COMPUTING WITH QUADRATIC FORMS OVER NUMBER FIELDS

PRZEMYSŁAW KOPROWSKI AND ALFRED CZOGAŁA

Abstract. This paper presents fundamental algorithms for computational theory of quadratic forms over number fields. In the first part of the paper, we present algorithms for checking if a given non-degenerate quadratic form over a fixed number field is either isotropic (respectively locally isotropic) or hyperbolic (respectively locally hyperbolic). Next, we give a method of computing the dimension of the anisotropic part of a quadratic form. Finally, we present algorithms computing the level of a number field and verifying whether two number fields have isomorphic Witt rings (i.e. are Witt equivalent).

1. Introduction

The algebraic theory of quadratic forms is an important branch of mathematics, which has already achieved its maturity. Yet still, the computational side of this theory is seriously under-developed. Little has been done so far to develop algorithms for fundamental problems on quadratic forms. There are a few papers that treat computational aspects of the theory but they mostly concentrate on forms over the rationals (see e.g. [3, 10]). While the algebraic theory of quadratic forms over number fields (i.e. finite extensions of $\mathbb{Q}$) are very like the theory over the rationals, the computational approach seems to be rudimentary here. The aim of this article is to partially fill this gap, as well as provoke further discussion and future research.

This paper is organized as follows: in Section 2 we present an algorithm (see Algorithm 7) checking if a given form (over a fixed number field $K$) is isotropic. This algorithm uses sub-procedures (Algorithms 5 and 6) deciding whether the form is isotropic at a non-archimedean prime of $K$ (respectively odd or even). These two algorithm may be of independent interest to the reader. Next, in Section 3 we show Algorithm 8 determining if a quadratic form is hyperbolic. It is known that any non-degenerate form can be uniquely decomposed into an orthogonal sum of its anisotropic part and a hyperbolic form (one of these two parts may of course be void if the form in question is either anisotropic or hyperbolic itself). Half of the dimension of the hyperbolic part is an important invariant of the quadratic form, known as the Witt index. In Section 4 we shows a procedure effectively computing the Witt index of a form.

In Sections 5 and 6 we go a step further and develop algorithm computing invariants of the ground fields, that play important roles in the algebraic theory of quadratic forms. Algorithm 12 computes the level $s(K)$ of a number field $K$, which is the length of the shortest representation of $-1$ as a sum of squares.

Recall that the set $ WK$ of similarity classes of non-degenerate symmetric bilinear forms over a given base field $K$ is a ring with operations induced by the orthogonal sum and the tensor product. It is called the Witt ring of the field $K$. Because a bilinear form defines an orthogonal geometry on the vector space on which it is
defined, thus the Witt ring can be viewed as an algebraic structure encoding information on all possible orthogonal geometries over a given base field. Two fields are said to be Witt equivalent, if their Witt rings are isomorphic. In [12] K. Szymiczek described the set of global field invariants fully determining its Witt equivalence class. In Section 9 we present Algorithm 13 computing all these invariants. In particular the algorithm may be used to verify whether two number fields are Witt equivalent.

The authors implemented all the algorithms presented in this paper in a computer algebra system Sage [11]. Using this implementation, we were able to find representatives of Witt classes of number fields of low degrees. The results are presented in Tables 1–3. Moreover, we used our implementation to empirically analyze the distribution of different Witt classes among number fields of a fixed degree. The quantitative results are gathered in Tables 4–6.

In Sections 2 through 4, we described the set of global field invariants fully determining its Witt equivalence class. In Section 6 we present Algorithm 14 computing all these invariants. In particular the algorithm may be used to verify whether two number fields are Witt equivalent.

In order to simplify frequent references we write the algorithm explicitly.

Algorithm 1. Given a element $a$ of a number field $K$, this algorithm returns true if and only if $a$ is a square, otherwise it returns false.

1. Use [15] §5.4 to check if $x^2 - a$ factors over $K$, if so return true, if not then return false.

2. Similarly we often need to check if an element is a square in the completion $K_p$ of the number field $K$ at some finite prime $p$. We present two separate procedures respectively for odd and even primes.

Algorithm 2. Given an element $a$ of a number field $K$ and an odd prime $p$. This algorithm returns true if and only if $a$ is a square in $K_p$. Otherwise it returns false.

1. Compute the valuation $\text{ord}_p a$. If it is odd then return false and quit.
2. Use [1] Algorithm 4.8.17 to find a uniformizer $\pi$ of $p$.
3. Check if the class of $a \cdot \pi^{-\text{ord}_p a}$ is a square in the residue field $K/p$. Return true if it is a square and false otherwise.

The proof of correctness of this algorithm is straightforward. Listing 2 contains the pseudo-code. The case of even primes is a bit more tricky.

Algorithm 3. Let $\mathfrak{d}$ be an even prime of $K$ and $a \in \mathcal{O}_K$ be a non-zero element. This algorithm returns true if and only if $a$ is a square in the completion $K_\mathfrak{d}$, otherwise it returns false.

1. Use Algorithm 1 to check if $a$ is a square in $K$, if so return true and quit.
2. Take $L = K(\sqrt{a})$. Use [2] Algorithm 2.4.13 to decompose $\mathfrak{d}$ in $\mathcal{O}_L$. If $\mathfrak{d}$ splits into two primes in $\mathcal{O}_L$, then return true, otherwise return false.

Proof of correctness. It is clear that if $x$ is already a square in $K$, then it is a square in $K_\mathfrak{d}$ as well. This justifies step (1). Suppose that it is not a square in $K$.

Take $L = K(\sqrt{a})$ and let $\mathfrak{D}$ be a prime of $L$ dominating $\mathfrak{d}$. Assume that $\mathfrak{d}$ splits in $L$, it follows that both: the relative ramification index $e(\mathfrak{D}/\mathfrak{s})$ and the relative inertia degree $f(\mathfrak{D}/\mathfrak{s})$ are equal 1 and so is the local degree $(L_\mathfrak{D} : K_\mathfrak{d})$. Consequently, $\sqrt{a} \in L \subseteq L_\mathfrak{D} = K_\mathfrak{d}$, as desired.

Conversely, assume that $a$ is a square in $K_\mathfrak{d}$. Let $\mathfrak{D}$ be any prime of $L$ laying over $\mathfrak{d}$. Then $L_\mathfrak{D} = K_\mathfrak{d}(\sqrt{a})$ by [9] Theorem 5.5, hence $L_\mathfrak{D} = K_\mathfrak{d}$ and so $e(\mathfrak{D}/\mathfrak{s}) = f(\mathfrak{D}/\mathfrak{s}) = 1$. This shows that, for every prime $\mathfrak{D}$ of $L$ over $\mathfrak{d}$ we have $(L_\mathfrak{D} : K_\mathfrak{d}) = 1$. 


But \((L:K) = 2\) is the sum of the local degrees \((L_L:K_L)\) for all the primes \(D\) dominating \(d\). It follows that there are exactly two such primes. □

Listing 3 contains the pseudo-code of the algorithm.

**Observation 1.1.** There exists an algorithm which, for a non-degenerate quadratic form presented as a symmetric matrix, finds a list \(\{a_1, \ldots, a_d\}\) with entries in \(O_K\) and such that the form \(\langle a_1, \ldots, a_d \rangle\) is isometric to \(\varphi\) over \(K\).

**Proof.** The Gram-Schmidt orthogonalization algorithm is well known. Once the form is diagonalized, with entries expressed using the power basis, it is straightforward to clear denominators using the fact that \(a/b\) and \(a \cdot b\) belong to the same square class. □

From now on we will always assume that all quadratic forms are non-degenerate and, as inputs to Algorithms 4–11, they are given as lists of entries from \(O_K\).

### 2. Isotropy of a Quadratic Form

In this section, we present an algorithm that checks if a given form \(\varphi\) over a number field \(K\) is isotropic or not. The organization of this section reflects the general idea of solving the problem locally. Hence, Algorithm 6 checks if the form is isotropic at an odd prime, further reducing the task to the residue field (see Algorithm 4). Next, Algorithm 6 verifies whether the form is isotropic at an even prime. Finally, Algorithm 7 checks if the form is globally isotropic, using the two above-mentioned algorithms as sub-procedures. Recall (see e.g. [12, Definition 15.2.1]) that the discriminant of a quadratic form \(\varphi\) is defined by the formula

\[
\text{disc } \varphi := (-1)^{d(d-1)/2} \det \varphi,
\]

where \(d = \dim \varphi\).

**Algorithm 4.** Let \(p\) be an odd prime of \(K\) and \(\varphi = \langle a_1, \ldots, a_d \rangle\) be a quadratic form with all its entries being \(p\)-adic units. This algorithm returns true if and only if the residual form \(\varphi \otimes K/p\) is isotropic, otherwise it returns false.

1. If \(\dim \varphi = 1\), return false.
2. If \(\dim \varphi = 2\), return true when \(\text{disc } \varphi\) is a square in \(K/p\), otherwise return false.
3. If \(\dim \varphi > 2\), return true.

The correctness of the above algorithm follows immediately from [7, Theorem I.3.2]. Listing 4 contains the pseudo-code of the algorithm.

**Algorithm 5.** Let \(p\) be an odd prime of a number field \(K\). Given a quadratic form \(\varphi\), this algorithm returns true if \(\varphi \otimes K_p\) is isotropic and false otherwise.

1. If \(\dim \varphi = 1\), return false.
2. If \(\dim \varphi \geq 5\), return true.
3. Let \(\{a_1, \ldots, a_d\}\) be the list of coefficients of \(\varphi\), all \(a_i \in O_K\). Partition this list into two sublists depending on the parity of the \(p\)-adic valuation:

\[
\varphi_0 := \left\{ a_i \cdot \pi^{-\ord_p a_i} \mid \ord_p a_i \equiv 0 \pmod{2} \right\},
\]

\[
\varphi_1 := \left\{ a_i \cdot \pi^{-\ord_p a_i} \mid \ord_p a_i \equiv 1 \pmod{2} \right\}.
\]

Here \(\pi\) is a uniformizer of \(p\) computed using [11, Algorithm 4.8.17].

4. Use Algorithm 4 to verify whether any of \(\varphi_0, \varphi_1\) is isotropic over \(K/p\). Return true if Algorithm 4 returned true at least once, otherwise return false.
The correctness of the algorithm follows from [7, Proposition VI.1.9]. The pseudo-code of the algorithm is presented in Listing 5. Now, we consider even primes. Recall (see e.g. [7, Definition V.3.17]) that the Hasse invariant of a quadratic form $\varphi = \langle a_1, \ldots, a_d \rangle$ at a prime $p$ is:

$$s_p(\varphi) := \prod_{1 \leq i < j \leq d} (a_i, a_j)_p,$$

where $(a_i, a_j)_p$ denotes the $p$-adic Hilbert symbol. Recently J. Voight in [14] presented an algorithm for computing the Hilbert symbol in a completion of a number field. We use it to verify whether a quadratic form is isotropic over a dyadic completion of $K$.

Algorithm 6. Let $\mathfrak{o}$ be an even prime of $K$ and $\varphi$ be a quadratic form over $K$. This algorithm returns true if and only if $\varphi \otimes K_\mathfrak{o}$ is isotropic, otherwise it returns false.

1. If $\dim \varphi \leq 1$, then return false and quit.
2. If $\dim \varphi = 2$, then use Algorithm 3 to check whether $\text{disc} \varphi$ is a square in $K_\mathfrak{o}$. If so, then return true and quit, otherwise return false and quit.
3. If $\dim \varphi = 3$, then proceed as follows:
   a. Use [14, Algorithm 6.6] to compute the Hilbert symbol $((-1, -\det(\varphi))_\mathfrak{o}$.
   b. Use Eq. (1) and [14, Algorithm 6.6] to compute the Hasse invariant $s_\mathfrak{o}(\varphi)$ of $\varphi$ at $\mathfrak{o}$.
   c. If $((-1, -\det(\varphi))_\mathfrak{o} = s_\mathfrak{o}(\varphi)$, then return true otherwise return false.
4. If $\dim \varphi = 4$, then proceed as follows:
   a. Use Algorithm 5 to check if $\det \varphi$ is a square in $K_\mathfrak{o}$. If not, then return true and quit.
   b. If $\det \varphi \in (K_\mathfrak{o}^\times)^2$, then use Eq. (1) and [14, Algorithm 6.6] to compute the Hasse invariant $s_\mathfrak{o}(\varphi)$ and the Hilbert symbol $((-1, -1)_\mathfrak{o}$. Return true if they are equal, return false if they are not.
5. If $\dim \varphi \geq 5$, then return true.

Proof of correctness. An unary form is never isotropic and a quintic or higher-dimensional form over a dyadic field is always isotropic by the means of [7, Theorem VI.2.12]. This justifies steps 1 and 5. Next, it is well known that a binary form is isotropic if and only if its determinant is a minus square, which proves step 2. On the other hand, if the form has dimension three, then [7, Proposition V.3.22] asserts that it is isotropic if and only if $((-1, -\det(\varphi))_\mathfrak{o} = s_\mathfrak{o}(\varphi)$.

This leaves us with quaternary forms. Now, [7, Corollary VI.2.15] asserts that over a local field there is only one anisotropic form of dimension 4 and its determinant is a square. Thus, if $\det \varphi \notin (K_\mathfrak{o}^\times)^2$, then $\varphi \otimes K_\mathfrak{o}$ is necessarily isotropic. On the other hand, if $\det \varphi \in (K_\mathfrak{o}^\times)^2$, then [7, Proposition V.3.23] provides us with a needed criterion for isotropy. □

We present the pseudo-code of this algorithm in Listing 6. Now, we are ready to present the main algorithm of this section, that checks if a form is isotropic over a given number field.

Algorithm 7. Given a quadratic form $\varphi = \langle a_1, \ldots, a_d \rangle$ over $K$ with $a_i \in \mathcal{O}_K$, this algorithm returns true if and only if $\varphi$ is isotropic and false if it is not.

1. If $\dim \varphi \leq 1$, then return false and quit.
2. If $\dim \varphi = 2$, then use Algorithm 1 to check if $\text{disc} \varphi$ is a square in $K$. If so, then return false; if not, return true.
3. Use a real-root separation algorithm (e.g. [8, §8.5.2]) on the generating polynomial of $K$ to find all the real embeddings $\rho_1, \ldots, \rho_r$ of $K$. Compute
the signature $s_j := \text{sgn} \rho_j(\varphi)$ of image of $\varphi$ under each embedding. If $|s_j| = \dim \varphi$ for any $1 \leq j \leq r$, then return false and quit.

(4) Use [11, §6.2.5] to factor $2\mathcal{O}_K$ into prime ideals $2\mathcal{O}_K = \mathfrak{d}_1^e \cdots \mathfrak{d}_n^e$ in $\mathcal{O}_K$. For each $\mathfrak{d}_i$ use Algorithm 5 to check if $\varphi \otimes K_{\mathfrak{d}_i}$ is isotropic. If the algorithm returns false, for at least one $\mathfrak{d}_i$, then return false and quit.

(5) Use [2, 2.3.22] to find all odd primes $p$ of $K$ dividing any of the coefficients $a_i$ of $\varphi$. For each such a prime $p$ call Algorithm 5. If the procedure returns false at least once, then return false and quit.

(6) Return true.

Proof of correctness. The cases of unary and binary forms are trivial. For forms of higher dimension we use the local-global principle [7, Principle VI.3.1]. The form is isotropic over $K$ if it is isotropic over all the completions of $K$. Now $\varphi$, having dimension at least three, is trivially isotropic at all odd primes that do not divide any of the coefficients. These are almost all primes of $K$. Thus, we are left with only finitely many cases to check: finitely many real places treated in step 3, finitely many dyadic places covered by step 4, and finitely many non-dyadic primes considered in step 5. □

The pseudo-code of Algorithm 7 is presented in Listing 7.

Remark 2.1. In order to compute in step 3 the signatures of $\varphi$ under real embeddings of $K$, one may proceed as follows. Write all the roots $\vartheta_1, \ldots, \vartheta_r$ of the defining polynomial $f$ in the interval representation as explained in [8, §8.5]. Let $a_j = a_{j,0} + a_{j,1}(\vartheta + \cdots + a_{j,n-1} \vartheta^{n-1})$ be a coefficient of $\varphi$ expressed in the power basis representation. Take $g_j := a_{j,0} + a_{j,1}(x + \cdots + a_{j,n-1} x^{n-1})$ and use [8, §8.5, Sign evaluation] to compute $s_{ji} = \text{sgn}(g_j(\vartheta_i))$ for every $1 \leq i \leq r$. Then $s_i = \sum_j s_{ji}$.

3. Hyperbolicity of a quadratic form

In this section we present an algorithm checking another fundamental property of a quadratic form, namely whether it is hyperbolic (hence, a zero element in the Witt group). The general idea is similar to the one adopted in the previous section. Again, we treat the problem locally, separately for odd primes, even primes and real embeddings of $K$.

Algorithm 8. Given a quadratic form $\varphi = \langle a_1, \ldots, a_d \rangle$ over $K$ with $a_i \in \mathcal{O}_K$, this algorithm returns true if and only if $\varphi$ is hyperbolic, otherwise it returns false.

(1) If $\dim \varphi$ is odd, then return false and quit.

(2) Compute the discriminant $\text{disc} \varphi$. Use Algorithm 1 to check if $\text{disc} \varphi$ is a square in $K$. If it is not, then return false.

(3) Use a real-root separation algorithm (e.g. [8, §8.5.2]) on the generating polynomial of $K$ to find all the real embeddings $\rho_1, \ldots, \rho_r$ of $K$. Compute the signature $s_j := \text{sgn} \rho_j(\varphi)$ of image of $\varphi$ under each embedding. If $s_j \neq 0$ for any $1 \leq j \leq r$, then return false and quit.

(4) Find all odd primes of $K$ dividing any of the coefficients $a_i$ of $\varphi$, using [2, 2.3.22]. For each such a prime $p$ proceed as follows:

(a) Let $\pi$ be the uniformizer of $p$ found using [11, Algorithm 4.8.17].

(b) Take $\varphi_1 := \{a_i \pi^{-\text{ord}_p a_i} : \text{ord}_p a_i \equiv 1 \pmod{2}\}$.

(c) If $\dim \varphi_1$ is odd, then return false and quit.

(d) If the residue class of $\text{disc} \varphi_1$ is not a square in the residue field $K/p$, then return false and quit.

(5) Factor 2 over $\mathcal{O}_K$ using [11, §6.2.5] to find all even primes of $K$. For each even prime $\mathfrak{d}$ compute the Hasse invariant $s_\mathfrak{d}(\varphi)$ using Eq. 1 and 13.
Algorithm 6.6]. If \( s_\varphi(\varphi) \neq 1 \) for at least one prime \( \varphi \), then return false and quit.

(6) Return true.

Proof of correctness. It is well known that the discriminant of a hyperbolic form is a square and its dimension has to be even. Moreover, the signature of a hyperbolic form over any real closed field is zero. Therefore, the form \( \varphi \) over \( K \) must have signature zero under every real embedding of \( K \), in order to be hyperbolic over \( K \).

Now, suppose that a form \( \varphi \) of an even dimension and with discriminant being a square has a zero signature in every ordering of \( K \). Fix an odd prime \( p \) and let \( \varphi_1 := \{ a_i \cdot \pi^{-\ord_p a_i} \mid \ord_p a_i \equiv 1 \pmod{2} \} \). It follows from [13] Theorem 9.2.3, that the second residue homomorphism of \( \varphi \), with respect to \( p \), is zero if and only if the form \( \varphi_1 \otimes K/p \) is even-dimensional and its discriminant is a square (in \( K/p \)).

If the second residual homomorphisms with respect to all odd primes of \( K \) are null, then the Witt class of \( \varphi \) sits in \( \mathcal{W}(O_K) \cap I^2K \) by [5] Ch. IV, Corollary 4.5], where \( \mathcal{W}(O_K) \) denotes the nilradical of the Witt ring of \( O_K \) and \( I \) is the fundamental ideal of the Witt ring \( WK \). Clearly one needs to check only these primes that divide any of the coefficients of \( \varphi \) as we do in step [4]. Now, let \( \varphi_1, \ldots, \varphi_d \) be all the dyadic primes of \( K \), [4] Proposition 3.5 asserts that the map \( \varphi \mapsto (s_{\varphi_1}(\varphi), \ldots, s_{\varphi_d}(\varphi)) \) as an isomorphism from \( \mathcal{W}(O_K) \cap I^2K \) onto \( \{\pm1\}^{g-1} \). Thus, \( \varphi \) is hyperbolic if and only if all its dyadic Hasse invariants vanish. This proves the correctness of the algorithm.

\[ \square \]

In the previous algorithm we did not need a separate sub-procedure for checking if a given form is hyperbolic at a given finite prime \( p \) of \( K \). Nevertheless, we present such an algorithm below. First of all, it nicely mirrors Algorithms [5] and [6] from the previous section. More importantly, it is used in the next section.

Algorithm 9. Let \( p \) be a finite prime of a number field \( K \) (either even or odd). Given a quadratic form \( \varphi \), this algorithm returns true if the form \( \varphi_p := \varphi \otimes K_p \) is hyperbolic and false otherwise.

(1) If \( \dim \varphi \) is odd, then return false and quit.

(2) Compute the discriminant \( \text{disc} \varphi \) and check if it is a square in the completion \( K_p \) (use either Algorithm [2] or [3]). If it is not a square, then return false and quit.

(3) use Eq. (1) and [4] Algorithm 6.6] to compute the Hasse invariant \( s_p(\varphi) \) and the power \((-1, -1)_p^{(m-1)/2}\) of \( p \)-adic Hilbert symbol, where \( 2m = \dim \varphi \). Return true if they are equal, return false if they are not.

Proof of correctness. Take a form \( \varphi \) of an even dimension. If the discriminant \( \text{disc} \varphi \) is a square in \( K_p \) and the Hasse invariant \( s_p(\varphi) \) equals \((-1, -1)_p^{(m-1)/2}\), then \( \varphi \) is isometric to the hyperbolic space \( m(1, -1) \) by [7] Proposition V.3.25.

\[ \square \]

4. Witt index of a quadratic form

Recall (see e.g. [7] Chapter i, §4] that any non-degenerate quadratic form \( \varphi \) can be uniquely (up to an isometry) decomposed as \( \varphi = \psi \perp H \), where \( \psi \) is an anisotropic form, called the anisotropic part of \( \varphi \) and \( H \) is hyperbolic. The number of hyperbolic planes constituting \( H \) (i.e. half of the dimension of \( H \)) is called the Witt index of \( \varphi \) and denoted \( \text{ind}(\varphi) \). In this section we present an algorithm that computes the dimension of the anisotropic part of \( \varphi \). It can be also used to educe the Witt index since clearly \( \text{ind} \varphi \equiv \frac{1}{2} \cdot (\dim \varphi - \dim \psi) \). Again, the problem is first solved locally (see Algorithm [10] and then the local solution is used to derive the global one in Algorithm [11].
Algorithm 10. Given a non-degenerate quadratic form \( \varphi \) over a number field \( K \) and a finite prime \( p \), this algorithm computes the dimension of the anisotropic part of \( \varphi_p := \varphi \otimes K_p \) over the completion \( K_p \).

1. \( \dim \varphi \) is even, proceed as follows:
   (a) Use Algorithm [9] to check if \( \varphi_p \) is hyperbolic. If it is, then return 0 and quit.
   (b) Check if \( \text{disc} \varphi \) is a square in \( K_p \) using either Algorithm [2] (if \( p \) is odd) or Algorithm [3] (when \( p \) is even). If so, then return 2 and quit.
   (c) Return 4.

2. \( \dim \varphi \) is odd, proceed as follows:
   (a) Let \( n := \dim \varphi \) and take \( \psi := \varphi \perp \langle (-1)^{n+1/2}, \det \varphi \rangle \).
   (b) Use Algorithm [9] to check if \( \psi \otimes K_p \) is hyperbolic. If it is, then return 1 and quit.
   (c) Return 3.

Proof of correctness. First assume that \( \varphi \) is an even-dimensional form, so \( \varphi_p \in IK_p \).
If it is not hyperbolic, then its class in the Witt ring \( W/K_p \) is not zero. Suppose that \( \text{disc} \varphi \) is a square in \( K_p \). It follows that the anisotropic part of \( \varphi_p \) has dimension 4.
Conversely, suppose that \( \text{disc} \varphi \) is not a square in \( K_p \). Therefore \( \varphi_p \in IK_p \setminus I^2K_p \) and so the anisotropic part of \( \varphi_p \) has dimension 2.

Now assume that the dimension of \( \varphi \) is odd. Hence, the form \( \psi \) constructed in step [24] is an even dimensional form and its discriminant is a square in \( K_p \). Consequently, the Witt class of \( \psi_p := \psi \otimes K_p \) sits in \( I^2K_p \). If \( \psi_p \) is hyperbolic then \( \psi_p \cong \frac{n+1}{2}(1,-1) \), hence \( \varphi_p \perp (1,-1) \cong \langle c \rangle \perp \frac{n+1}{2}(1,-1) \) for \( c = -(-1)^{n+1/2} \det \varphi \).
This implies that the anisotropic part of \( \varphi_p \) is unary. Conversely, suppose that \( \psi_p \) is not hyperbolic. As in the first part of the proof, this leads to \( \psi_p = \eta_p \) in the Witt ring \( W/K_p \). In particular, the Witt classes of \( \varphi_p \) and \( \langle c, 1, u, \pi, u\pi \rangle \) are equal. But a quintic form over a local field is necessarily isotropic and so it is similar to either ternary or unary form. We claim that the unary case is impossible. Indeed, if \( \langle c, 1, u, \pi, u\pi \rangle \cong \langle x \rangle \perp 2(1,-1) \), then square classes of \( c \) and \( x \) are equal and the Witt cancellation theorem asserts that the forms \( \langle 1, u, \pi, u\pi \rangle \) and \( 2(1,-1) \) are isometric over \( K_p \) contradicting [7] Corollary VI.2.15. All in all, \( \langle c, 1, u, \pi, u\pi \rangle \) has a ternary anisotropic part and so has \( \varphi_p \).

Algorithm 11. Given a non-degenerate quadratic form \( \varphi \) over a number field \( K \), this algorithm computes the dimension of the anisotropic part of \( \varphi \).

1. Use a real-root separation algorithm (e.g. [3] §8.5.2) on the generating polynomial of \( K \) to find all the real embeddings \( \rho_1, \ldots, \rho_r \) of \( K \) and compute the maximum of the signatures

\[
N := \max_{1 \leq j \leq r} |\text{sgn} \rho_j(\varphi)|.
\]

2. If \( N \geq 3 \), then return \( N \) and quit.
3. Find all primes of \( K \) dividing any of the coefficients \( a_i \) of \( \varphi \), using [2] 2.3.22.
4. Factor 2 over \( \mathcal{O}_K \) using [1] §6.2.5 to find all even primes of \( K \).
5. Let \( \mathcal{L} \) be the set consisting of all even primes of \( K \) and all odd primes dividing any of the coefficients of \( \varphi \).
6. For every \( p \in \mathcal{L} \) compute the dimension \( \begin{array}{|c|c|} d_p \end{array} \) of the anisotropic part of \( \varphi \otimes K_p \) using Algorithm [10] and let \( M = \max \{ d_p \mid p \in \mathcal{L} \} \).
8 PRZEMYSŁAW KOPROWSKI AND ALFRED CZOGAŁA

(7) Return $\max\{M, N\}$.

**Proof of correctness.** Let $\psi$ be the anisotropic part of $\varphi$. Obviously
\[
\dim \psi \equiv \dim \varphi \equiv |\sgn \rho_j(\varphi)| \pmod{2}
\]
for any real embedding $\rho_j$ of $K$. Take $N$ to be the maximum of the signatures of $\varphi$ at all the real places. Now, $\psi$ being anisotropic must be anisotropic at some place of $K$, either finite or infinite. Therefore, clearly $\dim \psi$ is the maximum of the dimensions of the anisotropic parts of the localizations of $\varphi$ at all the places of $K$. However, if $N < 3$, then we do not need to consider the finite primes at all. Indeed, if $\dim \psi \geq 5$, then [7, Theorem VI.2.2] implies that it must be an infinite, hence real, place. Therefore in this case $\dim \psi$ is the maximum of the dimensions of the anisotropic parts of the localizations of $\varphi$ at all the places of $K$. On the other hand, it cannot be strictly greater, since otherwise $\psi$ would have to be anisotropic at some finite place contrary to the already mentioned [7, Theorem VI.2.2]. □

5. LEVEL OF A NUMBER FIELD

In this section we present an algorithm determining an important invariant of a number field, namely its level. Recall that a level of a field $K$, denoted $s(K)$ is the minimal number of terms needed to represent $-1$ as a sum of squares in $K$. When $-1$ cannot be expressed as a sum of squares (i.e. $K$ is formally real), $s(K) = \infty$.

**Algorithm 12.** Given a number field $K = \mathbb{Q}(\vartheta)$ specified by its defining polynomial $f$, this algorithm computes the level $s(K)$.

1. Use Sturm’s sequence to check if $f$ has any real roots. If so, then return $\infty$ and quit.
2. Use Algorithm 1 to check if $-1$ is a square in $K$. If so, then return 1 and quit.
3. Use [1, §6.2.5] to find the factorization of 2 in $\mathcal{O}_K$ in the form of a list $\mathcal{L}$ consisting of triples $(d_j, e_j, f_j)$, where $d_j$ is a prime of $K$ dominating 2 with the ramification index $e_j$ and the inertia degree $f_j$.
4. If for any $j$, both $e_j$ and $f_j$ are odd, then return 4 and quit.
5. Return 2.

**Proof of correctness.** The standard Sturm’s sequence method counts the number of real embeddings of $K$. If this number is positive, then $K$ is formally real and consequently its level equals $s(K) = \infty$. Next, Algorithm 1 verifies if $-1$ is a square in $K$, if so then $s(K) = 1$. Otherwise, $s(K)$ is either 2 or 4. In order to distinguish between these two cases, [7, Proposition XI.2.11] comes in handy. It asserts that $s(K) = 4$ if and only if there is $1 \leq j \leq k$ such that $d_j = e_j f_j$ is odd. Otherwise, $s(K) = 2$. This is precisely what step 4 at the end of the algorithm is for. □

The pseudo-code of this algorithm is presented in Listing 10.

6. WITT EQUIVALENCE

The ultimate problem in the algebraic theory of quadratic forms is to find criteria for an existence of an isomorphism between the Witt rings of two fields. Such fields are then called **Witt equivalent** if the above-mentioned isomorphism exists. In this section we present an algorithm computing the complete set of Witt equivalence invariants of a given number field. In particular, comparing the results returned by the algorithm one can check whether two number fields are Witt equivalent or not. It is known (see e.g. [12]) that the following invariants fully determine the Witt class of a number field $K$:

```python
# Pseudo-code for Algorithm 12
```

```python
# Pseudo-code for Algorithm 1
```
Let \( d = (K : \mathbb{Q}) \) the degree of \( K \) over \( \mathbb{Q} \);

- \( r \) the number of real embeddings of \( K \);
- \( s = s(K) \) the level of \( K \);
- \( k \) the number of dyadic primes of \( K \);
- for each dyadic prime \( \mathfrak{q} \), with \( 1 \leq j \leq k \), the pair \((d_j, s_j)\) consisting of a local degree \( d_j = (K_{\mathfrak{q}} : \mathbb{Q}_2) \) and the local level \( s_j = s(K_{\mathfrak{q}}) \).

We claim that all these invariants are computable.

Let again \( K = \mathbb{Q}(\varpi) \) be a fixed number field specified by the minimal polynomial \( f \in \mathbb{Q}[x] \) of the generator \( \varpi \). The first two invariants \( d \) and \( r \) are trivially computable. The degree \( d \) is just the degree \( \text{deg } f \) of the defining polynomial. In order to compute \( r \) one simply counts the number of real roots of \( f \) using Sturm’s sequence (see e.g. [8 §8.4]). In the previous section we showed how to compute the level of \( K \). This leaves us only with the local invariants. Assume that the principal ideal \( 2\mathcal{O}_K \) factors into prime ideals as:

\[
2\mathcal{O}_K = \mathfrak{d}_1^{e_1} \cdots \mathfrak{d}_h^{e_h}
\]

and let \( f_j = (\mathcal{O}_K / \mathfrak{d}_j : \mathbb{F}_2) \) be the inertia degree of \( \mathfrak{d}_j \) \((1 \leq j \leq k)\). The local degree \( d_j = (K_{\mathfrak{d}_j} : \mathbb{Q}_2) \) is the product \( d_j = e_j f_j \). What we need is to determine the local level \( s_j = s(K_{\mathfrak{d}_j}) \). Fix an even prime \( \mathfrak{d} = \mathfrak{d}_j \).

**Algorithm 13.** Let \( \mathfrak{d} \) be an even prime of a number field \( K \), \( e \) be the ramification and \( f \) the inertia degree of \( \mathfrak{d} \). This algorithm computes the level \( s(K_\mathfrak{d}) \) of the dyadic completion \( K_\mathfrak{d} \) of \( K \).

1. Use Algorithm \[1\] to check if \(-1\) is a square in \( K \), if so then return 1 and quit.
2. If both \( e \) and \( f \) are odd, then return 4.
3. If \( e \) is odd but \( f \) is even, then return 2.
4. If \( e \) is even, use Algorithm \[2\] to check whether \(-1\) is a square in \( K_\mathfrak{d} \), if so then return 1, if not then return 2.

**Proof of correctness.** It is clear that if \(-1\) is a square already in \( K \), then it is also a square in \( K_\mathfrak{d} \) and so \( s(K_\mathfrak{d}) = 1 \). This justifies the first step. Suppose that \( e \) is odd. Let \( \mathbb{Q}_2(i) \) be the (unique) maximal unramified extension of \( \mathbb{Q}_2 \) contained in \( K_\mathfrak{d} \). Since the quadratic extension \( \mathbb{Q}_2(i)/\mathbb{Q}_2 \) is totally ramified (see e.g. [9 Ch. V §2]), it follows that \( i \notin \mathbb{Q}_2(\eta) \). Now, \( (\mathbb{Q}_2(\eta) : \mathbb{Q}_2) = f \) and \( (K_\mathfrak{d} : \mathbb{Q}_2) = ef \). Hence the relative degree \( (K_\mathfrak{d} : \mathbb{Q}_2(\eta)) \) equals \( e \) and so is odd. In particular \( i \notin K_\mathfrak{d} \), either. Thus \( s(K_\mathfrak{d}) \geq 2 \). Finally [7] Example XI.2.4 asserts that \( s(K_\mathfrak{d}) = 4 \) if and only if \( (K_\mathfrak{d} : \mathbb{Q}_2) \) is odd.

Conversely, assume that \( e \) is even and so is the degree \( (K_\mathfrak{d} : \mathbb{Q}_2) \). It follows from [7] Example XI.2.4 that \( s(K_\mathfrak{d}) \leq 2 \). It equals one if and only if \(-1\) is a square in \( K_\mathfrak{d} \). \( \square \)

The pseudo-code of this algorithm is contained in Listing 11. Having all the necessary ingredients ready we may now present the last algorithm of this paper that construct the complete set of Witt equivalence invariants.

**Algorithm 14.** If \( K = \mathbb{Q}(\varpi) \) is a number field specified (up to an isomorphism) by the minimal polynomial \( f \in \mathbb{Q}[t] \) of its generator, then this algorithm computes the complete set of Witt equivalence invariants of \( K \). In particular, two fields are Witt equivalent if and only if the outputs of the algorithm are the same for both fields.

1. Let \( d = \text{deg } f \).
2. Use Sturm’s sequence to compute the number \( r \) of real roots of \( f \).
3. Use Algorithm [12] to compute the level \( s = s(K) \).
(4) Use [11] §6.2.5 to factor $2\mathcal{O}_K$, let $\mathcal{L} = \{(\mathfrak{d}_j, e_j, f_j)\}$ be the output of this algorithm. Take an empty list $\mathcal{S}$.

(5) For each even prime $\mathfrak{d}_j \in \mathcal{L}$ let $d_j = e_j f_j$. Use Algorithm 13 to compute the local level $s_j = s(K_{\mathfrak{d}_j})$. Append the pair $(d_j, s_j)$ to the list $\mathcal{S}$.

(6) Sort the list $\mathcal{S}$ lexicographically.

(7) Return $(d, r, s, k, \mathcal{S})$.

Observe that the expensive step of decomposing 2 in $\mathcal{O}_K$ appears twice: in Algorithm 14 as well as in its subroutine computing the level of $K$. The same can be said about reducibility of $x^2 + 1$ in $K[x]$. Thus in an actual implementation one can combine algorithms 12–14 into a single procedure compromising the readability.

7. Example applications

In order to verify the correctness of the algorithm as well as to allow experimentation, we implemented the presented algorithm in a computer algebra system Sage [11]. A formula for the number of Witt classes of number fields of a fixed degree was developed in [12]. Nevertheless, actual representatives for these classes was only found for quadratic and cubic fields in [12] and for quartic fields in [6]. The first test for usability of our implementation was to find new representatives of all 29 classes of quartic fields. It turned out that just by a random search, our implementation was able to find representatives with generating polynomials having smaller coefficients (the biggest coefficient in our case is 752 vs. 208 042 in [6]). The results are presented in Table 1. Next, we found the representatives of all 36 classes of quintic fields (see Table 2) and all 95 classes of sextic fields (see Table 3). These two results are completely new, showing the usability of the developed algorithms.

Finally, we used the implementation to test empirically how the different Witt classes are distributed. To this end, for each of the degrees 3, 4 and 5, we randomly generated 5000 fields and counted the numbers of different Witt classes among the outputs. Tables 4–6 and charts 1–3 summarize the findings.

Remark 7.1. Observe that some of Witt classes are extremely rare and virtually impossible to be found by a blind random search (in particular, such a blind search was not able to find a single representatives of classes (xxxvi), (xxiv), (xii) of quintic fields, even after 24 hours of computing time). These are mostly the classes of fields where 2 splits completely. In order to find the representatives of these classes one may proceed as follows. Denote $|a|_2 := 2^{-\operatorname{ord}_2 a}$ the canonical dyadic norm and let $\| (a_0, \ldots, a_d) \|_2 := \max \{|a_0|_2, \ldots, |a_d|_2\}$ be the associated norm of the vector space $\mathbb{Q}_2^{d+1}$. Take a polynomial $f$ with $d$ distinct integral roots. For example, for quintic fields we used just $f = (x - 1)(x - 3)(x - 5)(x - 7)(x - 11)$.

Write it as a dot product $f = V \cdot X$, where $V$ is the vector of coefficients of $f$ and $X = (1, x, x^2, \ldots, x^d)^T$ are the powers of $x$. Take now some random vector $W$ and let $\tilde{f} = (V + W) \cdot X$. If the norm $\|W\|_2$ of $W$ is small enough, then $\tilde{f}$ still has $d$ distinct roots in $\mathbb{Q}_2$ but there is a good chance that it is irreducible over $\mathbb{Q}$. It follows that 2 splits completely in the field $K = \mathbb{Q}[x]/(\tilde{f})$, as desired.

References

[1] H. Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.

[2] H. Cohen. *Advanced topics in computational number theory*, volume 193 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2000.

[3] J. E. Cremona and D. Rusin. Efficient solution of rational conics. *Math. Comp.*, 72(243):1417–1441 (electronic), 2003.
[4] A. Czogala. Witt rings of Hasse domains of global fields. *J. Algebra*, 244(2):604–630, 2001.
[5] D. H. J. Milnor. *Symmetric Bilinear Forms*, volume 73 of *Ergebnisse der Matematik und ihrer Grenzgebiete*. Springer Verlag, New York, 1973.
[6] S. Jakubec and F. Marko. Witt equivalence classes of quartic number fields. *Math. Comp.*, 58(197):355–368, 1992.
[7] T. Y. Lam. *Introduction to quadratic forms over fields*, volume 67 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2005.
[8] B. Mishra. *Algorithmic algebra*. Texts and Monographs in Computer Science. Springer-Verlag, New York, 1993.
[9] W. Narkiewicz. *Elementary and analytic theory of algebraic numbers*. Springer-Verlag, Berlin, second edition, 1990.
[10] D. Simon. Solving quadratic equations using reduced unimodular quadratic forms. *Math. Comp.*, 74(251):1531–1543 (electronic), 2005.
[11] W. Stein et al. *Sage Mathematics Software (Versions: 4.7.1 and 5.3)*. The Sage Development Team, 2010–12. [http://www.sagemath.org](http://www.sagemath.org).
[12] K. Szymiczek. Witt equivalence of global fields. *Comm. Algebra*, 19(4):1125–1149, 1991.
[13] K. Szymiczek. *Bilinear algebra*, volume 7 of *Algebra, Logic and Applications*. Gordon and Breach Science Publishers, Amsterdam, 1997. An introduction to the algebraic theory of quadratic forms.
[14] J. Voight. Identifying the matrix ring: algorithms for quaternion algebras and quadratic forms. to appear in Quadratic and higher degree forms.
[15] F. Winkler. *Polynomial algorithms in computer algebra*. Texts and Monographs in Symbolic Computation. Springer-Verlag, Vienna, 1996.

Przemysław Koprowski (corresponding author)
*E-mail address: pkoprowski@member.ams.org*
Faculty of Mathematics
University of Silesia
ul. Bankowa 14
PL-40-007 Katowice, Poland

Alfred Czogala
*E-mail address: alfred.czogala@us.edu.pl*
Faculty of Mathematics
University of Silesia
ul. Bankowa 14
PL-40-007 Katowice, Poland
Table 1. Witt classes of quartic fields

| No. | defining polynomial | $r$ | $s$ | dyadic degrees and levels |
|-----|---------------------|-----|-----|---------------------------|
| i   | $x^4 + 2x^3 - x^2 - 2x + 2$ | 0   | 1   | $\{(2, 1), (2, 1)\}$ |
| ii  | $x^4 + 1$            | 0   | 1   | $\{(4, 1)\}$            |
| iii | $x^4 + 2x^3 + 27x^2 + 2x + 2$ | 0   | 2   | $\{(2, 1), (2, 1)\}$ |
| iv  | $x^4 + x^3 + x^2 - 4x + 2$ | 0   | 2   | $\{(2, 1), (2, 2)\}$ |
| v   | $x^4 - x^3 + x + 1$    | 0   | 2   | $\{(2, 2), (2, 2)\}$ |
| vi  | $x^4 - 2x^3 - 3x^2 + 4x + 9$ | 0   | 2   | $\{(4, 1)\}$ |
| vii | $x^4 + x + 1$         | 0   | 2   | $\{(4, 2)\}$ |
| viii| $x^4 - 29x^3 + 402x^2 + 752x + 608$ | 0 | 4 | $\{(1, 4), (1, 4), (1, 4), (1, 4)\}$ |
| ix  | $x^4 - x^3 + 10x^2 - 2x + 4$ | 0 | 4 | $\{(1, 4), (1, 4), (2, 1)\}$ |
| x   | $x^4 + x + 2$         | 0   | 4   | $\{(1, 4), (1, 4), (2, 2)\}$ |
| xi  | $x^4 + x^2 - x + 1$   | 0   | 4   | $\{(1, 4), (3, 4)\}$ |
| xii | $x^4 + x^3 - 83x^2 - 70x - 96$ | 2 | $\infty$ | $\{(1, 4), (1, 4), (1, 4), (1, 4)\}$ |
| xiii| $x^4 - 6x^3 + x^2 - 2x - 2$ | 2 | $\infty$ | $\{(1, 4), (1, 4), (2, 1)\}$ |
| xiv | $x^4 - 2x^2 + x - 2$  | 2   | $\infty$ | $\{(1, 4), (1, 4), (2, 2)\}$ |
| xv  | $x^4 + x^3 - x - 2$   | 2   | $\infty$ | $\{(1, 4), (3, 4)\}$ |
| xvi | $x^4 + 6x^3 - x^2 + 2x + 10$ | 2 | $\infty$ | $\{(2, 1), (2, 1)\}$ |
| xvii| $x^4 - x^3 - x^2 - 2$ | 2 | $\infty$ | $\{(2, 1), (2, 2)\}$ |
| xviii| $x^4 + x^3 + x^2 - 2$  | 2 | $\infty$ | $\{(2, 2), (2, 2)\}$ |
| xix | $x^4 - 3x^2 - 2x - 3$ | 2   | $\infty$ | $\{(4, 1)\}$ |
| xx  | $x^4 - x^2 - 1$       | 2   | $\infty$ | $\{(4, 2)\}$ |
| xxi | $x^4 - 92x^3 + 39x^2 + 56x + 4$ | 4 | $\infty$ | $\{(1, 4), (1, 4), (1, 4), (1, 4)\}$ |
| xxii| $x^4 - 3x^3 - 10x^2 + 10x + 4$ | 4 | $\infty$ | $\{(1, 4), (1, 4), (2, 1)\}$ |
| xxiii| $x^4 + 2x^3 - 10x^2 - 5x + 2$ | 4 | $\infty$ | $\{(1, 4), (1, 4), (2, 2)\}$ |
| xxiv| $x^4 + x^3 - 4x^2 - x + 2$ | 4 | $\infty$ | $\{(1, 4), (3, 4)\}$ |
| xxv | $x^4 - 10x^3 - 5x^2 + 6x + 2$ | 4 | $\infty$ | $\{(2, 1), (2, 1)\}$ |
| xxvi| $x^4 - x^3 - 6x^2 - x + 1$ | 4 | $\infty$ | $\{(2, 1), (2, 2)\}$ |
| xxvii| $x^4 + 3x^3 - 4x^2 - 3x + 1$ | 4 | $\infty$ | $\{(2, 2), (2, 2)\}$ |
| xxviii| $x^4 + 4x^3 - 4x + 1$ | 4 | $\infty$ | $\{(4, 1)\}$ |
| xxix| $x^4 + x^3 - 3x^2 - x + 1$ | 4 | $\infty$ | $\{(4, 2)\}$ |
Table 2. Witt classes of quintic fields

| No. | defining polynomial                      | r | dyadic degrees and levels |
|-----|------------------------------------------|---|--------------------------|
| i   | $x^5 + x^2 + 1$                          | 1 | $\{ (5, 4) \}$          |
| ii  | $x^5 - 2x^4 + 3x^3 + x + 2$              | 1 | $\{ (1, 4), (4, 1) \}$  |
| iii | $x^5 + x^4 + x^3 + 2x + 1$              | 1 | $\{ (1, 4), (4, 2) \}$  |
| iv  | $x^5 - 3x^3 + x^2 - 2$                  | 1 | $\{ (2, 1), (3, 4) \}$  |
| v   | $x^5 - x + 1$                           | 1 | $\{ (2, 3), (3, 4) \}$  |
| vi  | $x^5 - x^3 + x^2 - x + 2$                | 1 | $\{ (1, 4), (1, 4), (3, 4) \}$ |
| vii | $x^5 + 3x^4 - 2x - 4$                   | 1 | $\{ (1, 4), (2, 1), (2, 1) \}$ |
| viii| $x^5 - 2x^4 - x^3 + 2x - 2$             | 1 | $\{ (1, 4), (2, 1), (2, 2) \}$ |
| ix  | $x^5 - x^2 - 2$                         | 1 | $\{ (1, 4), (2, 2), (2, 2) \}$ |
| x   | $x^5 - 4x^4 - 3x^3 + 3x - 1$             | 1 | $\{ (1, 4), (1, 4), (1, 4), (2, 1) \}$ |
| xi  | $x^5 - x^4 - 2x^3 - 3x^2 + x - 1$       | 1 | $\{ (1, 4), (1, 4), (1, 4), (2, 2) \}$ |
| xii | $x^5 + 37x^4 + 774x^3 - 1058x^2 + 2169x - 643$ | 1 | $\{ (1, 4), (1, 4), (1, 4), (1, 4) \}$ |
| xiii| $x^5 + x^4 + x^2 - x - 1$                | 3 | $\{ (5, 4) \}$          |
| xiv | $x^5 + x^4 - x^3 - x^2 - 3x + 1$        | 3 | $\{ (1, 4), (4, 1) \}$  |
| xv  | $x^5 - 2x^3 - 2x - 1$                   | 3 | $\{ (1, 4), (4, 2) \}$  |
| xvi | $x^5 - 2x^3 - x^2 - 2x + 2$             | 3 | $\{ (2, 1), (3, 4) \}$  |
| xvii| $x^5 - x^4 - 2x^3 + 1$                  | 3 | $\{ (2, 2), (3, 4) \}$  |
| xviii| $x^5 - x^3 + x^2 - 3x - 2$             | 3 | $\{ (1, 4), (1, 4), (3, 4) \}$ |
| xix | $x^5 - 5x^3 - 2x - 4$                   | 3 | $\{ (1, 4), (2, 1), (2, 2) \}$ |
| xx  | $x^5 - 2x^4 - 3x^3 + 2x + 4$             | 3 | $\{ (1, 4), (2, 2), (2, 2) \}$ |
| xxi | $x^5 + x^4 + x^2 - 3x - 2$              | 3 | $\{ (1, 4), (2, 1), (2, 2) \}$ |
| xxii| $x^5 + 26x^4 + 19x^3 + 24x^2 - 26x - 12$ | 3 | $\{ (1, 4), (1, 4), (1, 4), (2, 1) \}$ |
| xxiii| $x^5 + 10x^4 - 11x^3 - 34x - 16$       | 3 | $\{ (1, 4), (1, 4), (1, 4), (2, 2) \}$ |
| xxiv| $x^5 + 101x^4 + 326x^3 + 926x^2 + 2105x + 893$ | 3 | $\{ (1, 4), (1, 4), (1, 4), (1, 4) \}$ |
| xxv | $x^5 + 4x^4 - 5x^2 + 1$                 | 5 | $\{ (5, 4) \}$          |
| xxvi| $x^5 - x^4 - 11x^3 - x^2 + 7x + 1$     | 5 | $\{ (1, 4), (4, 1) \}$  |
| xxvii| $x^5 - 5x^4 + 13x^2 - 5$               | 5 | $\{ (1, 4), (4, 2) \}$  |
| xxviii| $x^5 - 15x^4 + 18x^3 + 37x^2 - 16x - 12$ | 5 | $\{ (1, 4), (1, 4) \}$  |
| xxix| $x^5 - 7x^4 + 6x^2 - 1$                 | 5 | $\{ (1, 4), (3, 4) \}$  |
| xxx | $x^5 + x^4 + 3x^3 - 12x^2 + x + 2$     | 5 | $\{ (1, 4), (1, 4), (3, 4) \}$ |
| xxxi | $x^5 + 31x^4 + 244x^3 + 524x^2 + 99x - 19$ | 5 | $\{ (1, 4), (2, 1), (2, 1) \}$ |
| xxxii| $x^5 + 26x^4 - 5x^3 - 93x^2 + 42x + 17$ | 5 | $\{ (1, 4), (2, 1), (2, 2) \}$ |
| xxxiii| $x^5 + x^4 - 11x^3 + 14x - 4$        | 5 | $\{ (1, 4), (2, 2), (2, 2) \}$ |
| xxxiv| $x^5 + 247x^4 + 2064x^3 + 4144x^2 + 1887x + 233$ | 5 | $\{ (1, 4), (1, 4), (1, 4), (2, 1) \}$ |
| xxxv | $x^5 + 9x^4 + 14x^3 + 29x^2 - 21x + 2$ | 5 | $\{ (1, 4), (1, 4), (1, 4), (2, 2) \}$ |
| xxxvi| $x^5 + 997x^4 + 33030x^3 + 31646x^2 + 6137x - 131$ | 5 | $\{ (1, 4), (1, 4), (1, 4), (1, 4) \}$ |
| No. | defining polynomial | r | s | dyadic degrees and levels |
|-----|---------------------|---|---|--------------------------|
| i   | $x^6 + x^3 - 2x^3 + 1$ | 0 | 1 | $\{ (6, 1) \}$          |
| ii  | $x^6 + x^3 + 2x^3 + x^2 + 2$ | 0 | 1 | $\{ (2, 1), (4, 1) \}$  |
| iii | $x^6 + 5x^4 + 19x^4 + 2x^2 + 32x + 64$ | 0 | 1 | $\{ (2, 1), (2, 1), (2, 1) \}$ |
| iv  | $x^6 + x^3 + 2x^3 + 8x^2 + 1$ | 0 | 2 | $\{ (6, 1) \}$          |
| v   | $x^6 + x + 1$ | 0 | 2 | $\{ (6, 2) \}$          |
| vi  | $x^6 + x^4 + 15x^2 + 2x + 2$ | 0 | 2 | $\{ (2, 1), (4, 1) \}$  |
| vii | $x^6 + x^3 - x^2 + 2$ | 0 | 2 | $\{ (2, 1), (4, 2) \}$  |
| viii| $x^6 + 2x^5 + x^4 - 2x^3 + 2x^2 + 2$ | 0 | 2 | $\{ (2, 2), (4, 1) \}$  |
| ix  | $x^6 + x^3 + x^2 + x + 1$ | 0 | 2 | $\{ (2, 2), (4, 2) \}$  |
| x   | $x^6 + 15x^3 - 30x^3 + 86x^2 - 72x + 72$ | 0 | 2 | $\{ (2, 1), (2, 1), (2, 1) \}$ |
| xi  | $x^6 - x^5 + 6x^4 - x^3 - x^2 + 2$ | 0 | 2 | $\{ (2, 1), (2, 1), (2, 2) \}$ |
| xii | $x^6 - x^5 + x^3 - x^2 + 2$ | 0 | 2 | $\{ (2, 1), (2, 2), (2, 2) \}$ |
| xiii| $x^6 + x^3 - x^2 + 2$ | 0 | 2 | $\{ (2, 2), (2, 2), (2, 2) \}$ |
| xiv | $x^6 + x^3 + x^2 + 1$ | 0 | 4 | $\{ (1, 4), (5, 4) \}$  |
| xv  | $x^6 + x^3 - x^3 + 1$ | 0 | 4 | $\{ (3, 4), (3, 4) \}$  |
| xvi | $x^6 - x^5 - x^4 + x^3 - x^2 + x - 2$ | 0 | 4 | $\{ (1, 4), (1, 4), (4, 1) \}$ |
| xvii| $x^6 - x + 2$ | 0 | 4 | $\{ (1, 4), (1, 4), (1, 4), (4, 2) \}$ |
| xviii| $x^6 - x^3 + x^3 + x^2 + 2$ | 0 | 4 | $\{ (1, 4), (2, 1), (3, 4) \}$ |
| xix | $x^6 + x^5 + x^2 + 1$ | 0 | 4 | $\{ (1, 4), (2, 2), (3, 4) \}$ |
| xx  | $x^6 - x^3 + 5x^4 - 2x^2 + x + 4$ | 0 | 4 | $\{ (1, 4), (1, 4), (1, 4), (3, 4) \}$ |
| xxi | $x^6 + 8x^5 + 26x^4 + 70x^3 + 122x^2 + 186x + 72$ | 0 | 4 | $\{ (1, 4), (1, 4), (2, 1), (2, 1) \}$ |
| xxii| $x^6 - x^3 + 3x^3 + 17x^2 + 4$ | 0 | 4 | $\{ (1, 4), (1, 4), (2, 1), (2, 2) \}$ |
| xxiii| $x^6 + x^3 + 3x^3 + 2x^2 + x + 2$ | 0 | 4 | $\{ (1, 4), (1, 4), (2, 2), (2, 2) \}$ |
| xxiv| $x^6 + 36x^3 + 1011x^4 + 224x^2 + 95x^2 - 4x + 173$ | 0 | 4 | $\{ (1, 4), (1, 4), (1, 4), (1, 4), (2, 1) \}$ |
| xxv | $x^6 + 112x^5 + 8256x^4 + 160x^3 + 7179x^2 + 2544x + 2731$ | 0 | 4 | $\{ (1, 4), (1, 4), (1, 4), (1, 4), (2, 2) \}$ |
| xxvi| $x^6 + 10x^3 - 16x^3 + 148x^2 + 352x + 512$ | 0 | 4 | $\{ (1, 4), (1, 4), (1, 4), (1, 4), (1, 4), (1, 4) \}$ |
| xxvii| $x^6 + x^3 - 2x^3 - 3$ | 2 | $\infty$ | $\{ (6, 1) \}$          |
| xxviii| $x^6 + x^3 - 1$ | 2 | $\infty$ | $\{ (6, 2) \}$          |
| xxix| $x^6 - x^5 - x^3 - x^2 + x - 1$ | 2 | $\infty$ | $\{ (1, 4), (5, 4) \}$  |
| xxx  | $x^6 - 4x^5 - 2x^4 + 2x^3 - 2x^2 - 2x + 1$ | 2 | $\infty$ | $\{ (2, 1), (4, 1) \}$  |
| xxxi | $x^6 + x^5 + x^4 + x^3 - x^2 - 2$ | 2 | $\infty$ | $\{ (2, 1), (4, 2) \}$  |
| xxxii| $x^6 - 3$ | 2 | $\infty$ | $\{ (2, 2), (4, 1) \}$  |
| xxxiii| $x^6 - x^5 - 1$ | 2 | $\infty$ | $\{ (2, 2), (4, 2) \}$  |
| xxxiv| $x^6 + x^5 + x^3 - 1$ | 2 | $\infty$ | $\{ (3, 4), (3, 4) \}$  |
| xxxv | $x^6 - x^5 - x^3 - x^2 + x^2 + x - 4$ | 2 | $\infty$ | $\{ (1, 4), (1, 4), (4, 1) \}$ |
| xxxvi| $x^6 - x^4 - x^3 - x + 2$ | 2 | $\infty$ | $\{ (1, 4), (1, 4), (4, 2) \}$ |
| xxxvii| $x^6 + 2x^5 - x^4 - x^3 - x^2 - 2$ | 2 | $\infty$ | $\{ (1, 4), (1, 4), (3, 4) \}$ |
| xxxviii| $x^6 - x^3 + x^2 - x - 2$ | 2 | $\infty$ | $\{ (1, 4), (2, 2), (3, 4) \}$ |
| xxxix| $x^6 - 2x^5 - 2x^4 + 3x^3 + 10x^2 + 25x^2 - 28x - 28$ | 2 | $\infty$ | $\{ (2, 1), (2, 1), (2, 1) \}$ |
| xlx | $x^6 + 3x^3 - x^3 + x^2 - 2$ | 2 | $\infty$ | $\{ (2, 1), (2, 1), (2, 2) \}$ |
| xli | $x^6 - x^3 + x^3 + x^2 - 2x - 2$ | 2 | $\infty$ | $\{ (2, 1), (2, 2), (2, 2) \}$ |

Continued on the next page
Table 3: Witt classes of sextic fields

| No. | defining polynomial | r      | s      | dyadic degrees and levels |
|-----|---------------------|--------|--------|---------------------------|
| xl   | $x^6 - x^4 - 2x^3 - x^2 + x^2 - 2x - 2$ | 2      | $+\infty$ | $(2, 2), (2, 2), (2, 2)$ |
| xlii | $x^6 + 5x^5 + x^4 + 2x^3 + x^2 - 2$ | 2      | $+\infty$ | $(1, 4), (1, 4), (1, 4), (3, 4)$ |
| xliii| $x^6 + 6x^4 - 2x^3 + x^2 - 2x - 8$ | 2      | $+\infty$ | $(1, 4), (1, 4), (2, 1), (2, 1)$ |
| xliv | $x^6 - 2x^5 - 2x^4 + x^3 - 2x^2 + x$ | 2      | $+\infty$ | $(1, 4), (1, 4), (2, 1), (2, 2)$ |
| xlv  | $x^6 - 3x^5 - x^4 + x^3 - x^2 + x - 2$ | 2      | $+\infty$ | $(1, 4), (1, 4), (2, 2), (2, 2)$ |
| xlvii| $x^6 - 232x^5 - 479x^4 - 440x^3 - 502x^2 - 348x - 64$ | 2      | $+\infty$ | $(1, 4), (1, 4), (1, 4), (1, 4), (2, 1)$ |
| xlviii| $x^6 - 45x^5 - 38x^4 - x^3 + 41x^2 - 6x - 16$ | 2     | $+\infty$ | $(1, 4), (1, 4), (1, 4), (1, 4), (2, 2)$ |
| xlix| $x^6 - 53x^5 - 16x^4 + 868x^3 - 800x^2 - 4288$ | 2     | $+\infty$ | $(1, 4), (1, 4), (1, 4), (1, 4), (1, 4)$ |
| li   | $x^6 + 6x^5 + 2x^4 - 2x^3 - 5x^2 - 2x + 1$ | 4      | $+\infty$ | $(6, 1)$ |
| lii  | $x^6 - 2x^4 - x^3 - 2x^2 + 7x + 1$ | 4      | $+\infty$ | $(6, 2)$ |
| liii | $x^6 + 6x^5 - x^4 - 4x^3 - 4x^2 + 2$ | 4      | $+\infty$ | $(1, 4), (5, 4)$ |
| liv  | $x^6 - 2x^5 - 3x^4 + 3x^3 - 3x^2 + 2$ | 4      | $+\infty$ | $(2, 1), (4, 1)$ |
| lv   | $x^6 - 3x^5 + 4x^4 - 3x^2 + x - 1$ | 4      | $+\infty$ | $(2, 2), (4, 1)$ |
| lvii | $x^6 - 2x^5 - x^4 - x^2 + 2x + 1$ | 4      | $+\infty$ | $(2, 2), (4, 2)$ |
| lviii| $x^6 - 29x^4 - 6x^3 - 4x^2 - 4x + 1$ | 4      | $+\infty$ | $(3, 4), (3, 4)$ |
| lvii | $x^6 - 2x^5 + x^4 - 3x^3 - 2x^2 + 3x + 1$ | 4     | $+\infty$ | $(1, 1), (4, 4), (4, 1)$ |
| lx   | $x^6 - 2x^5 - 10x^4 - 2x^3 + 1$ | 4      | $+\infty$ | $(2, 1), (2, 1), (2, 1)$ |
| lxii | $x^6 + 29x^5 + 2x^4 - 3x^3 + 2x^2 + 3x + 1$ | 4     | $+\infty$ | $(1, 1), (1, 4), (1, 4), (1, 4)$ |
| lxiii| $x^6 - 10x^5 + 3x^4 + 4x^3 + 2x^2 - 3x + 1$ | 4     | $+\infty$ | $(1, 1), (1, 4), (1, 4), (1, 4)$ |
| lxiv | $x^6 - 10x^5 + 3x^4 + 4x^3 + 2x^2 - 3x + 1$ | 4     | $+\infty$ | $(1, 1), (3, 3)$ |
| lxv | $x^6 - 2x^5 - 10x^4 - 2x^3 + 1$ | 4      | $+\infty$ | $(2, 1), (2, 1), (2, 1)$ |
| lxvi | $x^6 + 29x^5 + 2x^4 - 3x^3 + 2x^2 + 3x + 1$ | 4     | $+\infty$ | $(1, 1), (1, 4), (1, 4), (1, 4)$ |
| lxvii| $x^6 - 2x^5 - 10x^4 - 2x^3 + 1$ | 4      | $+\infty$ | $(2, 1), (2, 1), (2, 1)$ |
| lxviii| $x^6 - 10x^5 + 3x^4 + 4x^3 + 2x^2 - 3x + 1$ | 4     | $+\infty$ | $(1, 1), (1, 4), (1, 4), (1, 4)$ |
| lxix | $x^6 - 10x^5 + 3x^4 + 4x^3 + 2x^2 - 3x + 1$ | 4     | $+\infty$ | $(1, 1), (1, 4), (1, 4), (1, 4)$ |
| lxx | $x^6 - 2x^5 - 10x^4 - 2x^3 + 1$ | 4      | $+\infty$ | $(2, 1), (2, 1), (2, 1)$ |
| lxxi | $x^6 - 10x^5 + 3x^4 + 4x^3 + 2x^2 - 3x + 1$ | 4     | $+\infty$ | $(1, 1), (1, 4), (1, 4), (1, 4)$ |
| lxxii| $x^6 - 2x^5 - 10x^4 - 2x^3 + 1$ | 4      | $+\infty$ | $(2, 1), (2, 1), (2, 1)$ |
| lxxiii| $x^6 - 10x^5 + 3x^4 + 4x^3 + 2x^2 - 3x + 1$ | 4    | $+\infty$ | $(1, 1), (1, 4), (1, 4), (1, 4)$ |
| lxxiv| $x^6 - 2x^5 - 10x^4 - 2x^3 + 1$ | 4      | $+\infty$ | $(2, 1), (2, 1), (2, 1)$ |
| lxxv | $x^6 - 2x^5 - 10x^4 - 2x^3 + 1$ | 4      | $+\infty$ | $(2, 1), (2, 1), (2, 1)$ |

Continued on the next page
Table 3: Witt classes of sextic fields

| No. | defining polynomial | $r$ | $s$ | dyadic degrees and levels |
|-----|---------------------|-----|-----|---------------------------|
| lxxvi | $x^6 - 29x^4 - 2x^3 + 163x^2 - 34x - 62$ | 6 | $+\infty$ | (2,1), (3,4) |
| lxxvii | $x^6 - 57x^5 - 137x^4 + 325x^3 + 557x^2 - 372x - 86$ | 6 | $+\infty$ | (2,1), (4,2) |
| lxxviii | $x^6 - 20x^4 - 4x^3 + 61x^2 + 4x - 46$ | 6 | $+\infty$ | (2,2), (4,1) |
| lxxix | $x^6 + 16x^5 - x^3 + 15x^2 - 3x - 1$ | 6 | $+\infty$ | (2,2), (3,4) |
| lxxx | $x^6 - 42x^5 + 11x^4 + 414x^3 - 464x^2 - 386x + 423$ | 6 | $+\infty$ | (2,1), (2,1) |
| lxxxi | $x^6 - 270x^5 - 478x^4 + 836x^3 + 890x^2 - 252x + 9$ | 6 | $+\infty$ | (1,1), (4,1) |
| lxxii | $x^6 - 69x^5 + 320x^4 + 388x^3 - 252x^2 - 122x + 8$ | 6 | $+\infty$ | (2,1) |
| lxxiii | $x^6 - 140x^5 + 285x^4 + 507x^3 - 508x^2 - 470x - 64$ | 6 | $+\infty$ | (2,1), (2,2) |
| lxxiv | $x^6 - 2x^5 - 35x^4 + 20x^3 + x - 1$ | 6 | $+\infty$ | (2,1), (2,2) |
| lxxv | $x^6 - 35x^5 - 4x^4 + 196x^3 - 56x + 4$ | 6 | $+\infty$ | (2,2), (2,2) |
| lxxvi | $x^6 + 64x^5 + 132x^4 + 2x^3 - 43x^2 + 2x + 2$ | 6 | $+\infty$ | (2,2), (2,2) |
| lxxvii | $x^6 + 128x^5 + 2049x^4 + 8192x^3 + 8151x^2 + 128x - 41$ | 6 | $+\infty$ | (2,1), (2,2) |
| lxxviii | $x^6 - 10x^5 - 34x^4 + 127x^3 + 169x^2 - 88x - 54$ | 6 | $+\infty$ | (2,2), (2,2) |
| lxxix | $x^6 + 8x^5 - 105x^4 + 8x^3 + 912x^2 + 16x - 768$ | 6 | $+\infty$ | (1,1), (2,2) |
| xc | $x^6 + 2x^5 - 77x^4 + 4x^3 + 910x^2 + 4x - 768$ | 6 | $+\infty$ | (2,1) |
| xci | $x^6 + 4x^5 - 105x^4 + 4x^3 + 910x^2 + 4x - 768$ | 6 | $+\infty$ | (2,2) |
| xcii | $x^6 - 283x^5 + 131x^4 + 448x^3 - 238x^2 - 28x + 16$ | 6 | $+\infty$ | (1,4), (2,2) |
| xciii | $x^6 + 1028x^5 + 131059x^4 + 4194272x^3 + 8388639x^2 + 4194268x + 131117$ | 6 | $+\infty$ | (1,4), (4,4), (1,4) |
| xciv | $x^6 + 496x^5 + 65611x^4 + 2097152x^3 + 2096311x^2 + 6032x - 643$ | 6 | $+\infty$ | (2,2), (2,2) |
| xciv | $x^6 - 85x^5 - 16x^3 + 1156x^2 - 544x + 64$ | 6 | $+\infty$ | (1,4), (1,4) |
Table 4. Number of different Witt classes in 5000 random cubic fields.

| No. | \( r \) | dyadic degrees and levels | \# in 5000 random fields |
|-----|--------|--------------------------|--------------------------|
| i   | 1      | \{3, 4\}                | 1647                     |
| ii  | 1      | \{(1, 4), (2, 1)\}      | 71                       |
| iii | 1      | \{(1, 4), (2, 2)\}      | 908                      |
| iv  | 1      | \{(1, 4), (1, 4), (1, 4)\} | 23                      |
| v   | 3      | \{3, 4\}                | 470                      |
| vi  | 3      | \{(1, 4), (2, 1)\}      | 36                       |
| vii | 3      | \{(1, 4), (2, 4)\}      | 202                      |

Table 5. Number of different Witt classes in 5000 random quartic fields.

| No. | \( r \) | \( s \) | dyadic degrees and levels | \# in 5000 random fields |
|-----|--------|--------|--------------------------|--------------------------|
| i   | 0      | 1      | \{4, 1\}                | 46                       |
| ii  | 0      | 1      | \{(2, 1), (2, 1)\}      | 0                        |
| iii | 0      | 2      | \{4, 1\}                | 5                        |
| iv  | 0      | 2      | \{4, 2\}                | 321                      |
| v   | 0      | 2      | \{(2, 1), (2, 1)\}      | 0                        |
| vi  | 0      | 2      | \{(2, 1), (2, 2)\}      | 16                       |
| vii | 0      | 2      | \{(2, 2), (2, 2)\}      | 108                      |
| viii| 0      | 4      | \{(1, 4), (3, 4)\}      | 315                      |
| ix  | 0      | 4      | \{(1, 4), (1, 4), (2, 1)\} | 3                      |
| x   | 0      | 4      | \{(1, 4), (1, 4), (2, 2)\} | 48                      |
| xi  | 0      | 4      | \{(1, 4), (1, 4), (1, 4)\} | 0                      |
| xii | 2      | +\infty | \{4, 1\}                | 33                       |
| xiii| 2      | +\infty | \{4, 2\}                | 1087                     |
| xiv | 2      | +\infty | \{(1, 4), (3, 4)\}      | 804                      |
| xv  | 2      | +\infty | \{(2, 1), (2, 1)\}      | 2                        |
| xvi | 2      | +\infty | \{(2, 1), (2, 2)\}      | 85                       |
| xvii| 2      | +\infty | \{(2, 2), (2, 2)\}      | 287                      |
| xviii| 2      | +\infty | \{(1, 4), (1, 4), (2, 1)\} | 12                   |
| xix | 2      | +\infty | \{(1, 4), (1, 4), (2, 2)\} | 121                   |
| xx  | 2      | +\infty | \{(1, 4), (1, 4), (1, 4)\} | 0                      |
| xxi | 4      | +\infty | \{4, 1\}                | 6                        |
| xxii| 4      | +\infty | \{4, 2\}                | 82                       |
| xxiii| 4      | +\infty | \{(1, 4), (3, 4)\}      | 48                       |
| xxiv| 4      | +\infty | \{(2, 1), (2, 1)\}      | 0                        |
| xxv | 4      | +\infty | \{(2, 1), (2, 2)\}      | 2                        |
| xxvi| 4      | +\infty | \{(2, 2), (2, 2)\}      | 16                       |
| xxvii| 4      | +\infty | \{(1, 4), (1, 4), (2, 1)\} | 0                      |
| xxviii| 4      | +\infty | \{(1, 4), (1, 4), (2, 2)\} | 3                      |
| xxix| 4      | +\infty | \{(1, 4), (1, 4), (1, 4)\} | 0                      |
Table 6. Number of different Witt classes in 5000 random quintic fields.

| No. | r | dyadic degrees and levels | # in 5000 random fields |
|-----|---|---------------------------|-------------------------|
| i   | 1 | \{(5, 4)\}                | 768                     |
| ii  | 1 | \{(1, 4), (4, 1)\}        | 25                      |
| iii | 1 | \{(1, 4), (4, 2)\}        | 598                     |
| iv  | 1 | \{(2, 1), (3, 4)\}        | 53                      |
| v   | 1 | \{(2, 2), (3, 4)\}        | 513                     |
| vi  | 1 | \{(1, 4), (1, 4), (3, 4)\}| 132                     |
| vii | 1 | \{(1, 4), (2, 1), (2, 1)\}| 0                       |
| viii| 1 | \{(1, 4), (2, 1), (2, 2)\}| 24                      |
| ix  | 1 | \{(1, 4), (2, 2), (2, 2)\}| 91                      |
| x   | 1 | \{(1, 4), (1, 4), (1, 4), (2, 1)\}| 0 |
| xi  | 1 | \{(1, 4), (1, 4), (1, 4), (2, 2)\}| 7 |
| xii | 1 | \{(1, 4), (1, 4), (1, 4), (1, 4), (1, 4)\}| 0 |
| xiii| 3  | \{(5, 4)\}                | 497                     |
| xiv | 3  | \{(1, 4), (4, 1)\}        | 17                      |
| xv  | 3  | \{(1, 4), (4, 2)\}        | 311                     |
| xvi | 3  | \{(2, 1), (3, 4)\}        | 33                      |
| xvii| 3  | \{(2, 2), (3, 4)\}        | 292                     |
| xviii| 3  | \{(1, 4), (1, 4), (3, 4)\}| 53                      |
| xix | 3  | \{(1, 4), (2, 1), (2, 1)\}| 3                       |
| xx  | 3  | \{(1, 4), (2, 1), (2, 2)\}| 10                      |
| xxi | 3  | \{(1, 4), (2, 2), (2, 2)\}| 43                      |
| xxii| 3  | \{(1, 4), (1, 4), (1, 4), (2, 1)\}| 0 |
| xxiii| 3  | \{(1, 4), (1, 4), (1, 4), (2, 2)\}| 2 |
| xxiv| 3  | \{(1, 4), (1, 4), (1, 4), (1, 4)\}| 0 |
| xxv | 5  | \{(5, 4)\}                | 6                       |
| xxvi| 5  | \{(1, 4), (4, 1)\}        | 1                       |
| xxvii| 5  | \{(1, 4), (4, 2)\}       | 4                       |
| xxviii| 5  | \{(2, 1), (3, 4)\}       | 0                       |
| xxix| 5  | \{(2, 2), (3, 4)\}       | 5                       |
| xxx | 5  | \{(1, 4), (1, 4), (3, 4)\}| 0                       |
| xxxi| 5  | \{(1, 4), (2, 1), (2, 1)\}| 0                       |
| xxxii| 5  | \{(1, 4), (2, 1), (2, 2)\}| 0                       |
| xxxiii| 5  | \{(1, 4), (2, 2), (2, 2)\}| 0                       |
| xxxiv| 5  | \{(1, 4), (1, 4), (1, 4), (2, 1)\}| 0 |
| xxxv| 5  | \{(1, 4), (1, 4), (1, 4), (2, 2)\}| 0 |
| xxxvi| 5  | \{(1, 4), (1, 4), (1, 4), (1, 4)\}| 0 |
Figure 1. Number of different Witt classes in 5000 random cubic fields.

Figure 2. Number of different Witt classes in 5000 random quartic fields.
Figure 3. Number of different Witt classes in 5000 random quintic fields.
Appendix A. Pseudo-codes of presented algorithms

For convenience of the readers, especially those wishing to implement the presented procedures, this section gathers the pseudo-codes of all described algorithms.

**Input:** $a$ and element of a number field $K$

**Output:**

- true if $a$ is a square in $K$
- false otherwise

```plaintext
// Use [15, §5.4]
if is_irreducible($x^2 - a$) then
  return true;
return false;
```

**Listing 1:** is_square

**Input:**
- $a$: a non-zero element $K$
- $p$: an odd prime of a number field $K$

**Output:**

- true if $a$ is a square in $K_p$
- false otherwise

```plaintext
if ord_p $a$ ≡ 1 (mod 2) then
  return false;
// Use [1, Algorithm 4.8.17]
π ← uniformizer(p);
a' ← $a$ · $π^{-\text{ord}_p a}$;
if is_square(a', $K/p$) then
  return true;
else
  return false;
```

**Listing 2:** is_nondyadic_square

**Input:**
- $a$: a non-zero element $K$
- $d$: an even prime of a number field $K$

**Output:**

- true if $a$ is a square in $K_d$
- false otherwise

```plaintext
// Use Algorithm 1
if is_square(a; $K$) then
  return true;
L ← $K[x]/(x^2 - a)$;
// Use [2, Algorithm 2.4.13]
$L$ ← prime_decomposition($d; L$);
if #L = 2 then
  return true;
else
  return false;
```

**Listing 3:** is_dyadic_square
\[
\varphi = \langle a_1, \ldots, a_d \rangle \quad \text{a non-degenerate quadratic form over } K,
\]

**Input:**
- \( a_i \) are \( p \)-adic units
- \( p \) is an odd prime of a number field \( K \)

**Output:**
- `true` if \( \varphi \otimes K/p \) is isotropic
- `false` otherwise

```
switch \text{dim}(\varphi) \text{ do }
\text{case 1}\n\quad \text{return false;}
\text{case 2}\n\quad \text{if is\_square(disc(\varphi), K/p) then}
\quad \text{return true;}
\quad \text{else}
\quad \text{return false;}
\text{otherwise return true;}
```

Listing 4: `is\_residually\_isotropic`

**Input:**
- \( \varphi = \langle a_1, \ldots, a_d \rangle \) a non-degenerate quadratic form over \( K \)
- \( p \) an odd prime of a number field \( K \)

**Output:**
- `true` if \( \varphi \otimes K_p \) is isotropic
- `false` otherwise

```
if \text{dim}(\varphi) = 1 \text{ then}
\quad \text{return false;}
if \text{dim}(\varphi) \geq 5 \text{ then}
\quad \text{return true;}
// Use [Algorithm 4.8.17]
\pi \leftarrow \text{uniformizer}(p);
\varphi_0 \leftarrow \{ a_i \cdot \pi^{-\text{ord}_p a_i} \mid \text{ord}_p a_i \equiv 0 \text{ (mod 2)} \};
\varphi_1 \leftarrow \{ a_i \cdot \pi^{-\text{ord}_p a_i} \mid \text{ord}_p a_i \equiv 1 \text{ (mod 2)} \};
// Use Algorithm 4
A \leftarrow \text{is\_residually\_isotropic(\varphi_0; p)};
B \leftarrow \text{is\_residually\_isotropic(\varphi_1; p)};
\text{return } A \lor B;
```

Listing 5: `is\_locally\_isotropic\_non\_dyadic`
Input: \( \varphi = \langle a_1, \ldots, a_d \rangle \) a non-degenerate quadratic form over \( K \)
\( \mathfrak{p} \) an even prime of a number field \( K \)

Output: \[
\begin{align*}
\text{true} & \quad \text{if } \varphi \otimes K_{\mathfrak{p}} \text{ is isotropic} \\
\text{false} & \quad \text{otherwise}
\end{align*}
\]

switch \( \dim(\varphi) \) do

\begin{enumerate}
\item case 1
  \begin{align*}
  & \text{return false;}
  \end{align*}
\item case 2
  \begin{align*}
  & \text{// Use Algorithm 3}
  \\
  & \text{if is\_dyadic\_square}(\langle -a_1a_2, \mathfrak{p} \rangle) \text{ then}
  \\
  & \quad \text{return true;}
  \text{else}
  \\
  & \quad \text{return false;}
  \end{align*}
\item case 3
  \begin{align*}
  & \text{// Use Eq. (1) and [14, Algorithm 6.6]}
  \\
  & s \leftarrow \prod_{i<j} \text{hilbert\_symbol}(a_i, a_j; \mathfrak{p});
  \\
  & \text{// Use [7, Proposition V.3.22]}
  \\
  & \text{if } s = \text{hilbert\_symbol}(\langle -1, -\det(\varphi); \mathfrak{p} \rangle) \text{ then}
  \\
  & \quad \text{return true;}
  \text{else}
  \\
  & \quad \text{return false;}
  \end{align*}
\item case 4
  \begin{align*}
  & \text{// Use Algorithm 3}
  \\
  & \text{if is\_dyadic\_square}(\langle \det(\varphi), \mathfrak{p} \rangle) \text{ then}
  \\
  & \quad \text{// Use Eq. (1) and [14, Algorithm 6.6]}
  \\
  & s \leftarrow \prod_{i<j} \text{hilbert\_symbol}(a_i, a_j; \mathfrak{p});
  \\
  & \text{// Use [7, Proposition V.3.23]}
  \\
  & \text{if } s = \text{hilbert\_symbol}(\langle -1, -1; \mathfrak{p} \rangle) \text{ then}
  \\
  & \quad \text{return true;}
  \text{else}
  \\
  & \quad \text{return false;}
  \text{else}
  \\
  & \quad \text{return true;}
  \end{align*}
\item otherwise \text{return true;}
\end{enumerate}

Listing 6: \text{is\_locally\_isotropic\_dyadic}
Input: $\varphi = \langle a_1, \ldots, a_d \rangle$ a non-degenerate quadratic form over $K$

Output: \[
\begin{cases}
\text{true} & \text{if } \varphi \text{ is isotropic} \\
\text{false} & \text{otherwise}
\end{cases}
\]

if $\dim(\varphi) = 1$
\[\text{return false;}
\]
if $\dim(\varphi) = 2$
\[// \text{Use Algorithm 1}
\]
\[\text{if is_square}(-a_1a_2; K) \text{ then}
\]
\[\text{return true;}
\]
\[\text{return false;}
\]
// Check all real places
$L \leftarrow \text{real_embeddings}(K)$;
for $\rho \in L$ do
\[s \leftarrow \text{sgn } \rho(\varphi);
\]
\[\text{if } |s| = \dim \varphi \text{ then}
\]
\[\text{return false;}
\]
// Check all dyadic primes
$L \leftarrow \text{prime_decomposition}(2, K)$;
for $\vartheta \in L$ do
\[// \text{Use Algorithm 6}
\]
\[\text{if not is_locally_isotropic_dyadic}(\varphi; \vartheta) \text{ then}
\]
\[\text{return false;}
\]
// Find all odd primes that matter
$L \leftarrow \emptyset$;
for $1 \leq i \leq \dim \varphi$ do
\[L \leftarrow L \cup \text{prime_decomposition}(a_i; K);
\]
$L \leftarrow L \setminus \text{prime_decomposition}(2, K)$;
// Check the odd primes
for $p \in L$ do
\[// \text{Use Algorithm 5}
\]
\[\text{if not is_locally_isotropic_non-dyadic}(\varphi; p) \text{ then}
\]
\[\text{return false;}
\]
return true;

Listing 7: is_globally_isotropic
Input: $\varphi = \langle a_1, \ldots, a_d \rangle$ a non-degenerate quadratic form over $K$

Output: $\begin{cases} \text{true} & \text{if } \varphi \text{ is hyperbolic} \\ \text{false} & \text{otherwise} \end{cases}$

if $\dim \varphi \equiv 1 \pmod{2}$ then
  return false;
// Use Algorithm 1
if not is_square($\text{disc } \varphi; K$) then
  return false;
// Check all real places
$L \leftarrow \text{real_embeddings}(K)$;
for $\rho \in L$ do
  $s \leftarrow \text{sgn } \rho(\varphi)$;
  if $s \neq 0$ then
    return false;
// Find all odd primes that matter
$L \leftarrow \emptyset$;
for $1 \leq i \leq \dim \varphi$ do
  $L \leftarrow L \cup \text{prime_decomposition}(a_i; K)$;
$L \leftarrow L \setminus \text{prime_decomposition}(2, K)$;
// Check the odd primes
for $p \in L$ do
  // Use [1], Algorithm 4.8.17
  $\pi \leftarrow \text{uniformizer}(p)$;
  $\varphi_1 \leftarrow \{ a_i \cdot \pi^{-\text{ord}_p a_i} \mid \text{ord}_p a_i \equiv 1 \pmod{2} \}$;
  if $\dim \varphi_1 \equiv 1 \pmod{2}$ or not is_square($\text{disc } \varphi_1; K/p$) then
    return false;
// Check all dyadic primes
$L \leftarrow \text{prime_decomposition}(2, K)$;
for $d \in L$ do
  // Use Eq. [1] and [14], Algorithm 6.6
  $s \leftarrow \prod_{i<j} \text{hilbert_symbol}(a_i, a_j; d)$;
  if $s \neq 1$ then
    return false;
return true;

Listing 8: $\text{is\_globally\_hyperbolic}$
Input: $\varphi = \langle a_1, \ldots, a_d \rangle$ a non-degenerate quadratic form over $K$
Output: $d$ the dimension of the anisotropic part of $\varphi$

// Find the signatures
$L \leftarrow \text{real embeddings}(K)$;
$N \leftarrow \max \{ \text{sgn}\, \rho(\varphi) \mid \rho \in L \}$;
if $N \geq 3$ then
    _return $N$;
if $N = 0 \lor N = 2$ then
    if $N = 0$ then
        // Use Algorithm 8
        if is_globally_hyperbolic($\varphi$) then
            _return 0;
        // Find all primes that matter
        $L \leftarrow \emptyset$;
        for $1 \leq i \leq \dim \varphi$ do
            $L \leftarrow L \cup \text{prime decomposition}(a_i; K)$;
        $L \leftarrow L \cup \text{prime decomposition}(2, K)$;
        // Check these primes
        for $p \in L$ do
            // Use Eq. (1) and [14, Algorithm 6.6]
            $s \leftarrow \prod_{i < j} \text{hilbert symbol}(a_i, a_j; p)$;
            if $s \neq 1$ then
                _return 4;
        if $N = 1$ then
            $\psi \leftarrow \varphi \perp \langle \det \varphi \rangle$;
        _return ($\dim \text{anisotropic part}(\psi) - 1$);
    // Find the signatures
    $L \leftarrow \text{real embeddings}(K)$;
    $N \leftarrow \max \{ \text{sgn}\, \rho(\varphi) \mid \rho \in L \}$;
    if $N \geq 3$ then
        _return $N$;
    if $N = 0 \lor N = 2$ then
        if $N = 0$ then
            // Use Algorithm 8
            if is_globally_hyperbolic($\varphi$) then
                _return 0;
            // Find all primes that matter
            $L \leftarrow \emptyset$;
            for $1 \leq i \leq \dim \varphi$ do
                $L \leftarrow L \cup \text{prime decomposition}(a_i; K)$;
            $L \leftarrow L \cup \text{prime decomposition}(2, K)$;
            // Check these primes
            for $p \in L$ do
                // Use Eq. (1) and [14, Algorithm 6.6]
                $s \leftarrow \prod_{i < j} \text{hilbert symbol}(a_i, a_j; p)$;
                if $s \neq 1$ then
                    _return 4;
        if $N = 1$ then
            $\psi \leftarrow \varphi \perp \langle \det \varphi \rangle$;
        _return ($\dim \text{anisotropic part}(\psi) - 1$);
    return 2;

Listing 9: dimension of anisotropic part

Input: $f \in \mathbb{Q}[x]$ the minimal polynomial of the generator $\vartheta$
Output: $s = s(K)$ the level of the number field $K = \mathbb{Q}(\vartheta)$

// Use Sturm sequence to compute the number of real roots of $f$:
$r \leftarrow \text{count real roots}(f)$;
if $r \neq 0$ then
    _return $\infty$;
if is_square($-1; K$) then
    _return 1;
// Use [1], §6.2.5 to decompose 2 in $\mathcal{O}_K$,
// $\mathcal{L}$ consists of triples $(\delta_i, e_i, f_i)$.
$\mathcal{L} \leftarrow \text{prime decomposition}(2, K)$;
// Use [7], Proposition XI.2.11
for $(\delta, e, f) \in \mathcal{L}$ do
    if $(e \cdot f) \equiv 1 \pmod{2}$ then
        _return 4;
return 2;

Listing 10: level
Input:
\[
\begin{aligned}
\{ & 0 \text{ an even prime of } K \\
& e \text{ the ramification index of } p \\
& f \text{ the inertia degree of } p \\
\end{aligned}
\]

Output: \( s = s(K_\mathfrak{d}) \) level of the completion of \( K \) at \( \mathfrak{d} \)

// Use Algorithm 1
if not is_square(\(-1, K\)) then
  return 1;
if \( e \equiv 1 \pmod{2} \) then
  if \( f \equiv 1 \pmod{2} \) then
    return 4;
  else
    return 2;
else
  // Use Algorithm 3
  if is_dyadic_square(\(-1, \mathfrak{d}\)) then
    return 1;
  else
    return 2;

Listing 11: dyadic_level

Input: \( f \in \mathbb{Q}[x] \) the minimal polynomial of the generator of a number field \( K \)
Output: The complete set of Witt equivalence invariants of \( K \) in the form
\[
(d, r, k, \{(d_1, s_1), \ldots, (d_k, s_k)\})
\]

// The degree is just \((K : \mathbb{Q}) = \deg f\).
\( d \leftarrow \deg f; \)
// Use Sturm sequence to compute the number of real roots of \( f \).
\( r \leftarrow \text{count_real_roots}(f); \)
// Use Algorithm 12 to compute the level of \( K \).
\( s \leftarrow \text{level}(f); \)
// Use [II, §6.2.5] to decompose 2 in the maximal order of \( K \).
\( \mathcal{L} \leftarrow \text{prime_decomposition}(2, K); \)
// The number of dyadic primes:
\( k \leftarrow \# \mathcal{L}; \)
\( S \leftarrow \{\}; \)
for \( 1 \leq i \leq k \) do
  // The local degree:
  \( d_i \leftarrow e_i \cdot f_i; \)
  // Use Algorithm 13 to compute the local level.
  \( s_i \leftarrow \text{dyadic_level}(K, p_i, e_i, f_i); \)
  \( S \leftarrow S \cup \{(d_i, s_i)\}; \)
// Sort the dyadic degrees and levels lexicographically.
sort \( S; \)
return \((d, r, s, k, S)\);

Listing 12: Witt_equivalence_invariants