Using Gröbner bases for determining the equivalence of linear codes and solving the decoding problem ⋆

M. Borges-Quintana [a] M.A. Borges-Trenard [a,1]
E. Martínez-Moro [b,2] F. Winkler [c]

[a] *Departamento de Matemática, Facultad de Matemática y Computación Universidad de Oriente, Santiago de Cuba, Cuba*

[b] *Departamento de Matemática Aplicada, Universidad de Valladolid Valladolid, Castilla, Spain*

[c] *Research Institute for Symbolic Computation, J.Kepler Universitaet Linz Linz, Austria*

Abstract

Two linear codes are permutation-equivalent if they are equal up to a permutation on the codeword coordinates. We present an invariant for the class of linear codes, i.e. a mapping so that two codes lead to the same value if and only if they are permutation equivalent. The properties of that invariant remember the linear algebra techniques into the framework of the Gröbner bases tools. An application of this mapping to the decoding problem is presented, with particular emphasis on the binary case.

Key words: Gröbner basis, code equivalence.

⋆ A preliminary version of this paper was presented at the conference Applications of Computer Algebra 2003.

Email addresses: `mijail@mbq.uo.edu.cu` (M. Borges-Quintana), `mborges@mabt.uo.edu.cu` (M.A. Borges-Trenard), `edgar.martinez@ieee.org` (E. Martínez-Moro), `Franz.Winkler@jku.at` (F. Winkler).

[1] MECD grant SAB 345-1234 Department of Álgebra, Geometría y Topología, Universidad de Valladolid, Spain.

[2] Supported by MCyT under project BFM2001-2251 and Junta de Castilla y León under project VA020/02.

[3] Supported by the Austrian Fonds zur Förderung der wiss. Forschungs (FWF) under project F1304 of the SFB 013.

Introduction

In this paper, we investigate a new approach to the problem of knowing whether two given linear codes are permutation-equivalent. In (E. Petrank, R. M. Roth (1997)) the authors proved that the *Code Equivalence Problem* is not an NP-complete problem, but it is at least as hard as the *Graph Isomorphism Problem*. Some partial answers have been given by using the weight enumerator, but computing the weight enumerator of a code becomes intractable when its size grows. Also N. Sendrier proposed a procedure for finding a fully discriminant signature for most linear codes (see N. Sendrier (2000)).

On the other hand, the connection between Gröbner bases and linear algebra comes from the very beginning, i.e. from Buchberger PhD thesis. In (B. Buchberger, H.M. Möller (1982)) it was the first time when an algorithm was formulated in order to compute a Gröbner basis for an ideal given in a different way than a set of polynomials [4]. In (Faugère et al (1993); M. G. Marinari et al. (1993)) these techniques were generalized to different settings (change of orderings, ideal defined by functionals). In (M.A. Borges et al. (2000)) the authors formulated a pattern algorithm, which included the predecessors and generalized the techniques to free associative algebras. In that paper, the algorithm introduced for monoid and groups algebras is particularly interesting. In (M. Borges et al. (2001, 2000)) the algorithm for monoid and groups algebras was specialized for the case of algebras associated to linear codes. For a concise introduction to the theory of Gröbner bases we refer to F.Winkler (1996).

All the papers and algorithms mentioned above have in common that the Gröbner bases tools make use mainly of linear algebra. In this paper we go further and show how these Gröbner basis tools can be used to determine if two linear codes are equivalent or not. The structure of binary codes permits us to use the notion of "reduced bases", which will allow to solve the problems of the equivalence and decoding for binary linear codes. The structure of this work is as follows: First in Section 1, we will introduce some basic notions on linear codes and show the relationship between them and some reduction techniques of Gröbner bases. In Section 2 the notion of equivalence of codes is given. Section 3 shows how to build the invariant matphi (ϕ) for equivalent codes. The construction of the reduced bases is given in Section 4, and the specific case of binary codes results almost like a straight-forward application of reduced Gröbner bases, although the term ordering used is not admissible. Section 5 takes as a starting point the results in Section 3, and adapts the

[4] The algorithm computed a Gröbner basis from a set of points which are the zeros of the ideal.

situation of deciding equivalence of linear codes using the associated reduced basis.

A reduced basis is in general considerably smaller than the invariant matphi, it is quite similar as comparing the border basis and the reduced Gröbner basis; therefore, we take the advantage of applying the reduced basis for the case of binary codes, for which some strategies are given and also, some examples with different situations that can come out. Section 6 corresponds to an application of the reduced basis in decoding for an arbitrary binary code. An algorithm similar to the computation of the canonical form using Gröbner basis is getting as the solution to decode. Some worked examples with GAP (GAP v4.3 (2002)) are shown in Section 7. Finally, in the appendix, we give various additional information, such as definitions of codes used in the examples, specific details of proofs, reduced basis for some codes, etc. All these items are quoted in different sections of the paper.

1 Linear codes and monoids

In this section we will show some basic notions of linear codes as well as we introduce the relationship between linear codes and some FGLM techniques. Most of the material can be found in a detailed way in (M. Borges et al. (2001)). For an acount on FGLM techniques the reader can consult (M.A. Borges, M. Borges (1998); M.A. Borges et al. (2000); Faugère et al (1993)). Let $\mathbb{F}_q$ a finite field with $q = p^m$ elements (p a prime number). A linear code $\mathcal{C}$ of dimension k and lengh n is the image of a linear mapping $L : \mathbb{F}_q^k \to \mathbb{F}_q^n$, where $k \leq n$, i.e. $\mathcal{C} = L(\mathbb{F}_q^k)$. There exists a $n \times (n-k)$ matrix H such that $c \cdot H = 0$ for all $c \in \mathcal{C}$, that is called *parity check matrix*. The weight of a codeword is its Hamming distance to the word 0, and the minimun distance d of a code is the minimun weight among all the non-zero words. The error correcting capacity of a code is $t = \left[\frac{d-1}{2}\right]$. Let $B(\mathcal{C}, t) = \{y \in \mathbb{F}_q^n \mid \exists c \in \mathcal{C} \text{ s.t. } d(c, y) \leq t\}$, it is well known that the equation:

$$eH = yH$$

has a unique solution e with $weight(e) \leq t$ for $y \in B(\mathcal{C}, t)$.

From now on we will consider the elements of $\mathbb{F}_q$ represented as $a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1}$, where α is a root of an irreducible polynomial of degree m over $\mathbb{F}_p$ and $a_i \in \mathbb{F}_p$ for all i. Let us consider the free commutative monoid $[X]$ generated by the nm variables $X := \{x_{11}, \ldots, x_{1m}, \ldots, x_{n1}, \ldots, x_{nm}\}$. We have the following map from X to to $\mathbb{F}_q^n$:

$$\begin{aligned}\varphi : X \rightarrow & \mathbb{F}_q^n \\ x_{ij} \mapsto & (0, \ldots, 0, \underbrace{\alpha^{j-1}}_{i}, 0, \ldots, 0)\end{aligned} \tag{1}$$

The map φ can be extended to a morphism from $[X]$ onto $\mathbb{F}_q^n$, where:

$$\varphi\Big(\prod_{i=1}^{n}\prod_{j=1}^{m} x_{ij}^{\beta_{ij}}\Big) = \Big(\Big(\sum_{j=1}^{m}\beta_{1j}\alpha^{j-1}\Big) mod\, p, \ldots, \Big(\sum_{j=1}^{m}\beta_{nj}\alpha^{j-1}\Big) mod\, p\Big)$$

For the sake of simplicity, from now on we will consider the variables x_{ij} as the set of nm variables x_k, where $k = (i-1)m + j$.

A linear code $\mathcal{C}$ defines an equivalence relation $R_{\mathcal{C}}$ in $\mathbb{F}_q^n$ by $(x, y) \in R_{\mathcal{C}} \Leftrightarrow x - y \in \mathcal{C}$. If we define $\xi(u) := \varphi(u)H$, where $u \in [X]$, the above congruence can be translated to $[X]$ by the morphims φ as $u \cong_{\mathcal{C}} w \Leftrightarrow (\varphi(u), \varphi(w)) \in R_{\mathcal{C}} \Leftrightarrow \xi(u) = \xi(w)$. The morphism ξ represents the transition of the syndromes from $\mathbb{F}_q^n$ to $[X]$, so that, $\xi(w)$ is the syndrome of w, which is equal to the syndrome of $\varphi(w)$.

Let $w \in [X]$. We will denote by $\mathrm{Supp}(w)$ the set of variables that divide w and by $\mathrm{Ind}(w)$ the set of indices associated to w, i.e.

$$\mathrm{Ind}(w) := \{i \in \{1, \ldots, n\} \mid \exists j \in \{1, \ldots, m\} \text{ such that } x_{ij} \in \mathrm{Supp}(w)\}.$$

Definition 1 (The error vector term ordering). We say that u is less than w w.r.t. the error term ordering, and denote it by $u <_e w$, if one of the following conditions holds:

(1) $|\mathrm{Ind}(u)| < |\mathrm{Ind}(w)|$.
(2) $|\mathrm{Ind}(u)| = |\mathrm{Ind}(w)|$ and $u \prec w$, where $\prec$ denotes an arbitrary but fixed admissible term ordering on $[X]$.

Remark 2. It is easy to prove that $<_e$ is a total ordering on $[X]$. Although it is not admissible (see (2) and (3) in the Example 26), it has the following properties[5] :

(1) $1 <_e u$ for all $u \neq 1$.
(2) $u <_e u \cdot x$, for all $x \in X$.

Definition 3. The word $w := \prod_{i=1}^{n}\prod_{j=1}^{m} x_{ij}^{\beta_{ij}}$ is said to be **standard** if $\beta_{ij} < p$, for every $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. Given $y \in \mathbb{F}_q^n$ we say that w is the **standard representation** of y if $\varphi(w) = y$ and w is standard .

[5] See M. Borges et al. (2001) for more details.

Note that φ is surjective but not injective, but considering the standard words in $[X]$, then the correspondence with $\mathbb{F}_q^n$ is biyective.

Now we will define axiomatically two objetcs, a set N of "canonical forms" and a function ϕ that will play a central role in the following section. Both objects are basic in some Gröbner basis techniques (see for example M. Borges et al. (2001); M.A. Borges et al. (2000); Faugère et al (1993)).

Definition 4 (Canonical forms). We define a set of canonical forms $N \subset [X]$ by the following properties:

(1) $1 \in N \subseteq [X]$.
(2) $|N| = q^{n-k}$.
(3) Two different words of N determine different cosets modulo $\equiv_{\mathcal{C}}$.
(4) For all $w \in N \setminus \{1\}$ there exists $x \in X$ such that $w = w'x$ and $w' \in N$.

The second property means that N has as many elements as syndromes has the code, and the third one that different words in N correspond to different syndromes.

Definition 5 (matphi). Let matphi be a function ϕ from $N \times X$ onto N, such that:

$$\forall x \in X \ \forall w \in [X], \ \xi(\phi(w,x)) = \xi(wx).$$

Remark 6. Note that matphi states that the representative element of wx, in the set N of canonical forms, is $\phi(w,x)$. This function can be extended to $[X] \times X$ by using the same definition.

The following algorithm for building N and ϕ can be found in (M. Borges et al. (2001)). There are three funtions needed to understand the algorithm:

- InsertNext$[t, List]$ inserts the products xt, where $x \in X$, in $List$ and keeps the encreasing order of $List$ w.r.t. the order $<_e$.
- NextTerm$[List]$ returns the first element from $List$ and deletes it from that set.
- Member$[obj, G]$ returns the position j of obj in G, if $obj \in G$, and false otherwise.

Algorithm 1.
Input: p, n, m, H parameters for a given linear code.
Output: N, ϕ.

1. $List := \{1\}, N := \emptyset, r := 0;$
2. While $List \neq \emptyset$ do
3. $\quad w :=$ NextTerm$[List]$;

4. $\quad v' := \xi(w)$;
5. $\quad j :=$ Member$[v', \{v_1, \ldots, v_r\}]$;
6. $\quad$ If $j \neq$ false
7. $\quad$ them for each k such that $w = ux_k$ with $u \in N$ do
8. $\qquad \phi(u, x_k) := w_j$;
9. $\quad$ else $r := r + 1$;
10. $\qquad v_r := v'$;
11. $\qquad w_r := w, \quad N := N \cup \{w_r\}$;
11.1 $\qquad$ $List :=$ InsertNext$[w_r, List]$;
11.2 $\qquad$ For k such that $w = ux_k$ with $u \in N$ do
11.3 $\qquad\quad \phi(u, x_k) := w$;
12. Return $[N, \phi]$.

Theorem 7 (Correctness of the algorithm). *Algorithm 1 returns N and ϕ that fullfill Definitions 4 and 5.*

Proof. By construction we see that $|N| \leq q^{n-k}$; moreover, each time a new word is included in N, mn products are included in $List$. Consequently, the procedure ends because $|List| \leq nmq^{n-k}$.

Lets us see N has the properties of Definition 4:

(1) $1 \in N \subseteq [X]$ is a direct consequence of the Steps 1 and 11.
(2) Consider the following recursive funtion:

$$\begin{aligned} cf : [X] &\longrightarrow N \\ 1 &\mapsto 1 \\ w &\mapsto \phi(cf(u), x_k) \end{aligned} \tag{2}$$

where $w = ux_k$, and $u \in [x_1, \ldots, x_k]$. It is easy to see that $cf(w)$ is an element of N with the same syndrome as w because if $w = x_{i_1}x_{i_2}\ldots x_{i_k}$ where $x_{i_j} <_e x_{i_{j+1}}$:

$$\begin{aligned} \xi\left(cf(w)\right) &= \xi\left(\phi\left(cf(x_{i_1}x_{i_2}\ldots x_{i_{k-1}}), x_{i_k}\right)\right) \\ &= \xi\left(cf(x_{i_1}x_{i_2}\ldots x_{i_{k-1}})x_{i_k}\right). \end{aligned} \tag{3}$$

The second equality in 3 holds by the definition of ϕ. Thus, by recursion we get:

$$\xi\left(cf(w)\right) = \xi\left(cf(1)x_{i_1}x_{i_2}\ldots x_{i_k}\right) = \xi\left(x_{i_1}x_{i_2}\ldots x_{i_k}\right).$$

On the other hand, $cf(w)$ is reachable by the above procedure, for every $w \in [X]$ (see Steps 10,11, 11.1); hence, $|N| = q^{n-k}$.

(3) Step 4 in the algorithm computes the coset associated to w, Step 5 checks whether that coset has already been considered, it guarantees the condition (3) of Definition 4.
(4) The last property of N it is a consequence of Steps 11 and 11.1.

Now let us see that the function matphi computed by the algorithm fullfils the property in Definition 5. Let $u \in N$ and $x \in X$. There are two choices in Algorithm 1 for the pair (u, x):

- If $\xi(ux)$ has already been considered (Step 6), then $\phi(u, x)$ is defined in Step 8 and holds $\xi(\phi(u, x)) = \xi(w_j) = \xi(ux)$.
- Otherwise $\phi(u, x) = ux$, and this is ensured by Step 11.3.

□

Remark 8. The set N computed by Algorithm 1 has the property that the representative elements in N of the syndromes corresponding to vectors in $B(C, t)$, are the smallest words in $[X]$ with respect to $<_e$, that is, the image by φ of the words that are the error vectors (see Lemma 1 in M. Borges et al. (2001, 2000)).

From now on we will be intrested in such sets of canonical forms with this additional property.

Remark 9 (Property 5 of a set of canonical forms). For all $w \in N$ such that $\varphi(w) \in B(C, t)$, then $weight(\varphi(w)) \leq t$. That is, w is the standard representation of the error vector $\varphi(w)$.

2 Equivalence of codes

Definition 10 (Permutation Equivalent Codes). Let $\mathcal{C}$ be a code of length n over $\mathbb{F}_q$. Let $\sigma \in S_n$, we define:

$$\sigma(\mathcal{C}) = \{(x_{\sigma^{-1}(i)})_{i=1}^n \mid (x_i)_{i=1}^n \in \mathcal{C}\}.$$

We say that $\mathcal{C}$ and $\sigma(\mathcal{C})$ are permutation-equivalent or σ-equivalent and we denote it by $\mathcal{C} \sim \sigma(\mathcal{C})$.

Throughout the paper the words "equivalent codes" means "permutation equivalent codes" unless otherwise specified. Let $\mathcal{L}_n$ denote the set of all codes of lenght n and $\mathcal{L} = \cup_{n>0}\mathcal{L}_n$.

Definition 11. An invariant over a set E is a mapping $\mathcal{L} \longrightarrow E$ such that any two permutation equivalent codes take the same value.

There are many of such invariants, for instance, the length, the minimun Hamming weight, the weight enumerator, etc.

3 The invariant matphi

In this section we will show that the mapping ϕ defined in the section above is an invariant for the class of linear codes. We begin with the following definition of equivalence between two functions ϕ.

Definition 12. S_n acts on $[X]$ as follows:

$$\forall(\sigma \in S_n)\ \ \sigma\left(\prod_i \prod_j x_{ij}^{\beta_{ij}}\right) = \prod_{\sigma^{-1}(i)} \prod_j x_{\sigma^{-1}(i)j}^{\beta_{\sigma^{-1}(i)j}}.$$

From the definition above we have:

$$\varphi(\sigma(w)) = \sigma(\varphi(w)). \tag{4}$$

The following scheme shows the relationship between the objects in $\mathbb{F}_q^n$ and $[X]$, and the images by φ and σ expressed by the former equality.

$$\begin{array}{ccc} \mathcal{C} & & [X] \\ c = \varphi(w) & \longleftarrow & w \\ \downarrow & \# & \downarrow \\ \sigma(c) & \longleftarrow & \sigma(w) \end{array}$$

It follows straightforward that:

Lemma 13. *The action of Definition 12 preserves the addition on $\mathbb{F}_q^n$, consequently, the multiplication on $[X]$, i.e.:*

$$\forall(\sigma \in S_n, u, v \in \mathbb{F}_q^n\ (u, v \in [X]))\ \sigma(v+w) = \sigma(v)+\sigma(w)\ (\sigma(vw) = \sigma(v)\sigma(w)).$$

Moreover:

Lemma 14.

1. *If N if a set of representantives of the cosets of $\mathcal{C}$ then $N^\star = \sigma(N)$ is a set of representatives of the cosets of $\mathcal{C}^\star = \sigma(\mathcal{C})$.*

2. *If* $v \in N$ *and* $weight(v) \leq t$ *then* $\sigma(v) \in N^\star = \sigma(N)$ *and* $weight(\sigma(v)) \leq t$ [6].
3. *Let* $x_i \in X$, *then* $\sigma(x_i) \in X$.

Definition 15. Let $\phi : N \times X \longrightarrow N$ be a matphi function, $\sigma \in S_n$. Let $N^\star = \sigma(N)$ and $\phi^\star : N^\star \times X \longrightarrow N^\star$ the corresponding matphi for the code $\mathcal{C}^\star = \sigma(C)$. Then

$$\phi \sim \phi^\star \Longleftrightarrow \forall (v \in N,\, i \in [1, mn]) \quad \phi^\star(\sigma(v),\, \sigma(x_i)) = \sigma(\phi(v,\, x_i)).$$

Theorem 16. $\mathcal{C} \sim C^\star \Longleftrightarrow \phi \sim \phi^\star$.

Proof. Suppose $\mathcal{C} \sim \mathcal{C}^\star$ and $\sigma \in S_n$ such that $\mathcal{C}^\star = \sigma(\mathcal{C})$. Let now w be equal to $\phi(v,\, x_i)$, hence, $\varphi(vx_i) = c + \varphi(w)$, where $c \in \mathcal{C}$ (see Definition 5); consequently, $\sigma(\varphi(vx_i)) = \sigma(c) + \sigma(\varphi(w))$ (Lemma 13), thus $\varphi(\sigma(v)\sigma(x_i)) = \sigma(c) + \varphi(\sigma(w))$ (Equality 4 and Lemma 13).
But $\sigma(w) \in N^\star$ and $\sigma(c) \in \mathcal{C}^\star$ (Lemma 14); therefore, $\phi^\star(\sigma(v),\, \sigma(x_i)) = \sigma(w)$ [7], which means that $\phi \sim \phi^\star$.
Conversely, let $\phi \sim \phi^\star$, $\sigma \in S_n$ such that $\phi^\star(\sigma(v),\, \sigma(x_i)) = \sigma(\phi(v,\, x_i))$ and $c \in \mathcal{C}$ such that : $c = \varphi(w)$, $w = x_{i_1} \ldots x_{i_l}$.

Therefore:

$$\sigma(c) = \varphi(\sigma(x_{i_1}) \ldots \sigma(x_{i_l})).$$

Set $w_0 := 1$, $w_0^\star := 1$, Due to $\phi \sim \phi^\star$:

$$\text{If } \phi(w_{k-1}, x_{i_k}) = w_k \text{ then } \phi^\star(\sigma(w_{k-1}), \sigma(x_{i_k})) = \sigma(w_k),\; k = 1, \ldots, l.$$

Since $c \in \mathcal{C}$ it follows $w_l = 1$ [8]; by the equality above:

$$\xi(\sigma(c)) = \xi(w_l^\star), \text{ and } w_l^\star = \sigma(w_l) = \sigma(1) = 1,$$

i.e., $c^\star = \sigma(c) \in \mathcal{C}^\star$. $\square$

Remark 17. Consequently, the function ϕ is an invariant in the class of linear codes.

Example 18. Consider the code $\mathcal{C}$ over $\mathbb{F}_2^6$ generated by the matrix:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

---

6 Taking into account that $weight(v) = weight(\sigma(v))$.
7 $\phi(w, x)$ is the only element in N such that $\xi(\phi(w, x)) = \xi(wx)$ (see Remark 6).
8 Remark: $c \in \mathcal{C}$ iff $w_l = 1$ (see M. Borges et al. (2001)).

The number of variables is 6, $\prec$ is set to be the plex. order with $x_i \prec x_{i+1}$. We can compute $N = \{1, x_1, \ldots, x_6, x_2x_3\}$ and ϕ correspond to:

$$
\begin{array}{l}
[\,[[0,0,0,0,0,0],1,[2,3,4,5,6,7]],[[1,0,0,0,0,0],1,[1,6,5,4,3,8]]\\
[[0,1,0,0,0,0],1,[6,1,8,7,2,5]],[[0,0,1,0,0,0],1,[5,8,1,2,7,6]]\\
[[0,0,0,1,0,0],1,[4,7,2,1,8,3]],[[0,0,0,0,1,0],1,[3,2,7,8,1,4]]\\
[[0,0,0,0,0,1],1,[8,5,6,3,4,1]],[[0,1,1,0,0,0],0,[7,4,3,6,5,2]]\,]
\end{array}
$$

This representation corresponds to the following: In each triple the first entry correspond to the elements $\varphi(w)$ where $w \in N$ ($w = N[i]$), the second one is 1 if $\varphi(w) \in B(\mathcal{C}, t)$ or 0 otherwise, and the third component points to the values $\phi(w, x_j)$ for $j = 1, \ldots, nm$ ($\phi(w, x_j) = N[\phi[i][3][j]]$).

Consider now the permutation $\sigma = (5, 6) \in S_6$ and the code $\mathcal{C}^\star = \sigma(\mathcal{C})$. Now for $\mathcal{C}^\star$ we got $N^\star = \{1, x_1, \ldots, x_6, x_1x_5\}$ and $\phi^\star$ is:

$$
\begin{array}{l}
[\,[0,0,0,0,0,0],1,[2,3,4,5,6,7]],[[1,0,0,0,0,0],1,[1,7,5,4,8,3]]\\
[[0,1,0,0,0,0],1,[7,1,8,6,5,2]],[[0,0,1,0,0,0],1,[5,8,1,2,7,6]]\\
[[0,0,0,1,0,0],1,[4,6,2,1,3,8]],[[0,0,0,0,1,0],1,[8,5,7,3,1,4]]\\
[[0,0,0,0,0,1],1,[3,2,6,8,4,1]],[[1,0,0,0,1,0],0,[6,4,3,7,2,5]]
\end{array}
$$

First note that $\sigma(N) = \{1,\, x_1,\, x_2,\, x_3,\, x_4,\, x_5,\, x_6,\, x_2x_3\} \neq N^\star$, but the difference is at the last element. In both cases they do not belong to $B(C, t)$ and $B(C^\star, t)$, respectively. It can be checked that $\phi^\star(x_2, x_3) = x_1x_5$, which means that the syndrome (corresponding to $\mathcal{C}^\star$) of x_1x_5 and x_2x_3 is the same. Then it is possible to do $N^\star := \sigma(N)$. Now we could apply Definition 15. For example, let us see that the second element of $\phi^\star$ corresponds to the second element of ϕ, that is $[[1,0,0,0,0,0],1,[1,7,5,4,8,3]]$ corresponds to $[[1,0,0,0,0,0],1,[1,6,5,4,3,8]]$. Note that in $\phi^\star[2][3][2]$ there is a 7 instead of 6, by Definition 15 we have

$$
\begin{aligned}
\phi^\star[2][3][2] = \phi^\star(x_1, x_2) = \phi^\star(\sigma(x_1), \sigma(x_2)) = N^\star[\underline{7}] = x_6 = \sigma(x_5) = \sigma(N[\underline{6}]) = \\
\sigma(N[\phi[2][3][2]]) = \sigma(\phi(x_1, x_2)).
\end{aligned}
$$

Note also that the last two positions on $\phi[2][3]$ and $\phi^\star[2][3]$ are exchanged:

- $\phi^\star[2][3][\underline{5}] = \phi^\star(x_1, x_5) = \phi^\star(\sigma(x_1), \sigma(x_6)) = N^\star[8] = x_2x_3 = \sigma(x_2x_3) = \sigma(N[8]) = \sigma(N[\phi[2][3][\underline{6}]]) = \sigma(\phi(x_1, x_6))$,
- $\phi^\star[2][3][\underline{6}] = \phi^\star(x_1, x_6) = \phi^\star(\sigma(x_1), \sigma(x_5)) = N^\star[3] = x_2 = \sigma(x_2) = \sigma(N[3]) = \sigma(N[\phi[2][3][\underline{5}]]) = \sigma(\phi(x_1, x_5))$.

Example 19. Neil Sloane is very fond of the two codes below $\mathcal{C}_1$ and $\mathcal{C}_2$ over

$\mathbb{F}_2^6$, with parity check matrix:

$$H_1^t = \begin{pmatrix} 1\,1\,0\,0\,0\,0 \\ 0\,0\,1\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1 \end{pmatrix} \quad H_2^t = \begin{pmatrix} 1\,1\,1\,1\,1\,1 \\ 0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,1\,0\,0 \end{pmatrix}$$

It is well known that they are isoespectral (i.e., they have the same weight distribution $1, 0, 3, 0, 3, 0, 1$) but they are not equivalent because the sum of the weight-2 codewords in $\mathcal{C}_1$ is $(1, 1, 1, 1, 1, 1)$ while in $\mathcal{C}_2$ is $(0, 0, 0, 0, 0, 0)$.

In the first case $N_1 = \{1, x_1, x_3, x_5, x_1x_3, x_1x_5, x_3x_5, x_1x_3x_5\}$ and ϕ_1 corresponds to:

[[[0, 0, 0, 0, 0, 0], 1, [2, 2, 3, 3, 4, 4]],[[1, 0, 0, 0, 0, 0], 1, [1, 1, 5, 5, 6, 6]]
[[0, 0, 1, 0, 0, 0], 1, [5, 5, 1, 1, 7, 7]],[[0, 0, 0, 0, 1, 0], 1, [6, 6, 7, 7, 1, 1]]
[[1, 0, 1, 0, 0, 0], 0, [3, 3, 2, 2, 8, 8]],[[1, 0, 0, 0, 1, 0], 0, [4, 4, 8, 8, 2, 2]]
[[0, 0, 1, 0, 1, 0], 0, [8, 8, 4, 4, 3, 3]],[[1, 0, 1, 0, 1, 0], 0, [7, 7, 6, 6, 5, 5]]]

In the second case $N_2 = \{1, x_1, x_2, x_4, x_6, x_1x_2, x_1x_4, x_1x_6\}$ and ϕ_2 is:

$$\begin{array}{l}
[[0,0,0,0,0,0],1,[2,3,2,4,2,5]][[1,0,0,0,0,0],1,[1,6,1,7,1,8]] \\
[[0,1,0,0,0,0],1,[6,1,6,8,6,7]][[0,0,0,1,0,0],1,[7,8,7,1,7,6]] \\
[[0,0,0,0,0,1],1,[8,7,8,6,8,1]][[1,1,0,0,0,0],0,[3,2,3,5,3,4]] \\
[[1,0,0,1,0,0],0,[4,5,4,2,4,3]][[1,0,0,0,0,1],0,[5,4,5,3,5,2]]
\end{array}$$

It is easy to conclude in this case that both codes are not equivalent, note that the elements in $\phi_1[i][3]$, $i = 1, \ldots, 8$, points to three different positions and each of them are repeated twice; as a consequence, this property should not change under the action of the permutation σ; however, one can see that it does not hold for ϕ_2.

In order to find the permutation there is much to be done, in the sequel we will approach to that problem by showing some partial results and strategies. Note that for finding the permutation the structure of matphi have to be checked, then it would be very useful to be able to study just a part of it, which already determine the whole structure. This would be something similar to the connection between a big Gröbner basis (the border basis for example) and the reduced Gröbner basis for a given order, the last one have the smaller size for such an structure with the desired properties.

4 Reduced basis

Let $\mathcal{C}$ be a code associated to the free commutative monoid $[X]$ by means of the morphisms φ and ξ. Let also be $<_e$ the error vector ordering defined on $[X]$; N a set of canonical forms, and ϕ the corresponding matphi function defined over $N \times X$. The notation $T(f)$ will denote as usual the maximal term of a polynomial f with respect to the order $<_e$, $T\{F\}$ the set of maximal terms of the set of polynomials F, $T(F)$ denote the semigroup ideal generated by $T\{F\}$, and $\langle F\rangle$ the polynomial ideal generated by the set F.

Definition 20 (Reduced basis). The reduced basis on $[X]$ for the code $\mathcal{C}$ and the order $<_e$ is the subset G of the set of binomials:

$B(\mathcal{C}) = \{wx - w' \, : \, w, w' \in N,\, x \in X,\, wx \neq w' \text{ and } \xi(wx) = \xi(w')\}$,

which satisfies the following two properties:

i. For all $(w, x) \in N \times X$ such that $wx \in T\{B\}$ [9], there exists $w' \in T\{G\}$ such that $w' \mid wx$.
ii. Given $w_1, w_2 \in T\{G\}$ then $w_1 \not| \, w_2$ and $w_2 \not| \, w_1$.

Remark 21.

(1) Note that the definition of reduced basis follows the same idea as a reduced Gröbner basis, just one has to be aware of the non admissibility of $<_e$, so, the performance of the reduced basis is in general affected, i.e, they can not always be used for an effective reduction process.
(2) Algorithm 1 can compute G instead of matphi. It is enough to check before Step 4, whether w is not a multiple of an element of $T\{G\}$ (for the constructed G). If w is a multiple, then the next element is taken from *List*; otherwise, Step 4 follows, Step 7 is replaced by $G := G \cup \{w - w_j\}$, while steps 11.2 and 11.3 disappear. Because of cosets determined by $\mathcal{C}$ are finite, the algorithm will always gave a reduced basis, if the changes described are made. The matter -considering the previous item- is whether this reduced basis is useful or not and how to use it.

The following algorithm computes the reduced basis for a code $\mathcal{C}$ and the order $<_e$.

Algorithm 2.
Input: p, n, m, H parameters for a given linear code.
Output: N, G.

1. $List := \{1\}, N := \emptyset, r := 0, G = \{\}$;

[9] Note that in this case $wx \neq \phi(w, x)$, i.e., $wx \notin N$.

2. While $List \neq \emptyset$ do
3. $\quad w := \text{NextTerm}[List]$;
3.1 $\quad$ If $w \in T(G)$ go to Step 2;
4. $\quad v' := \varphi(w)$;
5. $\quad (\Lambda, j) := \text{Member}[v', \{v_1, \ldots, v_r\}]$;
6. $\quad$ If $\Lambda =$true
7. $\quad$ them $G := G \cup \{w - w_j\}$;
8. $\quad$ else $r := r + 1$;
9. $\qquad v_r := v'$;
10. $\qquad w_r := w, \quad N := N \cup \{w_r\}$;
11. $\qquad List := \text{InsertNext}[w_r, List]$;
12. Return $[N, G]$

4.1 Binary case

In the binary case, it is essential that $x_i^2 - 1 \in I(C)$, for all $x_i \in X$. For a code with at least 1 error-correcting capability, $x_i^2 - 1 \in G$. In the case of a code with 0 error-correcting capability, $x_i^2 - 1 \in G$ or $x_i \in T\{G\}$; therefore, the other elements of $T\{G\}$ will have the variables with exponent one, so they will be standard words (see Definition 3). Note that, by using these relations in G, every word can be reduced to an standard word, which is the **standard representation** of the initial word. Note that for standard words, the order $<_e$ and the total degree term ordering are exactly the same. So, Algorithm 2 will compute the reduced basis w.r.t. $<_e$, which is exactly the reduced Gröbner basis w.r.t. the total degree term ordering corresponding to the ideal associated with the code $\mathcal{C}$ ($I(C) = \langle G \rangle$). As a consequence the following result holds.

Definition 22. The reduction in one step ($\longrightarrow$) using the reduced basis G is deffined as follows.
For any $w \in [X]$:

i. Reduce w to its standard form w' (it is enough to use the relations $x_i^2 \longrightarrow 1$, for all $x_i \in X$).
ii. Reduce w' with respect to G by the usual one step reduction.

Theorem 23. *Let $\mathcal{C}$ be a binary code and G the reduced basis with respect to $<_e$. Let $w \in [X]$ an arbitrary word, then*

i. The reduction process $\xrightarrow{*}$ [10] is notherian.
ii. The irreducible elements belong to N.
iii. If $w \xrightarrow{*} w'$, then $\xi(w) = \xi(w')$.

[10] Transitive clousure of $\longrightarrow$.

iv. *If* $w \xrightarrow{*} w_1$ *and* $w \xrightarrow{*} w_2$*, and* w_1*,* w_2 *are irreducible words modulo* $\xrightarrow{*}$*, then* $w_1 = w_2$*.*

Remark 24. By convention we will refer to the reduction process $\xrightarrow{*}$ as the reduction modulo G. The irreducible element corresponding to w will be called the canonical form of w and it will be denoted by $Can(w, G)$. By the Proposition iii, $\xi(w) = \xi(Can(w, G))$.

Proof. Let us assume that $w = w_0 \longrightarrow w_1 \ldots \longrightarrow w_k \ldots$ is a descending chain of reductions. Note that $w_i \longrightarrow w_{i+1}$ means (see Deffinition 22) that w_i is reduced to its standard form w_i', and w_i' is reduced in one step to w_{i+1} by the usual one step reduction modulo G. It is easy to see that $w_i >_e w_i'$ (every word is greater than its standard form, it is enough to notice that w_i' is a subword of w_i, see Remark 2). It can be proved that $w_{i+1} <_e w_i'$ (see 8.1.1 in Appendix). Then, after k steps of reductions we got $w_0' >_e \ldots >_e w_k'$, taken into consideration that the w_i''s are standard words, this chain will be also a descending chain for the total degree ordering; consequently, this chain have to stop (the total degree ordering is admissible, so there is no infinite descending chain of elements). Therefore, $\xrightarrow{*}$ is noetherian and i follows.

In order to prove ii, by construction of Algorithm 2 one can see that w belongs to $T(G) \cup N$, then because the reduction is noetherian an irreducible element belongs to N (moreover $T(G) \cap N = \emptyset$).

It is easy to see that the one step reduction keeps the syndrome invariant (see 8.1.2 in Appendix); therefore, iii follows. As a consequence, every irreducible word obtained by the reductions has the same syndrome as w, i.e., $\xi(w_1) = \xi(w_2) = \xi(w)$. By the Property 3 of N, there exists only one canonical form with a given syndrome, so, $w_1 = w_2$, and iv is proved. ☐

Remark 25 (The two keys of the success for $\mathbb{F}_2$). Let $\mathcal{C}$ be a code defined over a finite field $\mathbb{F}_{p^m}$.

(1) If $p \neq 2$, standard words could contain variables with exponents different from one.
(2) If $m \neq 1$, there could be more than one variable for each component of an n-vector in $\mathbb{F}_{p^m}^n$.

As a consequence, if $q = p^m \neq 2$ it is no longer true that over standard words the orders $<_e$ and the total degree ordering are the same. It will not be possible even to guarantee a descending chain of words in standard forms as a consequence of the reduction process.

Example 26. By $w \xrightarrow{G_i} v$ we mean w reduces to v modulo the i-th polynomial of the reduced basis G. The definitions of the codes which are used in this example can be found in Section 8.3 and the corresponding set N and G in 8.4.

(1) **CF2:** Let us reduce $w := x_1x_2x_3x_7x_8x_9$ modulo G:

$$w = x_1x_2x_3x_7x_8x_9 \xrightarrow{G_{45}} x_5x_7x_9^2x_{10},\;\; x_5x_7x_9^2x_{10} \xrightarrow{G_9} x_5x_7x_{10},$$
$$x_5x_7x_{10} \xrightarrow{G_{21}} x_1x_3x_{10}.$$

(2) **CF3:** Let $w := x_1x_5x_7$, then, for this word a cycle in the reduction process modulo G is getting.

$$w = x_1x_5x_7 \xrightarrow{G_{35}} x_3x_7^2,\;\; x_3x_7^2 \xrightarrow{G_{10}} x_1x_5x_7.$$

In this case the non admissibility of $<_e$ under standard words causes the cycle ($p \neq 2$ is the cause of the non admissibility, observe that $x_1x_5 <_e x_3x_7$ but $x_1x_5x_7 >_e x_3x_7^2$, and $x_3x_7^2$ is an standard word because the exponent 2 is allowed). This will not happen reducing w by means of matphi because it contains the whole set $B(\mathcal{C})$.

$\phi(1, x_1) = x_1$, now take the next variable in w, $\phi(x_1, x_5) = x_1x_5$. Finally, we take x_7, $\phi(x_1x_5, x_7) = x_3x_7^2$, so $w \xrightarrow{*} x_3x_7^2$. The reader can check the definition of the function cf in the proof of Theorem 7 (the function that associates to each $w \in [X]$ its canonical form, which consists in a sequence of applications of ϕ that depends on the sequence of letters that form w).

(3) **CF4:** Let $w = x_2x_4x_7$, for this word we will get a cycle in the reductions by G.

$$x_2x_4x_7 \xrightarrow{G_{37}} x_6x_7x_8 \xrightarrow{G_{23}} x_1x_8x_{10} \xrightarrow{G_{35}} x_6x_7x_8.$$

The non admissibility of $<_e$ under standard words in this example is because $m \neq 1$ ($m = 2$, $x_1x_{10} <_e x_6x_7$ but $x_1x_8x_{10} >_e x_6x_7x_8$ because $\mathrm{Ind}(x_7) = \mathrm{Ind}(x_8)$ then $|\mathrm{Ind}(x_1x_8x_{10})| = 3 > 2 = |\mathrm{Ind}(x_6x_7x_8)|$).

Let us see now the reduction process using matphi:

$$\phi(1, x_2) = x_2,\;\; \phi(x_2, x_4) = x_2x_4,\;\; \phi(x_2x_4, x_7) = x_6x_7x_8.$$

As a conclusion, matphi always lead to a finite sequence of reductions in a number of steps equal to the length of the word; however, a much bigger structure than the reduced basis its needed. For binary codes, it is always possible to apply reduction using the reduced basis instead of matphi.

5 Reduced basis and equivalence of binary codes

Let $\mathcal{C}$ be a binary code of length n, $\sigma \in S_n$, and $\mathcal{C}^\star = \sigma(C)$. Section 3, Definition 12 show how to apply a permutation to a word, it is then not difficult to extent the action of σ onto a polynomial set in a natural way. Assume the error vector order $<_e$ is defined using an admissible ordering $\prec$.

Theorem 27. *Let G be a reduced basis, and let $x_1 \prec \ldots \prec x_n$. Then $G^\star = \sigma(G)$ is a reduced basis for $\mathcal{C}^\star$ and the order $<_e$ with the admissible order $\prec$ so that $x_{\sigma(1)} \prec \ldots \prec x_{\sigma(n)}$.*

Taking into consideration the computation of the reduced basis for $\mathcal{C}^\star$, for the order $<_e$ and the admissible order $\prec$ taking $x_{\sigma(1)} \prec \ldots \prec x_{\sigma(n)}$, one can infer that: the elements generated in $List^\star$ are such that, at each step, $List^\star = \sigma(List)$. As a consequence, $N^\star = \sigma(N)$, and the reduced basis computed by the algorithm will be $G^\star = \sigma(G)$.

Note that when σ is not known, $G^\star$ is also unknown, but one can compute the reduced basis for $\mathcal{C}^\star$ (G'), by considering the natural order for the variables $x_1 \prec \ldots \prec x_n$.

Theorem 28. *Given $\sigma \in S_n$, G a reduced basis for a code $\mathcal{C}$, G' a reduced basis of a code $\mathcal{C}'$. Then the following three propositions are equivalent.*

i. $\mathcal{C}' = \sigma(\mathcal{C})$.
ii. $G^\star := \sigma(G)$ *is a reduced basis for the code $\mathcal{C}'$ w.r.t. $<_e$, for some ordering of the variables.*
iii. *Every binomial of $G^\star$ is reduced to zero modulo G' and, vice versa, every binomial of G' is reduced to zero modulo $G^\star$.*

This result is the analogous for reduced bases of Theorem 16 for matphi, it is a consequence of Theorem 23 and Theorem 27, and the fact that $G^\star$ and G' are both reduced basis of $\mathcal{C}^\star$, but w.r.t. different orderings.

Proof. *i* $\Rightarrow$ *ii*: Is a consequence of Theorem 27.

ii $\Rightarrow$ *iii*: It follows from the definition of reduced basis (Definition 20) and the fact that G' and $G^\star$ are both reduced basis of the same code.

iii $\Rightarrow$ *i*: Let $c' \in \mathcal{C}'$ and $w \in [X]$ the standard word such that $\varphi(w) = c$. *iii* guarantee that a vector $c' \in \mathcal{C}'$ if and only if $c' \in \sigma(\mathcal{C})$ (a word w such that $\varphi(w) = c$ is reduced to 1 modulo G' if and only if it is reduced to 1 modulo $\sigma(G)$); therefore, $\mathcal{C}' = \sigma(\mathcal{C})$ (see 8.6 in Appendix for more details). $\square$

Remark 29 (Finding the permutation).

(1) The comparison of the structures of G and G', and the application of Theorem 28, are the main tools for finding a suitable permutation or concluding that the codes are not equivalent.
(2) Let $S^\star$ be the subgroup of S_n that keeps invariant the code $\mathcal{C}^\star$. Then, it is well known that $\{\sigma \circ \rho \mid \rho \in S^\star\}$, is the set of permutations that transform $\mathcal{C}$ in $\mathcal{C}^\star$. Let us observe that there could be possible the existence of $\rho' \in S^\star$, so that $\sigma'(G) = G'$, where $\sigma' = \sigma \circ \rho'$. Hence, we are interested in finding such a σ' (if there is any).

5.1 Finding the permutation using the reduced basis

We will work with the binary code **CF2** defined in Section 8.3 of the Appendix, $\mathcal{C} := CF2$.

Example 30. Let $\sigma := (1, 10, 2, 7, 9, 6, 4, 3, 5)$ and $\mathcal{C}^\star = \sigma(C)$. See the definition of $\mathcal{C}^\star$ in Section 8.3 in Example 42. Let us suppose σ is unknown, therefore, $N^\star$ and $G^\star$, given in Section 8.5, are also unknown. However, one can compute the set of canonical forms N' and the reduced basis G' for the code $\mathcal{C}^\star$, see the results on Section 8.5.1. It will be shown how there exists a permutation σ' such that it transforms G in G' and, as a consequence, the code $\mathcal{C}^\star$ it is also gotten as $\sigma'(\mathcal{C})$.

Let us get for G (Example 46), the list $Heads(2)$ of length 10, such that

$$Heads(2)[i] := |\{w \in T\{G\} \, : \, i \in \mathrm{Ind}(w) \text{ and } |\mathrm{Ind}(w) = 2\}|,$$

for $i = 1, \ldots, 10$.

Note that a reduced basis can be divided by levels according to $|\mathrm{Ind}(T\{G\})|$; for example, the level one of the reduced basis G, is composed by the set of binomials $\{x_i^2 - 1 \, : \, i = 1, \ldots, 10\}$. Let us compute also $Heads'(2)$ for G' (see Section 8.5.3).

$Heads(2) = [0, 2, 3, 4, 6, 6, 5, 4, 0, 0]$, $Heads' = [0, 0, 2, 3, 4, 0, 5, 6, 4, 6]$.

The same reasoning can be done with the irreducible elements in the levels of a reduced bases; for example, at the level 2 of G:

$Irreds(2)[i] := |\{w - v \in G \, : \, i \in \mathrm{Ind}(v) \text{ and } |\mathrm{Ind}(w)| = 2\}|$, for $i = 1, \ldots, 10$.
Therefore we compute the lists $Irreds(2)$ for G and $Irreds'(2)$ for G'.
$Irreds(2) = [9, 5, 4, 3, 3, 1, 2, 3, 0, 0]$, $Irreds'(2) = [9, 0, 5, 4, 3, 0, 2, 1, 3, 3]$.

Note that if such a permutation σ' exists, it should transform $Heads(2)$ in $Heads'(2)$ and $Irreds(2)$ in $Irreds'(2)$. From this two conditions it is possible the get the following.

By considering $Heads(2)$ and $Heads'(2)$, it is obtained that $\sigma'(7) = 7$, $\sigma'(2) = 3$, and $\sigma'(3) = 4$. Taking into account $Irreds(2)$ and $Irreds'(2)$, we got, $\sigma'(1) = 1$ and $\sigma'(6) = 8$. Then $\sigma'(6)$ is already determined, so, $Heads(2)[5] = Heads(2)[6]$ and $\sigma'(6) = 8$ implies $\sigma'(5) = 10$. So, it remains to find $\sigma'(4)$, $\sigma'(8)$, $\sigma'(9)$, $\sigma'(10)$. It is clear that $\sigma'(\{4, 8\}) \in \{5, 9\}$, and $\sigma'(\{9, 10\}) \in \{2, 6\}$.

Looking at the level 4 of G, $x_1x_2x_3x_8 \in T\{G\}$ then, $x_1x_3x_4\sigma'(x_8) \in T\{G'\}$; therefore, $\sigma'(x_8) = x_5$, which implies $\sigma'(8) = 5$ and $\sigma'(4) = 9$. On the other hand, $\sigma'(\{9, 10\}) \in \{2, 6\}$ are enough conditions for the images of 9 and 10. As a consequence, a permutation σ' can be given (actually two permutations can be given) such that $\mathcal{C}^\star = \sigma'(\mathcal{C})$,

$\sigma' = [1, 3, 4, 9, 10, 8, 7, 5, 2, 6]$ (in list notation).

The reader can check that σ' transforms G in G', which is sufficient [11] (not necessary) to decide that $\mathcal{C}$ and $\mathcal{C}^\star$ are equivalent.

Example 31. In the Example 19, we have already shown (by comparing the matphi's) that the code $\mathcal{C}_1$ and $\mathcal{C}_2$ are not equivalent. Let us see this result now by using the reduced basis. See in the Appendix, on Example 44 and 45, the reduced basis for both codes. Note the number of elements of both sets are not the same, which it says that at least there is not a direct transformation from G_1 to G_2, but still they can be equivalent (see Theorem 28 and Remark 29).

If they were equivalent codes, then it should exist a reduced basis $G_2^\star$ for $\mathcal{C}_2$ such that it can be obtained from G_1 by the transformation using a permutation $\sigma \in S_6$. Then every binomial in $G_2^\star$ most be reduced to zero modulo G_2. Looking the structure of G_1, there are three binomials of the form $x_i - x_j$, so, in the reduced basis $G_2^\star$ there will be also three binomials of that form. Then, these three binomials of $G_2^\star$ must be reduced to zero modulo G_2, as a consequence, three different variables have to be divisible for elements in $T(G_2)$, but there are only two variables belonging to $T(G_2)$. Therefore, at least one of the binomials in $G_2^\star$ can not be reduced to zero, and that implies $\mathcal{C}_1$ and $\mathcal{C}_2$ are not equivalent.

Example 32. Let $\sigma = (1, 2, 6, 9, 10, 4, 5, 3, 7, 8)$ and $\mathcal{C}^\star = \sigma(C)$. See the definition of $\mathcal{C}^\star$ in Section 8.3 on Example 43. Let suppose σ is unknown, therefore, $N^\star$ and $G^\star$, given in Section 8.5.2, are also unknown. However, one can compute the set of canonical forms N' and the reduced basis G' for the code $\mathcal{C}^\star$, see the results on Section 8.5.3. It will be shown how in this case there is no permutation σ' such that it transforms G in G'; however, Theorem 28 holds and consequently, the codes are equivalent.

The reader can see that the number of elements in G is 46, and the number of elements in G' is 45. So there can not exist a permutation that transforms G into G'. One can also note that at the level 4, G has two elements, while G'

[11] The existence of such a permutation.

has only one.

In this case, one can check that Theorem 28 holds for $G^\star$ and G', which it would confirm that $\mathcal{C}^\star$ is gotten as $\sigma(C)$. In **GBLA_LC** there is a function ***basesequivalents*** that determined whether two reduced basis are reduced basis for the same code.

The most complecated case occurs when there is not a direct transformation from G to G'. In order to find a suitable permutation, it could be helpful to know some images, this is of course an heuristic approach.

Fortunately, the use of the reduced basis for binary codes in the decoding problem, has a complete solution for an arbitrary binary code, an arbitrary reduced basis for the given code, and for any given received vector of this code.

6 Decoding binary codes using the reduced basis

In M. Borges et al. (2001, 2000) authors showed how to apply the structure matphi for decoding general linear codes. Although the fast decoding process using matphi can not be compared with other quite efficient algorithms for decoding some class of linear codes (Reed-Solomom Codes, BCH Codes, Algebraic Geometric Codes) because those algorithms require a very little preprocessing. In (W.W. Peterson, E.J. Weldon (1972)) it is shown how a variant of the syndrome decoding algorithm can be obtained, the Step-by-Step algorithm, this algorithm allows to use the same syndrome decoding idea but using an smaller structure than the syndrome table, in essence, it is shown that it is enough to know for each coset the smaller weight instead of storing the candidate error vector. Following the same desire of reducing the needed structure for decoding any arbitrary linear code, we give an step further for an arbitrary binary code. In this case, it will be enough to have the reduced basis, which is always smaller than the whole structure matphi (whose size is proportional to the syndrome table) and many times can be considerable smaller than matphi and the syndrome table.

Theorem 33. *Let $\mathcal{C}$ be a binary code and G the reduced basis with respect to $<_e$. Let $w \in [X]$ an arbitrary word, if $weight(\varphi(Can(w, G))) \leq t$ then $\varphi(Can(w, G))$ is the error vector corresponding to $\varphi(w)$. Otherwise, if $weight(\varphi(Can(w, G))) > t$, $\varphi(w)$ contains more than t errors.*

Remark 34.

(1) Note that every vector in F_2^n can be translated to a word in standard representation, therefore, applying the previous theorem it is possible to

decode an arbitrary vector in F_2^n. It is also possible to determine when the vector contains more than t errors.

(2) The proof of this theorem is a direct consequence of Theorem 23 and the property 5 of a set of canonical forms (see Remark 9). First, 23.i said that the element $Can(w, G)$ exists and can be computed for any $w \in [X]$. Second, 23.ii guarantees that $Can(w, G)$ belongs to N and 23.iii states that the syndrome of w is the same of $Can(w, G)$. If $\varphi(w) \in B(C, t)$ (i.e. $\varphi(w)$ contains at most t errors), by the property 5 of N, $weight(\varphi(Can(w, G))) \leq t$, that is, $\varphi(Can(w, G))$ is the error vector, and $\varphi(w) - \varphi(Can(w, G))$ is the codeword corresponding to $\varphi(w)$. If $\varphi(w) \notin B(C, t)$ it is clear that $\varphi(Can(w, G)) \notin B(C, t)$ (they both have the same syndrome), consequently, if $weight(\varphi(Can(w, G))) > t$ it means that more than t errors were done.

Example 35. In the Example 26, the reader can see the application of the reduced basis for the binary code **CF2** (see the definition of the code on Section 8.3). The vector $(1, 1, 1, 0, 0, 0, 1, 1, 1, 0) \in F_2^{10}$ (corresponding to the word $x_1x_2x_3x_7x_8x_9$) was reduced to $(1, 0, 1, 0, 0, 0, 0, 0, 0, 1)$ (corresponding to the word $x_1x_3x_{10}$). But the weight of the resulting vector is 3, and the error correcting capability of **CF2** is 1. So the result is not reliable.

Let us take now the vector $v = (1, 1, 1, 1, 0, 0, 0, 0, 1, 1)$, the corresponding word is $w = x_1x_2x_3x_4x_9x_{10}$. Let us reduce now w using the reduced basis of **CF2** given on Example 46.

$$w = x_1x_2x_3x_4x_9x_{10} \xrightarrow{G_{26}} x_1x_9^2x_{10}^2 \xrightarrow{G_9} x_1x_{10}^2 \xrightarrow{G_{10}} x_1.$$

$weight(\varphi(x_1)) = weight((1, 0, 0, 0, 0, 0, 0, 0, 0, 0)) = 1$, then

$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ is the error vector corresponding to v, and the codeword is $(0, 1, 1, 1, 0, 0, 0, 0, 1, 1)$.

7 GAP examples

All the computation was done in Gap 4.3, with the GAP package **GBLA_LC** [12].

The order used in the examples shown below is $\prec := <_{TL}$.

Example 36. Code over $\mathbb{F}_2^6$, with parity check matrix (see in Example 19 the

[12] **GBLA_LC** "Gröbner Basis by Linear Algebra and Linear Codes" is a collection of programms made in GAP, it consist in an extended and improved version of the work presented in Y. Azze (2003), which was made under of guidance of Borges-Quintana.

definition of the code $\mathcal{C}_1$):

$$H_1^t = \begin{pmatrix} 1\ 1\ 0\ 0\ 0\ 0 \\ 0\ 0\ 1\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 1\ 1 \end{pmatrix}$$

Definition of the code:

```
n:=6;
k:=3;
m:=1;
p:=2;
F:=GF(2);
alpha_prim:=RootOfDefiningPolynomial(GF(2));
alpha_ext:=RootOfDefiningPolynomial(F);
a:=alpha_ext;
H:=[[One(a),One(a),Zero(a),Zero(a),Zero(a),Zero(a)],
    [Zero(a),Zero(a),One(a),One(a),Zero(a),Zero(a)],
    [Zero(a),Zero(a),Zero(a),Zero(a),One(a),One(a)]];
R:=PolynomialRing(Rationals,n*m);
x:=IndeterminatesOfPolynomialRing(R);
```

GBLA output (see ϕ_1 in Page 10):

```
gap>GBLA_LC(H,m,n,k,p);
[[1,x_1,x_3,x_5,x_1*x_3,x_1*x_5,
          x_3*x_5,x_1*x_3*x_5],
[[[0*Z(2),0*Z(2),0*Z(2),0*Z(2),0*Z(2),0*Z(2)],
   [ 2, 2, 3, 3, 4, 4 ] ],
[[Z(2)^0,0*Z(2),0*Z(2),0*Z(2),0*Z(2),0*Z(2)],
   [ 1, 1, 5, 5, 6, 6 ] ],
[[0*Z(2),0*Z(2),Z(2)^0,0*Z(2),0*Z(2),0*Z(2)],
   [ 5, 5, 1, 1, 7, 7 ] ],
[[0*Z(2),0*Z(2),0*Z(2),0*Z(2),Z(2)^0,0*Z(2)],
   [ 6, 6, 7, 7, 1, 1 ] ],
[[Z(2)^0,0*Z(2),Z(2)^0,0*Z(2),0*Z(2),0*Z(2)],
   [ 3, 3, 2, 2, 8, 8 ] ],
[[Z(2)^0,0*Z(2),0*Z(2),0*Z(2),Z(2)^0,0*Z(2)],
   [ 4, 4, 8, 8, 2, 2 ] ],
[[0*Z(2),0*Z(2),Z(2)^0,0*Z(2),Z(2)^0,0*Z(2)],
   [ 8, 8, 4, 4, 3, 3 ] ],
[[Z(2)^0,0*Z(2),Z(2)^0,0*Z(2),Z(2)^0,0*Z(2)],
   [ 7, 7, 6, 6, 5, 5 ] ] ], 0 ]
```

The output is given as list of three components. The first one, is the set of vectors $\varphi(N)$ corresponding to the images by φ of the canonical forms. The second one, is the function matphi, each component of matphi have the structure $[\varphi(w), [\phi(w, x_1), \ldots, \phi(w, x_n)]]$, i.e., the 0-1 second component is omitted [13]. The third one, corresponds to the error-correcting capability of the code.

Example 37. See Example 19 the definition of the code $\mathcal{C}_2$.

```
n:=6;
k:=3;
m:=1;
p:=2;
F:=GF(2);
alpha_prim:=RootOfDefiningPolynomial(GF(2));
alpha_ext:=RootOfDefiningPolynomial(F);
a:=alpha_ext;
H:=[[One(a),One(a),One(a),One(a),One(a),One(a)],
     [Zero(a),Zero(a),Zero(a),One(a),Zero(a),One(a)],
     [Zero(a),One(a),Zero(a),One(a),Zero(a),Zero(a)]];
```

Computing the reduced basis for $\mathcal{C}_2$:

```
gap> Greduce1(H,m,n,k,p);
[ [ 1, x_1, x_2, x_4, x_6, x_1*x_2, x_1*x_4, x_1*x_6 ],
  [ [ 1, [ x_3, x_1 ], [ x_5, x_1 ], [ x_1^2, 1 ], [ x_2^2, 1 ],
         [ x_4^2, 1 ], [ x_6^2, 1 ] ],
      [ 2, [ x_2*x_4, x_1*x_6 ], [ x_2*x_6, x_1*x_4 ],
           [ x_4*x_6, x_1*x_2 ] ] ], 0 ]
```

The output is given as list of three components. The first one, is the set N of canonical forms. The second one, is the reduced basis, given as a list of pairs (w, v) such that $w \in T\{G\}$ and $v \in N$, the pairs are organized in sublist according to the $|\mathrm{Ind}(w)|$. The third one, corresponds to the error-correcting capability of the code.

In **GBLA_LC** there are built in functions which allow to convert the reduced basis from a set of pairs to a set of binomials.

[13] Note that this information can be easily gotten from the weight of $\varphi(w)$, if $weight(\varphi(w)) \leq t$ the corresponding second component would be 1, and 0 otherwise.

8 Appendix

8.1 Proof of Theorem 23

1- $w_{i+1} <_e w'_i$.
$w'_i = u_1u_2$ and let $u_1 - v_1 \in G$, then, w'_i is reduced to $w_{i+1} = v_1u_2$, but $u_1 <_e v_1$. There are two cases:
Case 1: $|\mathrm{Ind}(u_1)| > |\mathrm{Ind}(v_1)|$, then, since w'_i is an standard word, it is obvious that $|\mathrm{Ind}(w'_i)| > |\mathrm{Ind}(w_{i+1})|$ and then, $w_{i+1} <_e w'_i$.
Case 2: $|\mathrm{Ind}(u_1)| = |\mathrm{Ind}(v_1)|$ and $v_1 \prec u_1$, where $\prec$ denotes an admissible ordering. If $\mathrm{Ind}(v_1) \cap \mathrm{Ind}(u_2) \neq \emptyset$ then $|\mathrm{Ind}(w_{i+1})| \leq |\mathrm{Ind}(w'_i)| - 1$ and therefore, $w_{i+1} <_e w'_i$. So, let assume that $\mathrm{Ind}(v_1) \cap \mathrm{Ind}(u_2) = \emptyset$, as a consequence, $|\mathrm{Ind}(w_{i+1})| = |\mathrm{Ind}(w'_i)|$. Taking into consideration the admissibility of $\prec$, $w_{i+1} = v_1u_2 \prec u_1u_2 = w'_i$ which implies $w_{i+1} <_e w'_i$.

□

2- The one step reduction of w by G keeps the syndrome.
$w = w_1w_2$ and let $w_1 - v_1 \in G$, then, w is reduced to v_1w_2, but $\xi(w_1) = \xi(v_1)$, therefore, $\xi(w) = \xi(w_1w_2) = \varphi(w_1w_2)H = (\varphi(w_1) + \varphi(w_2))H = \varphi(w_1)H + \varphi(w_2)H = \varphi(v_1)H + \varphi(w_2)H = \varphi(v_1w_2)H = \xi(v_1w_2)$.

□

8.2 A definition of a total degree ordering

The following well known order will play an important role, in the computation of the reduced basis and the connection with the equivalence problem. Assume the monoid $[X]$ is the free conmutative monoid of n variables, and let $\sigma \in S_n$.

Definition 38. Let $<_{TL}$ be the total degree ordering defined as follows. Assume the order of the variables is determined by the permutation σ ($x_{\sigma(1)} <_{TL} \ldots <_{TL} x_{\sigma(n)}$). Let be $w_1, w_2 \in [X]$, such that
$w_1 = x_{i_1} \ldots x_{i_k}$ ($x_{i_l} <_{TL} x_{i_{l+1}}$), $w_2 = x_{j_1} \ldots x_{j_m}$ ($x_{j_l} <_{TL} x_{j_{l+1}}$) then $w_1 <_{TL} w_2$ if one of the following condition is satisfied:

i. $k < m$.
ii. $k = m$ and $x_{i_S} <_{TL} x_{j_S}$, where $s = min\,\{l = 1, \ldots, p : i_l \neq j_l\}$.

Note that condition ii express a lexicographic order, it said that the order is determined by the first difference in the sequence of variables of each word, then the variable less with respect to $<_{TL}$ will decide which word is less with respect to $<_{TL}$. It is not difficult to prove that $<_{TL}$ is admissible.

An (n, k, t)-code will be a code of length n, dimension k, and t error correcting capability.

Example 39 (CF2). A $(10, 4, 1)$-code over $\mathbb{F}_2$. Number of codewords: 16, number of canonical forms: 64;

$$\mathbf{H} := \begin{vmatrix} 1&0&0&0&1&0&0&0&0&0 \\ 1&0&1&1&0&1&0&0&0&0 \\ 1&1&0&1&0&0&1&0&0&0 \\ 1&1&1&0&0&0&0&1&0&0 \\ 1&1&1&1&0&0&0&0&1&0 \\ 1&1&1&1&0&0&0&0&0&1 \end{vmatrix}$$

Example 40 (CF3). A $(7, 3, 1)$-code over $\mathbb{F}_3$. Number of codewords: 27,number of canonical forms: 81;

$$\mathbf{H} := \begin{vmatrix} 1&0&1&2&0&0&0 \\ 1&1&0&0&2&0&0 \\ 1&1&1&0&0&1&0 \\ 0&0&1&0&0&0&2 \end{vmatrix}$$

Example 41 (CF4). A $(5, 2, 1)$-code over $\mathbb{F}_4$. Let a be a primitive element in $\mathbb{F}_4$. Number of codewords: 16, number of canonical forms: 64;

$$\mathbf{H} := \begin{vmatrix} 1&1&1&1&1 \\ 0&1&a&a^2&0 \\ 1&a&a^2&0&0 \end{vmatrix}$$

Example 42 ($\boldsymbol{\sigma}$(CF2)-1). $(\sigma := (1, 10, 2, 7, 9, 6, 4, 3, 5) \in S_{10})$ [14]
A $(10, 4, 1)$-code over $\mathbb{F}_2$. Number of codewords: 16, number of canonical forms:

[14] The permutation σ is given in cycle notation.

64;

$$\mathbf{H} := \begin{vmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Example 43 ($\boldsymbol{\sigma}$(CF2)-2). ($\sigma = (1, 2, 6, 9, 10, 4, 5, 3, 7, 8) \in S_{10}$).
A $(10, 4, 1)$-code over $\mathbb{F}_2$. Number of codewords: 16, number of canonical forms: 64;

$$\mathbf{H} := \begin{vmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

8.4 Set of canonical forms and reduced basis

In all cases it was used the order $<_{TL}$ with the identity permutation (see Section 8.2). All the computation was done in Gap 4.3, with the **GBLA_LC**[15] group of programms.

Example 44 (Code $\mathcal{C}_1$ of the Example 19).

$$N_1 = \{1,\ x_1,\ x_3,\ x_5,\ x_1x_3,\ x_1x_5,\ x_3x_5,\ x_1x_3x_5\},$$

$$G_1 = \{x_2 - x_1,\ x_4 - x_3,\ x_6 - x_5,\ x_1^2 - 1,\ x_3^2 - 1,\ x_5^2 - 1\}.$$

[15] **GBLA_LC** "Gröbner Basis by Linear Algebra and Linear Codes" is a group of programms made in GAP, it consist in an extended an improved version of a Mathematic Diploma Work of the course 2002-2003, which was tutored by Mijail Borges Quintana.

Example 45 (Code $\mathcal{C}_2$ of the Example 19).

$$N_2 = \{1,\ x_1,\ x_2,\ x_4,\ x_6,\ x_1x_2,\ x_1x_4,\ x_1x_6\},$$

$$\begin{aligned} G_2 = \{\ & x_3 - x_1,\ x_5 - x_1,\ x_1^2 - 1,\ x_2^2 - 1,\ x_4^2 - 1,\ x_6^2 - 1, \\ & x_2x_4 - x_1x_6,\ x_2x_6 - x_1x_4,\ x_4x_6 - x_1x_2\}. \end{aligned}$$

Example 46 (CF2).

$$\begin{aligned} N = \{\ & 1,\ x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7,\ x_8,\ x_9,\ x_{10}, \\ & x_1x_2,\ x_1x_3,\ x_1x_4,\ x_1x_5,\ x_1x_6,\ x_1x_7,\ x_1x_8,\ x_1x_9,\ x_1x_{10},\ x_2x_3,\ x_2x_4, \\ & x_2x_7,\ x_2x_8,\ x_2x_9,\ x_2x_{10}, x_3x_4,\ x_3x_8,\ x_3x_9,\ x_3x_{10},\ x_4x_9,\ x_4x_{10},\ x_5x_9, \\ & x_5x_{10},\ x_6x_9,\ x_6x_{10},\ x_7x_9,\ x_7x_{10},\ x_8x_9,\ x_8x_{10},\ x_9x_{10}, \\ & x_1x_2x_3,\ x_1x_2x_4,\ x_1x_2x_7,\ x_1x_2x_8,\ x_1x_2x_9,\ x_1x_2x_{10},\ x_1x_3x_4,\ x_1x_3x_8, \\ & x_1x_3x_9,\ x_1x_3x_{10},\ x_1x_4x_9,\ x_1x_4x_{10},\ x_1x_5x_9,\ x_1x_5x_{10},\ x_1x_6x_9,\ x_1x_6x_{10}, \\ & x_1x_7x_9,\ x_1x_7x_{10},\ x_1x_8x_9,\ x_1x_8x_{10},\ x_1x_9x_{10},\ x_2x_3x_8,\ x_5x_9x_{10}\}, \end{aligned}$$

$$\begin{aligned} G = \{\ & x_1^2 - 1,\ x_2^2 - 1,\ x_3^2 - 1,\ x_4^2 - 1,\ x_5^2 - 1,\ x_6^2 - 1,\ x_7^2 - 1,\ x_8^2 - 1, \\ & x_9^2 - 1, x_{10}^2 - 1, \\ & x_2x_5 - x_1x_6,\ x_2x_6 - x_1x_5,\ x_3x_5 - x_1x_7,\ x_3x_6 - x_2x_7,\ x_3x_7 - x_1x_5, \\ & x_4x_5 - x_1x_8,\ x_4x_6 - x_2x_8,\ x_4x_7 - x_3x_8,\ x_4x_8 - x_1x_5,\ x_5x_6 - x_1x_2, \\ & x_5x_7 - x_1x_3,\ x_5x_8 - x_1x_4,\ x_6x_7 - x_2x_3,\ x_6x_8 - x_2x_4,\ x_7x_8 - x_3x_4, \\ & x_2x_3x_4 - x_9x_{10},\ x_2x_3x_9 - x_4x_{10},\ x_2x_3x_{10} - x_4x_9,\ x_2x_4x_9 - x_3x_{10}, \\ & x_2x_4x_{10} - x_3x_9,\ x_2x_7x_9 - x_8x_{10},\ x_2x_7x_{10} - x_8x_9,\ x_2x_8x_9 - x_7x_{10}, \\ & x_2x_8x_{10} - x_7x_9,\ x_2x_9x_{10} - x_3x_4,\ x_3x_4x_9 - x_2x_{10},\ x_3x_4x_{10} - x_2x_9, \\ & x_3x_8x_9 - x_6x_{10},\ x_3x_8x_{10} - x_6x_9,\ x_3x_9x_{10} - x_2x_4,\ x_4x_9x_{10} - x_2x_3, \\ & x_6x_9x_{10} - x_3x_8,\ x_7x_9x_{10} - x_2x_8,\ x_8x_9x_{10} - x_2x_7, \\ & x_1x_2x_3x_8 - x_5x_9x_{10},\ x_1x_5x_9x_{10} - x_2x_3x_8\}. \end{aligned}$$

Example 47 (CF3).

$$N = \{\ 1,\ x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7,\ x_1^2,\ x_2^2,\ x_3^2,\ x_4^2,\ x_5^2,\ x_6^2,\ x_7^2,$$

$x_1x_2,\ x_1x_3,\ x_1x_5,\ x_1x_6,\ x_1x_7,\ x_2x_3,\ x_2x_4,\ x_2x_6,\ x_2x_7,\ x_3x_4,\ x_3x_5,$

$x_3x_6,\ x_4x_5,\ x_4x_6,\ x_4x_7,\ x_5x_6,\ x_5x_7,\ x_6x_7,\ x_1^2x_3,\ x_1^2x_5,\ x_1^2x_7,\ x_2^2x_3,$

$x_1x_5^2,\ x_1x_6^2,\ x_1x_7^2,\ x_2^2x_7,\ x_2x_3^2,\ x_1x_3^2,\ x_2x_7^2,\ x_3^2x_4,\ x_3^2x_7,\ x_3x_4^2,\ x_4^2x_5,$

$x_3x_6^2,\ x_4x_5^2,\ x_4^2x_7,\ x_4x_6^2,\ x_4x_7^2,\ x_6^2x_7,\ x_3x_7^2,\ x_1^2x_3^2,\ x_1^2x_6^2,\ x_1^2x_7^2,\ x_2^2x_3^2,$

$x_2^2x_7^2,\ x_3^2x_6^2,\ x_4^2x_6^2,\ x_6^2x_7^2,$

$x_1x_2x_3,\ x_1x_2x_7,\ x_1x_3x_6,\ x_1x_6x_7,\ x_2x_3x_4,\ x_2x_3x_6,\ x_2x_4x_7,\ x_2x_6x_7,$

$x_3x_4x_5,\ x_3x_5x_6,\ x_4x_5x_7,\ x_5x_6x_7,\ x_1^2x_3x_5,\ x_1^2x_5x_7,\ x_1x_3x_6^2,\ x_1x_6^2x_7,$

$x_3x_4^2x_5,\ x_4^2x_5x_7\},$

$$
\begin{aligned}
G = \{\ & x_1^3-1,\ x_2^3-1,\ x_3^3-1,\ x_4^3-1,\ x_5^3-1,\ x_6^3-1,\ x_7^3-1,\\
& x_1x_4-x_2,\ x_2x_5-x_6,\ x_3x_7-x_1x_5,\ x_1^2x_2-x_4,\ x_1^2x_6-x_4x_5,\\
& x_1x_2^2-x_4^2,\ x_2^2x_4-x_1^2,\ x_2^2x_6-x_5,\ x_2x_4^2-x_1,\ x_2x_6^2-x_5^2,\ x_3^2x_5-x_1^2x_7,\\
& x_3^2x_6-x_4x_7,\ x_3x_5^2-x_1x_7^2,\ x_4^2x_6-x_1x_5,\ x_5^2x_6-x_2,\ x_5^2x_7-x_1x_3^2,\\
& x_5x_6^2-x_2^2,\ x_5x_7^2-x_1^2x_3,\ x_6x_7^2-x_3x_4,\ x_1^2x_5^2-x_4x_6^2,\ x_3^2x_4^2-x_6^2x_7,\\
& x_3^2x_7^2-x_4x_6^2,\ x_4^2x_5^2-x_1x_6^2,\ x_4^2x_7^2-x_3x_6^2,\\
& x_1x_2x_6-x_4^2x_5,\ x_1x_3x_5-x_3^2x_7,\ x_1x_5x_6-x_4^2x_6^2,\ x_1x_5x_7-x_3x_7^2,\\
& x_2x_4x_6-x_1^2x_5,\ x_3x_4x_6-x_6^2x_7^2,\ x_4x_5x_6-x_1^2x_6^2,\ x_4x_6x_7-x_3^2x_6^2,\\
& x_1x_2x_7^2-x_3x_5x_6,\ x_1x_2x_3^2-x_5x_6x_7,\ x_2x_3^2x_4-x_1x_6^2x_7,\\
& x_2x_4x_7^2-x_1x_3x_6^2\}.
\end{aligned}
$$

Example 48 (CF4).

$$
\begin{aligned}
N = \{\ & 1,\ x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7,\ x_8,\ x_9,\ x_{10},\\
& x_1x_2,\ x_3x_4,\ x_5x_6,\ x_7x_8,\ x_9x_{10},\\
& x_1x_3,\ x_1x_4,\ x_1x_5,\ x_1x_6,\ x_1x_7,\ x_1x_8,\ x_1x_9,\ x_1x_{10},\ x_2x_3,\ x_2x_4,\ x_2x_5,\\
& x_2x_6,\ x_2x_7,\ x_2x_8,\ x_2x_9,\ x_2x_{10},\ x_3x_6,\ x_4x_7,\ x_4x_{10},\ x_5x_8,\ x_7x_9,\ x_8x_{10},\\
& x_1x_2x_3,\ x_1x_2x_4,\ x_1x_2x_5,\ x_1x_2x_6,\ x_1x_2x_7,\ x_1x_2x_8,\ x_1x_2x_9,\ x_1x_2x_{10},\\
& x_1x_3x_4,\ x_1x_7x_8,\ x_1x_9x_{10},\ x_2x_3x_4,\ x_2x_5x_6,\ x_3x_4x_5,\ x_2x_7x_8,\ x_2x_9x_{10},\\
& x_3x_4x_8,\ x_3x_4x_{10},\ x_1x_5x_6,\ x_5x_6x_7,\ x_3x_7x_8,\ x_6x_7x_8,\\
& x_1x_2x_3x_4,\ x_1x_2x_5x_6,\ x_1x_2x_9x_{10},\\
& x_1x_2x_3x_6\},
\end{aligned}
$$

$$G = \{\ x_1^2-1,\ x_2^2-1,\ x_3^2-1,\ x_4^2-1,\ x_5^2-1,\ x_6^2-1,\ x_7^2-1,\ x_8^2-1,\ x_9^2-1,$$
$$x_{10}^2-1,$$
$$x_3x_5-x_1x_7,\ x_3x_7-x_1x_5,\ x_3x_8-x_2x_9,\ x_3x_9-x_2x_8,\ x_3x_{10}-x_5x_6,$$
$$x_4x_5-x_9x_{10},\ x_4x_6-x_2x_8,\ x_4x_8-x_2x_6,\ x_4x_9-x_3x_6,\ x_5x_7-x_1x_3,$$
$$x_5x_9-x_4x_{10},\ x_5x_{10}-x_3x_6,\ x_6x_7-x_1x_{10},\ x_6x_8-x_2x_4,\ x_6x_9-x_3x_4,$$
$$x_6x_{10}-x_1x_7,\ x_7x_{10}-x_1x_6,\ x_8x_9-x_2x_3,\ x_1x_2x_7x_8-x_4x_{10},$$
$$x_1x_3x_6-x_5x_6x_7,\ x_1x_4x_7-x_3x_4x_5,\ x_1x_4x_{10}-x_2x_7x_8,$$
$$x_1x_5x_8-x_3x_7x_8,\ x_1x_7x_9-x_3x_4x_{10},\ x_1x_8x_{10}-x_6x_7x_8,$$
$$x_2x_3x_6-x_3x_4x_8,\ x_2x_4x_7-x_6x_7x_8,\ x_2x_4x_{10}-x_1x_7x_8,$$
$$x_2x_5x_8-x_3x_4x_{10},\ x_2x_7x_9-x_3x_7x_8,\ x_2x_8x_{10}-x_3x_4x_5\}.$$

8.5 *Computing $N^\star = \sigma(N)$, $G^\star = \sigma(G)$, N' and G' for the code $\mathcal{C}^\star = \sigma(CF2)$*

Let $\sigma := (1, 10, 2, 7, 9, 6, 4, 3, 5)$ a permutation in cycle notation belonging to S_{10}. **GBLA_LC** contains functions which allows to compute, given a permutation σ, and a definition for a code $\mathcal{C}$ (a generator matrix or a check matrix), a definition of the code $\sigma(\mathcal{C})$ (see Section 8.3 of the Appendix for the definition of σ(CF2)). After having the definition of the code, then one can compute the set N' and G' for $\sigma(\mathcal{C})$.

$$N^\star = \{\ 1,\ x_{10},\ x_7,\ x_5,\ x_3,\ x_1,\ x_4,\ x_9,\ x_8,\ x_6,\ x_2,$$
$$x_7x_{10},\ x_5x_{10},\ x_3x_{10},\ x_1x_{10},\ x_4x_{10},\ x_9x_{10},\ x_8x_{10},\ x_6x_{10},\ x_2x_{10},\ x_5x_7,$$
$$x_3x_7,\ x_7x_9,\ x_7x_8,\ x_6x_7,\ x_2x_7,\ x_3x_5,\ x_5x_8,\ x_5x_6,\ x_2x_5,\ x_3x_6,\ x_2x_3,$$
$$x_1x_6,\ x_1x_2,\ x_4x_6,\ x_2x_4,\ x_6x_9,\ x_2x_9,\ x_6x_8,\ x_2x_8,\ x_2x_6,$$
$$x_5x_7x_{10},\ x_3x_7x_{10},\ x_7x_9x_{10},\ x_7x_8x_{10},\ x_6x_7x_{10},\ x_2x_7x_{10},\ x_3x_5x_{10},$$
$$x_5x_8x_{10},\ x_5x_6x_{10},\ x_2x_5x_{10},\ x_3x_6x_{10},\ x_2x_3x_{10},\ x_1x_6x_{10},\ x_1x_2x_{10},$$
$$x_4x_6x_{10},\ x_2x_4x_{10},\ x_6x_9x_{10},\ x_2x_9x_{10},\ x_6x_8x_{10},\ x_2x_8x_{10},\ x_2x_6x_{10},$$
$$x_5x_7x_8,\ x_1x_2x_6\}$$

$$G^\star = \{\ x_1^2-1,\ x_2^2-1,\ x_3^2-1,\ x_4^2-1,\ x_5^2-1,\ x_6^2-1,\ x_7^2-1,\ x_8^2-1,$$
$$x_9^2-1,\ x_{10}^2-1,$$

$x_1x_7 - x_4x_{10},\ x_4x_7 - x_1x_{10},\ x_1x_5 - x_9x_{10},\ x_4x_5 - x_7x_9,\ x_5x_9 - x_1x_{10},$

$x_1x_3 - x_8x_{10},\ x_3x_4 - x_7x_8,\ x_3x_9 - x_5x_8,\ x_3x_8 - x_1x_{10},\ x_1x_4 - x_7x_{10},$

$x_1x_9 - x_5x_{10},\ x_1x_8 - x_3x_{10},\ x_4x_9 - x_5x_7,\ x_4x_8 - x_3x_7,\ x_8x_9 - x_3x_5,$

$x_3x_5x_7 - x_2x_6,\ x_5x_6x_7 - x_2x_3,\ x_2x_5x_7 - x_3x_6,\ x_3x_6x_7 - x_2x_5,$

$x_2x_3x_7 - x_5x_6,\ x_6x_7x_9 - x_2x_8,\ x_2x_7x_9 - x_6x_8,\ x_6x_7x_8 - x_2x_9,$

$x_2x_7x_8 - x_6x_9,\ x_2x_6x_7 - x_3x_5,\ x_3x_5x_6 - x_2x_7,\ x_2x_3x_5 - x_6x_7,$

$x_5x_6x_8 - x_2x_4,\ x_2x_5x_8 - x_4x_6,\ x_2x_5x_6 - x_3x_7,\ x_2x_3x_6 - x_5x_7,$

$x_2x_4x_6 - x_5x_8,\ x_2x_6x_9 - x_7x_8,\ x_2x_6x_8 - x_7x_9,$

$x_5x_7x_8x_{10} - x_1x_2x_6,\ x_1x_2x_6x_{10} - x_5x_7x_8\}.$

8.5.1 N' and G' for $\sigma(CF2)$

$$\begin{aligned}N' = \{\ & 1,\ x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7,\ x_8,\ x_9,\ x_{10},\\
& x_1x_2,\ x_1x_3,\ x_1x_4,\ x_1x_5,\ x_1x_6,\ x_1x_7,\ x_1x_8,\ x_1x_9,\ x_1x_{10},\ x_2x_3,\ x_2x_4,\\
& x_2x_5,\ x_2x_6,\ x_2x_7,\ x_2x_8,\ x_2x_9,\ x_2x_{10},\ x_3x_4, x_3x_5,\ x_3x_6,\ x_3x_7,\ x_3x_9,\\
& x_4x_5,\ x_4x_6,\ x_4x_9,\ x_5x_6,\ x_6x_7,\ x_6x_8,\ x_6x_9,\ x_6x_{10},\\
& x_1x_2x_3,\ x_1x_2x_4,\ x_1x_2x_5,\ x_1x_2x_6,\ x_1x_2x_7,\ x_1x_2x_8,\ x_1x_2x_9,\ x_1x_2x_{10},\\
& x_1x_3x_4,\ x_1x_3x_5,\ x_1x_3x_6,\ x_1x_3x_7,\ x_1x_3x_9,\ x_1x_4x_5,\ x_1x_4x_6,\ x_1x_4x_9,\\
& x_1x_5x_6,\ x_1x_6x_7,\ x_1x_6x_8,\ x_1x_6x_9,\ x_1x_6x_{10},\ x_2x_6x_{10},\ x_3x_4x_5\},\end{aligned}$$

$$\begin{aligned}G' = \{\ & x_1^2-1,\ x_2^2-1,\ x_3^2-1,\ x_4^2-1,\ x_5^2-1,\ x_6^2-1,\ x_7^2-1,\ x_8^2-1,\\
& x_9^2-1,\ x_{10}^2-1,\\
& x_3x_8 - x_1x_{10},\ x_3x_{10} - x_1x_8,\ x_4x_7 - x_1x_{10},\ x_4x_8 - x_3x_7,\ x_4x_{10} - x_1x_7,\\
& x_5x_7 - x_4x_9,\ x_5x_8 - x_3x_9,\ x_5x_9 - x_1x_{10},\ x_5x_{10} - x_1x_9,\ x_7x_8 - x_3x_4,\\
& x_7x_9 - x_4x_5,\ x_7x_{10} - x_1x_4,\ x_8x_9 - x_3x_5,\ x_8x_{10} - x_1x_3,\ x_9x_{10} - x_1x_5,\\
& x_2x_3x_4 - x_6x_9,\ x_2x_3x_5 - x_6x_7,\ x_2x_3x_6 - x_4x_9,\ x_2x_3x_7 - x_5x_6,\\
& x_2x_3x_9 - x_4x_6,\ x_2x_4x_5 - x_6x_8,\ x_2x_4x_6 - x_3x_9,\ x_2x_4x_9 - x_3x_6,\\
& x_2x_5x_6 - x_3x_7,\ x_2x_6x_7 - x_3x_5,\ x_2x_6x_8 - x_4x_5,\ x_2x_6x_9 - x_3x_4,\\
& x_3x_4x_6 - x_2x_9,\ x_3x_4x_9 - x_2x_6,\ x_3x_5x_6 - x_2x_7,\ x_3x_6x_7 - x_2x_5,\end{aligned}$$

$x_3x_6x_9 - x_2x_4,\ x_4x_5x_6 - x_2x_8,\ x_4x_6x_9 - x_2x_3,$
$x_1x_2x_6x_{10} - x_3x_4x_5,\ x_1x_3x_4x_5 - x_2x_6x_{10}\}.$

8.5.2 $N^\star$ and $G^\star$ for $\sigma(CF2)$ ($\sigma = (1, 2, 6, 9, 10, 4, 5, 3, 7, 8) \in S_{10}$)

$$N^\star = \{\ 1,\ x_2,\ x_6,\ x_7,\ x_5,\ x_3,\ x_9,\ x_8,\ x_1,\ x_{10},\ x_4,$$
$$x_2x_6,\ x_2x_7,\ x_2x_5,\ x_2x_3,\ x_2x_9,\ x_2x_8,\ x_1x_2,\ x_2x_{10},\ x_2x_4,\ x_6x_7,$$
$$x_5x_6,\ x_6x_8,\ x_1x_6,\ x_6x_{10},\ x_4x_6,\ x_5x_7,\ x_1x_7,\ x_7x_{10},\ x_4x_7,\ x_5x_{10},$$
$$x_4x_5,\ x_3x_{10},\ x_3x_4,\ x_9x_{10},\ x_4x_9,\ x_8x_{10},\ x_4x_8,\ x_1x_{10},\ x_1x_4,\ x_4x_{10},$$
$$x_2x_6x_7,\ x_2x_5x_6,\ x_2x_6x_8,\ x_1x_2x_6,\ x_2x_6x_{10},\ x_2x_4x_6,\ x_2x_5x_7,$$
$$x_1x_2x_7,\ x_2x_7x_{10},\ x_2x_4x_7,\ x_2x_5x_{10},\ x_2x_4x_5,\ x_2x_3x_{10},\ x_2x_3x_4,$$
$$x_2x_9x_{10},\ x_2x_4x_9,\ x_2x_8x_{10},\ x_2x_4x_8,\ x_1x_2x_{10},\ x_1x_2x_4,\ x_2x_4x_{10},$$
$$x_1x_6x_7,\ x_3x_4x_{10}\},$$

$$G^\star = \{\ x_1^2 - 1,\ x_2^2 - 1,\ x_3^2 - 1,\ x_4^2 - 1,\ x_5^2 - 1,\ x_6^2 - 1,\ x_7^2 - 1,\ x_8^2 - 1,$$
$$x_9^2 - 1,\ x_{10}^2 - 1,$$
$$x_3x_6 - x_2x_9,\ x_6x_9 - x_2x_3,\ x_3x_7 - x_2x_8,\ x_7x_9 - x_6x_8,\ x_7x_8 - x_2x_3,$$
$$x_3x_5 - x_1x_2,\ x_5x_9 - x_1x_6,\ x_5x_8 - x_1x_7,\ x_1x_5 - x_2x_3,\ x_3x_9 - x_2x_6,$$
$$x_3x_8 - x_2x_7,\ x_1x_3 - x_2x_5,\ x_8x_9 - x_6x_7,\ x_1x_9 - x_5x_6,\ x_1x_8 - x_5x_7,$$
$$x_5x_6x_7 - x_4x_{10},\ x_6x_7x_{10} - x_4x_5,\ x_4x_6x_7 - x_5x_{10},\ x_5x_6x_{10} - x_4x_7,$$
$$x_4x_5x_6 - x_7x_{10},\ x_6x_8x_{10} - x_1x_4,\ x_4x_6x_8 - x_1x_{10},\ x_1x_6x_{10} - x_4x_8,$$
$$x_1x_4x_6 - x_8x_{10},\ x_4x_6x_{10} - x_5x_7,\ x_5x_7x_{10} - x_4x_6,\ x_4x_5x_7 - x_6x_{10},$$
$$x_1x_7x_{10} - x_4x_9,\ x_1x_4x_7 - x_9x_{10},\ x_4x_7x_{10} - x_5x_6,\ x_4x_5x_{10} - x_6x_7,$$
$$x_4x_9x_{10} - x_1x_7,\ x_4x_8x_{10} - x_1x_6,\ x_1x_4x_{10} - x_6x_8,$$
$$x_1x_2x_6x_7 - x_3x_4x_{10},\ x_2x_3x_4x_{10} - x_1x_6x_7\}.$$

8.5.3 N' and G' for $\sigma(CF2)$ ($\sigma = (1, 2, 6, 9, 10, 4, 5, 3, 7, 8) \in S_{10}$):

$$N' = \{\ 1,\ x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7,\ x_8,\ x_9,\ x_{10},$$
$$x_1x_2,\ x_1x_3,\ x_1x_4,\ x_1x_5,\ x_1x_6,\ x_1x_7,\ x_1x_8,\ x_1x_9,\ x_1x_{10},\ x_2x_4,\ x_2x_6,$$

$x_2x_7,\ x_2x_8,\ x_2x_9,\ x_2x_{10},\ x_3x_4,\ x_3x_{10},\ x_4x_5,\ x_4x_6,\ x_4x_7,\ x_4x_8,\ x_4x_9,$

$x_4x_{10},\ x_5x_{10},\ x_6x_7,\ x_6x_8,\ x_6x_{10},\ x_7x_{10},\ x_8x_{10},\ x_9x_{10},$

$x_1x_2x_4,\ x_1x_2x_6,\ x_1x_2x_7,\ x_1x_2x_8,\ x_1x_2x_9,\ x_1x_2x_{10},\ x_1x_3x_4,\ x_1x_3x_{10},$

$x_1x_4x_5,\ x_1x_5x_{10},\ x_1x_6x_7,\ x_2x_4x_6,\ x_2x_4x_7,\ x_2x_4x_8,\ x_2x_4x_9,\ x_2x_4x_{10},$

$x_2x_6x_7,\ x_2x_6x_8,\ x_2x_6x_{10},\ x_2x_7x_{10},\ x_2x_8x_{10},\ x_2x_9x_{10},\ x_3x_4x_{10}\},$

$$
\begin{aligned}
G' = \{\ & x_1^2-1,\ x_2^2-1,\ x_3^2-1,\ x_4^2-1,\ x_5^2-1,\ x_6^2-1,\ x_7^2-1,\ x_8^2-1,\\
& x_9^2-1,\ x_{10}^2-1,\\
& x_2x_3-x_1x_5,\ x_2x_5-x_1x_3,\ x_3x_5-x_1x_2,\ x_3x_6-x_2x_9,\ x_3x_7-x_2x_8,\\
& x_3x_8-x_2x_7,\ x_3x_9-x_2x_6,\ x_5x_6-x_1x_9,\ x_5x_7-x_1x_8,\ x_5x_8-x_1x_7,\\
& x_5x_9-x_1x_6,\ x_6x_9-x_1x_5,\ x_7x_8-x_1x_5,\ x_7x_9-x_6x_8,\ x_8x_9-x_6x_7,\\
& x_1x_4x_6-x_8x_{10},\ x_1x_4x_7-x_9x_{10},\ x_1x_4x_8-x_6x_{10},\ x_1x_4x_9-x_7x_{10},\\
& x_1x_4x_{10}-x_6x_8,\ x_1x_6x_8-x_4x_{10},\ x_1x_6x_{10}-x_4x_8,\ x_1x_7x_{10}-x_4x_9,\\
& x_1x_8x_{10}-x_4x_6,\ x_1x_9x_{10}-x_4x_7,\ x_4x_5x_{10}-x_6x_7,\ x_4x_6x_7-x_5x_{10},\\
& x_4x_6x_8-x_1x_{10},\ x_4x_6x_{10}-x_1x_8,\ x_4x_7x_{10}-x_1x_9,\ x_4x_8x_{10}-x_1x_6,\\
& x_4x_9x_{10}-x_1x_7,\ x_6x_7x_{10}-x_4x_5,\ x_6x_8x_{10}-x_1x_4,\\
& x_1x_2x_6x_7-x_3x_4x_{10}\}.
\end{aligned}
$$

8.6 Proof of iii $\Rightarrow$ i in Theorem 28

Let $c' \in \mathcal{C}'$ and $w \in [X]$ the standard word such that $\varphi(w) = c$. Assume w is reduced to w' modulo G', *iii* implies that w is reduced to $w^\star$ modulo $G^\star$ and $\varphi(w') - \varphi(w^\star) \in \mathcal{C}^\star$. Let us prove that.

$w \xrightarrow{*} w' \Rightarrow w = \sum_{i=1}^{k} g_i' t_i + w'$, where each $g_i' \in G'$. *iii* implies that $g_i' \in I(G^\star)$ for all $i = 1, \ldots, k$, i.e. $w - w' \in G^\star$. We have that $w - w^\star \in I(G^\star)$ (because w is reduced to $w^\star$ modulo $G^\star$). Therefore, $w' - w^\star \in I(G^\star)$, consequently,

$$\varphi(w') - \varphi(w^\star) \in C^\star. \tag{5}$$

The equality above implies that the canonical forms of w modulo G' and $G^\star$ have the same syndrome. The canonical form of w modulo G' is 1 (because $\varphi(w) \in \mathcal{C}'$), then by the equality (5), it has to be also 1 w.r.t. $\sigma(G)$ (1 is the only canonical form corresponding to the codewords).

References

Y. Azze-Bauta. *Construcción y Aplicación de Técnicas de Bases de Gröbner en la Teoría de Códigos Lineales.* Diploma thesis, Universidad de Oriente, Santiago de Cuba, Cuba, 2003.

M. A. Borges-Trenard and M. Borges-Quintana. Gröbner bases property on elimination ideal in the noncommutative case. In *Gröbner bases and applications (Linz, 1998)*, p. 323–337. Cambridge Univ. Press, Cambridge, 1998.

M. Borges-Quintana, M. A. Borges-Trenard, and F. Winkler. An application of the FGLM techniques to linear codes. In *Fourth Italian-Latin American Conference on Applied and Industrial Mathematics (Havana, 2001)*, p. 280–286. Inst. Cybern. Math. Phys., Havana, 2001.

M. A. Borges-Trenard, M. Borges-Quintana, and T. Mora. Computing Gröbner bases by FGLM techniques in a non-commutative setting. *J. Symbolic Comput.*, 30(4), p. 429–449, 2000.

M. Borges-Quintana, F. Winkler, and M. Borges-Trenard. FGLM Techniques Applied to Linear Codes – An Algorithm for Decoding Linear Codes. Techn. Rep. *RISC-Linz, RISC - 00-14*, J. Kepler Univ., Linz, Austria, 2000.

B. Buchberger, H.M. Möller. The construction of Multivariate Polynomials with Preassigned Zeros. In: EUROCAM'82, Marseille. Springer Verlag, Berlin Heidelberg, New York, *Lecture Notes in Computer Science 144*, p. 24-31, 1982.

J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symbolic Comput.*, 16(4), p. 329–344, 1993.

The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.3*; 2002, (`http://www.gap-system.org`).

M. G. Marinari, H. M. Möller, T. Mora. Gröbner Bases of Ideals Defined by Functionals with an Application to Ideals of Projective Points. *Applicable Algebra in Engeneering, Communication and Computing*, vol. 4, p. 103-145, 1993.

Erez Petrank and Ron M. Roth. Is code equivalence easy to decide? *IEEE Trans. Inform. Theory*, 43(5), p. 1602–1604, 1997.

Nicolas Sendrier. Finding the permutation between equivalent linear codes: the support splitting algorithm. *IEEE Trans. Inform. Theory*, 46(4), p. 1193–1203, 2000.

Patric R.J. Östegar, Classifying subespaces of Hamming Spaces. *Designs, Codes and Cryptography*, 27, p. 297–305, 2002.

W. W. Peterson, E. J. Jr. Weldon. *Error-Correcting Codes (2nd ed.).* MIT Press, Cambridge, Massachusetts, London. England, 1972.

F.Winkler. *Polynomial Algorithms in Computer Algebra.* Springer-Verlag, Viena, New York, 1996.