Exploring the benefits of a general abstract formalization

Thaynara Arielly de Lima
Universidade Federal de Goiás

Funded by CNPq Universal grant No. 313290/2021-0,
FAPEG Research grant No. 202310267000223

EuroProofNet Workshop 2024
2nd Workshop on the development, maintenance, refactoring and search of large libraries of proofs

September 13, 2024

Joint Work With

Bruno Berto de Oliveira Ribeiro

André Luiz Galdino

Andréia Borges Avelar

Mauricio Ayala-Rincón

Motivation

- Ring theory has a wide range of applications in several fields of knowledge:
  - combinatorics, algebraic cryptography and coding theory apply finite (commutative) rings [1];
  - ring theory forms the basis for algebraic geometry, which has applications in engineering, statistics, biological modeling, and computer algebra [7].

  A complete formalization of ring theory would make possible the formal verification of elaborated theories involving rings in their scope.

- Formalizing rings will enrich the mathematical libraries of PVS:

  https://github.com/nasa/pvslib/tree/master/algebra

algebraic theories $\rightsquigarrow$ algebraic structures

1^{st} 2^{nd}

Formalization approach

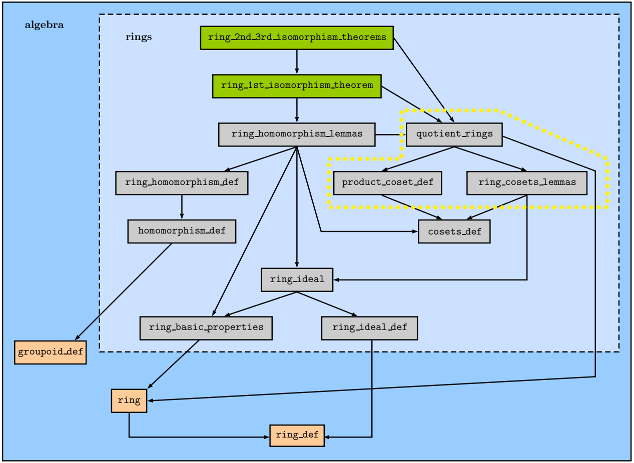


Figure: Hierarchy of the sub-theories for the three isomorphism theorems for rings (Taken from [2])

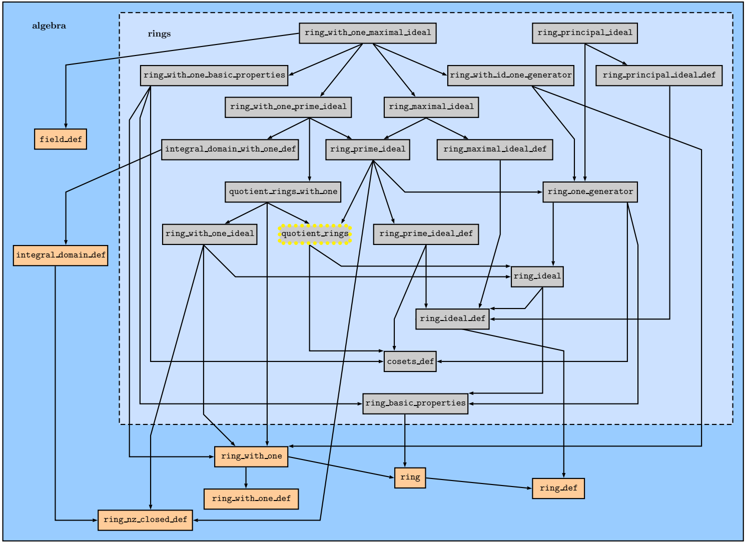
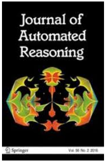


Figure: Hierarchy of the sub-theories related with principal, prime and maximal ideals (Taken from [2])



[2] de Lima, Galdino, Avelar, Ayala-Rincón
Formalization of Ring Theory in PVS: Isomorphism Theorems, Principal, Prime and Maximal Ideals, Chinese Remainder Theorem
Journal of Automated Reasoning, 2021

https://doi.org/10.1007/s10817-021-09593-0

- Formalization of the general algebraic-theoretical version of the Chinese remainder theorem (CRT) for the theory of rings, proved as a consequence of the first isomorphism theorem.
- The number-theoretical version of CRT for the structure of integers is obtained as a consequence.

CRT for rings $\rightsquigarrow$ CRT for $\mathbb{Z}$

1^{st} 2^{nd}

Formalization approach

CRT for integers

Consider m a positive integer such that $m = m_1 \cdot m_2 \ldots \cdot m_r$, where $gcd(m_i, m_j) = 1, i \neq j$. Then

$$Z_m \cong Z_{m_1} \times Z_{m_2} \times \ldots \times Z_{m_r}$$

↑

CRT for (non-necessarily commutative) rings

Let R be a ring and $A_1, A_2, \ldots A_r$ comaximal ideals of R $(A_i + A_j = R, i \neq j)$. Then

$$R/A_1 \cap A_2 \ldots \cap A_r \cong R/A_1 \times R/A_2 \times \ldots \times R/A_r$$

$$a = b * q + r \qquad 0 \leqslant r < b$$

$$19 = 5 * 3 + 4$$

$$5 = 4 * 1 + 1$$

$$4 = 1 * 4 + 0$$

gcd(a,b) = gcd(b,r)

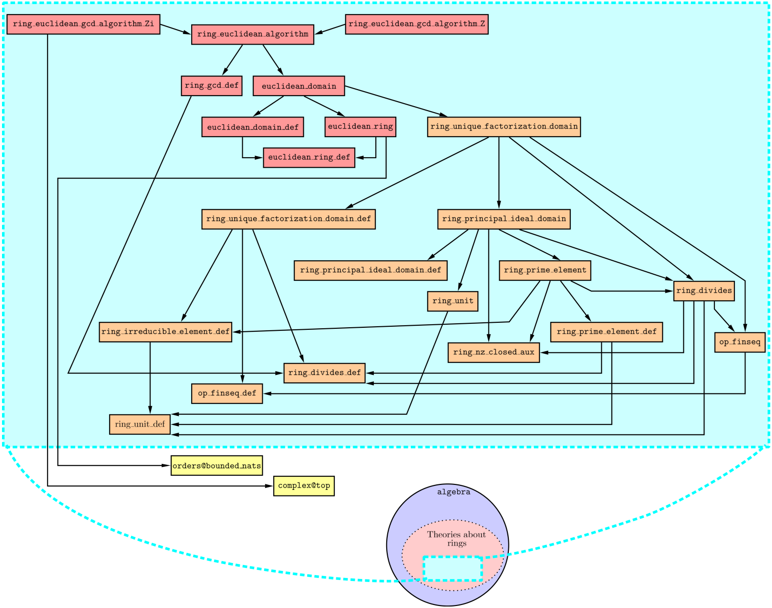


Figure: Euclidean Domains and Algorithms (Taken from [3])

A Euclidean ring is a commutative ring R equipped with a norm φ over $R \setminus \{zero\}$, 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`.

```
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)
```

The theory `Euclidean_ring_def` includes two additional definitions to allow abstraction of acceptable Euclidean norms, ϕ, and associated functions, f_ϕ, fulfilling the properties of Euclidean rings.

```
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
```

Using the previous two relations, a general abstract recursive Euclidean gcd algorithm is specified in the sub-theory `ring_euclidean_algorithm` as the definition `Euclidean_gcd_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)
```

The termination of the algorithm is guaranteed proving that proof obligations (termination Type Correctness Conditions - TCCs) generated by PVS hold. For instance:

```
euclidean_gcd_algorithm_TCC9: OBLIGATION
FORALL (R: (euclidean_domain?[T, +, *, zero, one]),
          (phi: [(difference(R, singleton(zero))) -> nat]
               | euclidean_pair?[T, +, *, zero](R, phi)),
          (f_phi: [[(R), (remove(zero, R))] -> [(R), (R)]]
               | euclidean_f_phi?[T, +, *, zero](R, phi)(f_phi)),
          a: (R), b: (remove[T](zero, R))):
    NOT a = zero AND phi(a) >= phi(b) IMPLIES
     FORALL (rem: (R)):
       rem = (f_phi(a, b))`2 AND NOT rem = zero IMPLIES
        lex2(phi(rem), IF b = zero THEN 0 ELSE phi(b) ENDIF) <
         lex2(phi(b), IF a = zero THEN 0 ELSE phi(a) ENDIF)
```

It uses the lexicographical `MEASURE` provided in the specification. The measure decreases after each possible recursive call.

The `Euclid_theorem` establishes the correctness of each recursive step regarding the abstract definition of `gcd`. It states that given adequate ϕ and f_ϕ, the `gcd` of a pair (a, b) is equal to the `gcd` of the pair (rem, b), where rem is computed by f_ϕ. Notice that since Euclidean rings allow a variety of Euclidean norms and associated functions (e.g., [6], [4]), `gcd` is specified as a relation.

```
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)
```

```
 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))
```

Finally, the theorem `Euclidean_gcd_alg_correctness` formalizes the correctness of the abstract Euclidean algorithm. The proof is by induction. 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 the lexicographic measure decreases. Otherwise, when the recursive call is `Euclidean_gcd_algorithm`$(R, \phi, f_\phi)(b, rem)$ the measure decreases and by application of `Euclid_theorem`, one concludes.

```
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))
```

Corollary `Euclidean_gcd_alg_correctness_in_Z` gives the Euclidean algorithm correctness for the Euclidean ring of integers, $\mathbb{Z}$. It states that the parameterized abstract algorithm, `Euclidean_gcd_algorithm[int,+,*,0,1]` satisfies the relation `gcd?[int,+,*,0]`, for any $i, j \in \mathbb{Z}, j \neq 0$.

It follows from the correctness of the abstract Euclidean algorithm and requires proving that $\phi_{\mathbb{Z}}$ and $f_{\phi_{\mathbb{Z}}}$ fulfill the definition of Euclidean rings. The latter is formalized as lemma `phi_Z_and_f_phi_Z_ok`.

```
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))
```

Correctness of the Euclidean algorithm for the Euclidean ring $\mathbb{Z}[i]$ of Gaussian integers.

The Euclidean norm of a Gaussian integer $x = (\mathtt{Re}(x) + i\,\mathtt{Im}(x)) \in \mathbb{Z}[i]$, $\phi_{\mathbb{Z}[i]}(x)$, is selected as the natural given by the multiplication of x by its conjugate ($\bar{x} = \mathtt{conjugate}(x) = \mathtt{Re}(x) - i\,\mathtt{Im}(x)$): $\mathtt{Re}(x)^2 + \mathtt{Im}(x)^2$.

```
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)
```

Step 1:

The auxiliary function `div_rem_appx` is used to specify the associated function $f_{\phi_{\mathbb{Z}[i]}}$ for the Euclidean ring $\mathbb{Z}[i]$.

- Consider $a, b \in \mathbb{Z}$, $b \neq 0$.
- Computes the pair of integers (q, r) such that $a = q\,b + r$, and $|r| \leq |b|/2$

```
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
```

Step 2:

- Consider $y \in \mathbb{Z}[i]$ and $x \in Z_{+}^{*}$;
- $\mathtt{Re}(y) = q_1 x + r_1$, where $|r_1| \leq |x/2|$;
- $\mathtt{Im}(y) = q_2 x + r_2$, where $|r_2| \leq |x/2|$;
- Let $q = q_1 + iq_2$ and $r = r_1 + ir_2$, then $y = q(x + 0i) + r$ and $r_1^2 + r_2^2 < |x|^2 = \phi(x + 0i)$.

Step 3:

- Consider $y, x \in \mathbb{Z}[i]$, $x \neq 0 + 0i$;

? $y = qx + r$, $\phi(r) < \phi(x)$;

- $y\,\bar{x} = q\,(x\,\bar{x}) + r\,\bar{x}$;
- Take $r = y - q\,x$.

```
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)
```

Corollary `Euclidean_gcd_alg_ in_Zi` gives the correctness of the Euclidean algorithm for the Euclidean ring $\mathbb{Z}[i]$.

This is a consequence of the correctness of the abstract Euclidean algorithm and lemma `phi_Zi_and_f_phi_Zi_ok` that states that $\phi_{\mathbb{Z}[i]}$ and $f_{\phi_{\mathbb{Z}[i]}}$ are adequate for $\mathbb{Z}[i]$: $\texttt{Euclidean\_f\_phi?}[\texttt{complex}, +, *, 0](\mathbb{Z}[i], \phi_{\mathbb{Z}[i]})(f_{\phi_{\mathbb{Z}[i]}})$.

```
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))
```

Euclidean algorithm for rings $\rightsquigarrow$ { Euclidean algorithm for $\mathbb{Z}$; Euclidean algorithm for $\mathbb{Z}[i]$ }

1^{st} — 2^{nd}

Formalization approach

The theory `quaternions_def[T:Type+,+,*:[T,T->T],zero,one,a,b:T]` uses an abstract type `T`, and assumes `group[T,+,zero]`, and axioms:

```
i = (zero, one, zero, zero)
j = (zero, zero, one, zero)
k = (zero, zero, zero, one)
a_q = (a, zero, zero, zero)
b_q = (b, zero, zero, zero)
```

```
conjugate(v) = (v`x, inv(v`y),inv(v`z),inv(v`t))
red_norm(v) = v*conjugate(v)
+(u,v):quat=(u`x+v`x, u`y+v`y, u`z+v`z, u`t+v`t);
*(c,v):quat=(c * v`x, c * v`y, c * v`z, c * v`t);
*:[quat,quat -> quat]; %quat multiplication

sqr_i          :AXIOM i * i = a_q
sqr_j          :AXIOM j * j = b_q
ij_is_k        :AXIOM i * j = k
ji_prod        :AXIOM j * i = inv(k)
sc_quat_assoc  :AXIOM c*(u*v) = (c*u)*v
sc_comm        :AXIOM (c*u)*v = u*(c*v)
sc_assoc       :AXIOM c*(d*u) = (c*d)*u
q_distr        :AXIOM distributive?[quat](*, +)
q_distrl       :AXIOM (u + v) * w = u * w + v * w
q_assoc        :AXIOM associative?[quat](*)
one_q_times    :AXIOM one_q * u = u
times_one_q    :AXIOM u * one_q = u
```

The PVS theory `quaternions` assumes `field[T,+,*,zero,one]` and formalizes several basic properties.

```
q_prod_charac: LEMMA FORALL (u,v:quat):
 u * v = (u`x * v`x + u`y * v`y * a + u`z * v`z * b + u`t * v`t * inv(a) * b,
         u`x * v`y + u`y * v`x + (inv(b)) *  u`z * v`t + b* u`t * v`z,
         u`x * v`z + u`z * v`x +a * u`y * v`t + inv(a) * u`t * v`y,
         u`x * v`t + u`y * v`z + inv(u`z * v`y) + u`t * v`x )
```

```
quat_is_ring_w_one: LEMMA
 ring_with_one?[quat,+,*,zero_q,one_q](fullset[quat])
```

```
red_norm_charac: LEMMA FORALL (q: quat):
    red_norm(q) = (q`x * q`x + inv(a) * (q`y * q`y) +
                   inv(b) * (q`z * q`z) + (a * b) * (q`t * q`t),
                   zero, zero, zero)
```

```
quat_div_ring_char: LEMMA
charac(fullset[T]) /= 2 IMPLIES
((FORALL (x,y:T): a*(x*x) + b*(y*y) /= one) IFF
division_ring?[quat,+,*,zero_q,one_q](fullset[quat]))
```

Formalization of Hamilton's Quaternion

Hamilton's quaternions are obtained by importing the theory of quaternions using the field of reals as a parameter, and the real -1 for the parameters a and b:

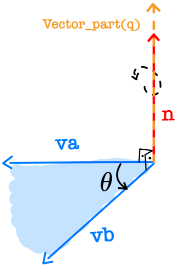
```
IMPORTING quaternions[real,+,*,0,1,-1,-1]
```

The formalization approach follows the principle:

$$\underbrace{\boxed{\text{Quaternions theory over any field}}_{1^{st}} \quad \rightsquigarrow \quad \boxed{\text{Hamilton's Quaternions (over } \mathbb{R})}_{2^{nd}}}_{\text{Formalization approach}}$$

Rotation by Hamilton's Quaternions

```
Quaternions_Rotation: THEOREM
FORALL (a:(pure_quat), b:(pure_quat) |
  norm(Vector_part(a)) = norm(Vector_part(b)) AND
  linearly_independent?(Vector_part(a), Vector_part(b))):
  LET q = rot_quat(a,b) IN
  b = T_q(q)(a)
```

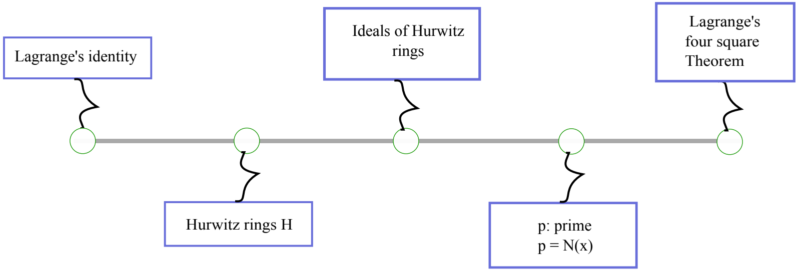


de Lima, Galdino, de Oliveira Ribeiro, Ayala-Rincón
A Formalization of the General Theory of Quaternions
In 15th International Conference on Interactive Theorem Proving (ITP 2024).
Leibniz International Proceedings in Informatics (LIPIcs).
https://doi.org/10.4230/LIPIcs.ITP.2024.11

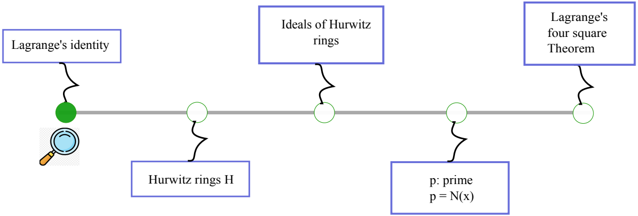
Work in progress

Lagrange's four-square theorem

Given a positive integer number x there are four non-negative integers a, b, c, d such that $x = a^2 + b^2 + c^2 + d^2$.



Work in progress - Lagrange's four-square theorem

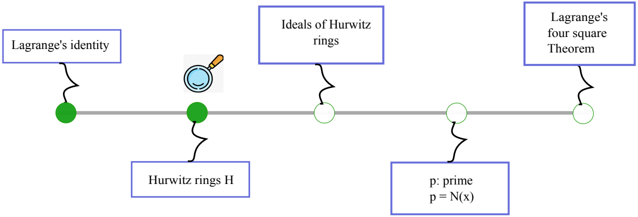



```
Lagrange_identity: LEMMA FORALL (a0, a1, a2, a3, b0, b1, b2, b3: real):
            (a0*a0 + a1*a1 + a2*a2+ a3*a3) * (b0*b0 + b1*b1 + b2*b2 + b3*b3) =
            (a0*b0 - a1*b1 - a2*b2 - a3*b3) * (a0*b0 - a1*b1 - a2*b2 - a3*b3)+
            (a0*b1 + a1*b0 + a2*b3 - a3*b2) * (a0*b1 + a1*b0 + a2*b3 - a3*b2)+
            (a0*b2 - a1*b3 + a2*b0 + a3*b1) * (a0*b2 - a1*b3 + a2*b0 + a3*b1)+
            (a0*b3 + a1*b2 - a2*b1 + a3*b0) * (a0*b3 + a1*b2 - a2*b1 + a3*b0)
```

Consider the Hamilton's Quaternions $x = (a_0, a_1, a_2, a_3)$ and $y = (b_0, b_1, b_2, b_3)$. Then

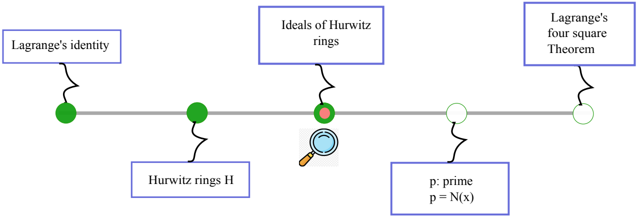
$$N(x) \cdot N(y) = N(x \star y)$$

Work in progress - Lagrange's four-square theorem



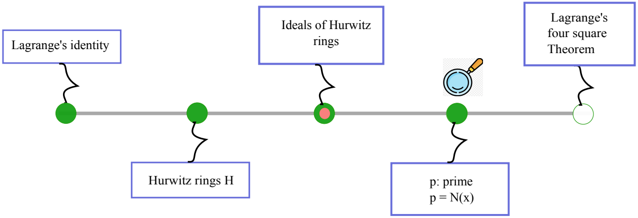

```
IMPORTING algebra@quaternions[rational,+,*,0,1,-1,-1]
Hurwitz_ring: set[quat] = {q: quat | EXISTS (x, y, z, t: int):
(q`x = x/2 AND q`y = x/2 + y AND q`z = x/2 + z AND q`t = x/2 + t)}

Hurwitz_ring_is_ring_w_one: THEOREM
    ring_with_one?[quat,+,*,zero_q, one_q](Hurwitz_ring)

Hurwitz_red_norm_charac: LEMMA FORALL (q: Hurwitz_ring):
    red_norm(q) = (q`x * q`x + q`y * q`y + q`z * q`z + q`t * q`t, 0, 0, 0)

Hurwitz_red_norm_is_posint: LEMMA FORALL (q: Hurwitz_ring):
    integer?((red_norm(q))`x) AND (red_norm(q))`x >= 0
```

Work in progress - Lagrange's four-square theorem



- For every ideal I of a Hurwitz ring H, if x in I then there exists $u \in I$ and $r \in H$ such that $x = r * u$.
- (Prime Hurwitz ideal) $V(p : prime) = \{(p * x, p * y, p * z, p * t)\} \subset H$.
- There exists L ideal of H such that $L \neq H$, $L \neq V$ and $V \subset L$.
  - $W(p) = \{(a_0, a_1, a_2, a_3) | a_i \in Z_p\}$ is not a division ring;
  - $H/V \cong W(p)$.

Work in progress - Lagrange's four-square theorem

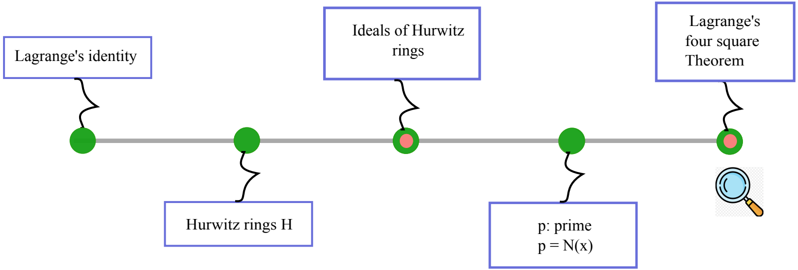


- If L is ideal of H such that $L \neq H$, $L \neq V$ and $V \subset L$, there exists $r \in H$ and $u \in L$ such that $p = r \star u$, and $N(r) > 1$ and $N(u) > 1$.
- $N(p, 0, 0, 0) = p^2 = N(r) \cdot N(u)$.
- There exists $x, y, z, t \in \mathbb{Z}$ such that $x^2 + y^2 + z^2 + t^2 = p$.

Work in progress

Lagrange's four-square theorem

Given a positive integer number x there are four non-negative integers a, b, c, d such that $x = a^2 + b^2 + c^2 + d^2$.



By induction on x.

Conclusion

Our formalizations follow the principles: first, formalize abstract theories with their generic properties; second, obtain particular structures as instantiations of the general theory and proceed with the formalization of their specialized properties.

algebraic theories $\rightsquigarrow$ algebraic structures

1^{st} 2^{nd}

Formalization approach

- Completing the theory of rings.
- Enriching automation of PVS strategies for abstract structures.

References I

- Bini, G., Flamini, F.: Finite commutative rings and their applications, vol. 680. Springer Science & Business Media (2012)
- de Lima, T.A., Avelar, A.B., Galdino, A.L., Ayala-Rincón, M., Formalization of Ring Theory in PVS: Isomorphism Theorems, Principal, Prime and Maximal Ideals, Chinese Remainder Theorem. Journal of Automated Reasoning, vol. 65. p. 1231–1263 (2021)
- de Lima, T.A., Avelar, A.B., Galdino, A.L., Ayala-Rincón, M., Formalizing Factorization on Euclidean Domains and Abstract Euclidean Algorithms. In Proceedings LSFA 2023. EPTCS 402, 2024, pp. 18-33
- Fraleigh, John B., A First Course in Abstract Algebra, Pearson, 2003 (1967).
- Galdino, André Luiz: Quatérnions e Rotações. Lecture Notes (in Portuguese). (2022)
- Hungerford, Thomas W., Algebra, Graduate Texts in Mathematics, vol. 73, 1980 (1974).
- Putinar, M. and Sullivant, S.,Emerging Applications of Algebraic Geometry. Springer New York (2008)

References II

Voight, John: Quaternion Algebras, ed.1. Springer Cham (2021)

Zeitlhöfler, Julian.:Nominal and observation-based attitude realization for precise orbit determination of the Jason satellites. PhD thesis. (2019)

Don't Get Lost in Deep Space: Understanding Quaternions. All about circuits, 2017. Available in https://www.allaboutcircuits.com/technical-articles/dont-get-lost-in-deep-space-understanding-quaternions/. Accessed on Feb.,13th, 2023.

File: Inscription on Broom Bridge (Dublin) regarding the discovery of Quaternions multiplication by Sir William Rowan Hamilton.jpg, 2017. Available in https://commons.wikimedia.org/wiki/File:Inscription_on_Broom_Bridge_%28Dublin%29_regarding_the_discovery_of_Quaternions_multiplication_by_Sir_William_Rowan_Hamilton.jpg. Accessed on Feb.,13th, 2023.