Scientific Issues
Jan Długosz University
in Częstochowa
Mathematics XXIII (2018)
67–74
DOI http://dx.doi.org/10.16926/m.2018.23.06

AUTOMATIC SEARCH OF RATIONAL SELF-EQUIVALENCES

LIDIA STĘPIEŃ AND MARCIN RYSZARD STĘPIEŃ

Abstract

Two Witt rings that are not strongly isomorphic (i.e., two Witt rings over two fields that are not Witt equivalent) have different groups of strong automorphisms. Therefore, the description of a group of strong automorphisms is different for almost every Witt ring, which requires the use various tools in proofs. It is natural idea to use computers to generate strong automorphisms of the Witt rings, which is especially effective in the case of the finitely generated Witt rings, where a complete list of strong automorphisms can be created. In this paper we present the algorithm that was used to generate strong automorphisms from the infinite group of strong automorphisms of the Witt ring of rational numbers $W(\mathbb{Q})$.

Keywords: algebra, rational self-equivalences, Witt ring, strong automorphism, algorithm, automatic search

1. Introduction

One of fundamental notions in algebraic number theory of quadratic forms is introduced in [11] ring called nowadays Witt ring. This ring carries information about the behaviour of all quadratic forms over fixed field, hence the structure of Witt ring depends strongly on the field. Two fields are said to be *Witt equivalent* if their Witt rings are isomorphic and considerd isomorpism preserves dimension of quadratic forms (*strong isomorphism*). We consider strong automorphisms of Witt rings and from above inndicate that two non-isomorphic Witt rings have different groups of strong automorphisms. Therefore the investigation of strong automorphisms of Witt rings is a difficult task because of variety of structure of Witt rings. It is a little easier to determine the groups of strong automorphisms of the Witt rings, which are generated by the finite groups of squares classes. In simple

• *Lidia Stępień* — e-mail: l.stepien@ujd.edu.pl
Jan Długosz University in Częstochowa.
• *Marcin Ryszard Stępień* — e-mail: mstepien@tu.kielce.pl
Kielce University of Technology.

cases one can list all strong automorphisms by hand. In rings with more complex structures, the natural idea is to use a computer to generate all strong automorphisms. Previous attempts have shown the effectiveness of algorithmic methods in linear algebra (see for example [3]). In literature there are descriptions of groups of strong automorphisms for many wide classes of Witt rings: [4], [5], [7], [8]. Some of the results were verified using computer programs [9].

The search for strong automorphisms is more difficult in the case of Witt rings, which are not finitely generated. The first step in this field may be the result from [1], where it has been shown that the group of strong automorphisms of global fields is uncountable. In this article, we deal with strong automorphisms of the Witt ring $W(\mathbb{Q})$ of the field of rational numbers as a special case of Witt ring of a global field. We present the algorithms used in the computer program that was used in [6] to generate strong automorphisms of the Witt ring $W(\mathbb{Q})$.

2. Algebraic background

In [2] authors showed that two global fields are Witt equivalent (and their Witt ring are strong isomorphic) if and only if they are *Hilbert-symbol equivalennt*. A *Hilbert-symbol equivalence* of two global fields K and L is a pair (T, t), where $T\colon \Omega(K) \to \Omega(L)$ is a bijection between the sets of primes of these fields and $t\colon K^*/K^{*2} \to L^*/L^{*2}$ is an isomorphism of their square class groups which preserves Hilbert symbols with respect to the corresponding primes, i.e.

$$(a, b)_{\mathfrak{p}} = (t(a), t(b))_{T(\mathfrak{p})} \quad \text{for all } \ a, b \in K^*/K^{*2},\ \mathfrak{p} \in \Omega(K).$$

The Hilbert symbol equivalence, where $K = L$ is called *Hilbert-symbol self-equivalence* of K.

We conscider the case $K = L = \mathbb{Q}$. Using results from [2] we conclude that for every pair (T, t), which is a Hilbert-symbol self equivalence of the field $\mathbb{Q}$ (called *rational self-equivalence*), the map $\langle a_1, \ldots, a_n\rangle \to \langle t(a_1), \ldots, t(a_n)\rangle$ induces a strong automoorphism of Witt ring $W(\mathbb{Q})$ of the field of rational numbers. Conversely, every strong automorphism of $W(\mathbb{Q})$ determines uniquely a rational self-equivalence (T, t).

In this case we can deal with prime numbers instead of prime ideals and Hilbert symbols depends only on Legendre symbols ([6], Lemma 2.1). The construction of rational self-equivalences presented in [6] bases on the notion of *small equivalence* introduced in [2]. To make the reading of the next part easier, we will cite some notions and several facts proved in [6].

Let $\mathbb{P}$ denotes the set of prime numbers together with the symbol ∞. For every prime number there is defined a completion $\mathbb{Q}_p$ of the field $\mathbb{Q}$ with

the help of valuation v_p called *p-adic number field*. Moreover we agree, that $\mathbb{Q}_\infty = \mathbb{R}$ is a completion of the field $\mathbb{Q}$ at the usual absolute value.

A finite, nonempty set $S \subset \mathbb{P}$ containing 2 and ∞ is called *sufficiently large*. Let S be sufficiently large set of prime numbers $S = \{p_1 \dots, p_n\}$ and assume that $p_1 = \infty$, $p_2 = 2$. The *set of S-singular elements* is defined as follows:

$$E_S = \{x \in \mathbb{Q}^* \colon \ v_p(x) \equiv 0 \pmod 2 \ \text{ for all } \ p \notin S\}.$$

Notice that E_S is a subgroup of the multiplicative group of the field $\mathbb{Q}$ containing all squares of rational numbers. Therefore the quotient group $E_S/\mathbb{Q}^{*2}$ is a subgroup of the group $\mathbb{Q}^*/\mathbb{Q}^{*2}$. By the definition of the set E_S every element $x \in \mathbb{Q}$ has the factorization

$$x = (-1)^{e_1} 2^{2k_2+e_2} p_3^{2k_3+e_3} \cdots p_n^{2k_n 3+e_n} q_1^{2l_1} \cdots q_m^{2l_m},$$

where $q_1, q_2, \dots, q_m \notin S$ are prime numbers, $k_i, l_i \in \mathbb{Z}$ and $e_i \in \{0, 1\}$. Then

$$x\mathbb{Q}^{*2} = (-1)^{e_1} 2^{e_2} p_3^{e_3} \cdots p_n^{e_n} \mathbb{Q}^{*2}.$$

It follows that the elements of the group $E_S/\mathbb{Q}^{*2}$ are represented by the integers of the form $(-1)^{e_1} 2^{e_2} p_3^{e_3} \cdots p_n^{e_n}$ in the unique way.

For every $p \in \mathbb{P}$ the natural imbedding of the field $\mathbb{Q}$ in the field $\mathbb{Q}_p$ induces the group homomorphism $i_p \colon \mathbb{Q}^*/\mathbb{Q}^{*2} \to \mathbb{Q}_p^*/\mathbb{Q}_p^{*2}$, which is surjective. For the finite set $S = \{p_1, \dots, p_n\} \subset \mathbb{P}$ we get the dual homomorphism $\operatorname{diag}_S \colon \mathbb{Q}^*/\mathbb{Q}^{*2} \to \prod_{p \in S} \mathbb{Q}_p^*/\mathbb{Q}_p^{*2}$ defined by

$$\operatorname{diag}_S(a) = [i_{p_1}(a), \dots, i_{p_n}(a)] = [a\mathbb{Q}_{p_1}^{*2}, \dots, a\mathbb{Q}_{p_n}^{*2}].$$

Definition 1. *Let S be sufficiently large set of prime numbers defined as above. A* small S-equivalence *is a pair $\mathcal{R} = ((t_p)_{p \in S}, T)$, where*
1) $T \colon S \to T(S)$ is a bijection,
2) there exists the isomorphism of the group of square classes $t_S : E_S/\mathbb{Q}^{*2} \to E_{T(S)}/\mathbb{Q}^{*2}$,
3) $(t_p)_{p \in S}$ is a family of local isomorphisms $t_p : \mathbb{Q}_p^*/\mathbb{Q}_p^{*2} \to \mathbb{Q}_{T(p)}^*/\mathbb{Q}_{T(p)}^{*2}$ preserving Hilbert symbols, i.e.

$$(a, b)_p = (t_p(a), t_p(b))_{T(p)} \quad \text{for all} \quad a, b \in \mathbb{Q}_p^*/\mathbb{Q}_p^{*2},$$

4) the following diagram commutes

$$\begin{array}{ccc} E_S/\mathbb{Q}^{*2} & \xrightarrow{i_S} & \prod\limits_{p \in S} \mathbb{Q}_p^*/\mathbb{Q}_p^{*2} \\ \downarrow t_S & & \downarrow \prod t_p \\ E_{T(S)}/\mathbb{Q}^{*2} & \xrightarrow{i_{T(S)}} & \prod\limits_{p \in S} \mathbb{Q}_{T(p)}^*/\mathbb{Q}_{T(p)}^{*2} \end{array}$$

It was shown in [6] that any small S_k-equivalence $\mathcal{R}_{S_k} = ((t_p)_{p\in S_k}, T)$, where $S_k = \{\infty, 2, p_3, p_4, \ldots, p_k\}$ is sufficiently large set of prime numbers can be extended to some small S'_{k+1}-equivalence, where $S'_{k+1} := S'_k \cup \{q_{k+1}\}$ and there is infinitely many prime numbers, which can be choosen as q_{k+1} provided they fulfill the following two (sufficent) condoitions:
1) $p_{k+1} \equiv q_{k+1} \pmod 8$,
2) $\left(\frac{p_i}{p_{k+1}}\right) = \left(\frac{q_i}{q_{k+1}}\right)$ for all $3 \leq i \leq k$
and the last Hilbert symbols depend only on Legendre symbols. Above conditions ensure comutativity of suitable diagrams (cf. [6]). In conclusion any small equivalence can be extended to some rational self-equivalence, which induces strong automorphism of Witt ring $W(\mathbb{Q})$ of the field of rational numbers.

3. Algorithm for building of sufficiently large sets

The computer program that performs the search of rational self-equivalen-ces consists of several stages and must be stopped at some point (because it is not possible to generate prime numbers infinitely).

We start from the sufficiently large sets $S' = SP' = \{\infty, 2\}$. Let us first remark that the definition of small equivalence imposes some restrictions on the mapping of T. Namely $T(\infty) = \infty$ and $T(2) = 2$. Then we take the smallest prime number $p_3 \notin S'$, i.e. $p_3 = 3$ and now we get expanded set $S' = \{\infty, 2, 3\}$. Then we search for prime number q_3 which is outside of SP' and fulfills $p_3 \equiv q_3 \pmod 8$. It turns out to be prime number 11. Let us denote this step of construction in the following way:
1) $p_3 = 3 \to 11 = q_3$.
(Notice, that we have to assume that $p_3 \neq q_3$. If we take $p_3 = q_3$ and continue in this way, we get identity).

Next we take the smallest prime number q_4, which was not used in the sequence $SP' = \{q_i\}_{i=1}^{\infty}$. It is number 3. We search for $q_4 = 3$ the smallest prime number p_4, which has required properties:
i) $q_4 \equiv p_4 \pmod 8$ and
ii) $\left(\frac{q_3}{q_4}\right) = \left(\frac{p_3}{p_4}\right)$.
It is the number 19. Hence we denote the second step of the construction:
2) $p_4 = 19 \leftarrow 3 = q_4$.
Further steps of construction lead to the following sequences of prime numbers
$S : 3, 19, 5, 13, 7, 1103, 11, 6329, 17, 347, 23, 77551, 29, 138581, 31,$
$SP : 11, 3, 13, 5, 223, 7, 283, 17, 2689, 19, 31159, 23, 109229, 29, 1010903,$
what gives the following sufficiently large sets:
$S' = \{\infty, 2, 3, 19, 5, 13, 7, 1103, 11, 6329, 17, 347, 23, 77551, 29, 138581, 31\},$

$SP' = \{\infty, 2, 3, 11, 3, 13, 5, 223, 7, 283, 17, 2689, 19, 31159, 23, 109229, 29, 1010903\}$
and the map T:
$T(\infty) = \infty$,
$T(2) = 2$,
$T(3) = 11$,
$T(19) = 3$,
$T(5) = 13$,
$T(13) = 5$,
$T(7) = 223$,
$T(1103) = 7$,
$T(11) = 283$,
$T(6329) = 17$,
$T(17) = 2689$,
$T(347) = 19$,
$T(23) = 31159$,
$T(77551) = 23$,
$T(29) = 109229$,
$T(138581) = 29$,
$T(31) = 1010903$
which easily shows how the next small equivalences are constructed. The limitation to 15 steps is due to the rapid increase of searched prime numbers. This process, continued into infinity, gives us a rational self-equivalence.

Of course the choice of another q_3 gives another sequences of prime numbers p_k and q_k and the different sequences of small equivalences (for enother examples of rational self-equivalences searched in this way see [6]).

Now we show how we construct two sequences S and SP of prime numbers using Algorithm 1.

Algorithm **built_sequences**() inputs the set P of prime numbers generated by sieve of Eratosthenes. It uses the function $FindElement()$ as defined in Algorithm 2. First we add 3 to the set S as a smallest prime number (line 2). The variable p is initilized as 3 (line 4). (The variable p and q are used to build the sets S and SP, respectively.) q is initialized as 0 (line 3). As long as the variable i is less than 15 the algorithm performs the following: for odd runs it searches a smallest prime number q by using function $FindElement()$ (line 7) and adds it to the set SP; next finds the first free number prime by using function $FirstFree()$ (line 9); gets it to q and adds it to the set SP; for even runs the algorithm performs steps described above for the variable p and the set S. Algorithm **built_sequences**() terminates when i is greater then 15 and returns two sets S and SP.

The function $FindElement()$ inputs the set P of prime numbers, the sets S and SP, the element el and the variable i. It uses the function

Algorithm 1: function **built_ sequences**(P)
Variables:
S (sequence of prime numbers, initialized as $\emptyset$)
SP (sequence of prime numbers, initialized as $\emptyset$
i, q, p (integer)
Returned values:
SP, S /* two sequences of prime numbers*/

1: $i \leftarrow 1$
2: $S \leftarrow S \cup \{3\}$
3: $q \leftarrow 0$
4: $p \leftarrow 3$
5: **while** $i \leqslant 15$ **do**
6: **if** $i \ mod \ 2 == 1$ **then**
7: $q \leftarrow FindElement(P, S, SP, p, i)$
8: $SP \leftarrow SP \cup \{q\}$
9: $q \leftarrow FirstFree(P, SP)$
10: $SP \leftarrow SP \cup \{q\}$
11: **else**
12: $p \leftarrow FindElement(P, S, SP, q, i)$
13: $S \leftarrow S \cup \{p\}$
14: $p \leftarrow FirstFree(P, S)$
15: $S \leftarrow S \cup \{p\}$
16: **end if**
17: $i \leftarrow i + 1$
18: **end while**
19: **return** S,SP

Legrende() defined as one of standard algorithm calculated of Legendre symbol [10]. Algorithm searches for the prime number j (line 3) such that $j \neq el$ AND $(j - el) \ mod \ 8 == 0$ (line 4). Algorithm terminates and returns j when for j and el and sets S and SP all Legrende symbols $Left$ and $Right$ are equal (lines 5-17), respectively.

The algorithms were implemented in C++. The experiments were carried out on an notebook Intel Core i5-5200U CPU 2.20 GHz, 8 GB RAM with Linux operation system.

4. Final remarks

In this case the obtained results have shown the usefulness of the computer. The value of the greatest searched prime number in the example described in previous section shows that it would be extremely time-consuming

Algorithm 2: function **FindElement**(P,S,SP,el,i)
Variables:
result (boolean variable, initialized as **False**)
j (integer)
Returned values:
j /* prime number*/

```
1: result ← False
2: while NOT result do
3:   if i mod 2 == 1 then
4:     j ← NextPrime(P, SP)
5:   else
6:     j ← NextPrime(P, S)
7:   end if
8:   if j ≠ el AND (j − el) mod 8 == 0 then
9:     result ← True
10:    k ← 1
11:    while k < i AND result do
12:      if i mod 2 == 1 then
13:        Left ← Legendre(S[k], el)
14:        Right ← Legendre(SP[k], j)
15:      else
16:        Left ← Legendre(S[k], j)
17:        Right ← Legendre(SP[k], el)
18:      end if
19:      result ← Left == Right
20:      k ← k + 1
21:    end while
22:  end if
23: end while
24: return j
```

or impossible at all to do the calculations without a computer. This allows usu to think that the computer would be useful in solving similar problems in the future.

References

[1] A. Czogała, M. Kula, *Automorphisms of Witt rings of global fields*, Acta Arithmetica, 163 (1) (2014), 1-13.

[2] R. Perlis, K. Szymiczek, P.E. Conner, R. Litherland, *Matching Witts with global fields*, In: W.B. Jacob, T.-Y. Lam, R.O. Robson (Eds.), *Recent Advances in Real Algebraic Geometry and Quadratic Forms*, (Proceedings

of the RAGSQUAD Year, Berkeley 1990–1991), Contemporary Mathematics, Amer. Math. Soc. Providence, Rhode Island, 155 (1994), 365–387, .

[3] M. Srebrny, L. Stepień, *SAT as a Programming Environment for Linear Algebra*, Fundamenta Informaticae 102 (1) (2010), 115-127.

[4] M. Stępień, *Automorphisms of products of Witt rings of local type*, Acta Mathematica et Informatica Universitatis Ostraviensis 10 (2002), 125-131.

[5] M. Stępień, *Automorphisms of Witt rings of elementary type*, Mathematica. Proc. XI^{th} Slovak-Polish-Czech Mathematical School, Catholic University in Ružomberok, 10, (2004), 62-67.

[6] M. Stępień. *A construction of infinite set of rational self-equivalences*, Scientific Issues of Jan Długosz University in Częstochowa. Mathematics, XIV (2009), 117-132.

[7] M. R. Stępień, *Automorphisms of Witt rings of finite fields*, Scientific Issues of Jan Długosz University in Częstochowa. Mathematics, XVI (2011) 67-70.

[8] M. R. Stępień, *Automorphisms of Witt rings of local type*, Journal of Applied Mathematics and Computational Mechanics, The Publishing Office of Czestochowa University of Technology 14 (3), (2015) 109-119.

[9] L. Stępień, M. R. Stępień, *Automatic search of automorphisms of Witt rings*, Scientific Issues of Jan Długosz University in Częstochowa. Mathematics, XVI, (2011) 141-146.

[10] M. Thoma, *How to calculate the Legendre symbol*, http://martin-thoma.com/how-to-calculate-the-legendre-symbol/ (2018.12.05)

[11] E. Witt, *Theorie der quadratischen Formen in beliebigen Körpern*, J. Reine Angew. Math. 176 (1937), 31-44.

Received: December 2018

Lidia Stępień
Jan Długosz University in Częstochowa,
Institute of Mathematics and Computer Science,
Częstochowa, ul. Armii Krajowej 13/15
Email address: l.stepien@ujd.edu.pl

Marcin Ryszard Stępień
Kielce University of Technology,
Department of Mathematics and Physics,
Al. Tysiąclecia Państwa Polskiego 7
Email address: mstepien@tu.kielce.pl