Jan Długosz University in Częstochowa
Scientific Issues, Mathematics XVI, Częstochowa 2011

AUTOMATIC SEARCH OF AUTOMORPHISMS OF WITT RINGS

Lidia Stępień[a], **Marcin Ryszard Stępień**[b]

[a] *Institute of Mathematics and Computer Science*
Jan Długosz University of Częstochowa
al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland
e-mail: l.stepien@ajd.czest.pl

[b] *Kielce University of Technology*
al. Tysiąclecia Państwa Polskiego 7, 25-314 Kielce, Poland
e-mail: mstepien@tu.kielce.pl

Abstract. The investigation of strong automorphisms of Witt rings is a difficult task because of variety of their structures. Cordes Theorem, known in literature as Harrison-Cordes criterion (cf. [1, Proposition 2.2], [3, Harrison's Criterion]), makes the task of describing all the strong automorphisms of a given (abstract) Witt ring $W = (G, R)$ easier. By this theorem, it suffices to find all such automorphisms σ of the group G that map the distiguished element -1 of the group G into itself (i.e. $\sigma(-1) = -1$) in which the value sets of 1-fold Pfister forms are preserved in the following sense: $\sigma(D(1, a)) = D(1, \sigma(a))$ for all $a \in G$. We use the above criterion and the well-known structure of the group G as a vector space over two-element field $\mathbb{F}_2$ for searching all automorphisms of this group. Then we check Harrison-Cordes criterion for found automorpisms and obtain all the automorpisms of a Witt ring W.

The task is easy for small rings (with small groups G). For searching of all strong automorpisms of bigger Witt rings we use a computer which automatizes the procedure described above. We present the algorithm for finding strong automorphisms of a Witt rings with finite group G and show how this algorithm can be optimized.

1. Searching of automorphisms of Witt rings

Consider Witt rings in terminology of Marshall [2]. Let $W = (R, G)$ be a Witt ring, where the group G is finite. We are interested in finding all automorphisms of the given finitely generated Witt ring W. By definition, the map σ is an automorphism of a Witt ring W if σ is such an automorphism of

the ring R that $\sigma(G) = G$. Cordes in [1] has formulated the usefull criterion for σ to be an automorphism of a Witt ring: any $\sigma \in \mathrm{Aut}(G)$ induces an automorphism of a Witt ring W iff

1) $\sigma(-1) = 1$;

2) $D(1, \sigma(a)) = \sigma(D(1, a))$ for all $a \in G$,

where by $D(1, a)$ we denote the value set of a 1-fold Pfister form $(1, a)$. The above statement, called nowadays the Harrison-Cordes criterion (cf. [3]), allows us to investigate automorphisms of simpler structure of the group G instead of automorphisms of the ring R.

As we know, G is a group of exponent 2, so it can be consisered as a vector space over $\mathbb{F}_2$. Hence, we can consider automorphisms of vector space $G(\mathbb{F}_2)$ (see Algorithm 1). For this purpose we choose a basis $\mathcal{B}$ of that vector space (step 1). If $|G| = 2^n$, then $\mathcal{B}$ of vector space $G(\mathbb{F}_2)$ consists of n elements of G. If we choose another basis $\mathcal{B}'$ (step 2), we can create a map between $\mathcal{B}$ and $\mathcal{B}'$. Finding all such bases we can build all maps from $\mathcal{B}'$ to other bases including their permutations of bases (step 3). Then we extend the obtained maps to a whole group G via known representation of vectors of G as a combinations of elements of the basis $\mathcal{B}$. Finally, we have to check whether the obtained automorphisms of the group G fullfil the Harrison-Cordes criterion. As a result, we get all such automorphisms of the vector space $G(\mathbb{F}_2)$ which can be extended to automorphisms of Witt ring $W = (R, G)$. This is equivalent to the case that we have found all strong automorphisms of W.

Algorithm 1 Search for automorphisms of vector space $G(\mathbb{F}_2)$

INPUT: $\dim A = n$, $|G|$;
OUTPUT: A set of all automorphisms of $G(\mathbb{F}_2)$

1: Finding a basis $\mathcal{B}$ of $G(\mathbb{F}_2)$.
2: Search for every bases of $G(\mathbb{F}_2)$.
3: Make maps from basis $\mathcal{B}$ to every bases of $G(\mathbb{F}_2)$ (including $\mathcal{B}$ and permutations of all bases).
4: Extend created maps to all group G.
5: Check if the obtained maps fulfill the Harrison-Cordes criterion.

The algorithm 1 is easy to handle in the case when the cardinality of the group G generating a Witt ring $W = (R, G)$ is small. Then we can calculate every automorphisms of W by hand (see example 1). When cardinality of G grows up, the task becomes much complicated and takes a lot of time (compare example 2). In order to accelerate calculation, we have written a computer program which realizes the above algorithm. Thanks to the program, we can

find automatically all automorphisms for all non-isomorphic Witt rings with the group G being finite. The only limitation is the power of a computer and the time needed for its work.

Example 1. 1. Let $W \cong \mathbb{Z}/2\mathbb{Z}[C_4]$ be the group ring of 4-element cyclic group C_4 with coefficients in the 2-element ring $\mathbb{Z}/2\mathbb{Z}$. Then W is a Witt ring with 4-element group G. The vector space $G(\mathbb{F}_2)$ has two-element basis, and it is easy to calculate all their 6 automorphisms.
2. Take $W \cong \mathbb{Z}/2\mathbb{Z}[C_2]^4$ – a Witt ring, which is the 4th power of a Witt ring $\mathbb{Z}/2\mathbb{Z}$. Then a suitable vector space $G(\mathbb{F}_2)$ has cardinality 16 and 4-element basis. It turns out that $|\mathrm{Aut}(G(\mathbb{F}_2))| = 20160$, and it is rather difficult task to calculate all this automorphisms by hand.

2. Optimalisation of algorithm and experimental results

In this section we shall show how we have optimized algorithm 1 in order to accelerate searching automorphisms with the help of computer. We make some rationalization in the step 1.

We start from the following equivalence relation $\sim$ which determines the equivalence classes of elements of a group G with respect to equicardinality of the value sets of 1-fold Pfister forms. We say that $g_1 \sim g_2$ iff $|D(1, g_1)| = |D(1, g_2)|$. The relation $\sim$ introduces the partition of the set of all elements of group G into the equivalence classes, which we call *types* (of elements) and denote by T (with subscripts when needed). For the sake of simplicity of notation we index them with m consecutive natural numbers, where m is the number of all the equivalence classes. Let $\mathcal{B} = \{b_1, \dots, b_n\}$ be a basis of the space $G(\mathbb{F}_2)$. Then $\overline{T} = \{T_{j1}, \dots, T_{jn}\}$ is called the *type of basis* B if the elements b_i are of type T_{ji} for each $1 \leq i \leq n$. In general, a system $(w_1, w_2, \dots, w_n)$ of elements of G is of type $\overline{T} = (T_{j1}, \dots, T_{jn})$ if $w_i \in T_{ji}$ for $1 \leq i \leq n$. Clearly, n and m do not have to be identical. We do not assume that the sets T_{ji} are pairwise different for $1 \leq i \leq n$. Repetitions are allowed.

It seems essential to start off with such a basis $\mathcal{B} = \{b_1, \dots, b_n\}$ for which the number of all possible tuples $(w_1, w_2, \dots, w_n)$ is the smallest possible. Hence, our goal is to find a basis of such a type $\overline{T} = \{T_{j1}, \dots, T_{jn}\}$ which is *minimal* in the sense that a system of types $\{T_{j1}, \dots, T_{jn}\}$ has the smallest possible cardinality.

The algebraic program AP (see algorithm 2) searches for a minimal type, in which a basis exist, by computing the determinants. Here the determinants are computed according to the Laplace expansion algorithm. The algebraic program AP takes the dimension n of a vector space $G(\mathbb{F}_2)$ as its input as

Algorithm 2 procedure **AP**$(n, \mathcal{T})$
Variable: $min \leftarrow 0$; $j \leftarrow 1$; $j \leftarrow 1$; $\overline{T} \leftarrow \emptyset$;
$< g >$ /* n-tuple of elements in type $\overline{T}$ */
Return values:
$AP(): \overline{T}$ /* a minimal type $\overline{T}$ in which a basis exists */
$\det(): \{1,0\}$ /* a value of determinant of matrix $\mathcal{M}$ */
$build\_matrix(): \mathcal{M}$ /* a matrix of scalars of n elements of group G */

INPUT: a dimension n and a set $\mathcal{T}$ of all types $\overline{T}$
OUTPUT: a minimal type $\overline{T} = (T_{j1}, T_{j2}, \ldots, T_{jn})$ in wich a basis exists

1: **repeat**
2: $\overline{T} \leftarrow \overline{T}_j \in \mathcal{T}$
3: $j \leftarrow j + 1$
4: $min \leftarrow 0$
5: **repeat**
6: $< g > \leftarrow < g >_i \in \overline{T}$
7: $\mathcal{M} \leftarrow build\_matrix(< g >)$
8: **if** $\det(\mathcal{M}) = 1$ **then**
9: min = 1;
10: **end if**
11: $i \leftarrow i + 1$
12: **until** (min = 1)
13: **until** (min = 1)
14: **return** $\overline{T}$

well as a set $\mathcal{T}$ of types $\overline{T}$ in the cardinality increasing order, represented as the lists of 0-1 sequences. The AP program gives an n-tuple of types (step 2), generates n elements of a group G belonging to these types (step 6). Then for each n-tuple it computes the determinant of the $n \times n$ matrix of scalars (step 8 and 7, respectively). If for a given type the determinant of all such matrices is equal 0, the AP program considers the next type. Otherwise, it returns a minimal type in which a basis exists and terminates.

Notice that if there is no linearly independent set of vectors of type $\overline{T}$, the algebraic AP program must perform all the computations of determinant of $|\overline{T}|$ possible systems of n vectors. Since the number of combinations depends exponentially on a dimension of vector space $G(\mathbb{F}_2)$, so the complexity of algebraic program AP depends exponentially on the dimension too.

Our algorithm is implemented in C++ language. Our experiments were carried out on an IBM PC machine with an Intel Pentium IV 3.2 GHz processor, 1024 MB RAM memory and Linux operating system.

Below we present an example involving a Witt ring with rather big group

G, which automorphisms cannot be calculated by hand and the group of all automorpisms is not desribed until now.

Example 2. Consider a Witt ring $W = (R, G)$ with the group $G = G_1 \times G_2 \times G_3$ such that $|G| = 2^9$, where

$G_1 = \{(1,1,1),(1,-1,1),(1,1,x),(1,1,y),(1,1,xy),(1,-1,x),(1,-1,y),(1,-1,xy)\}$,

$G_2 = \{(1,1,1),(1,-1,1),(1,1,x),(1,1,y),(1,1,xy),(1,-1,x),(1,-1,y),(1,-1,xy)\}$,

$G_3 = \{(1,1,1),(-1,-1,-1),(1,1,-1),(1,-1,1),(1,-1,-1),(-1,1,1),(-1,1,-1),(-1,-1,1)\}$.

According to Algorithm 1, we choose some basis of $G(\mathbb{F}_2)$, namely $\mathcal{B} = (g_2, g_3, g_4, g_9, g_{17}, g_{25}, g_{74}, g_{129}, g_{257})$. Let us check what type has that basis. Knowing the value sets of 1-fold Pfister forms we decompose the group G into 5 types. For the sake of simplicity of notation, we denote the type T_j by its index j, for $1 \leq j \leq m$; for example, the type T_3 will be denoted by 3. Hence, the type T of n-elements of group G will be n-tuple of indices $(j_1, j_2, \ldots, j_n)$ of types, where $\forall_{i=1}^{n} 1 \leq j_i \leq m$ and $n = \dim G(\mathbb{F}_2)$.

Type	1	2	3	4	5
Cardinality	56	336	8	104	8

It follows that the type of our basis $\mathcal{B}$ is $\overline{T} = (1,1,1,4,2,2,5,1,1)$ and we must find all bases in $|\overline{T}| = 178\,862\,731\,407\,360$ possible systems of n vectors.

With the help of algorithm 2, we have found a basis in minimal type. Last 10 from 117 checked types are presented in Table 1. As we can see, the type $(5,4,4,4,4,3,3,3,3)$ has the smallest cardinality between all the types in which basis of $G(\mathbb{F}_2)$ exists.

		AP	
Type $\overline{T}$	$\lvert\overline{T}\rvert$	sec.	result
(5,3,3,3,1,1,1,1,1)	213909696	293.07	NOT EXISTS
(5,5,5,5,4,3,3,1,1)	251130880	269.96	NOT EXISTS
(5,5,5,4,3,3,3,1,1)	251130880	302.70	NOT EXISTS
(5,5,5,5,4,4,4,3,3)	285539072	314.83	NOT EXISTS
(5,5,5,4,4,4,3,3,3)	285539072	165.72	NOT EXISTS
(5,5,5,3,3,1,1,1,1)	287955360	316.66	NOT EXISTS
(5,5,5,5,5,3,2,1,1)	289766400	265.74	NOT EXISTS
(5,5,3,3,3,3,2,1,1)	289766400	343.82	NOT EXISTS
(5,5,5,5,5,4,4,4,4)	321868820	294.82	NOT EXISTS
(5,4,4,4,4,3,3,3,3)	321868820	153.97	EXISTS
	Sum:	3850.03	

Table 1: The results for the group of example 2.1.

The algorithm 1 has been improved yet. In [4] the autors showed that for Witt rings, which have $-1 \neq 1$ in G, we can choose such a basis of minimal

type that the distinguished element -1 is one of the vector of that basis. In the next step all bases of minimal type with one fixed vector -1 are searched. Therefore, we can map a choosen basis into all another ones such that the map preserves the vector -1. In that way the first condition of the Harrison-Cordes criterion is fulfilled and we reduce thea number of choosen vectors in each basis to $n-1$.

Another rationalization proposed in [4] consists in coding the problem of searching of minimal type as a propositional formula and using newest SAT-solvers – very effective tool for veryfing satisfiability of that formula. The autors have showed that the time of finding minimal type with the help of SAT-solvers is shorter than in our algorithm 2.

We claim that algorithm 1 can be improved in another way too. We leave it for our future work.

References

[1] C. M. Cordes. The Witt group and the equivalence of fields with respect to quadratic forms. *J. Algebra*, **26**, 400–421, 1973.

[2] M. Marshall. *Abstract Witt Rings*. Queen's Papers in Pure and Applied Math., **57**, Queen's University, Ontario 1980.

[3] R. Perlis, K. Szymiczek, P.E. Conner, R. Litherland. Matching Witts with global fields. In: *Recent Advances in Real Algebraic Geometry and Quadratic Forms (Proceedings of the RAGSQUAD Year, Berkeley 1990–1991)*, W.B. Jacob, T.Y. Lam, R.O. Robson (Eds), Amer. Math. Soc., Contemporary Mathematics, **155**, Providence, Rhode Island 1994.

[4] M. Srebrny, L. Stępień. A propositional programming environment for linear algebra. *Fundamenta Informaticae*, **81**, 325–345, 2007.