): divides?(R)(x, a*b) IMPLIES
    divides?(R)(x, a) OR divides?(R)(x, b))
```

ring with identity, the lemma `commutative_id_one_gen_charac` [↗](#) provides a much simpler characterization of the set `one_gen(R)(a)`; indeed, such characterization simplifies the analysis of properties (i), (ii), and (iv) since (a) can be built as the set $aR = \{ar : r \in R\}$.

From the concepts of divisibility and unit, we specified prime and irreducible elements on a ring with identity as the predicates given in the Specification 2.

In the ring of integers, prime and irreducible elements are indistinguishable. However, this is not true for all rings. For instance, 2 is prime but not irreducible in \mathbb{Z}_6 . Theorem 2.2 gives some properties regarding prime and irreducible elements formalized in the subtheories `ring_prime_element` [↗](#) and `ring_principal_ideal_domain` [↗](#). Among others, it shows that prime and irreducible elements are equal over principal ideal domains.

Theorem 2.2 (Th.3.4, Hungerford [17]). *Let p and c be nonzero elements in an integral domain R .*

- (i) p is prime if and only if (p) is a nonzero prime ideal;
- (ii) c is irreducible if and only if (c) is maximal in the set S of all proper principal ideals of R .
- (iii) Every prime element of R is irreducible.
- (iv) If R is a principal ideal domain, then p is prime if and only if p is irreducible.
- (v) Every associate of an irreducible [resp. prime] element of R is irreducible [resp. prime].
- (vi) The only divisors of an irreducible element of R are its associates and the units of R .

Although the result is stated for integral domains, Hungerford advises that a weakened hypothesis can be considered in some parts of the theorem. We formalize the results using the minimum number of required conditions and detect that items (i) and (vi) of the Theorem 2.2 hold for commutative rings with identity.

Properties (i), (ii), and (iii) form the basis for the formalization of the characterization of primes as irreducible elements over principal ideal domains, given in property (iv) and specified as the lemma `PID_prime_el_iff_irreducible` [↗](#). The sufficiency of the property (iv), established in the property (iii), is verified as the lemma `prime_el_is_irreducible` [↗](#). It follows in a relatively straightforward manner from the definition of prime elements and the result that the multiplicative cancelation law holds for non-zero elements in integral domains (lemma `nzd_R_cancel_left` [↗](#)) since integral domains have not zero divisors.

On the other hand, the necessity of (iv) is trickier since it depends on properties (i), (ii), and other additional previous results developed for rings with identity and maximal ideals. Property (i) is specified by the lemma `prime_el_iff_prime_ideal` [↗](#). Its proof depends on the lemmas `prime_ideal_prop1`

Specification 3: Theory `ring_unique_factorization_domain_def` [↗](#) with the definition of unique factorization domain

```

fsIr?(R)(fsI: finseq[(R)]): bool = FORALL (i: below[length(fsI)]):
  R_irreducible_element?(R)(fsI(i))

unique_factorization_domain?(R): bool = integral_domain_w_one?(R) AND
FORALL(a: (R)): a /= zero AND NOT unit?(R)(a) IMPLIES
  EXISTS(fsI:(fsIr?(R))): a = op_fseq(fsI) AND
  FORALL(fsIp:fsIr(R)): a = op_fseq(fsIp) IMPLIES length(fsI) = length(fsIp) AND
  EXISTS(phi:[below[length(fsI)]->below[length(fsI)]]): (bijective?(phi)) AND
  FORALL(i:below[length(fsI)]): associates?(R)(fsIp(phi(i)),fsI(i))

```

and `prime_ideal_prop2` [↗](#) formalized in theory `ring_prime_ideal`, which provide a characterization of prime ideals over commutative rings. Lemma `el_irred_iff_one_gen_maximal` [↗](#) specifies property property (ii), which establishes that the principal ideal generated by an irreducible element is a maximal element in the set S of all proper principal ideals of a ring. It is important to stress here that in the pen-and-paper proof of property (iv) given in [17], Hungerford assumes the vital result that maximal elements in the previously mentioned set S are maximal ideals in R . We formalized this property without this assumption as the lemma `el_max_iff_one_gen_maximal` [↗](#) in the sub-theory `ring_principal_ideal_domain`. Finally, the necessity of property (iv) is concluded as follows. If p is an irreducible element, then (p) is a maximal element, according to `el_max_iff_one_gen_maximal`. Since R is a ring with identity, $R^2 = R$ by lemma `ring_w_one_is_idempotent` [↗](#), which is formalized in the sub-theory `ring_with_one_basic_properties`. Consequently, (p) is a prime ideal by the lemma `maximal_prime_ideal` [↗](#) and, by property (i), p is a prime element.

2.2 Unique Factorization Domains

The well-known Fundamental Theorem of Arithmetic for integers states that any positive integer greater than 1 can be factorized as a unique product of primes unless a permutation of such factors. Unique Factorization Domains (UFDs) are integral domains with an analogous of such theorem. The Specification 3 shows the definition of UFDs. It depends on a sequence of irreducible elements `fsIr?(R)(fsI)` on a ring R with identity and a recursive operator `op_fseq(fsI)`, as specified in the sub-theory `op_finseq_def` [↗](#), which multiplies the elements of such a sequence. The operator `op_fseq(fsI)` is specified over an abstract structure $(T, *, one)$ equipped with a binary operation $*$ and a constant one .

From the point of view of formalization, such a general specification is very useful for two reasons: firstly, it allows the use of the operator `op_fseq(fsI)` in a variety of abstract and concrete structures (monoids, monads, groups, rings, integers, reals) by only adequately parameterizing the sub-theory `op_finseq_def`; secondly, it avoids proof obligations, called in PVS *Type Correctness Conditions (TCCs)*, automatically generated by the system, since the operator is defined for elements of an abstract type, which provides more automation in our formal verification. Indeed, suppose such an operator was defined over elements of an algebraic structure, for example, a monad. To each application of that definition in a specific context, PVS will automatically generate a proof obligation to verify that `op_fseq(fsI)` acts on a sequence whose elements belong to a monad. This specification design would make the theory verification more onerous. It is advantageous to use polymorphism to formalize concepts and properties that hold for a non-interpreted type since it allows the reuse of such results in multiple contexts.

In sub-theory `ring_unique_factorization_domain`, we formalized the Theorem 2.3, which is a

Specification 4: Auxiliary function to build an ascending chain of ideals

```

phi(n:nat, R:principal_ideal_domain, a:(non_fact_el_set(R))):
RECURSIVE (non_fact_el_set(R)) =
  IF n = 0 then a
  ELSE choose ({x : (non_fact_el_set(R)) |
               strict_subset?(one_gen(R)(phi(n-1, R, a)), one_gen(R)(x))})
  ENDIF MEASURE n

```

landmark result about UFDs.

Theorem 2.3 (Th.3.7, Hungerford [17]). *Every principal ideal domain is a unique factorization domain.*

The formalization of the Theorem 2.3 has two main steps that we briefly comment on the following.

Step 1 - Existence of a factorization

First, previous subtheories established in the PVS theory algebra were enriched with auxiliary results. The new lemma `chain_ideal_union_ideal`, which states that the union of a chain of ideals in a ring R is an ideal, is included in the sub-theory `ring_ideal`. The new lemma `nonzero_ring_exists_maximal_ideal_aux`, which proves that every ideal in a ring R with identity, except R itself, is contained in a maximal ideal in R , is added to the sub-theory `ring_with_one_maximal_ideal`.

The formalization of this lemma considers an ideal $A \neq R$, $S = \{B \subset R; B \text{ is ideal in } R, B \neq R \text{ and } A \subset B\}$ and $\mathcal{C} = \{C_i \mid i \in I\}$ an arbitrary chain of ideals in S . We prove that the ideal $C = \bigcup C_i$ is an upper bound of the chain \mathcal{C} in S and, by using Zorn's lemma (available in the NASA PVS theory orders), we conclude that S has a maximal element, which is a maximal ideal in R . In the sub-theory `ring_principal_ideal`, we add the new lemma `stable_chain`, which states that if R is a principal ideal ring and $(a_1) \subset (a_2) \dots$ is a chain of ideals in R , then for some positive integer n , $(a_j) = (a_n)$ for all $j \geq n$. The new lemma `nonzero_nonunit_irreducible_divides`, formalized in the sub-theory `ring_principal_ideal_domain`, states that every nonzero and non-unit element in a principal ideal domain is divided by an irreducible element.

We conclude Step 1 by verifying that the subset of R below, a principal ideal domain, is empty.

$$\text{non_fact_el_set}(R) = \left\{ x : \begin{array}{l} x \text{ is a nonzero non-unit element} \\ \text{in } R \text{ and cannot be finitely} \\ \text{factorized into irreducible elements} \end{array} \right\}$$

In fact, if $a \in \text{non_fact_el_set}(R)$, we could build an ascending chain of ideals, $(a) \subset (a_1) \subset \dots$, which contradicts the lemma `stable_chain`. The key to verifying such fact was to specify the recursive function `phi(n, R, a)` showed in Specification 4 (sub-theory `ring_principal_ideal_domain`) and verify that it is well defined whenever `non_fact_el_set(R)` is non-empty.

Whenever $a \in \text{non_fact_el_set}(R)$, the choice of the element a_1 , obtained by the function `choose` in Specification 4, is guaranteed. In fact, the lemma `nonzero_nonunit_irreducible_divides` ensures that $a = ca_1$, where c is irreducible. It implies that a_1 belongs to `non_fact_el_set(R)` and satisfies the condition $(a) \subset (a_1)$ by Theorem 2.1(i).

Step 2: “Uniqueness” of a factorization

We mean “uniqueness”, the existence of a bijective function between the elements of two factorizations mapping associated elements. First, we formalized the lemma `prime_el_divides` (sub-theory `ring_prime_element`) which states if a prime element p in an integral domain divides the product $a_1 \dots a_n$ then there exists $1 \leq i \leq n$ such that p divides a_i . By 2.2(iii), it holds if p is an irreducible

Specification 5: Definitions of Euclidean rings and Euclidean domains

```

euclidean_ring?(R): bool = commutative_ring?(R) AND
EXISTS (phi: [(R - {zero}) -> nat]): FORALL(a,b: (R)):
  ((a*b /= zero IMPLIES phi(a) <= phi(a*b)) AND
   (b /= zero IMPLIES EXISTS(q,r:(R)):
     (a = q*b+r AND (r = zero OR (r /= zero AND phi(r) < phi(b))))))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

euclidean_domain?(R): bool = euclidean_ring?(R) AND integral_domain_w_one?(R)

```

element. From this, if $a_1 \dots a_n = a = b_1 \dots b_m$, where $a_i, 1 \leq n$ and $b_j, 1 \leq m$ are irreducible elements, then a_1 divides b_j , for some j . By Theorem 2.2(vi), a_1 and b_j are associates. Using induction on n , we prove that $n = m$ and establish the required bijective function.

2.3 Euclidean Rings

A Euclidean ring is a commutative ring R equipped with a norm φ over $R - \{zero\}$, where an abstract version of the well-known Euclid's division lemma holds. Euclidean rings and domains are specified in the subtheories `euclidean_ring_def` [↗](#) and `euclidean_domain_def` [↗](#) (Specification 5).

In sub-theory `euclidean_domain` [↗](#), we formalized that elements of Euclidean ring can be factorized as irreducible elements by verifying Theorem 2.4.

Theorem 2.4 (Th.3.9, Hungerford [17]). *A Euclidean ring R is a principal ideal ring with identity. Consequently, every Euclidean domain is a unique factorization domain.*

The verification makes use of the well-ordering principle over $\varphi(I^*) = \{\varphi(x) \in \mathbb{N}; x \in I - \{zero\}\}$, where I is a nonzero ideal in R and φ is a norm on $R - \{zero\}$. By choosing $a \in I$ such that $\varphi(a)$ is the minimum element of $\varphi(I^*)$, $b \in I$ satisfies $b = qa + r$, for some $q \in R$ and $r \in I$. From this, we infer that $r = 0$, since $r \neq 0$ contradicts the minimality of $\varphi(a)$. Consequently, $b = qa$ and $I \subset Ra \subset (a) \subset I$, guaranteeing that every ideal in R is a principal ideal. By Theorem 2.3, we have that a Euclidean principal ideal domain is a unique factorization domain.

In sub-theory `euclidean_domain` [↗](#), we also formalized the results stating that the ring of integers ([↗](#)) and any arbitrary field ([↗](#)) are Euclidean domains.

3 Formalization of gcd Algorithm for Euclidean Domains

The theory `Euclidean_ring_def` [↗](#) includes two additional definitions to allow abstraction of acceptable Euclidean norms and associated functions fulfilling the properties of Euclidean rings (see Specification 6).

The first definition is the relation `Euclidean_pair?` [↗](#). Given a Euclidean ring R and a Euclidean norm of non-zero elements over the naturals $\phi : R \setminus \{zero\} \rightarrow \mathbb{N}$, the predicate `Euclidean_pair?(R, ϕ)` holds whenever ϕ satisfies the constraints of a Euclidean norm over R .

The second definition is the curried relation given as `Euclidean_f_phi?(R, ϕ)(f_ϕ)` [↗](#). This relation holds whenever `Euclidean_pair?(R, ϕ)` holds, and f_ϕ is a function from $R \times R \setminus \{zero\}$ to $R \times R$, such that for all pair of elements of R in its domain, $f_\phi(a, b)$ gives a pair of elements, say (div, rem) satisfying the constraints of Euclidean rings regarding the norm ϕ : if $a \neq zero$, $a = div * b + rem$, and

Specification 6: Additional definitions in the sub-theory `Euclidean_ring_def`

```

Euclidean_pair?(R : (Euclidean_ring?), phi: [(R - {zero}) -> nat]) : bool =
  FORALL(a,b: (R)): ((a*b /= zero IMPLIES phi(a) <= phi(a*b)) AND
    (b /= zero IMPLIES
      EXISTS(q,r:(R)): (a = q*b+r AND
        (r = zero OR (r /= zero AND phi(r) < phi(b)))))

Euclidean_f_phi?(R : (Euclidean_ring?),
  phi : [(R - {zero}) -> nat] | Euclidean_pair?(R,phi))
(f_phi : [(R) , (R - {zero}) -> [(R),(R)]] : bool =
  FORALL (a : (R), b : (R - {zero})):
    IF a = zero THEN f_phi(a,b) = (zero, zero)
    ELSE LET div = f_phi(a,b)'1, rem = f_phi(a,b)'2 IN
      a = div * b + rem AND
      (rem = zero OR (rem /= zero AND phi(rem) < phi(b)))
  ENDIF

```

if $rem \neq zero$, $\phi(rem) < \phi(b)$. These definitions are correct since the existence of such a ϕ and f_ϕ is guaranteed by the fact that R is a Euclidean ring. Also, notice that the decrement of the norm, i.e., $\phi(rem) < \phi(b)$, is the key to building an abstract Euclidean terminating procedure.

Using the previous two relations, a general abstract recursive Euclidean gcd algorithm is specified in the sub-theory `ring_euclidean_algorithm` [↗](#) as the curried definition `Euclidean_gcd_algorithm` [↗](#) (See Specification 7). The correctness of this algorithm is guaranteed by the types of its arguments. Indeed, since allowed arguments R, ϕ , and f_ϕ should satisfy `Euclidean_f_phi?(R, \phi)(f_\phi)`, R is a Euclidean ring with associated Euclidean norm ϕ and adequate division and remainder functions given by f_ϕ . The termination of the algorithm is a proof obligation [↗](#) (termination TCC) automatically generated by PVS. Termination is proved using the lexicographical MEASURE of the algorithm provided in the specification. This measure decreases after each possible recursive call: for `Euclidean_gcd_algorithm(R, \phi, f_\phi)(a, b)`, if $a \neq zero$, $\phi(a) \geq \phi(b)$ and $rem \neq zero$, the recursive call is `Euclidean_gcd_algorithm(R, \phi, f_\phi)(b, rem)`; thus, the pair $(\phi(b), \phi(a))$ is lexicographically greater than $(\phi(rem), \phi(b))$, since $\phi(b) > \phi(rem)$.

In the other case, the recursive call is `Euclidean_gcd_algorithm(R, \phi, f_\phi)(b, a)`. This happens if $a \neq zero$, and $\phi(b) > \phi(a)$; therefore, $(\phi(b), \phi(a))$ is lexicographically greater than $(\phi(a), \phi(b))$.

It is worth mentioning that such termination TCCs are generated automatically by PVS, but in general, as in this case, the mandatory proof must be formalized manually.

The proof of correctness of the recursive algorithm is given as a straightforward corollary of the `Euclid_theorem` [↗](#) (in Specification 7) that establishes the correctness of each recursive step regarding the abstract definition of `gcd` [↗](#) given in Specification 8. Essentially, this theorem states that given an adequate Euclidean norm ϕ and associated function f_ϕ , the gcd of a pair (a, b) is equal to the gcd of the pair (rem, b) , where rem is computed through f_ϕ , i.e., rem is equal to the second projection of $f_\phi(a, b)$. Notice that since Euclidean rings allow a variety of Euclidean norms and associated functions (e.g., [17], [12]), the definition of gcd is not specified as a function but as the relation “gcd?”.

Finally, the proof of correctness of the abstract Euclidean algorithm is obtained by induction, using the lexicographic MEASURE of the algorithm. The theorem `Euclidean_gcd_alg_correctness` [↗](#) (in Specification 7) formalizes this fact. For an input pair (a, b) , in the inductive step of the proof, when $\phi(b) > \phi(a)$ and the recursive call swaps the arguments, one assumes that

$$\text{gcd?}(R)(\{b, a\}, \text{Euclidean_gcd_algorithm}(R, \phi, f_\phi)(b, a)),$$

Specification 7: Abstract gcd Euclidean algorithm for Euclidean rings in the sub-theory `ring_euclidean_algorithm` [↗](#)

```

Euclidean_gcd_algorithm(R : (Euclidean_domain?[T,+,*,zero,one]),
  (phi: [(R - {zero}) -> nat] | Euclidean_pair?(R,phi)),
  (f_phi: [(R),(R - {zero}) -> [(R),(R)]] |
    Euclidean_f_phi?(R,phi)(f_phi)))
(a: (R), b: (R - {zero})) : RECURSIVE (R - {zero}) =

IF a = zero THEN b
ELSIF phi(a) >= phi(b) THEN
  LET rem = (f_phi(a,b))'2 IN
  IF rem = zero THEN b
  ELSE Euclidean_gcd_algorithm(R,phi,f_phi)(b,rem)
ENDIF
ELSE Euclidean_gcd_algorithm(R,phi,f_phi)(b,a)
ENDIF
MEASURE lex2(phi(b), IF a = zero THEN 0 ELSE phi(a) ENDIF)

Euclid_theorem : LEMMA
FORALL(R:(Euclidean_domain?[T,+,*,zero,one]),
  (phi: [(R - {zero}) -> nat] | Euclidean_pair?(R, phi)),
  (f_phi: [(R),(R - {zero}) -> [(R),(R)]] |
    Euclidean_f_phi?(R,phi)(f_phi)),
  a: (R), b: (R - {zero}), g : (R - {zero})) :
  gcd?(R)({x : (R) | x = a OR x = b}, g) IFF
  gcd?(R)({x : (R) | x = (f_phi(a,b))'2 OR x = b}, g)

Euclidean_gcd_alg_correctness : THEOREM
FORALL(R:(Euclidean_domain?[T,+,*,zero,one]),
  (phi: [(R - {zero}) -> nat] | Euclidean_pair?(R, phi)),
  (f_phi: [(R),(R - {zero}) -> [(R),(R)]] |
    Euclidean_f_phi?(R,phi)(f_phi)),
  a: (R), b: (R - {zero}) ) :
  gcd?(R)({x : (R) | x = a OR x = b},
    Euclidean_gcd_algorithm(R,phi,f_phi)(a,b))

```

which means that $\text{Euclidean_gcd_algorithm}(R, \phi, f_\phi)(b, a)$ computes correctly the gcd of the pair (b, a) . From this assumption, one concludes that

$$\text{gcd?}(R)({a, b}, \text{Euclidean_gcd_algorithm}(R, \phi, f_\phi)(a, b)).$$

Otherwise, when the recursive call is $\text{Euclidean_gcd_algorithm}(R, \phi, f_\phi)(b, \text{rem})$, which happens if $\phi(a) \geq \phi(b)$, then $\text{rem} = (f_\phi(a, b))'2$, the second component of $f_\phi(a, b)$; by induction hypothesis one has that

$$\text{gcd?}(R)({b, \text{rem}}, \text{Euclidean_gcd_algorithm}(R, \phi, f_\phi)(b, \text{rem})).$$

Finally, by application of `Euclid_theorem`, one concludes that the abstract general Euclidean algorithm computes correctly a gcd for the pair (a, b) .

Now, we show how the correctness of the abstract algorithm `Euclidean_gcd_algorithm` is easily inherited, under adequate parameterizations, for the structures of integers \mathbb{Z} and Gaussian integers $\mathbb{Z}[i]$. The lines of reasoning follow those given in discussions on factorization in commutative rings and multiplicative norms in textbooks (e.g., Section 47 in [12], or Chapter 3, Section 3 in [17]).

The Specification 9 presents the case of the Euclidean ring \mathbb{Z} . The Euclidean norm $\phi_{\mathbb{Z}}$ is selected as the absolute value while the associated function $f_{\phi_{\mathbb{Z}}}$ is built using the integer division and remainder,

Specification 8: gcd definition for commutative rings - sub-theory ring_gcd_def [↗](#)

```

gcd?(R)(X: {X | NOT empty?(X) AND subset?(X,R)}, d:(R - {zero})): bool =
  (FORALL a: member(a, X) IMPLIES divides?(R)(d,a)) AND
  (FORALL (c:(R - {zero})):
    (FORALL a: member(a, X) IMPLIES divides?(R)(c,a)) IMPLIES
    divides?(R)(c,d))

```

Specification 9: Correctness of the parameterization of the abstract Euclidean algorithm for the Euclidean ring \mathbb{Z} - sub-theory ring_euclidean_gcd_algorithm_Z [↗](#)

```

phi_Z(i : int | i /= 0) : posnat = abs(i)

f_phi_Z(i : int, (j : int | j /= 0)) : [int, below[abs(j)]] =
  ((IF j > 0 THEN ndiv(i,j) ELSE -ndiv(i,-j) ENDIF), rem(abs(j))(i))

phi_Z_and_f_phi_Z_ok : LEMMA Euclidean_f_phi?[int,+,*,0](Z,phi_Z)(f_phi_Z)

Euclidean_gcd_alg_correctness_in_Z : COROLLARY
  FORALL(i: int, (j: int | j /= 0) ) :
    gcd?[int,+,*,0](Z)({x : (Z) | x = i OR x = j},
      Euclidean_gcd_algorithm[int,+,*,0,1](Z, phi_Z,f_phi_Z)(i,j))

```

specified in the PVS prelude libraries as `div` and `rem`: for $a \in \mathbb{Z}, b \in \mathbb{Z} \setminus \{0\}$, `div(a,b)` computes the integer division of a by b , and, for $b \in \mathbb{Z}^+ \setminus \{0\}$, `rem(b)(a)` computes the remainder of a by b .

The correctness of the Euclidean algorithm for the ring of integers is specified as the corollary `Euclidean_gcd_alg_correctness_in_Z` [↗](#). It states that for the Euclidean ring of integers \mathbb{Z} , and any $i, j \in \mathbb{Z}, j \neq 0$, the parameterized abstract algorithm, `Euclidean_gcd_algorithm[int,+,*,0,1]` satisfies the relation `gcd?[int,+,*,0]`:

$$\text{gcd?}[int, +, *, 0](\mathbb{Z})(\{i, j\}, \text{Euclidean_gcd_algorithm}[int, +, *, 0, 1](\mathbb{Z}, \phi_{\mathbb{Z}}, f_{\phi_{\mathbb{Z}}})(i, j))$$

The formalization of this corollary follows from the theorem of correctness for the abstract Euclidean algorithm, `Euclidean_gcd_alg_correctness` theorem (Specification 7), which essentially requires proving that the chosen Euclidean measure $\phi_{\mathbb{Z}}$, and the associated function $f_{\phi_{\mathbb{Z}}}$ fulfill the conditions in the definition of Euclidean rings. The latter is formalized as lemma `phi_Z_and_f_phi_Z_ok` [↗](#) : `Euclidean_f_phi?[int,+,*,0](Z,phi_Z)(f_phi_Z)`.

The Specification 10 presents the formalization of correctness of the Euclidean algorithm for the Euclidean ring $\mathbb{Z}[i]$ of Gaussian integers. The Euclidean norm of a Gaussian integer $x = (\text{Re}(x) + i \text{Im}(x)) \in \mathbb{Z}[i]$, $\phi_{\mathbb{Z}[i]}(x)$, is selected as the natural given by the multiplication of x by its conjugate (`conjugate(x) = Re(x) - i Im(x)`): $\text{Re}(x)^2 + \text{Im}(x)^2$. The construction of an adequate associated function $f_{\phi_{\mathbb{Z}[i]}}$ (`f_phi_Zi` in Specification 10) requires additional explanations and is specified through the auxiliary function `div_rem_appx` [↗](#). For a pair of integers (a,b) , $b \neq 0$, this function computes the pair of integers (q,r) such that $a = qb + r$, and $|r| \leq |b|/2$; thus, qb is the integer closest to a . The equality $a = qb + r$ is formalized as lemma `div_rev_appx_correctness` [↗](#). Several properties on the field of complex numbers are imported from the PVS complex theory.

Now, we explain the construction of the function $f_{\phi_{\mathbb{Z}[i]}}$ [↗](#). For y , a Gaussian integer and x , a positive integer, let $\text{Re}(y) = q_1x + r_1$ and $\text{Im}(y) = q_2x + r_2$, where (q_1, r_1) and (q_2, r_2) are computed with the

Specification 10: Correctness of the parameterization of the abstract Euclidean algorithm for $\mathbb{Z}[i]$ - sub-theory `ring_euclidean_gcd_algorithm_Zi` [↗](#)

```

Zi: set[complex] = {z : complex | EXISTS (a,b:int): a = Re(z) AND b = Im(z)}

Zi_is_ring: LEMMA ring?[complex,+,*,0](Zi)

Zi_is_integral_domain_w_one: LEMMA integral_domain_w_one?[complex,+,*,0,1](Zi)

phi_Zi(x:(Zi) | x /= 0): nat = x * conjugate(x)

phi_Zi_is_multiplicative: LEMMA
  FORALL((x: (Zi) | x /= 0), (y: (Zi) | y /= 0)):
    phi_Zi(x * y) = phi_Zi(x) * phi_Zi(y)

div_rem_appx(a: int, (b: int | b /= 0)) : [int, int] =
  LET r = rem(abs(b))(a),
      q = IF b > 0 THEN ndiv(a,b) ELSE -ndiv(a,-b) ENDIF IN
  IF r <= abs(b)/2 THEN (q,r)
  ELSE IF b > 0 THEN (q+1, r - abs(b))
  ELSE (q-1, r - abs(b))
  ENDIF
  ENDIF

div_rev_appx_correctness : LEMMA
  FORALL (a: int, (b: int | b /= 0)) :
    abs(div_rem_appx(a,b)^2) <= abs(b)/2 AND
    a = b * div_rem_appx(a,b)^1 + div_rem_appx(a,b)^2

f_phi_Zi(y: (Zi), (x: (Zi) | x /= 0)): [(Zi),(Zi)] =
  LET q = div_rem_appx(Re(y * conjugate(x)), x * conjugate(x))^1 +
      div_rem_appx(Im(y * conjugate(x)), x * conjugate(x))^1 * i,
      r = y - q * x IN (q,r)

phi_Zi_and_f_phi_Zi_ok: LEMMA
  Euclidean_f_phi?[complex,+,*,0](Zi,phi_Zi)(f_phi_Zi)

Euclidean_gcd_alg_in_Zi: COROLLARY
  FORALL(x: (Zi), (y: (Zi) | y /= 0)) :
    gcd?[complex,+,*,0](Zi)({z :(Zi) | z = x OR z = y},
      Euclidean_gcd_algorithm[complex,+,*,0,1](Zi, phi_Zi,f_phi_Zi)(x,y))

```

auxiliary function `div_rem_appx` (with respective inputs $(\text{Re}(y), x)$ and $(\text{Im}(y), x)$). Let $q = q_1 + iq_2$ and $r = r_1 + ir_2$, then $y = qx + r$. Also, notice that if $r \neq 0$ then $\phi_{\mathbb{Z}[i]}(r) \leq \phi_{\mathbb{Z}[i]}(x)$, since $r_1^2 + r_2^2 \leq x^2/2 \leq x^2$. For the case in which x is a non zero Gaussian integer, $\phi_{\mathbb{Z}[i]}(x) > 0$ holds.

Then, we can compute `div_rem_appx(y conjugate(x), x conjugate(x))`, obtaining $q, r' \in \mathbb{Z}[i]$ such that $y \text{ conjugate}(x) = q(x \text{ conjugate}(x)) + r'$, and $r' = 0$ or $\phi_{\mathbb{Z}[i]}(r') < \phi_{\mathbb{Z}[i]}(x \text{ conjugate}(x))$.

By selecting $r = y - qx$, we obtain $y = qx + r$ and $r \text{ conjugate}(x) = r'$.

Finally, when $r \neq 0$, since $\phi_{\mathbb{Z}[i]}(r \text{ conjugate}(x)) < \phi_{\mathbb{Z}[i]}(x \text{ conjugate}(x))$, by application of the lemma `phi_Zi_is_multiplicative` [↗](#), we conclude that $\phi_{\mathbb{Z}[i]}(r) < \phi_{\mathbb{Z}[i]}(x)$.

The formalization of correctness of the Euclidean algorithm for Gaussian integers obtained by parameterizations with $\mathbb{Z}[i]$, its Euclidean norm $\phi_{\mathbb{Z}[i]}$ and associated function $f_{\phi_{\mathbb{Z}[i]}}$ follows as the simple corollary `Euclidean_gcd_alg_in_Zi` [↗](#) in Specification 10. This is proved using the correctness of the abstract Euclidean algorithm (Specification 7) and lemma `phi_Zi_and_f_phi_Zi_ok` [↗](#). The latter states that the Euclidean norm $\phi_{\mathbb{Z}[i]}$ and its associated function $f_{\phi_{\mathbb{Z}[i]}}$ are adequate for the Euclidean ring $\mathbb{Z}[i]$: `Euclidean_f_phi?[complex,+,*,0](Z[i], phi_Z[i])(f_phi_Z[i])`.

4 Related Work, and work in progress

4.1 Related work

Several formalizations focus on specific ring structures as the ring of integers. Such developments range from simple formalization exercises, such as correctness proofs of gcd algorithms for \mathbb{Z} , to elaborated mechanical proofs of the Chinese Remainder theorem for \mathbb{Z} . The latter started from Zhang and Hua’s RRL (Rewrite Rule Laboratory) mechanization [29], followed by different approaches in Mizar, HOL Light, hol98, Coq [27], ACL2 [25], and VeriFun [28]. Nevertheless, the general algebraic abstract approach is followed by a few developments. In particular, such an approach is followed in the Isabelle/HOL Algebra Library (see [2], [1], and [3]); a library that provides a wide range of theorems on mathematical structures, including results on rings, groups, factorization over ideals, rings of integers and polynomial rings, as well as formalization of an algorithm to compute echelon forms over Euclidean domains, and so characteristic polynomials of matrices. Also, the Lean mathlib library [8] specifies unique factorization domains, prime and irreducible elements in commutative rings, and relations with principal ideal domains. In addition, it specifies the notion of gcd for Euclidean domains and formalizes several properties as the correctness of the extended Euclidean algorithm by applying Bézout’s gcd lemma. The library mathlib formalizes that a Euclidean domain is a principal ideal domain and a principal ideal domain is a unique factorization domain. The former is given as formally verified construction from a definition. From this instance, it is possible to infer that the Gaussian integers are a Euclidean domain and thus a principal ideal. Also, the Euclidean algorithm can be adapted to structures as the Gaussian integers. A recent extension of mathlib specifies the ring of Witt vectors and formalizes the isomorphism between the ring of Witt vectors over $\mathbb{Z}/p\mathbb{Z}$ and the ring of p -adic integers \mathbb{Z}_p , for a prime p [7].

In Coq, results about groups, rings, and ordered fields were formalized as part of the FTA project [14]; this work gave rise to the formalization of the Feit and Thompson’s proof of the Odd Order Theorem [15]. Also, there are formalizations in Coq of real ordered fields [6], finite fields [24], and rings with explicit divisibility [5]. In Nuprl and Mizar, there are proofs of the Binomial Theorem for rings in [18] and [26], respectively, and a Mizar formalization of the First Isomorphism Theorem for rings [19]. In ACL2, there exists a hierarchy of algebraic structures ranging from setoids to vector spaces that aims the formalization of computer algebra systems [16].

Regarding the paper-and-pen proofs in [17] and the formalization reported in this paper, the last one comprises about twenty pages of Hungerford’s textbook. We estimate it took ten months of human labor. Some results in the book appear as trivial remarks only. Nevertheless, they required the formalization of a significative sequence of auxiliary lemmas. An excellent example of the lack of details in this respect is a remark after Definition 3.5. in [17] that we have used in tutorials to motivate mathematicians to deal with proof assistants. It states that:

“every irreducible element in a unique factorization domain is necessarily prime by Definition 3.5. Consequently, irreducible and prime elements coincide, by Theorem 3.4.”

Indeed, the formalization of this remark required the application of additional properties related to bijective functions, the equivalence relation “associates”, and the composition of finite sequences, among others inherited from the abstract structure integral domain. Also, several particular cases had to be analyzed to ensure the result in the cases where the elements involved are units or equal to zero.

Finally, we would like to stress that the project is focused on the formalization side, but aspects related to code extraction can be explored through tools provided by PVS. For instance, PVSio is an animation tool that extends the ground evaluator of PVS with a predefined library of programming features

[22]. The evaluation is possible whenever algorithms are specified constructively. For our purposes, this means that we can run the formalized gcd algorithm with the help of the ground evaluator for any Euclidean domain for which the Euclidean norm ϕ and the associated function f_ϕ are specified constructively, which is the case of our specifications for \mathbb{Z} and $\mathbb{Z}[i]$. Elaborated approaches, based on PVSio, use this animation tool to evaluate the formal models on a set of randomly generated test cases comparing the computed results against output values obtained by actual software [10]. After applying such approaches, performance and comparison with other implementations would be possible.

4.2 Work in progress

Work in progress to be reported in the future includes formalizing the general theory of quaternions built from any abstract structure of fields specified in PVS as commutative division rings. The specification of quaternions is given from an abstract type T with binary operators for addition and multiplication, with constants `zero` and `one`, respectively. The type T with addition and `zero` is an Abelian group, and the multiplication is associative. The specification includes axioms for quaternion addition, and multiplication ($i^2 = a$, $j^2 = b$, for some given parameters a and b of T), associativity for quaternion multiplication, distributivity of quaternion addition and multiplication, and properties for the scalar product between elements of the field and of the quaternion. All that is provided in the theory `quaternion_def` [↗](#). Afterward, in the PVS theory `quaternions` [↗](#), using these axioms, a series of general properties of quaternions are provided, which range from the characterization of quaternion multiplication to the characterization of quaternions as division rings. Once again, following the general approach to specifying quaternions from abstract fields, we can obtain the specific structure of Hamilton's quaternions, using as the parameter to build the quaternions the specific field of reals. As far as we know, there are formalizations of Hamilton's quaternions in HOL Light and Isabelle/HOL (e.g., [13], [23]). In contrast, some elements of the general theory of quaternions built over any abstract field, as in our case, were developed as part of the Lean `mathlib` library [8].

We do not argue that any proof assistant is a better or worse framework than any other for formalizing algebraic notions and properties. However, we are confident that the current formalization work adequately explores the inductive and higher-order possibilities available in PVS and substantially contributes to completing the theory of the algebraic properties of rings by providing the most general and abstract possible presentation of such algebraic structures, as also given in some of the previous references, mainly as done by the approaches mentioned above in Isabelle/HOL and Lean ([1], [8]).

5 Conclusions and Future Work

In contrast to other works, restricted to specific ring structures, our formalization approach focuses on the theory of abstract rings, as done in the Lean- and Isabelle-related libraries (cf [8], and [3], respectively) discussed in the related work. Advantages of such an approach include increasing the interest of mathematicians in formalizations and having practical general presentations of computational algebraic properties portable to specific ring structures. In particular, in [21], the Chinese Remainder Theorem was formalized for (non-necessarily commutative) rings, obtaining, as a corollary, the CRT version for the ring of integers. This work substantially extends the `algebra` PVS library by specifying Euclidean rings and factorization domains, and formalizing the correspondence between principal ideal domains and unique factorization domains. Also, it proved the correctness of a general Euclidean gcd algorithm for Euclidean domains. The usefulness of such an abstract verified gcd algorithm is evident by its adapta-

tion to specific Euclidean domain structures. Indeed, this versatility is illustrated by showing how simple corollaries establish the correctness of the Euclidean algorithm (parameterized) for the rings of integers and Gaussian integers (\mathbb{Z} and $\mathbb{Z}[i]$).

In future work, we will include the specification of modular arithmetic and verification of generic versions of Euler’s Theorem and Fermat’s Little Theorem for Euclidean domains.

References

- [1] Jesús Aransay, Clemens Ballarin, Martin Baillon, Paulo Emílio de Vilhena, Stephan Hohe, Florian Kam-müller & Lawrence C. Paulson (2019): *The Isabelle/HOL Algebra Library*. Technical Report, Isabelle Li-brary, University of Cambridge Computer Laboratory and Technische Universität München. Available at <https://isabelle.in.tum.de/dist/library/HOL/HOL-Algebra/document.pdf>.
- [2] Jesús Aransay & Jose Divasón (2016): *Formalisation of the computation of the echelon form of a matrix in Isabelle/HOL*. *Formal Aspects Comput.* 28(6), pp. 1005–1026, doi:10.1007/s00165-016-0383-1.
- [3] Clemens Ballarin (2019): *Exploring the Structure of an Algebra Text with Locales*. *Journal of Automated Reasoning* 64, pp. 1093–1121, doi:10.1007/s10817-019-09537-9.
- [4] Ricky Butler & David Lester (2007): *A PVS Theory for Abstract Algebra*. Available at <http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/pvslib.html>. Accessed in March 31, 2019.
- [5] Guillaume Cano, Cyril Cohen, Maxime Dénès, Anders Mörtberg & Vincent Siles (2016): *Formalized linear algebra over Elementary Divisor Rings in Coq*. *Logical Methods in Computer Science* 12(2:7), pp. 1–23, doi:10.2168/LMCS-12(2:7)2016.
- [6] Cyril Cohen & Assia Mahboubi (2012): *Formal proofs in real algebraic geometry: from ordered fields to quantifier elimination*. *Logical Methods in Computer Science* 8(1:2), pp. 1–40, doi:10.2168/LMCS-8(1:2)2012.
- [7] Johan Commelin & Robert Y. Lewis (2021): *Formalizing the ring of Witt vectors*. In: *10th ACM SIGPLAN International Conference on Certified Programs and Proofs CPP*, ACM, pp. 264–277, doi:10.1145/3437992.3439919.
- [8] The mathlib Community (2020): *The Lean Mathematical Library*. In: *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020*, ACM, pp. 367–381, doi:10.1145/3372885.3373824.
- [9] David S. Dummit & Richard M. Foote (2003): *Abstract Algebra*, 3 edition. Wiley.
- [10] Aaron Dutle, César A. Muñoz, Anthony Narkawicz & Ricky W. Butler (2015): *Software Validation via Model Animation*. In: *Proc. 9th International Conference on Tests and Proofs (TAP@STAF)*, *Lecture Notes in Computer Science* 9154, Springer, pp. 92–108, doi:10.1007/978-3-319-21215-9_6.
- [11] Christian Eder, Gerhard Pfister & Adrian Popescu (2017): *On Signature-Based Gröbner Bases Over Euclidean Rings*. In Michael A. Burr, Chee K. Yap & Mohab Safey El Din, editors: *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2017, Kaiserslautern, Germany, July 25-28, 2017*, ACM, pp. 141–148, doi:10.1145/3087604.3087614.
- [12] John B. Fraleigh (2003): *A First Course in Abstract Algebra*, 7th edition. Pearson.
- [13] Andrea Gabrielli & Marco Maggesi (2017): *Formalizing Basic Quaternionic Analysis*. In: *Proc. 8th International Conference on Interactive Theorem Proving ITP*, *Lecture Notes in Computer Science* 10499, Springer, pp. 225–240, doi:10.1007/978-3-319-66107-0_15.
- [14] Herman Geuvers, Randy Pollack, Freek Wiedijk & Jan Zwanenburg (2002): *A Constructive Algebraic Hier-archy in Coq*. *Journal of Symbolic Computation* 34(4), pp. 271–286, doi:10.1006/jSCO.2002.0552.
- [15] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O’Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev,

- Enrico Tassi & Laurent Théry (2013): *A Machine-Checked Proof of the Odd Order Theorem*. In: *4th International Conference on Interactive Theorem Proving ITP, Lecture Notes in Computer Science 7998*, Springer, pp. 163–179, doi:10.1007/978-3-642-39634-2_14.
- [16] Jónathan Heras, Francisco Jesús Martín-Mateos & Vico Pascual (2015): *Modelling algebraic structures and morphisms in ACL2*. *Applicable Algebra in Engineering, Communication and Computing* 26(3), pp. 277–303, doi:10.1007/s00200-015-0252-9.
- [17] Thomas W. Hungerford (1980): *Algebra*. *Graduate Texts in Mathematics* 73, Springer-Verlag, New York-Berlin. Reprint of the 1974 original.
- [18] Paul Bernard Jackson (1995): *Enhancing the Nuprl Proof Development System and Applying it to Computational Abstract Algebra*. Ph.D. thesis, Cornell University. Available at <https://ecommons.cornell.edu/handle/1813/7167>.
- [19] Artur Kornilowicz & Christoph Schwarzweller (2014): *The First Isomorphism Theorem and Other Properties of Rings*. *Formalized Mathematics* 22(4), pp. 291–301, doi:10.2478/forma-2014-0029.
- [20] Daniel Lichtblau (2013): *Applications of Strong Gröbner Bases over Euclidean Domains*. *International Journal of Algebra* 7(8), pp. 369–390, doi:10.12988/ija.2013.13037.
- [21] Thaynara Arielly de Lima, André Luiz Galdino, Andréia Borges Avelar & Mauricio Ayala-Rincón (2021): *Formalization of Ring Theory in PVS*. *J. Autom. Reason.* 65(8), pp. 1231–1263, doi:10.1007/s10817-021-09593-0.
- [22] César. Muñoz & Richard Butler (2003): *Rapid prototyping in PVS*. Technical Report NASA/CR-2003-212418, NIA-2003-03, NASA Langley Research Center (NIA).
- [23] Lawrence C. Paulson (2018): *Quaternions*. *Arch. Formal Proofs* 2018. Available at <https://www.isa-afp.org/entries/Quaternions.html>.
- [24] Jade Philipoom (2018): *Correct-by-Construction Finite Field Arithmetic in Coq*. Master’s thesis, Master of Engineering in Computer Science, MIT. Available at <https://dspace.mit.edu/handle/1721.1/119582>.
- [25] David M. Russinoff (2000): *A Mechanical Proof of the Chinese Remainder Theorem*. UTCS Technical Report - no longer available - ACL2 Workshop 2000 TR-00-29, University of Texas at Austin. Available at <https://www.cs.utexas.edu/users/moore/acl2/workshop-2000/final/russinoff-short/paper.pdf>.
- [26] Christoph Schwarzweller (2003): *The Binomial Theorem for Algebraic Structures*. *Journal of Formalized Mathematics* 12(3), pp. 559–564. Available at <http://mizar.org/JFM/Vol12/binom.html>.
- [27] Christoph Schwarzweller (2009): *The Chinese Remainder Theorem, its Proofs and its Generalizations in Mathematical Repositories*. *Studies in Logic, Grammar and Rhetoric* 18(31), pp. 103–119. Available at <https://philpapers.org/rec/SCHTCR-12>.
- [28] Christoph Walther (2018): *A Machine Assisted Proof of the Chinese Remainder Theorem*. Technical Report VFR 18/03, FB Informatik, Technische Universität Darmstadt.
- [29] Hantao Zhang & Xin Hua (1992): *Proving the Chinese Remainder Theorem by the Cover Set Induction*. In: *11th International Conference on Automated Deduction CADE, Lecture Notes in Computer Science 607*, Springer, pp. 431–445, doi:10.1007/3-540-55602-8_182.