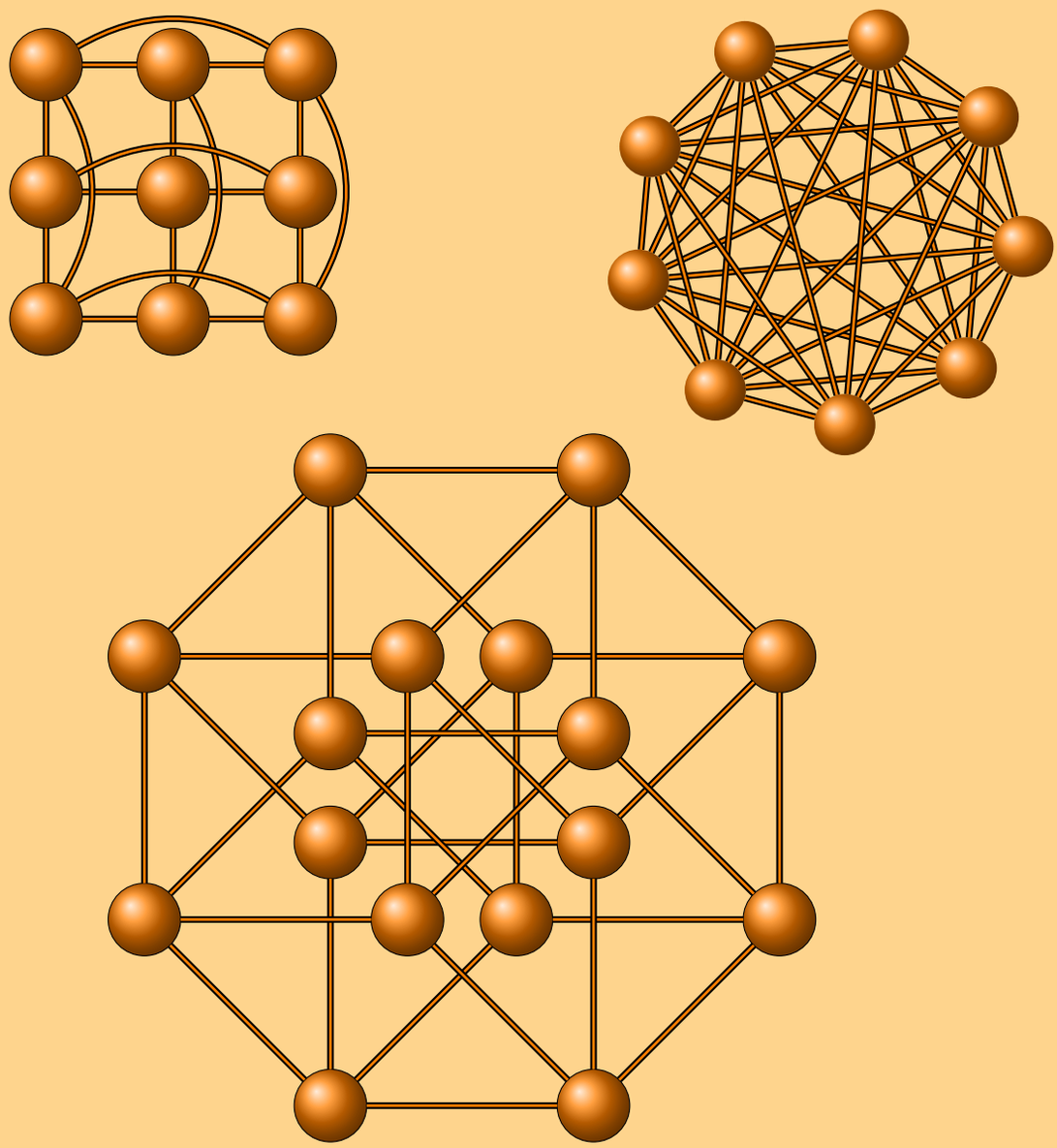
Irene Márquez-Corbella

Combinatorial Commutative Algebra Approach to Complete Decoding



Institute of Mathematics, University of Valladolid

April 2013

Universidad de Valladolid
Departamento de Álgebra, Análisis Matemático,
Geometría y Topología

A Combinatorial Commutative Algebra Approach to Complete Decoding

Presentada por *Irene Márquez-Corbella* para optar al grado de doctora por la Universidad de Valladolid

DIRIGIDA POR:

Dr. D. Antonio Campillo López
Dr. D. Edgar Martínez-Moro

Abril, 2013

Labor omnia vincit

Contents

Índice

Acknowledgments

En primer lugar quiero expresar mis más sinceros agradecimientos a mis directores, Dr. Antonio Campillo López y Dr. Edgar Martínez-Moro, por todo su apoyo, confianza en mi trabajo y sabios consejos. Han sido un referente no solamente para el desarrollo de esta tesis sino en mi formación, ofreciéndome una visión más amplia del mundo académico y permitiéndome descubrir cuánto me motiva.

I would also like to express my sincere gratitude to Prof. Ruud Pellikaan for his fundamental role in my work during the last four years. Prof. Ruud Pellikaan provided me with the guidance, the assitance and the expertise that I needed during my doctoral time. It has been a pleasure to work with him and I hope we will continue our collaboration in future projects.

Al Ministerio de Educación que ha financiado esta tesis a través de una beca de Formación de Profesorado Universitario (FPU) con referencia AP2008-01598. Y al Ministerio de Ciencia e Innovación por su contribución económica a través del proyecto *Geometría algebraica de las singularidades, computación e información* con referencia MTM2007-64704.

Me gustaría dar las gracias de manera especial a todos mis profesores de la ULL por su inestimable ayuda y, lo más importante, por enseñarme que con trabajo todo es posible.

A toda mi familia, a mis padres, a mi hermana y a Franchy, por su paciencia, por su apoyo incondicional y por su ayuda al equilibrar la balanza. Gracias porque nunca me he sentido lejos, por perder el miedo a viajar y por buscar conmigo soluciones a las adversidades.

Y por último, aunque no por ello menos importante, a mis amigos, que han sabido disculpar mis ausencias y siempre han tenido una palabra de ánimo. Si de algo puedo presumir es de haber hecho grandes amigos en todas mis paradas, soy muy afortunada.

*L*ast but not least I want to thank my friends who have been able to excuse all my absences and always had a word of encouragement. If I can presume of something is to have made great friends in all my stops, I'm very lucky.

*P*our terminer, mais non moins important, à mes amis qui ont excusé toutes mes absences et qui ont toujours trouver un mot d'encouragement. S'il est une chose que je peux affirmer, c'est d'avoir noué de belles amitiés partout où je suis allée, j'ai beaucoup de chance.

*E*d infine, ma non per importanza, ai miei amici, i quali sono sempre stati comprensivi quando non ho potuto essere presente e mi hanno sempre incoraggiata nei momenti piú difficili. Se posso vantarmi di qualcosa, è di aver stretto grandi amicizie durante questo percorso di studio e di questo mi ritengo molto fortunata.

Tot slot, maar zeker niet onbelangrijk, voor mijn vrienden, die in staat zijn geweest mij te vergeven voor mijn afwezigheid en altijd een bemoedigend woord hadden. Ik prijs mezelf gelukkig dat ik goede vrienden heb gemaakt tijdens al mijn verblijven.

Και τέλος, αλλά όχι λιγότερο σημαντικό, στους φίλους μου, που συγχωρούσαν τις απουσίες μου και ήξεραν πάντα να με ενθαρρύνουν. Αν μπορώ να πώ κάτι με βεβαιότητα είναι ότι έκανα πολύ καλούς φίλους σε όλες μου τις στάσεις, είμαι πολύ τυχερή.

و اخيران، مع ان مهم جدا، أنا أريد أن أشكر أصدقاءي.فكلهم فهموا غيابي ودعموني دائما. إذا يوجد شيء أنا سعيدة بها، هو أنني عندي أصدقاء في كل الأماكن التي سكنت فها.بالحقيقة، انا محظوظة جدا

Und nicht zuletzt danke ich meinen Freunden, die mir mein Fehlen immer verziehen und mich stets ermutigt haben. Wenn ich mit Recht behaupten kann, dass ich während aller meiner Aufenthalte wunderbare Freunde gefunden habe, so kann ich mich sehr glücklich schätzen.

Summary (Spanish)

Esta tesis pretende explorar el nexo de unión que existe entre la estructura algebraica de un código lineal y el proceso de descodificación completa. Sabemos que el proceso de descodificación completa para códigos lineales arbitrarios es NP-completo [6], incluso si se admite preprocesamiento de los datos [23]. Nuestro objetivo es realizar un análisis algebraico del proceso de la descodificación, para ello asociamos diferentes estructuras matemáticas a ciertas familias de códigos. Desde el punto de vista computacional, nuestra descripción no proporciona un algoritmo eficiente pues nos enfrentamos a un problema de naturaleza NP. Sin embargo, proponemos algoritmos alternativos y nuevas técnicas que permiten relajar las condiciones del problema reduciendo los recursos de espacio y tiempo necesarios para manejar dicha estructura algebraica.

A partir de ahora denotamos por $\mathbb{Z}$, $\mathbb{Z}_s$, $\mathbb{K}$ y $\mathbb{F}_q$, respectivamente, al anillo de enteros, al anillo de enteros módulo s, a un cuerpo arbitrario y a un cuerpo finito con q elementos donde q es potencia de un primo.

La descodificación completa tiene diversas aplicaciones no sólo en teoría de códigos sino también en distintas áreas asociadas a la seguridad de la información:

- La **Teoría de Códigos** tiene como objetivo enviar un mensaje con la mayor eficiencia y verosimilitud posible. En este campo, una de las principales aplicaciones de nuestra investigación consiste en definir el conjunto de palabras de soporte minimal de un código lineal. Este problema ha sido resuelto para ciertas familias de códigos lineales, como los códigos Hamming o los códigos Reed-Muller binarios de segundo orden. También ha habido intentos de caracterizar este subconjunto de palabras en los códigos BCH y en los códigos Reed-Muller de orden r, véanse [19] y las referencias que se proporcionan en este artículo. En general, se trata de un problema de complejidad NP pues está relacionado con la descodificación completa.

- La **Criptografía** posee un objetivo distinto y, en cierto sentido, opuesto, ya que pretende hacer confusa la información, de forma que, si un mensaje es interceptado, sea imposible comprenderlo. En particular, nuestra investigación tiene aplicaciones en esquemas para compartir secretos. Un esquema para compartir secretos es un protocolo criptográfico que, como su nombre indica, divide un determinado secreto en fragmentos repartidos entre los participantes,

de forma que sólo ciertos conjuntos autorizados lo pueden reconstruir. A la familia de los conjuntos de participantes que nos permite recuperar el secreto se le denomina estructura de acceso. Existen diferentes formas de definir esquemas para compartir secretos utilizando códigos lineales (véanse [31, 81, 83, 84, 96]). Las palabras de soporte minimal de un código determinan de forma completa la estructura de acceso de estos esquemas.

- La **Esteganografía** se define como la ciencia de ocultar la información de tal forma que, aparte del emisor y receptor, nadie pueda detectar la existencia del mensaje. Con el fin de encubrir un mensaje necesitamos principalmente dos factores. Por una parte, la elección de un medio inocuo $\mathbf{x}$, como puede ser una imagen digital; y, por otra, el diseño de algoritmos que nos permitan incrustar nuestro mensaje $\mathbf{m}$ en el medio elegido (función de modificación), sin que el medio apenas se altere y permita luego recuperar el contenido oculto en la misma (función de recuperación).

  En 1998, Crandall [36] propone crear modelos esteganográficos utilizando códigos lineales. La idea es modificar el medio de tal forma que el síndrome del medio alterado coincide con el mensaje que queremos enviar. Dado un código lineal $\mathscr{C}$ de parámetros $[n,k]$ definido en el cuerpo $\mathbb{F}_q$, del que conocemos un algoritmo de descodificación completo por mínimas distancias, y sea H una matriz de control de dicho código, consideremos que $\mathbf{x} \in \mathbb{F}_q^n$ es un vector extraído del medio y que $\mathbf{m} \in \mathbb{F}_q^k$ es el mensaje que queremos enviar. La función de modificación consiste en transformar el vector $\mathbf{x}$ en un vector $\mathbf{y} \in \mathbb{F}_q^n$ tal que $H\mathbf{y} = \mathbf{m}$, dicho con otras palabras, introducimos errores en el medio con la característica de que $\mathbf{y}$ sea una de las palabras con síndrome $\mathbf{m}$ más cercanas al vector $\mathbf{x}$. Mientras que la función de recuperación consiste en calcular el síndrome del vector $\mathbf{y}$, i.e. $H\mathbf{y} = \mathbf{m}$.

  Observemos que la función de descodificación se utiliza de diferente forma en Teoría de Códigos y en Esteganografía. En la primera el objetivo es corregir errores, por lo que es necesario garantizar que la palabra enviada y la que se obtiene del proceso de descodificación coincidan; sin embargo, en Esteganografía el objetivo es introducir errores (los menos posibles), así que la unicidad de la palabra más cercana no es importante, si existe más de una opción elegimos una de ellas de forma aleatoria. De ahí que los métodos de descodificación completa desempeñen un papel importante en esta ciencia (véanse [89] y las referencias que se dan en este texto).

Iniciamos la tesis con un capítulo de preliminares, una breve introducción a los temas de investigación, indispensable para comprender los resultados que se exponen en los siguientes capítulos. Incluimos en este apartado una breve descripción de la Teoría de Códigos, herramientas algebraicas como las Bases de Gröbner y las Bases del Borde, así como una sección que pone de manifiesto la relación entre Códigos lineales y Matroides que se pueden representar en un cuerpo finito.

El Capítulo 2 está dedicado en exclusiva a códigos binarios, es decir a subespacios vectoriales de $\mathbb{F}_2^n$. El objetivo es profundizar y generalizar los trabajos previos [11, 13, 14, 15, 16] sobre la representación de Gröbner de los códigos binarios. Demostramos que se trata de una estructura esencial en el proceso de descodificación pues resuelve el problema subyacente del cálculo del conjunto de líderes de un código.

Denotaremos por $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ a la base canónica de $\mathbb{F}_2^n$. Una representación de Gröbner de un código binario $\mathscr{C}$ es un par $(\mathscr{N}, \phi)$ donde $\mathscr{N}$ está formado por un conjunto de representantes de las clases de equivalencia de $\mathbb{F}_2^n/\mathscr{C}$ y ϕ es una aplicación en el dominio $\mathscr{N} \times \{\mathbf{e}_i\}_{i=1}^n$, de tal forma que la imagen de cada par $(\mathbf{n}, \mathbf{e}_i)$ es el representante en $\mathscr{N}$ de la clase $\mathbf{n} + \mathbf{e}_i$. La palabra *Gröbner* no es casual. De hecho, si consideramos un orden $\prec$ compatible con el peso de Hamming, el ideal binomial asociado a un código $I(\mathscr{C}) = \left\langle \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \mathbf{a} - \mathbf{b} \in \mathscr{C} \right\rangle \subseteq \mathbb{K}[\mathbf{X}]$ y calculamos $\mathscr{G}$ una base de Gröbner reducida de $I(\mathscr{C})$ respecto de $\prec$, entonces, los vectores asociados a las formas normales de $\mathscr{G}$ representan el conjunto $\mathscr{N}$, mientras que la función ϕ puede verse como una tabla de multiplicación entre las variables X_i y las formas normales en el anillo $\mathbb{K}[\mathbf{X}]/I(\mathscr{C})$. Por lo tanto, la representación de Gröbner de un código se puede obtener con una adaptación del algoritmo FGLM, tal y como se muestra en [11]. En el Capítulo 2 se presenta una modificación de este algoritmo que nos permite no sólo obtener la representación de Gröbner de un código binario, sino también nos aporta el conjunto de todos los vectores líderes e, incluso, un conjunto de prueba para nuestro código. Además, sin coste adicional, el algoritmo se puede adaptar para obtener otros parámetros del código, como por ejemplo el radio de Newton y el radio de recubrimiento.

Es sabido que un *conjunto de prueba* de un código $\mathscr{C}$ es un subconjunto de palabras $\mathscr{T}_{\mathscr{C}} \subseteq \mathscr{C}$, de forma que cualquier palabra recibida $\mathbf{y}$, o bien es un elemento líder de $\mathscr{C}$, o bien existe un elemento $\mathbf{t} \in \mathscr{T}_{\mathscr{C}}$ tal que $\mathrm{w}_H(\mathbf{y} - \mathbf{t}) \leq \mathrm{w}_H(\mathbf{t})$, donde $\mathrm{w}_H(\cdot)$ denota el peso Hamming de un vector.

Al final del segundo capítulo presentamos un enfoque unificado, a través de la representación de Gröbner de un código, de los dos algoritmos de descodificación por gradientes que se conocen para códigos binarios, y que fueron presentados de forma independiente por Liebler [71] y por Ashikhmin y Barg [2]. De estos algoritmos se decía que tenían diferente naturaleza, sin embargo demostramos que se trata de dos formas de entender la reducción de un monomio módulo el ideal asociado a un código binario.

En el Capítulo 3 trabajamos con códigos modulares definidos en $\mathbb{Z}_q$, es decir con subgrupos abelianos de $\left(\mathbb{Z}_q^n, +\right)$. A lo largo de este capítulo utilizamos las siguientes aplicaciones de *cambio de característica*:

$$\blacktriangledown : \ \mathbb{Z}^s \longrightarrow \mathbb{Z}_q^s \quad \text{y} \quad \blacktriangle \ \mathbb{Z}_q^s \longrightarrow \mathbb{Z}^s$$

donde s se determina por el contexto. Ambas aplicaciones actúan coordenada a coordenada sobre vectores y matrices. La aplicación $\blacktriangledown$ se corresponde con la reducción módulo q mientras que $\blacktriangle$ sustituye la clase de los elementos $0, 1, \ldots, q-1$ por el

mismo símbolo considerado como un entero. De esta forma podemos trabajar con monomios con exponentes en $\mathbb{Z}_q$ como si fueran monomios con exponentes enteros. Consideramos una matriz $A \in \mathbb{Z}_q^{m\times n}$ (llamada matriz de coeficientes), el vector $\mathbf{b} \in \mathbb{Z}_q^m$ y el vector de costes $\mathbf{w} \in \mathbb{R}^n$. El *problema de programación lineal entero modular* asociado, que denotamos por $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$, se define como un problema de programación lineal en aritmética modular cuyo objetivo es encontrar un vector $\mathbf{u} \in \mathbb{Z}_q^n$ que minimice la función de costos $\mathbf{w} \cdot \blacktriangle\mathbf{u}$ y que verifique que $A\mathbf{u}^T = \mathbf{b} \mod q$. Si expresamos este problema de forma esquemática se corresponde a:

$$\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b}) = \begin{cases} \text{Minimizar: } \mathbf{w} \cdot \blacktriangle\mathbf{u} \\ \text{Sujeto a: } \begin{cases} A\mathbf{u}^T \equiv \mathbf{b} \mod q \\ \mathbf{u} \in \mathbb{Z}_q^n \end{cases} \end{cases}$$

Es bien conocido que, en el caso binario, el método de Descodificación por Mínima Distancia (MDD) se puede ver como un problema de programación lineal entero modular. El método MDD (uno de los más utilizados como algoritmo de descodificación) consiste en, recibido un vector, encontrar una palabra del código que minimice la distancia de Hamming con la palabra recibida. En otros términos, resolver el problema modular $\mathrm{IP}_{H,\mathbf{w},2}$ es equivalente a la descodificación completa de la palabra $\mathbf{b}$ en $\mathscr{C}$, donde H es una matriz de paridad de nuestro código binario $\mathscr{C}$ y $\mathbf{w}$ es el vector $(1,\ldots,1)$.

En 2002 Ikegami y Kaji [60], adaptando las ideas de Conti y Traverso [33], propusieron un algoritmo eficiente para resolver el problema anterior utilizando Bases de Gröbner. En términos generales, el algoritmo define un ideal $I(A)$ asociado a la familia de problemas modulares cuya matriz de coeficientes es A, es decir $\mathrm{IP}_{A,\mathbf{w},q}$, de forma que los binomios de la Base de Gröbner reducida de dicho ideal respecto de un orden adecuado (compatible con el vector peso $\mathbf{w}$) proporcionan un conjunto de prueba para la familia de problemas $\mathrm{IP}_{A,\mathbf{w},q}$.

Un *conjunto de prueba* para un problema entero modular $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ es un subconjunto de $\mathbb{Z}_q$ con la propiedad de que, para cada solución $\mathbf{u}$ no óptima del problema, existe un elemento $\mathbf{t}$ en dicho conjunto, tal que $\mathbf{u}+\mathbf{t}$ nos aporta una nueva solución con mejor valor en la función objetivo del problema.

En resumen, estos conceptos en el caso binario nos describen un método para calcular un conjunto de prueba del código asociado. Además, resulta que la Base de Graver del ideal $I(A)$ proporciona un *conjunto de prueba universal* que, al interpretarlo en el correspondiente código modular, conduce a un conjunto que contiene el conjunto de palabras de soporte minimal.

Este resultado es el eje principal del Capítulo 3 en el que, además, se presentan algunas soluciones para mejorar la eficacia del algoritmo propuesto. Por una parte, se propone utilizar la filosofía expuesta por Di Biase-Urbanke en [38] para reducir el número de variables. La reducción que se formula es una generalización al caso modular no binario del artículo [11]. Además, como debemos calcular una Base de Gröbner de un ideal del que conocemos sus generadores, se recomienda utilizar técnicas FGLM. La complejidad de este algoritmo es $\mathscr{O}\left(n^2 q^{n-k}\right)$, donde k es el número

de filas de una matriz generatriz del código y n es la longitud del código, es decir, el número de variables involucradas en nuestro ideal. El procedimiento que se propone posee las siguientes ventajas:

- Todos los pasos se pueden llevar a cabo utilizando eliminación gaussiana sobre una matriz con elementos en $\mathbb{Z}_q$.
- El problema del crecimiento de los coeficientes no tiene que ser considerado pues, como toda la información del código se encuentra en los exponentes de los binomios que definen el ideal, podemos considerar como cuerpo base el cuerpo binario $\mathbb{F}_2$.
- El problema de crecimiento del grado total de los binomios tampoco tiene que ser tenido en cuenta, pues el grado total de los binomios involucrados está acotado por $n \times q$.

Otra forma de reducir la complejidad de los algoritmos definidos en el Capítulo 3 es utilizar la descomposición de un código como códigos de menor longitud. Para ello hacemos uso de la teoría de descomposición de matroides representables. Buscamos un procedimiento que:

1. Encuentre una descomposición especial, que llamamos m-gluing, de cualquier código como códigos de menor longitud.
2. Calcule el conjunto de palabras de soporte minimal de cada uno de los códigos que aparecen en esta descomposición.
3. Y deduzca, finalmente, el conjunto de palabras de soporte minimal del código original utilizando los conjuntos obtenidos en el paso anterior.

Observemos que, en el segundo apartado de este procedimiento, todos los cálculos son independientes. Por lo tanto resulta adecuado utilizar computación paralela en cada componente, reduciendo el tiempo de ejecución del algoritmo. En el caso binario, un proceso similar se puede definir para calcular conjuntos de prueba.

En el Capítulo 4 abordamos los problemas tratados en los capítulos anteriores para un código lineal arbitrario. Para ello es fundamental una nueva definición de nuestras funciones de cambio de característica planteadas ahora sobre los cuerpos finitos $\mathbb{F}_q$. Considerando que α es un elemento primitivo de dicho cuerpo y que $\left\{\mathbf{e}_1, \ldots, \mathbf{e}_{q-1}\right\}$ es una base estándar del anillo de enteros $\mathbb{Z}^{q-1}$, definimos:

$$\nabla: \quad \{0,1\}^{q-1} \quad \longrightarrow \quad \mathbb{F}_q \qquad \text{y} \qquad \Delta: \quad \mathbb{F}_q^n \quad \longrightarrow \quad \{0,1\}^{q-1}$$

La aplicación Δ reemplaza la clase del elemento $\mathbf{a} = \alpha^j \in \mathbb{F}_q^*$ por el vector $\mathbf{e}_j$ y el elemento cero por el vector cero en $\mathbb{Z}^{q-1}$, mientras que la aplicación ∇ recupera el elemento $j_1\alpha + \ldots j_{q-1}\alpha^{q-1}$ de $\mathbb{F}_q$ de la $(q-1)$-tupla de elementos binarios $\left(j_1, \ldots, j_{q-1}\right)$.

Supongamos que $\mathbf{X}$ denota n variables vectores $X_1, \ldots, X_n$ y que cada variable $\mathbf{X}_i$ se puede descomponer en $q-1$ componentes: $x_{i1}, \ldots, x_{iq-1}$. Dada una n-tupla $\mathbf{a} = (a_1, \ldots, a_n)$ de elementos en $\mathbb{F}_q$, fijamos la siguiente notación:

$$\mathbf{X}^{\mathbf{a}} \quad = \quad X_1^{a_1} \cdots X_n^{a_n} \quad = \quad \left(x_{11} \cdots x_{1q-1}\right)^{\Delta a_1} \cdots \left(x_{n1} \cdots x_{nq-1}\right)^{\Delta a_n}.$$

Esta relación nos permite trabajar con monomios de exponentes enteros, aunque sus exponentes sean elementos de $\mathbb{F}_q$. Además, con esta representación todo elemento $\mathbf{a} \in \mathbb{F}_q^n$ verifica que $\deg(\mathbf{X}^{\mathbf{a}}) = \mathrm{w}_H(\Delta\mathbf{a})$. Es decir, un orden compatible con el peso en $\mathbb{F}_q^n$ se puede ver como un orden compatible con el grado en $\mathbb{K}[\mathbf{X}]$. Esta es la idea clave de la generalización que realizamos en el Capítulo 4 y que pasamos a detallar en las siguientes líneas. Sea $\mathscr{C}$ un código lineal cualquiera definido en el cuerpo $\mathbb{F}_q$:

1. Probamos que el ideal binomial asociado a $\mathscr{C}$, que denotamos por $I_+(\mathscr{C})$, está generado por un conjunto finito de binomios definidos por las filas de una matriz generatriz de nuestro código y las relaciones dadas por la tabla aditiva del cuerpo $\mathbb{F}_q$.

2. Calculamos la representación de Gröbner de $\mathscr{C}$.

3. Mostramos que una base de Gröbner reducida del ideal $I_+(\mathscr{C})$ respecto de un orden compatible con el grado nos define un conjunto de prueba de nuestro código. Es decir, a través de la representación de Gröbner de un código lineal podemos definir dos procesos de reducción que nos permiten definir dos algoritmos de descodificación por gradiente similares a los dos algoritmos conocidos para códigos binarios.

4. Adaptamos el algoritmo FGLM para calcular una base de Gröbner reducida del ideal $I_+(\mathscr{C})$ respecto de un orden compatible con el grado. Se trata de una generalización de las técnicas FGLM presentadas en [11] para el caso binario. Además hacemos un estudio exhaustivo de la complejidad del algoritmo propuesto.

5. Proporcionamos un algoritmo para calcular las palabras de soporte minimal del código $\mathscr{C}$.

Aparte de esto, los resultados obtenidos en el Capítulo 4 se pueden generalizar para otra clase de códigos, como los códigos modulares, los códigos definidos en múltiples alfabetos o los códigos aditivos. Los códigos modulares ya fueron tratados en el capítulo anterior pero, con esta nueva aproximación, podemos calcular no sólo el conjunto de palabras de soporte minimal sino también un conjunto de prueba.

El último capítulo de la tesis ofrece un nuevo enfoque utilizando técnicas de semigrupos. El objetivo es asignar un conjunto de generadores al semigrupo relacionado con ciertas familias de códigos. La construcción que realizamos establece una fuerte conexión entre geometría tórica y códigos correctores de errores. Constituye, por tanto, una forma de aplicar resultados de la Teoría de Semigrupos para resolver

problemas de la Teoría de la Información. Estudiaremos con detalle diferentes representaciones del semigrupo asociado a códigos modulares y lineales, para concluir que la elección de representaciones digitales es la más adecuada para llevar a cabo la descodificación completa. Entendemos por representaciones digitales un conjunto de generadores $\{\mathbf{n}_1, \ldots, \mathbf{n}_r\}$ de nuestro semigrupo S que nos permite escribir cada palabra $\mathbf{m} \in S$ de la forma

$$\sum_{i=1}^{r} a_i \mathbf{n}_i \text{ con } a_1, \ldots, a_r \in \{0, 1\} \subseteq \mathbb{N}.$$

La ventaja de utilizar esta representación es la equivalencia que surge entre el grado de los monomios $\mathbf{X}^\gamma$ del ideal del semigrupo $I(S)$ y el peso de Hamming del vector γ, siguiendo los resultados del Capítulo 4.

La investigación que se presenta en esta tesis ha dado lugar a las siguientes publicaciones científico-técnicas (con peer-review):

[75] I. Márquez-Corbella and E. Martínez-Moro. *Decomposition of Modular Codes for Computing Test Sets and Graver Basis*. Mathematics in Computer Science, 6(2), pp. 147-165, 2012.

[74] I. Márquez-Corbella and E. Martínez-Moro. *Algebraic structure of the minimal support codewords set of some linear codes*. Adv. Math. Commun., 5(2), pp. 233-244, 2011.

[18] M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro, *An algebraic view to gradient descent decoding*. Information Theory Workshop (ITW), 2010 IEEE, Dublin (Ireland), August 30 - September 3, 2010, pp.1-4.

Otras publicaciones derivadas de colaboraciones mantenidas durante la ejecución de esta tesis y que han sido relevantes para la misma son:

[76] I. Márquez-Corbella and E. Martínez-Moro. *An Introduction to LDPC codes*. In Algebraic Geometry Modeling in Information Theory, Series on Coding Theory and Cryptology, pp. 129-166. ISBN: 978-981-4335-75-1, 2013.

[78] I. Márquez-Corbella, E. Martínez-Moro and G. R. Pellikaan. *On the unique representation of very strong algebraic geometry codes*. Designs, Codes and Cryptography, pp. 1-16, 2012.

[77] I. Márquez-Corbella, E. Martínez-Moro and G. R. Pellikaan. *The non-gap sequence of a subcode of a generalized Reed-Solomon code*. Designs, Codes and Cryptography, pp. 1-17, 2012.

[37] B. Debraize and I. Márquez-Corbella. *Fault Analysis of the Stream Cipher Snow 3G*. Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), Lausanne (Switzerland), September 6, 2009, pp. 103-110.

Otras publicaciones enviadas para su consideración son:

[79] I. Márquez-Corbella, E. Martínez-Moro, G. R. Pellikaan and D. Ruano. *Computational Aspects of Retrieving a Representation of an Algebraic Geometry Code*.

[81] I. Márquez-Corbella, E. Martínez-Moro and E. Suárez-Canedo. *On the Composition of Secret Sharing Schemes Related to Codes*.

[12] M. B. Quintana, M. A. B. Trenard, I. Márquez-Corbella and E. Martínez-Moro. *Computing coset leaders and leader codewords of binary codes*.

[82] I. Márquez-Corbella, E. Martínez-Moro and E. Suárez-Canedo. *On the ideal associated to a linear code.*

[80] I. Márquez-Corbella and R. Pellikaan. *Error-correcting pairs for public-key cryptosystem*.

Summary (English)

This thesis aims to explore the bridge between the algebraic structure of a linear code and the complete decoding process. Complete decoding is an NP-problem [6] for an arbitrary linear code, even if preprocessing is allowed [23]. Our goal is to make an algebraic analysis of the decoding process. To that end, we associate different mathematics structures to certain families of codes. From a computational point of view, our description does not provide an efficient algorithm since our problem is NP-complete. However we propose alternative algorithms and new techniques to reduce the space and time needed to handle such problem.

By $\mathbb{Z}$, $\mathbb{Z}_s$, $\mathbb{K}$ and $\mathbb{F}_q$ where q is a prime power, we denote the ring of integers, the ring of integers modulo s, an arbitrary field and a finite field with q elements, respectively.

Complete decoding has many applications not only in Coding Theory but also in other areas of Information Security:

- The goal of **Coding Theory** is to efficiently transfer reliable information. One of the main applications of complete decoding process is that it describes the set of codewords of minimal support. This problem has been solved for q-ary Hamming codes and for the second order binary Reed-Muller codes. There have been several attempts to characterize this set for other classes of codes, such as BCH codes and the r^{th}-order binary Reed-Muller code, see [19] and the references therein. However, in general it is difficult to describe them since it is related with the problem that concerns this thesis (complete decoding).

- **Cryptography**, is the science of secure communication in such a way that if a message is intercepted, it is not understood. In particular, our research has applications in secret sharing schemes which is a cryptographic protocol for distributing a secret amongst a group of participants, such that only specified subsets are able to determine the secret from joining the shares they hold. The subset of participants which are able to reconstruct the secret is called the access structure of the scheme. There are several ways to obtain a secret sharing scheme using a linear code $\mathscr{C}$, we refer the reader to [31, 84, 96]. The set of codewords of minimal support describe completely the minimal access structure of these schemes.

- **Steganography** is the science of stealth communication in such a way that no one, apart from the sender and the receiver, can detect the existence of a mes-

sage. In order to hide messages in innocuous-looking media we need mainly two factors: (i) the cover selection and (ii) the embedding and recovering methods. The embedding function should introduce a little distortion on the cover, while the extraction function recovers the hidden message.

In 1998, Crandall [36] proposed to create steganographic schemes with the use of error correcting codes. The key idea of this approach is to modify the cover to obtain a cover modification whose syndrome is precisely equal to the message to hide. In particular, let $\mathscr{C}$ be an $[n,k]$-linear code over $\mathbb{F}_q$ and H be a parity check matrix of $\mathscr{C}$, $\mathbf{x} \in \mathbb{F}_q^n$ be a cover-data and $\mathbf{m} \in \mathbb{F}_q^{n-k}$ be the message to be hidden in $\mathbf{x}$. The problem consist in finding $\mathbf{y} \in \mathbb{F}_q^n$ such that $H\mathbf{y}^T = \mathbf{m}$. Therefore, the embedding map is directly linked to a decoding function.

However, the decoding map is used differently in Coding Theory and Steganography. In Coding Theory the goal is to detect and correct errors so it is important that the original message and the decoded word match. On the contrary, in Steganography the objective is to introduce errors in the cover so the uniqueness of the nearest codeword is not necessary, if there more than one nearest codeword we arbitrary take one of them. Hence complete decoding plays an important role in steganography, see [89] and the references given there.

We begin the thesis with a preliminary chapter to give a brief introduction of the research topics necessary for understanding the new results presented in the following chapters. This memory is intended to be as much self contained as possible. Therefore, we provide an introduction to Coding Theory, to computational tools, namely Gröbner and Border basis, and a section highlighting the relationship between linear codes and representable matroids.

Chapter 2 is devoted to binary codes, i.e. k-dimensional linear subspace of $\mathbb{F}_2^n$. The goal of this chapter is to extend previous work on the Gröbner representation of binary codes [11, 13, 14, 15, 16], that we will see is an essential structure in the decoding process, to get a better understanding of the underlying coset enumeration. Let $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ be the canonical basis of $\mathbb{F}_2^n$. A Gröbner representation of a binary linear code $\mathscr{C}$ is a pair $(\mathscr{N}, \phi)$ where $\mathscr{N}$ is a transversal of the cosets in $\mathbb{F}_2^n/\mathscr{C}$ and ϕ is the function that maps each pair $(\mathbf{n}, \mathbf{e}_i)$ to the element of $\mathscr{N}$ representing the coset that contains $\mathbf{n}+\mathbf{e}_i$. As the reader may have guessed, the word *Gröbner* is not casual. Indeed, if we consider a total degree ordering $\prec$, the binomial ideal associated to $\mathscr{C}$ $I(\mathscr{C}) = \left\langle \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \mathbf{a} - \mathbf{b} \in \mathscr{C} \right\rangle \subset \mathbb{K}[\mathbf{X}]$ and we compute the reduced Gröbner basis $\mathscr{G}$ of $I(\mathscr{C})$ w.r.t. $\prec$. Then, we can take $\mathscr{N}$ as the vectors associated with the standard monomials of $\mathscr{G}$, while the function ϕ can be seen as multiplication tables of standard monomials times the variables X_i. Therefore, the Gröbner representation of $\mathscr{C}$ can be computed with a slight modification of the FGLM algorithm presented in [11]. In Chapter 2 we present an algorithm that not only provides a Gröbner representation of a binary linear code but also the whole set of all coset leaders and even more a test-set. Moreover, the algorithm could be adapted without incrementing the complexity

to get more information such as the Newton radius or the Covering radius.
For those not familiar with Coding Theory recall that a *test-set* of a code is a set of codewords such that every word $\mathbf{y}$ either belongs to the set of coset leaders or there is an element $\mathbf{t}$ from the test-set such that the $w_H(\mathbf{y}-\mathbf{t}) \leq w_H(\mathbf{t})$, where $w_H(\cdot)$ denotes the Hamming weight of a vector.
At the end of Chapter 2 we show a unified approach, via the Gröbner representation of a code, to the two gradient descent decoding algorithm known for binary codes proposed independently by Liebler [71] and Ashikhmin and Barg [2]. The algorithms were claimed to be of different nature but we will show that both algorithms can be seen as two ways of understanding the reduction of a monomial modulo the binomial ideal associated to the binary code.

In Chapter 3 we work with modular codes defined over $\mathbb{Z}_q$, or equivalently a submodule of $\left(\mathbb{Z}_q^n, +\right)$. Throughout this chapter we will use the following *characteristic crossing functions*:

$$\blacktriangledown : \quad \mathbb{Z}^s \longrightarrow \mathbb{Z}_q^s \qquad \text{y} \qquad \blacktriangle \quad \mathbb{Z}_q^s \longrightarrow \mathbb{Z}^s$$

where s is determined by context and both maps act coordinate-wise. The map $\blacktriangledown$ is reduction modulo q while the map $\blacktriangle$ replaces the class of $0, 1, \ldots, q-1$ by the same symbols regarded as integers. This relationship allows us to work with monomials with $\mathbb{Z}_q$-exponents as monomials with integer exponents.
Consider the matrix $A \in \mathbb{Z}_q^{m\times n}$ (namely the coefficient matrix), the vector $\mathbf{b} \in \mathbb{Z}_q^m$ and the cost vector $\mathbf{w} \in \mathbb{R}^n$. We define a *modular integer program*, denoted by $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$, as the problem of finding a vector $\mathbf{u} \in \mathbb{Z}_q^n$ that minimizes the inner product $\mathbf{w} \cdot \blacktriangle\mathbf{u}$ subject to $A\mathbf{u}^T \equiv \mathbf{b} \mod q$. If we express the problem in schematic form it becomes:

$$\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b}) = \begin{cases} \text{minimize: } \mathbf{w} \cdot \blacktriangle\mathbf{u} \\ \text{subject to: } \begin{cases} A\mathbf{u}^T \equiv \mathbf{b} \mod q \\ \mathbf{u} \in \mathbb{Z}_q^n \end{cases} \end{cases}$$

It is widely known that Minimum Distance Decoding (MDD) for binary codes can be regarded as a linear integer program with binary arithmetic conditions. From the point of transmission reliability, MDD is the most powerful decoding method. This method consists of finding a codeword that minimized the Hamming distance with the received word. In other words, solving the modular problem $\mathrm{IP}_{H,\mathbf{w},2}(\mathbf{b})$, where H is a parity check matrix of the linear code $\mathscr{C}$ and $\mathbf{w}$ is the all one vector, is equivalent to perform complete decoding of the received word $\mathbf{b}$ in $\mathscr{C}$.
In 2002, Ikegami and Kaji [60], extending the ideas of Conti and Traverso [33], proposed an efficient algorithm which uses Gröbner bases to solve modular integer program. Roughly speaking, the algorithm computes a Gröbner basis $\mathscr{G}$ of the binomial ideal associated to the family of modular problems $\mathrm{IP}_{A,\mathbf{w},q}$ with respect to an adequate term order compatible with the corresponding cost-vector $\mathbf{w}$. Therefore, the binomials involved in the Gröbner bases of $\mathscr{G}$ induced a uniquely defined test-set for the family of modular problems $\mathrm{IP}_{A,\mathbf{w},q}$.

A *test-set* for integer programming is a subset with the property that for each feasible but not optimal solution, there exists at least an element from the test set in which when added to the solution yields an improved value of the objective function.
These ideas gives a method for computing a test-set for a binary code. Furthermore the Graver basis associated to a modular integer programming problem provides a universal test-set which turns out to be a set containing the set of codewords of minimal support of the corresponding modular code.
These ideas are the main axis of Chapter 3. In brief, in order to obtain a test-set for the binary case or the set of codewords of minimal support of a modular code, we must compute a reduced Gröbner basis of an ideal from which we know a generating set. We propose several solutions to improve the efficiency of the algorithm. For example, we can use Urbanke-Di Biase [38] philosophy to reduce the number of variables. This reduction consists of a generalization to the non-modular case of what was exposed in [11]. Moreover, we recommend to use the extension of the FGLM techniques presented in [11] for the modular case. In that paper the algorithm was stated for the binary case but we present the generalization to the modulo q. Note that the complexity of the algorithm is $\mathcal{O}\left(n^2 q^{n-k}\right)$ where k is the dimension of the code and n is the length of the code, i.e. n is the number of variables involved. This procedure is completely general but it has the following advantages in our setting:

1. All steps can be carried out as Gaussian elimination steps.
2. The problem of growth of the coefficients does not have to be considered since we can encode all the information of the problem in the exponents, thus we can always take $\mathbb{K} = \mathbb{F}_2$.
3. The problem of growth of the total degree does not have to be taken into account since the total degree of the binomials concerned are bounded by $n \times q$.

Another way to reduce the complexity is by using the decomposition of a code as "gluing" of smaller ones. That is to say, our aim is to explicitly define a procedure that:

- Finds a decomposition of an arbitrary modular code $\mathscr{C}$ as m-gluing of two (or more) smaller codes.
- Computes the set of codewords of minimal support for each code that appeared on the decomposition.
- Finally computes the set of codewords of minimal support of the original code using the set obtained in the previous step.

Note that parallel computing is implicitly well-suited for the second step since the computation can be carried out in parallel for each component. A similar process can be defined in the binary case to compute a Gröbner test-set.

In Chapter 4 we study a generalization of the results discussed in the above chapters to an arbitrary linear code. For that purpose, we need new *characteristic crossing*

functions defined over a finite fields $\mathbb{F}_q$ where q is a prime power. Let α be a primitive element of $\mathbb{F}_q$ and $\left\{\mathbf{e}_1,\ldots,\mathbf{e}_{q-1}\right\}$ be the canonical basis of $\mathbb{Z}^{q-1}$, then we define:

$$\nabla: \quad \{0,1\}^{q-1} \quad \longrightarrow \quad \mathbb{F}_q \qquad \text{and} \qquad \Delta: \quad \mathbb{F}_q^n \quad \longrightarrow \quad \{0,1\}^{q-1}$$

The map Δ replace the class of the elements $\mathbf{a}=\alpha^j \in \mathbb{F}_q^*$ by the vector $\mathbf{e}_j$ and $0 \in \mathbb{F}_q$ by the zero vector $\mathbf{0} \in \mathbb{Z}^{q-1}$. While the map ∇ recovers the element $j_1\alpha+\ldots j_{q-1}\alpha^{q-1}$ of $\mathbb{F}_q$ from the $(q-1)$-tuple of binary elements $\left(j_1,\ldots,j_{q-1}\right)$.
Let $\mathbf{X}$ denote n vector variables $X_1,\ldots,X_n$ such that each variable X_i can be decomposed into $q-1$ components $x_{i1},\ldots,x_{iq-1}$ with $i=1,\ldots,n$. Let $\mathbf{a}=(a_1,\ldots,a_n)$ be an n-tuple of elements of the field $\mathbb{F}_q$. We fix the following notation

$$\mathbf{X}^{\mathbf{a}} \quad = \quad X_1^{a_1}\cdots X_n^{a_n} \quad = \quad \left(x_{11}\cdots x_{1q-1}\right)^{\Delta a_1}\cdots\left(x_{n1}\cdots x_{nq-1}\right)^{\Delta a_n}.$$

This relationship allows us to work with monomials whose exponents are formed by elements defined over the field $\mathbb{F}_q$ as monomials with integer exponents. Moreover, for all $\mathbf{a} \in \mathbb{F}_q^n$ we have that $\deg(\mathbf{X}^{\mathbf{a}}) = \mathrm{w}_H(\Delta\mathbf{a})$, that is, a weight compatible ordering on $\mathbb{F}_q^n$ can be viewed as a total degree ordering on $\mathbb{K}[\mathbf{X}]$. This is the key idea of the generalization made in Chapter 4 which we will detail in the following lines. Let $\mathscr{C}$ be a linear code defined over the finite field $\mathbb{F}_q$, then:

1. We give a finite set of generators of the binomial ideal associated to $\mathscr{C}$, denoted by $I_+(\mathscr{C})$, defined by the rows of a generating matrix of our code and the relations given by the additive table of the field $\mathbb{F}_q$.

2. We compute a Gröbner representation of $\mathscr{C}$.

3. We show that the binomials involved in the reduced Gröbner basis of $I_+(\mathscr{C})$ w.r.t. a degree compatible ordering define a test-set for our code.

4. We define via the Gröbner representation of a linear code two reduction processes which allow us to define two gradient descent decoding algorithms similar to the two algorithms known for binary codes.

5. We discuss an alternative for the computation of the Gröbner basis of $I_+(\mathscr{C})$, adapting the FGLM techniques presented in [11] for the binary case. In addition, a complete study of the complexity of the proposed algorithm is given.

6. We set an algorithm to compute the set of codewords of minimal support of $\mathscr{C}$.

Moreover, the result of this chapter could be generalized to other classes of codes such as modular codes, codes defined over multiple alphabets or additive codes. Modular codes have already been discussed in Chapter 3 but this new approach allows the computation of a Gröbner test-set.

Last chapter provides a new approach using techniques inspired by toric mathematics from semigroups. The purpose of Chapter 5 is to assign a generating set to the semigroup associated to an error-correcting code. This construction establishes a strong

relation between codes and semigroups and constitutes a means to apply numerous results in the field of semigroups to problems in information theory. We will study different representations for the semigroup S associated with modular and linear codes. However the choice of digital representations seems to be the best adapted to perform complete decoding on the selected codes. We understand by digital representation any generating set $\{\mathbf{n}_1, \ldots, \mathbf{n}_r\}$ of S such that every element $\mathbf{m} \in S$ can be written as

$$\sum_{i=1}^{r} a_i \mathbf{n}_i \text{ with } a_1, \ldots, a_r \in \{0, 1\} \subseteq \mathbb{N}.$$

The advantage of using digital representations lies on the equivalence between the degree of the monomials $\mathbf{X}^\gamma \in I(S)$ and the Hamming weight of the vectors γ, following the results obtained in Chapter 4.

Some research of this thesis can be found in the following publications (with peer-review):

[75] I. Márquez-Corbella and E. Martínez-Moro. *Decomposition of Modular Codes for Computing Test Sets and Graver Basis*. Mathematics in Computer Science, 6(2), pp. 147-165, 2012.

[74] I. Márquez-Corbella and E. Martínez-Moro. *Algebraic structure of the minimal support codewords set of some linear codes*. Adv. Math. Commun., 5(2), pp. 233-244, 2011.

[18] M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro, *An algebraic view to gradient descent decoding*. Information Theory Workshop (ITW), 2010 IEEE, Dublin (Ireland), August 30 - September 3, 2010, pp.1-4.

More publications and collaborative work during my PhD are:

[76] I. Márquez-Corbella and E. Martínez-Moro. *An Introduction to LDPC codes*. In Algebraic Geometry Modeling in Information Theory, Series on Coding Theory and Cryptology, pp. 129-166. ISBN: 978-981-4335-75-1, 2013.

[78] I. Márquez-Corbella, E. Martínez-Moro and G. R. Pellikaan. *On the unique representation of very strong algebraic geometry codes*. Designs, Codes and Cryptography, pp. 1-16, 2012.

[77] I. Márquez-Corbella, E. Martínez-Moro and G. R. Pellikaan. *The non-gap sequence of a subcode of a generalized Reed-Solomon code*. Designs, Codes and Cryptography, pp. 1-17, 2012.

[37] B. Debraize and I. Márquez-Corbella. *Fault Analysis of the Stream Cipher Snow 3G*. Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), Lausanne (Switzerland), September 6, 2009, pp. 103-110.

Other publications submitted for publications are:

[79] I. Márquez-Corbella, E. Martínez-Moro, G. R. Pellikaan and D. Ruano. *Computational Aspects of Retrieving a Representation of an Algebraic Geometry Code*.

[81] I. Márquez-Corbella, E. Martínez-Moro and E. Suárez-Canedo. *On the Composition of Secret Sharing Schemes Related to Codes*.

[12] M. B. Quintana, M. A. B. Trenard, I. Márquez-Corbella and E. Martínez-Moro. *Computing coset leaders and leader codewords of binary codes*.

[82] I. Márquez-Corbella, E. Martínez-Moro and E. Suárez-Canedo. *On the ideal associated to a linear code.*

[80] I. Márquez-Corbella and R. Pellikaan. *Error-correcting pairs for public-key cryptosystem*.

Notations

$\mathbb{K}$	An arbitrary field, appears on page 36.
$\mathbb{K}[\mathbf{X}]$	The polynomial ring in n variables over $\mathbb{K}$, appears on page 36.
$\mathbb{Z}$	The ring of integers, appears on page 126.
$\mathbb{Z}_s$	The ring of integer modulo s, appears on page 30.
$\mathbb{F}_q$	A finite field with q elements, appears on page 25.

Linear Codes

$\mathscr{C}$	An $[n,k]$ linear code over $\mathbb{F}_q$, appears on page 26.
n	Length of $\mathscr{C}$, appears on page 26.
k	Dimension of $\mathscr{C}$, appears on page 26.
d	Minimum distance of $\mathscr{C}$, appears on page 26.
$d_H(\mathbf{x},\mathbf{y})$	Hamming distance between two vectors $\mathbf{x},\ \mathbf{y} \in \mathbb{F}_q^n$, appears on page 26.
$\mathrm{w}_H(\mathbf{y})$	Hamming weight of the vector $\mathbf{y} \in \mathbb{F}_q^n$, appears on page 26.
$\mathrm{supp}(\mathbf{y})$	Support of the vector $\mathbf{y} \in \mathbb{F}_q^n$, appears on page 26.
t	error-correcting capacity of $\mathscr{C}$, appears on page 27.
$\mathscr{C}^\perp$	Dual code of $\mathscr{C}$, appears on page 28.
$S(\mathbf{x})$	Syndrome of a vector $\mathbf{x} \in \mathbb{F}_q^n$, appears on page 29.
$\mathrm{CL}(\mathscr{C})$	Set of coset leaders of the code $\mathscr{C}$, appears on page 28.
$\mathrm{CL}(\mathbf{y})$	Subset of coset leaders corresponding to the coset $\mathscr{C}+\mathbf{y}$, appears on page 28.
$\mathrm{CL}(\mathscr{C})_i^j$	The j-th element of the set of coset leaders of weight i, appears on page 57.

$\mathscr{T}_{\mathscr{C}}$	Test-set for $\mathscr{C}$, appears on page 30.
$\mathrm{D}(\mathbf{c})$	Voronoi region of a codeword $\mathbf{c} \in \mathscr{C}$, appears on page 30.
$\mathscr{M}_{\mathscr{C}}$	Set of codewords of minimal support of $\mathscr{C}$, appears on page 30.
$\nu(\mathscr{C})$	Newton radius of $\mathscr{C}$, appears on page 59.
$\rho(\mathscr{C})$	Covering radius of $\mathscr{C}$, appears on page 59.
WDCL	Weight Distribution of the Coset Leaders of a binary code $\mathscr{C}$, appears on page 59.
$\mathrm{L}(\mathscr{C})$	Set of leader codewords of $\mathscr{C}$, appears on page 62.
$\mathscr{X}(A)$	Set of words at Hamming distance 1 from A, appears on page 66.
$\delta(A)$	Boundary of A, appears on page 66.
$\mathscr{Z}(\mathscr{C})$	Set of all Zero-Neighbours of $\mathscr{C}$, appears on page 66.
$B(\mathscr{C})$	Border of the code $\mathscr{C}$, appears on page 74.
$R(\mathscr{C})$	Reduced Border of a code, appears on page 75.
$\mathrm{head}(\mathbf{b})$	First component of an element $\mathbf{b} \in B(\mathscr{C})$, appears on page 75.
$\mathrm{tail}(\mathbf{b})$	Second component of an element $\mathbf{b} \in B(\mathscr{C})$, appears on page 75.
$I_m(\mathscr{C})$	Binomial ideal of $\mathscr{C}$ by taking care of the arithmetic over the mathematical definition structure of $\mathscr{C}$, appears on page 90.
$I_+(\mathscr{C})$	Binomial ideal of $\mathscr{C}$ given by the additive rules of the mathematical definition structure of $\mathscr{C}$, appears on page 127.
$\mathscr{R}_{X_i}\left(T_+\right)$	Binomials on the variable X_i associated to the relations given by the additive table of the field $\mathbb{F}_q$, appears on page 128.
$\mathrm{cf}(\mathbf{w})$	Canonical form of $\mathbf{w} \in \mathbb{F}_q^n$, appears on page 135.
$\mathbf{x}_J$	The restriction of $\mathbf{x}$ to the coordinates indexed by J, appears on page 102.
$\overline{J}$	The relative complement of J in $\{1,\ldots,n\}$, appears on page 102.
$\mathscr{C}_J$	Punctured code at positions in J, appears on page 102.
$\mathscr{C}._J$	Shortened code at positions in J, appears on page 103.
$c^{(i)}$	i-th coordinate of $\mathbf{c}$, appears on page 103.
$\mathscr{C}_1 \oplus \mathscr{C}_2$	Sum code of two modular codes $\mathscr{C}_1$ and $\mathscr{C}_2$, appears on page 103.

$(\mathbf{c}_1 \parallel \mathbf{c}_2)$	Codewords of the sum code $\mathscr{C}_1 \oplus \mathscr{C}_2$, appears on page 103.
$(\mathbf{a} \mid \mathbf{b}) \in \mathbb{Z}_q^{m_1+m_2}$	Concatenation of $\mathbf{a} \in \mathbb{Z}_q^{m_1}$ and $\mathbf{b} \in \mathbb{Z}_q^{m_2}$, appears on page 103.
$S(\mathscr{C}_1, \mathscr{C}_2)$	New code from $\mathscr{C}_1 \oplus \mathscr{C}_2$ by shortening at the positions where $\mathscr{C}_1$ and $\mathscr{C}_2$ overlap, appears on page 104.
$\mathbf{c}_1 \parallel_m \mathbf{c}_2$	Codewords of $S(\mathscr{C}_1, \mathscr{C}_2)$, appears on page 104.
$\mathscr{C}_1 ⓖ_m \mathscr{C}_2$	m-gluing of $\mathscr{C}_1$ and $\mathscr{C}_2$, appears on page 104.

Gröbner bases, Border bases and Graver bases

$[\mathbf{X}]$	Set of terms in $\mathbb{K}[\mathbf{X}]$, appears on page 36.
$[\mathbf{X}]_d$	Set of terms of degree d, appears on page 42.
$\prec_T$	Total degree ordering, appears on page 36.
$\mathrm{supp}(f)$	Support of a polynomial f, appears on page 37.
$\mathrm{LT}_\prec(f)$	Leading term of f w.r.t. $\prec$, appears on page 37.
$\mathrm{LC}_\prec$	Leading coefficient of f w.r.t. $\prec$, appears on page 37.
$\mathrm{LT}_\prec\{F\}$	Set of leading terms of all non-zero polynomials in F, appears on page 37.
$\mathrm{LT}_\prec(F)$	Semigroup ideal generated by $\mathrm{LT}_\prec\{F\}$, appears on page 37.
$\mathrm{in}_\prec(I)$	Initial ideal of the ideal I w.r.t. $\prec$, appears on page 37.
$\delta\mathscr{O}$	Border of an order ideal $\mathscr{O}$, appears on page 42.
$\mathrm{ind}_\mathscr{O}(t)$	$\mathscr{O}$-index of $t \in [\mathbf{X}]$, appears on page 43.
$S_\prec(f,g)$	S-polynomial of f and g w.r.t. $\prec$, appears on page 41.
$N_\prec(I)$	Set of standard monomials of I w.r.t. $\prec$, appears on page 46.
$B_\prec(I)$	Border set of I w.r.t. $\prec$, appears on page 46.
$C_\prec(I)$	Corner set of I w.r.t. $\prec$, appears on page 46.
$G_\prec(I)$	Unique minimal basis of $\mathrm{in}_\prec(I)$, appears on page 46.
$\mathscr{G}_\prec$	Reduced Gröbner basis of I w.r.t. $\prec$, appears on page 46.
$\mathscr{B}_\prec$	Border basis of I w.r.t. $\prec$, appears on page 46.
$\mathrm{Red}_\prec(f,F)$	Remainder on the division of f by the list of polynomials F w.r.t. the term order $\prec$., appears on page 39.

$(\mathcal{N}, \phi)$	Gröbner representation of $\mathcal{C}$, appears on page 50.
$\mathbf{n} \longrightarrow_i \mathbf{n}'$	Reduction of an element $\mathbf{n}$ relative to the unit vector $\mathbf{u}_i$, appears on page 144.

Linear programming problems

$\mathcal{C}_A$	Set of circuits of $A \in \mathbb{Z}^{m\times n}$, appears on page 88.
Gr_A	Graver basis of $A \in \mathbb{Z}^{m\times n}$, appears on page 88.
$\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$	Integer LP problem, appears on page 82.
$\Lambda(A)$	Lawrence lifting of $A \in \mathbb{Z}^{m\times n}$, appears on page 89.
$\Lambda(A)_q$	Lawrence lifting for the modulo q of $A \in \mathbb{Z}_q^{m\times n}$, appears on page 90.
$T_{\succ_{\mathbf{w}}}$	Test-set for the family of problems $\mathrm{IP}_{A,\mathbf{w}}$, appears on page 82.
UGB_A	Universal Gröbner basis of $A \in \mathbb{Z}^{m\times n}$, appears on page 88.

Semigroups

S	Commutative semigroup with an identity element, appears on page 178.
$G(S)$	Associated commutative group of S, appears on page 179.
$\mathbb{K}[S]$	Semigroup algebra of S, appears on page 179.
$I(S)$	Semigroup ideal associated to S, appears on page 180.
$I_F(S)$	Semigroup ideal associated to S when S is defined by the generating set F, appears on page 180.
$\mathscr{L}$	Lattice, appears on page 180.
$I_{\mathscr{L}}$	Lattice ideal associated to $\mathscr{L}$, appears on page 180.

1 Preliminaries

Contents

This first chapter aims to give a brief introduction about the research topics of the thesis which include *Coding Theory* (Section 1.1), *Gröbner basis and Border basis* (Section 1.2) and an overview of the connections between codes and *Matroids* (Section 1.3). This chapter will provide the reader with the necessary background for interpreting the results within this study; and thus, improve the readability of the thesis. We will fix general notation, summarize definitions and recall known results. Indeed for these three topics we need several volumes for their proper coverage as well as their connections. Thus, this introduction is based in our purposes. Please refer to the cited references for a deeper insight.

Section 1.1 provides an introduction to Coding Theory. The approach to error correcting codes started in the late 1940's with the work of Shannon [101], which set the theoretical limits of reliable communication; and the work of Hamming [53] and Golay [50] who developed the first error correcting schemes. My goal is to

present the necessary aspects of this theory in a simple and understandable manner. For a more thorough treatment of the theory of error-correcting codes see Berlekamp [5], Blahut [9], Justesen and Høholdt [62], MacWilliams and Sloan [72], Hoffman [56] Huffman and Pless [58] and Van Lint [112].

Section 1.2 gives a very basic introduction to Gröbner Bases and Border Bases. Gröbner Bases were first introduced in 1965 in Burno Buchberger's PhD dissertation [24] who named his method after his advisor Wolfgang Gröbner. The idea can be traced back to Gauss-Jordan elimination for multivariate polynomial systems. Our notation on Gröbner Bases is from [98]. For a fuller treatment we refer the reader to [1, 34, 35, 52, 67]. Gröbner bases for toric ideals and their application to integer programming have been studied by several authors (including Conti-Traverso [33], Sturmfels [106, 107], Sturmfels-Thomas [108], Thomas [110]), note that Chapter 3 and Chapter 4 is an application of their results. We refer the reader to [40, 87, 103] for deeper discussion on combinatorial commutative algebra and binomial ideals.

Section 1.3 explores the connection between Coding Theory and Matroid Theory, see [61] and the references given there. Matroids were introduced independently by Whitney [120] and by Van der Waerden [117]. This theory enables the study of the concepts of linear algebra and graph theory in a more abstract way. For a more background reading on Matroid Theory see Oxley [90], Kung [68], Welsh [118] or White [119].

1.1 Coding Theory

Claude Shannon's paper from 1948 entitled *"A Mathematical Theory of Communication"* [101] gave birth to the disciplines of *information theory* and *coding theory*. The main goal of these disciplines is to efficiently transfer reliable information. To be **efficient** the transfer of information must not require a big amount of time and effort. Whereas to be **reliable**, the received and transmitted data must coincide. However, during the transmission over noisy channels, the information could be damaged so it has become necessary to develop ways of detecting when an error has occurred and how to correct it.

Given a communication channel that may corrupt information sent over it, Shannon proved that there exists a number which he identified with the capacity of the channel, such that reliable communication is possible at any rate below its capacity.

The fundamental problem of Coding Theory is to find optimal encoding and decoding schemes for reliable error correction of data sent through noisy transmission channels. The idea is to add redundant information to enable the detection and correction of errors after transmission. The theory has been developed for diverse applications, intersecting mathematics and engineering, such as the minimization of noise from compact disc recordings, the transmission across telephone and satellite lines, data storage and information transmission from a distant source. Diagram 1.1 provides an idea of a general communication channel.

We begin by selecting an alphabet which is a finite set of letters. Every original message is presented as a set of k-tuples of element of the chosen alphabet. In general

our alphabet is the finite field with q elements $\mathbb{F}_q$, so our messages will be vectors in $\mathbb{F}_q^k$.

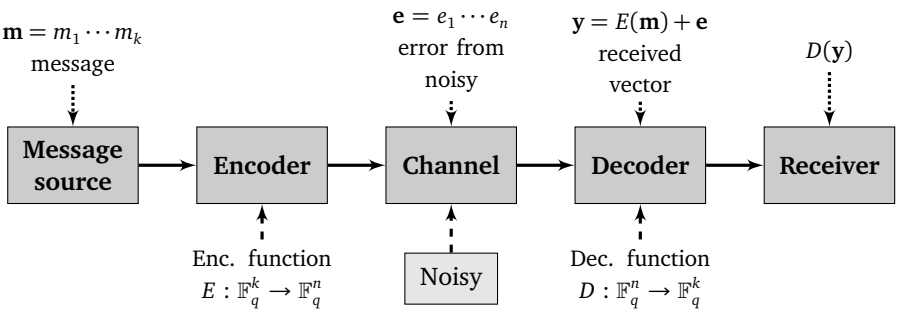


Figure 1.1: Block diagram of a communication system

An *encoding function* with parameters $[n, k]$ is a function $E : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$ that maps a message $\mathbf{m}$ consisting of k symbols into a longer redundant string $E(\mathbf{m})$ of length n over $\mathbb{F}_q^n$. An *error-correcting code* $\mathscr{C}$ is defined to be the image of an encoding function and the set of string $E(\mathbf{m})$ of the different messages are called the set of codewords of $\mathscr{C}$. Thus, the process of transforming the original message into codewords of a certain code of length n is known as *encoding*.

Every codeword is then sent through a noisy channel. We will assume that the channel noise is "nicely" behaved, i.e. is a *discrete memoryless channel*. Roughly speaking a *discrete channel* comprises an input $\mathscr{X}$ and an output $\mathscr{Y}$ alphabets (both finite alphabets, moreover in our particular case both alphabets will coincide) and a probability transition matrix which defines the probability that $y \in \mathscr{Y}$ was received given that $x \in \mathscr{X}$ was sent; the channel is *memoryless* if the error in each letter is independent of previous channel inputs or outputs.

Finally, the receiver gets a possibly distorted copy of the transmitted codeword and needs to figure out the original message. This process of correcting errors and obtaining back the message is called *decoding*. This is done via a *decoding function* $D : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^k$ that maps noisy received words $\mathbf{y}$ to an estimate $\hat{\mathbf{y}}$, hoping that $\hat{\mathbf{y}}$ and the transmitted message agree.

Shannon's Theorem guarantees that taking into account the characteristics of the channel and with the right encoding, our estimate will be very likely match the original message. However, Shannon does not provide a method to perform the required encoding and decoding schemes efficiently.

Let $\mathscr{A}$ be a fix alphabet with q elements. There are several parameters that indicate the quality of a code of length n over the alphabet $\mathscr{A}$, which contains M codewords:

- The *information rate* of the code $R = \frac{\log_q(M)}{n}$, which gives a measure of how much information is being transmitted.

- The *relative minimum distance* $\delta = \frac{d}{n}$ which is closely related to its error-correcting capability, where d is the minimum distance of the code or equivalently the least Hamming distance between any two distinct codewords (number of places where they differ).

Now it is clear that the aim of Coding Theory is to provide codes with high information rate, high relative minimum distance and low complexity on its encoding and decoding procedures.

1.1.1 Basics of Linear Codes

In order to define codes that can encode and decode efficiently, we can add some structure to the codespace. In this thesis we are mainly interested in linear codes.

Definition 1.1. A *linear code* $\mathscr{C}$ over $\mathbb{F}_q$ of length n and dimension k, or an $[n,k]_q$ code for short, is a k-dimensional subspace of $\mathbb{F}_q^n$.

We will call the vectors $\mathbf{v}$ in $\mathbb{F}_q^n$ words while the vectors in $\mathscr{C}$ are called *codewords*.

For a word $\mathbf{v} \in \mathbb{F}_q^n$ its *support* is defined as its support as a vector in $\mathbb{F}_q^n$, i.e. $\mathrm{supp}(\mathbf{v}) = \{i \mid v_i \neq 0\}$ and its *Hamming weight*, denoted by $\mathrm{w}_H(\mathbf{v})$ as the cardinality of $\mathrm{supp}(\mathbf{v})$.

It is clear that for an $[n,k]$ code over $\mathbb{F}_q$, its size, which is the number of codewords, is equal to $M = q^k$; the *information rate*, which gives a measure of how much information is being transmitted, is $R = \frac{k}{n}$; and the *redundancy* is $n-k$.

Definition 1.2. The *Hamming distance*, $d_H(\mathbf{x},\mathbf{y})$, between two vectors $\mathbf{x},\ \mathbf{y} \in \mathbb{F}_q^n$ is the number of places where they differ, or equivalently, $d_H(\mathbf{x},\mathbf{y}) = \mathrm{w}_H(\mathbf{x}-\mathbf{y})$.

The *minimum distance* d of a linear code $\mathscr{C}$ is defined as the minimum weight among all nonzero codewords. Therefore, the minimum distance is also called the *minimum weight* of the code. If $\mathscr{C}$ has minimum distance d, then we refer to the code as an $[n,k,d]$ linear code over $\mathbb{F}_q$.

The following bound called *Singleton bound* gives us the maximal minimum distance of an $[n,k]$ code.

Theorem 1.3. *If $\mathscr{C}$ is an $[n,k,d]_q$ code, then $d \leq n-k+1$.*

Proof. A proof can be found for example in [62, Theorem 5.1.1]. □

Any code with parameters which achieve the Singleton Bound is called a Maximum Distance Separable (MDS) code.

The minimum distance of a code is important in determining the error-correcting capability of the code, i.e. the higher the minimum distance, the more errors the code can correct.

Definition 1.4. A code $\mathscr{C}$ is t-error correcting if there exists a decoding function $D: \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^k$ such that for every message $\mathbf{m} \in \mathbb{F}_q^k$ and every received word $\mathbf{y} \in \mathbb{F}_q^n$ with $d_H(\mathbf{y}, E(\mathbf{m})) \leq t$, we have that $D(\mathbf{y}) = \mathbf{m}$. In other words, it is not possible to attain a certain received word from making at most t error in two different codewords.

Remark 1.5. Note that for any $[n,k,d]$ code and for any received word $\mathbf{y}$, if $t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$ then there is only one codeword $\mathbf{c}$ in $\mathscr{C}$ satisfying that $d_H(\mathbf{c},\mathbf{y}) \leq t$. Or equivalently, the nearest codeword to $\mathbf{y}$ is unique when $\mathbf{y}$ and $\mathscr{C}$ has maximum distance of $\left\lfloor \frac{d-1}{2} \right\rfloor$ to $\mathscr{C}$. The parameter $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ is called the *error-correcting capacity* of the code $\mathscr{C}$.

There are two standard ways to describe a linear subspace, either explicitly by giving a basis, or implicitly by the solution space of a set of homogeneous linear equations. Therefore, there are two ways to describe a linear code: explicitly as the row space of a matrix (generator matrix) or implicitly by the null space of a matrix (parity check matrix).

Let $\mathscr{C}$ be an $[n,k]$ linear code over $\mathbb{F}_q$, then $\mathscr{C}$ is a k-dimensional linear subspace of $\mathbb{F}_q^n$, i.e. there exists a basis that consists of k linearly independent codewords.

Definition 1.6. A $k \times n$ matrix G with entries in $\mathbb{F}_q$ is called a *generator matrix* of an $[n,k]$ linear code $\mathscr{C}$ if the rows of G form a basis of $\mathscr{C}$.

Given a generator matrix G of $\mathscr{C}$, the *encoding* from the message $\mathbf{m} \in \mathbb{F}_q^k$ to a codeword can be done efficiently by a matrix multiplication since

$$\mathscr{C} = \left\{ \mathbf{m}G \mid \mathbf{m} \in \mathbb{F}_q^n \right\}.$$

Thus, representing the code requires storing only k vectors of length n rather than the q^k vectors that would be required for a nonlinear code. Note also that the representation of the code provided by a generator matrix G is not unique since by performing row operations on G (nonzero linear combinations of the rows) another generator matrix of $\mathscr{C}$ can be obtained.

A generator matrix G is a *standard generator matrix* if its first k columns forms the identity matrix of size k. This form of the generator matrix gives a *systematic encoding* of the information, i.e. we simply let the first k coordinates be equal to the information symbols. The generator matrix G is *systematic* if among its columns we can find the identity matrix of size k, in which case G is said to be systematic on those columns. Any set of coordinates corresponding to k independent columns of G forms an *information-set* for $\mathscr{C}$.

Definition 1.7. An $(n-k) \times n$ matrix of rank $n-k$ with entries in $\mathbb{F}_q$ is called a *parity check matrix* of an $[n,k]$ linear code $\mathscr{C}$ if $\mathscr{C}$ is the null space of this matrix.

A generator matrix G and a parity check matrix H for a code $\mathscr{C}$ satisfy $GH^T = 0$. Moreover, a parity check matrix for a code provides information about the minimum distance of the code.

Proposition 1.8. *Let H be a parity check matrix of $\mathscr{C}$. Then the minimum distance d of $\mathscr{C}$ is the smallest integer d such that d columns of H are linearly independent.*

The next proposition gives a parity check matrix for $\mathscr{C}$ when $\mathscr{C}$ has a generator matrix in standard form.

Proposition 1.9. *If $G = \left(\begin{array}{c|c} I_k & A \end{array} \right)$ is a generator matrix for the $[n,k]_q$ code $\mathscr{C}$ in standard form, then $H = \left(\begin{array}{c|c} -A^T & I_{n-k} \end{array} \right)$ is a parity check matrix for $\mathscr{C}$, where I_m denotes the identity matrix of size m and A^T is the transpose of A.*

Definition 1.10. Let $\mathscr{C}$ be any code (not necessarily linear) defined over $\mathbb{F}_q^n$. The *dual code* of $\mathscr{C}$, denoted $\mathscr{C}^\perp$, is the code

$$\mathscr{C}^\perp = \left\{ \mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{x} \cdot \mathbf{c} = 0 \text{ for all } \mathbf{c} \in \mathscr{C} \right\},$$

where $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + \ldots + x_n y_n$ for $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ denotes the usual *inner product* or *dot product* on $\mathbb{F}_q^n$.

The dual of $\mathscr{C}$ is linear even if $\mathscr{C}$ is not. Indeed it is often a good way of proving that a given code is linear.

Lemma 1.11. *If $\mathscr{C}$ is an $[n,k]$ linear code, then its dual $\mathscr{C}^\perp$ is an $[n, n-k]$ linear code and $\left(\mathscr{C}^\perp\right)^\perp = \mathscr{C}$. Moreover, a parity check matrix of $\mathscr{C}$ is a generator matrix for the dual code $\mathscr{C}^\perp$.*

Let $\mathscr{C}$ be an $[n,k]$ code in $\mathbb{F}_q$ and $\mathbf{x}$ be any vector in $\mathbb{F}_q^n$, the set

$$\mathbf{x} + \mathscr{C} = \{\mathbf{x} + \mathbf{c} \mid \mathbf{c} \in \mathscr{C}\}$$

is called a *coset* of $\mathscr{C}$. Thus, two vectors $\mathbf{x}$ and $\mathbf{y}$ belong to the same coset if and only if $\mathbf{y} - \mathbf{x} \in \mathscr{C}$.

By Lagrange's Theorem, the cosets form a partition of the space $\mathbb{F}_q^n$ into q^{n-k} classes each containing q^k elements.

Theorem 1.12 (Lagrange's Theorem)**.** *Suppose $\mathscr{C}$ is an $[n,k]$ code in $\mathbb{F}_q^n$, then:*

- *Every vector of $\mathbb{F}_q^n$ is in some coset of $\mathscr{C}$.*
- *Every coset contains exactly q^k vectors.*
- *Any two cosets are either equivalent or disjoint.*

The *minimum weight of a coset* is the smallest Hamming weight among all vectors in the coset. Notice that while the minimum weight of a coset is well-defined, there may be more than one vector of that weight in the coset.

Definition 1.13. The words of minimal Hamming weight in the cosets of $\mathbb{F}_q^n/\mathscr{C}$ are the set of coset leaders for $\mathscr{C}$ in $\mathbb{F}_q^n$. We will denote by $\mathrm{CL}(\mathscr{C})$ the set of coset leaders of the code $\mathscr{C}$ and by $\mathrm{CL}(\mathbf{y})$ the subset of coset leaders corresponding to the coset $\mathscr{C} + \mathbf{y}$.

The zero vector is the unique coset leader of the code $\mathscr{C}$. Moreover, every coset of weight at most $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ has a unique coset leader.

Definition 1.14. Choose a parity check matrix H for $\mathscr{C}$. The *Syndrome* of a vector $\mathbf{x} \in \mathbb{F}_q^n$ with respect to the parity check matrix H is the vector $S(\mathbf{x}) = H\mathbf{x}^T \in \mathbb{F}_q^{n-k}$.

As the syndrome of a codeword is $\mathbf{0}$, then we have a way to test whether the vector belongs to the code. Moreover, two vectors $\mathbf{y}$ and $\mathbf{e}$ that differ by a codeword have the same syndrome, i.e.

$$H\mathbf{y}^T = H(\mathbf{c}+\mathbf{e})^T = \mathbf{0} + H\mathbf{e}^T = H\mathbf{e}^T.$$

Hence, there is a one-to-one correspondence between cosets of $\mathscr{C}$ and values of syndromes. The following is just a reformulation of our arguments:

Theorem 1.15. *Two vectors belong to the same coset if and only if they have the same syndrome.*

Since H has rank $n-k$, the number of cosets is q^{n-k}, that is, every vector in $\mathbb{F}_q^{n-k}$ is a syndrome.

Remark 1.16. It is well known that complete minimum distance decoding (CDP) over the code $\mathscr{C}$ has a unique solution for those vectors in the union of the Hamming balls of radius t around the codewords of $\mathscr{C}$ i.e. $B(\mathscr{C}, t) = \left\{\mathbf{y} \in \mathbb{F}_2^n \mid \mathbf{c} \in \mathscr{C} \text{ s.t. } d_H(\mathbf{c}, \mathbf{y}) \leq t\right\}$, where $t = \lfloor \frac{d-1}{2} \rfloor$ is the *error-correcting capacity* of $\mathscr{C}$ and $\lfloor \cdot \rfloor$ denotes the greatest integer function.

The following theorem gives us a nice relationship between the coset leaders.

Theorem 1.17. *Let* $\mathbf{w} \in \mathrm{CL}(\mathscr{C})$ *and* $\mathbf{y} \in \mathbb{F}_q^n$ *such that all nonzero components of* $\mathbf{y}$ *agree with the corresponding component of* $\mathbf{w}$*. Then* $\mathbf{y} \in \mathrm{CL}(\mathscr{C})$.

In other words, if there is a coset of weight s, there is also a coset of any weight less than s.

Proof. Consider $\mathbf{y} \in \mathbb{F}_q^n \setminus \{\mathbf{w}\}$ such that $y_i = w_i$ for all $i \in \mathrm{supp}(\mathbf{y})$. Therefore, $\mathrm{w}_H(\mathbf{w}) = \mathrm{w}_H(\mathbf{y}) + \mathrm{w}_H(\mathbf{w}-\mathbf{y})$. Suppose, contrary to our claim, that $\mathbf{y} \notin \mathrm{CL}(\mathscr{C})$. Then, there must exists a codeword $\mathbf{c} \in \mathscr{C}$ such that $\mathrm{w}_H(\mathbf{y}+\mathbf{c}) < \mathrm{w}_H(\mathbf{y})$. However,

$$\mathrm{w}_H(\mathbf{w}+\mathbf{c}) \leq \mathrm{w}_H(\mathbf{y}+\mathbf{c}) + \mathrm{w}_H(\mathbf{w}-\mathbf{y}) < \mathrm{w}_H(\mathbf{y}) + \mathrm{w}_H(\mathbf{w}-\mathbf{y}) = \mathrm{w}_H(\mathbf{w})$$

which contradicts the fact that $\mathbf{w} \in \mathrm{CL}(\mathscr{C})$.

□

In particular, for the binary case, we have the following result, where $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ denotes the canonical basis of $\mathbb{F}_2^n$.

Corollary 1.18. *Let* $\mathbf{w} \in \mathrm{CL}(\mathscr{C})$ *such that* $\mathbf{w} = \mathbf{y} + \mathbf{e}_i$ *for some word* $\mathbf{y} \in \mathbb{F}_2^n$ *and* $i \in \mathrm{supp}(\mathbf{w})$*, then* $\mathbf{y} \in \mathrm{CL}(\mathscr{C})$.

Proof. See [58, Corollary 11.7.7]. □

Definition 1.19. The *Voronoi region* of a codeword $\mathbf{c} \in \mathscr{C}$, denoted by $\mathrm{D}(\mathbf{c})$, is defined as:

$$\mathrm{D}(\mathbf{c}) = \left\{ \mathbf{y} \in \mathbb{F}_q^n \mid d_H(\mathbf{y}, \mathbf{c}) \leq d_H(\mathbf{y}, \mathbf{c}') \text{ for all } \mathbf{c}' \in \mathscr{C} \setminus \{\mathbf{c}\} \right\}.$$

Note that the set of Voronoi regions of a linear code $\mathscr{C}$ partitions the space $\mathbb{F}_q^n$. However, some points of $\mathbb{F}_q^n$ may be contained in several regions. Furthermore, the Voronoi region of the all-zero codeword $\mathrm{D}(\mathbf{0})$ coincides with the set of coset leaders of $\mathscr{C}$, i.e. $\mathrm{D}(\mathbf{0}) = \mathrm{CL}(\mathscr{C})$.

Definition 1.20. A *test-set* $\mathscr{T}_{\mathscr{C}}$ for a given code $\mathscr{C}$ is a set of codewords such that every word $\mathbf{y}$ either lies in the Voronoi region of the all-zero vector, denoted by $\mathrm{D}(\mathbf{0})$, or there exist $\mathbf{t} \in \mathscr{T}_{\mathscr{C}}$ such that $\mathrm{w}_H(\mathbf{y} - \mathbf{t}) < \mathrm{w}_H(\mathbf{y})$.

Definition 1.21. A non-zero codeword $\mathbf{m}$ in the code $\mathscr{C}$ is said to be a *minimal support codeword* if there is no other codeword $\mathbf{c} \in \mathscr{C}$ such that

$$\mathrm{supp}(\mathbf{c}) \subseteq \mathrm{supp}(\mathbf{m}).$$

We will denote by $\mathscr{M}_{\mathscr{C}}$ the set of codewords of minimal support of $\mathscr{C}$.

1.1.2 Generator matrices of modular codes

Throughout this section $\mathscr{C}$ will be a modular code defined over $\mathbb{Z}_m$ where $\mathbb{Z}_m$ denotes the ring of integers modulo m. In other words, $\mathscr{C}$ is a submodule of $\left(\mathbb{Z}_m^n, +\right)$.

Let $m = \prod p_i^{r_i}$ be the prime factorization of m. Along this thesis p will denote a prime number, unless otherwise specified. For any divisor q of m let

$$\Phi_q : \quad \mathbb{Z}_m \quad \longrightarrow \quad \mathbb{Z}_q$$

be the map defined by $\Phi_q(\mathbf{c}) = \mathbf{c} \mod q$.

Definition 1.22. The vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{Z}_{p^r}$ are *modular independent* if and only if $\sum_{i=1}^k a_i \mathbf{v}_i = \mathbf{0}$ implies that p / a_i for all i.

Therefore, the zero vector is modular dependent and any nonzero vector $\mathbf{v}$ is modular independent. Thus, if $\mathbf{v}_1, \ldots, \mathbf{v}_s$ are modular independent, then $\mathbf{v}_i \neq \mathbf{0}$ for all i. Moreover $\mathbf{v}_1, \ldots, \mathbf{v}_s$ are modular dependent if and only if some $\mathbf{v}_j$ can be written as a linear combination of the other vectors (for a proof see, for example, [39, Lemma 3.1]).

We use the definition of modular independence of vectors in $\mathbb{Z}_{p^r}$ to define modular independence in $\mathbb{Z}_m^n$ as follows:

Definition 1.23. The vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ in $\mathbb{Z}_m^n$ are *modular independent* if the vectors $\Phi_i(\mathbf{v}_1), \ldots, \Phi_i(\mathbf{v}_k)$ are modular independent for some i.

On [39, Theorem 3.2] it is shown that if $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are modular independent over $\mathbb{Z}_m$ and $\sum_{j=1}^k a_j \mathbf{v}_j = \mathbf{0}$, then all α_j are nonunits, but the converse is not true. This motivates the following definition:

Definition 1.24. The vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ in $\mathbb{Z}_m^n$ are *independent* if $\sum_{j=1}^{k} a_j \mathbf{v}_j = \mathbf{0}$ implies that $a_j \mathbf{v}_j = \mathbf{0}$ for all j.

Note that if $\mathbf{v}_1, \ldots, \mathbf{v}_r$ are independent, then so are $\mathbf{v}_1, \ldots, \mathbf{v}_r, \mathbf{0}$. Moreover if $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are independent and $\lambda \mathbf{w} \notin \langle \mathbf{v}_1, \ldots, \mathbf{v}_k \rangle$ for any $\lambda \neq 0$, then $\mathbf{v}_1, \ldots, \mathbf{v}_k, \mathbf{w}$ are independent.

It is easy to see that, if the nonzero vectors $\mathbf{v}_1, \ldots, \mathbf{v}_s$ in $\mathbb{Z}_{p^e}^n$ are independent, then they are modular independent over $\mathbb{Z}_{p^e}$. However the converse does not hold. Moreover, over $\mathbb{Z}_m$, modular independence does not imply independence nor does independence imply modular independence.

Example 1.25. Consider the vectors $(11,7)$ and $(3,9)$ in $\mathbb{Z}_{12}^2$. By definition, they are modular independent over $\mathbb{Z}_{12}$, since $\Phi_4((11,7)) = (3,3)$ and $\Phi_4((3,9)) = (3,1)$ are modular independent over $\mathbb{Z}_4$. However they are not independent over $\mathbb{Z}_{12}$ since $6(11,7) + 2(3,9) = (0,0)$ but $6(11,7) = 2(3,9) = (6,6) \neq (0,0)$.

If now we consider the vectors $(4,0)$ and $(0,3)$ in $\mathbb{Z}_{12}^2$. They are independent since $a(4,0) + b(0,3) = (0,0)$ implies that $4a \equiv 0$ and $3b \equiv 0$, i.e. $a \in \{0,3,6,9\}$ and $b \in \{0,4,8\}$, so $a(4,0) = (0,0)$ and $b(0,3) = (0,0)$. But $(4,0)$ and $(0,3)$ are modular dependent over $\mathbb{Z}_{12}$ since $\Phi_4((4,0)) = (0,0)$ and $\Phi_3((0,3)) = (0,0)$ so they are modular dependent over $\mathbb{Z}_4$ and $\mathbb{Z}_3$.

Definition 1.26. Let $\mathscr{C}$ be a code over $\mathbb{Z}_m$. The codewords $\mathbf{w}_1, \ldots, \mathbf{w}_k$ form a *basis* for $\mathscr{C}$ if they are independent, modular independent and generate $\mathscr{C}$.

This allow us to make the following definition:

Definition 1.27. A matrix G is called a *generator matrix* for $\mathscr{C}$ if the row vectors form a basis for $\mathscr{C}$.

From the fundamental theorem on finitely generated abelian groups, if $\mathscr{C}$ is a code of length n over $\mathbb{Z}_m$, then:

$$\mathscr{C} \cong \mathbb{Z}_m/\langle d_1 \rangle \oplus \ldots \oplus \mathbb{Z}_m/\langle d_r \rangle$$

where all d_i are nonunits and $\langle d_r \rangle \subseteq \langle d_{r-1} \rangle \subseteq \ldots \subseteq \langle d_1 \rangle$ with $r \leq n$ are uniquely determined. Or equivalently, $1 < d_1/d_2/\ldots/d_r/m$, with $r \leq n$.

We can use this notion of modular basis to develop the already-known results for the field case such as:

1. Every code has a basis. Indeed we have the following theorem:

   Theorem 1.28. *Suppose* $\phi : \mathscr{C} \longrightarrow \mathbb{Z}_m/\langle d_1 \rangle \oplus \ldots \oplus \mathbb{Z}_m/\langle d_r \rangle$ *as before. Let* $\mathbf{w}_i$ *be the codeword in* $\mathscr{C}$ *corresponding to* $(0, \ldots, 1, \ldots, 0)$ *in the direct product, where* $1 \in \mathbb{Z}_m/\langle d_i \rangle$ *is in the ith place. Then* $\mathbf{w}_1, \ldots, \mathbf{w}_r$ *form a basis for* $\mathscr{C}$.

   The proof follows from [39, Theorem 4.6] or [92, Theorem 4.12].

2. Any two bases have the same number of codewords, which enable us to define the *rank* to be the number of codewords in a basis, for a proof see [39, Theorem 4.7] or [92, Theorem 4.13].

However, note that r modular independent codewords in a code of rank r do not necessarily form a basis. Moreover, s modular independent codewords with $s < r$ are not always possible to be extended to a basis for a code of rank r.

Remark 1.29. Codes over the rings $\mathbb{Z}_{p^r}$ have generator matrices permutation equivalent to the standard form

$$G = \begin{pmatrix} I_{k_0} & A_{0,1} & A_{0,2} & A_{0,3} & \dots & A_{0,r-1} & A_{0,r} \\ 0 & pI_{k_1} & pA_{1,2} & pA_{1,3} & \dots & pA_{1,r-1} & pA_{1,r} \\ 0 & 0 & p^2 I_{k_2} & p^2 A_{2,3} & \dots & p^2 A_{2,r-1} & pA_{2,r} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & p^{r-1} I_{k_{r-1}} & p^{r-1} A_{r-1,r} \end{pmatrix}.$$

In [92] they define standard forms for codes over $\mathbb{Z}_m$ using the Chinese Remainder Theorem which does not necessarily coincide with a matrix with the previous form.

1.1.3 The general decoding problem

For the decoding problem we would like that $D(E(\mathbf{m})+\text{noise}) = \mathbf{m}$ for every message $\mathbf{m}$ and every reasonable noise pattern.

Maximum Likelihood decoding (MLD) is the most powerful decoding method from the point of view of transmission reliability. However, it has a complexity that grows exponentially with the length of the code. MLD can be described as the method where given a received word $\mathbf{y} \in \mathbb{F}_q^n$, try to find a codeword $\mathbf{x}$ that maximizes the probability that $\mathbf{y}$ was received given that $\mathbf{x}$ was sent. On a symmetric channel the MLD becomes the *Minimum distance decoding* (MDD) whose goal is to output the codeword closest in Hamming distance to the received word.

A decoding algorithm which decodes only up to the number of errors for which it is guaranteed to find a unique codeword within such a distance of the received word, is called a unique decoding sequel. In particular, a linear code with minimum distance d has a *unique decoding algorithm* that can correct up to $t = \frac{d-1}{2}$ errors (*error-correcting capability* of the code). When the number of errors is greater than t then the unique decoding algorithms could either output a wrong codeword or report a decoding failure and not output any codeword. However, when we have a decoder capable of finding all codewords nearest to the received vector then we have a *complete decoder*.

In general, *complete decoding* for a linear code has proved to be an NP-hard computational problem. For binary codes see [6] and for the q-ary case see [3, Theorem 4.1]. Recall that the complexity is measured by the number of operations (time complexity) and the amount of memory used (space complexity).

We are interested in *complete minimum distance decoding* (CDP), or equivalently given a received vector $\mathbf{y} \in \mathbb{F}_q^n$, find one of the closest codewords in $\mathscr{C}$. The first

idea to accomplish this procedure could be to compute the hamming distance of the received word with all the codewords (recall that a linear code has q^k codewords). The complexity of this brute force method is $\mathcal{O}\left(nq^k\right)$. Thus, large parameters make any brute force method wildly impractical.

Known decoding methods with complexity asymptotically less than that of exhaustive search can be divided mainly into three groups:

- **Syndrome decoding**.

  Its trivial implementation can be performed as follows:

  1. We construct the syndrome lookup table (i.e. we enumerate the cosets of $\mathscr{C}$ in $\mathbb{F}_q^n$, we choose a coset leader for each coset and we compute its syndrome).
  2. Compute the syndrome $S(\mathbf{y})$ of the received vector $\mathbf{y} \in \mathbb{F}_q^n$ and determine from the table which coset leader $\mathbf{e}$ satisfies that $S(\mathbf{y}) = S(\mathbf{e})$.
  3. Decode $\mathbf{y}$ as $\mathbf{y} - \mathbf{e}$.

  The space complexity of this method is $\mathcal{O}\left(nq^{n-k}\right)$. Therefore, the computation of the look-up table grows exponentially with the length of the code. However, if pre-computation is allowed, this algorithm is fast.

- **Gradient Descent decoding**.

  The general principle of these methods is the use of a certain set of codewords $\mathscr{T}_{\mathscr{C}}$ (namely test-set, formally described in Definition 1.20) which has been precomputed and stored in memory in advance. Then the algorithm can be accomplished by recursively inspecting the test-set for the existence of an adequate element which is subtracted from the current vector. Algorithm 1 describes a gradient-like decoding algorithm for binary codes, this algorithm appears in [3].

Algorithm 1: Gradient-like decoding

Data: The received word $\mathbf{y} \in \mathbb{F}_2^n$.
Result: A codeword $\mathbf{c} \in \mathscr{C}$ that minimized the Hamming distance $\mathrm{d}_H(\mathbf{c}, \mathbf{y})$.

```
Set c = 0;
while y ∉ D(0) do
    Look for z ∈ T_C such that w_H(y − z) < w_H(y);
    y ⟵ y − z;
    c ⟵ c + z;
endw
Return c = y.
```

The following result shows that Algorithm 1 performs complete minimum-distance decoding.

Theorem 1.30. *(Correctness of Algorithm 1) Algorithm 1 always converges to the nearest codeword.*

Proof. We follow [3, Theorem 3.11]. Let $\mathbf{y} \notin D(\mathbf{0})$. By construction, there must exists an element $\mathbf{z}_1 \in \mathscr{T}_{\mathscr{C}}$ such that $\mathrm{w}_H(\mathbf{y}-\mathbf{z}_1) < \mathrm{w}_H(\mathbf{y})$. We replace $\mathbf{y}$ by $\mathbf{y}_1 = \mathbf{y}-\mathbf{z}_1$.

Repeating the above process a finite number of times we must arrive to a vector $\mathbf{y}_m$ with the following property:

$$\mathbf{y}_m = \mathbf{y} - \sum_{i=1}^{m} \mathbf{z}_m \in D(\mathbf{0}).$$

Therefore $\mathbf{y} \in D\left(\sum_{i=1}^{m} \mathbf{z}_m\right)$, which completes the proof. □

The idea behind this is *step by step* decoding which is an old but quite recurrent technique. A primer study on it can be found in [94].

The time complexity of the algorithm is $\mathscr{O}\left(n^2|\mathscr{T}_{\mathscr{C}}|\right)$ and the space complexity is $\mathscr{O}\left(n|\mathscr{T}_{\mathscr{C}}|\right)$, see [3, Theorem 3.11]. Making use of the ideas of the zero-neighbours algorithm it is possible to reduce the number of codewords inspected by the gradient-like algorithm. Moreover, for binary codes if $\mathscr{T}_{\mathscr{C}} = \mathscr{M}_{\mathscr{C}}$ then this version is called *minimal-vector decoding*.

Theorem 1.31. *The minimal-vector algorithm for binary codes performs complete minimum distance decoding.*

Proof. This proof appears in [3, Theorem 3.13]. Assume that $\mathbf{y} \notin D(\mathbf{0})$, or equivalently, there exists a nonzero codeword $\mathbf{c} \in \mathscr{C}$ such that

$$\mathrm{w}_H(\mathbf{y}+\mathbf{c}) < \mathrm{w}_H(\mathbf{c}) \tag{1.1}$$

It follows easily that every nonzero codeword of $\mathscr{C}$ can be written as

$$\mathbf{c} = \sum_{i=1}^{m} \mathbf{z}_m \text{ with } \mathbf{z}_m \in \mathscr{T}_{\mathscr{C}} \text{ and } m \geq 1$$

where $\mathrm{w}_H(\mathbf{c}) > \mathrm{w}_H(\mathbf{c}-\mathbf{z}_1) > \ldots > \mathrm{w}_H(\mathbf{c}-\sum_{i=1}^{m} \mathbf{z}_m) = 0$.

In our case $\mathscr{T}_{\mathscr{C}} = \mathscr{M}_{\mathscr{C}}$, that is, $\mathbf{c}$ can be described as sum of codewords of minimal support. Note that on the binary case the supports of codewords from $\mathscr{M}_{\mathscr{C}}$ do not intersect. Hence there must exists at least one codewords $\mathbf{z} \in \mathscr{M}_{\mathscr{C}}$ satisfying Equation 1.1. Then the proof follows from Theorem 1.30. □

Gradient Descent Decoding has gained renewed interest over the past 20 years thanks to the efficient implementation achieved with turbo codes and low-density parity check (LDPC) codes. In fact, see [116] for gradient descent procedures based in bit flipping, particularly beneficial for LDPC codes.

- **Information-set decoding**

  The idea of these methods is to decode by localizing errors. Fundamentally, we look for a set of coordinates which are error-free in such a way that the restriction of a code's generator matrix to these positions is invertible and the original message can be retrieved by multiplying the received vector and the inverse of the mentioned submatrix.

  For any vector $\mathbf{x}$ in $\mathbb{F}_q^n$, we denote by $\mathbf{x}_I$ the restriction of $\mathbf{x}$ to the coordinates indexed by I. Similarly, let A be a matrix with at least n columns we denote by A_I the limitation of A to the columns indexed by I.

  Now we can define more precisely the principle of these procedures. Let $\mathbf{y}$ be the received vector and I some information-set of our code $\mathscr{C}$, then $\mathbf{y}$ can be written as $\mathbf{y} = \mathbf{c} + \mathbf{e}$ for some codeword $\mathbf{c}$ and some error vector $\mathbf{e}$ verifying that $\mathrm{supp}(\mathbf{e}) \cap I = \emptyset$ or equivalently $\mathbf{y}_I = \mathbf{c}_I$. Moreover, the restriction of the matrix $G_I^{(-1)}G$, where G denotes a generator matrix for $\mathscr{C}$, to the columns indexed by the set I match the identity matrix I_k of size k. Therefore, the original message can be found as $\mathbf{m} = \mathbf{y}_I G_I^{(-1)}$.

  The first approach to this method was introduced by Prange in [95]. Subsequent variants were devised by McEliece [86], by Lee and Brickell [69], by Leon [70] and by Stern [105]. Latter improvements to Stern's and Lee-Brickell's attack were presented independently by van Tilborg, Canteaut, Chabaud, Chabanne, Finiasz and Sendrier in [27, 28, 29, 43, 113, 114]. As a culmination of the previous improvement, Bernstein, Lange and Peters presented a new advance in [7]. Moreover, Peters in [93] generalizes Lee-Brickell's algorithm and Stern's algorithm to the non-binary case.

  More recent improvements include [4, 8, 44, 85] which give asymptotically exponential improvements over Stern's algorithm.

  Note that information-set decoding, though much more efficient than a brute-force search, still needs exponential time in the code length.

Definition 1.32. Given a received word $\mathbf{y}$ and an error-correcting capability w, a *list-decoding algorithm* outputs a (possibly empty) list of all codewords within a Hamming distance of w to $\mathbf{y}$. The decoding is considered successful if the transmitted codeword is included in the list.

This method was introduced independently by Elias [41] and Wozencraft [121]. An important parameter of list-decoding is the size of the list that the decoder is allowed to output. Note that if the size is equal to one, then list-decoding coincides with unique decoding, otherwise list-decoding only gives more options than unique decoding.

1.2 Gröbner Bases and Border Bases

Let $\mathbb{K}$ denote an arbitrary field and $\mathbb{K}[\mathbf{X}] = \mathbb{K}[X_1, \ldots, X_n]$ denote the polynomial ring in n variables over the field $\mathbb{K}$. Let $[\mathbf{X}]$ be the set of terms in $\mathbb{K}[\mathbf{X}]$, i.e.

$$[\mathbf{X}] = \left\{ \mathbf{X}^{\mathbf{a}} = X_1^{a_1} \cdots X_n^{a_n} \mid \mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n \right\}$$

which is a multiplicative version of the additive semigroup $\mathbb{N}^n$. Note that, $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ being elements in $\mathbb{N}^n$, implies that:

- $\mathbf{X}^{\mathbf{a}} \cdot \mathbf{X}^{\mathbf{b}} = \mathbf{X}^{\mathbf{c}} \iff \mathbf{a} + \mathbf{b} = \mathbf{c}$ i.e. $a_i + b_i = c_i$
- $\mathbf{X}^{\mathbf{a}} / \mathbf{X}^{\mathbf{b}} \iff \mathbf{a} \leq \mathbf{b}$ i.e. $a_i \leq b_i$

where $<$ is a natural partial ordering in $\mathbb{N}^n$.

A *term order* on $\mathbb{K}[\mathbf{X}]$ is a total order $\prec$ on the set $[\mathbf{X}]$ such that

$$\mathbf{X}^{\mathbf{a}} \prec \mathbf{X}^{\mathbf{b}} \quad \text{implies that} \quad \mathbf{X}^{\mathbf{a}+\mathbf{c}} \prec \mathbf{X}^{\mathbf{b}+\mathbf{c}}.$$

Moreover, if $1 \prec \mathbf{X}^{\mathbf{a}}$ for all $\mathbf{a} \in \mathbb{N}^n \setminus \{\mathbf{0}\}$, then $\prec$ is a *well-ordering* on $[\mathbf{X}]$.

Let us fix an ordering on the variables, such as $X_1 > X_2 > \ldots > X_n$. The most common examples of term order are:

- The *lexicographical* (lex) ordering defined by

$$\mathbf{X}^{\mathbf{a}} <_{\text{lex}} \mathbf{X}^{\mathbf{b}} \iff \exists j \text{ such that } \left(a_i = b_i \text{ for } i > j\right) \text{ and } \left(a_j < b_j\right)$$

- The *reverse lexicographical* (lexrev) ordering defined by

$$\mathbf{X}^{\mathbf{a}} <_{\text{lexrev}} \mathbf{X}^{\mathbf{b}} \iff \exists j \text{ such that } \left(a_i = b_i \text{ for } i < j\right) \text{ and } \left(a_j > b_j\right)$$

  It is not a well-ordering since $X_1 < 1$.

- The *degree lexicographical* (deglex) ordering defined by

$$\mathbf{X}^{\mathbf{a}} <_{\text{deglex}} \mathbf{X}^{\mathbf{b}} \iff \begin{cases} \deg(\mathbf{X}^{\mathbf{a}}) > \deg(\mathbf{X}^{\mathbf{b}}) \text{ or} \\ \deg(\mathbf{X}^{\mathbf{a}}) = \deg(\mathbf{X}^{\mathbf{b}}) \text{ and } \mathbf{X}^{\mathbf{a}} <_{\text{lex}} \mathbf{X}^{\mathbf{b}} \end{cases}$$

- The *degree lexicographical* (degrevlex) ordering defined by

$$\mathbf{X}^{\mathbf{a}} <_{\text{degrevlex}} \mathbf{X}^{\mathbf{b}} \iff \begin{cases} \deg(\mathbf{X}^{\mathbf{a}}) > \deg(\mathbf{X}^{\mathbf{b}}) \text{ or} \\ \deg(\mathbf{X}^{\mathbf{a}}) = \deg(\mathbf{X}^{\mathbf{b}}) \text{ and } \mathbf{X}^{\mathbf{a}} <_{\text{revlex}} \mathbf{X}^{\mathbf{b}} \end{cases}$$

More in general, given an ordering $\prec$ on $[\mathbf{X}]$ we define its *total degree ordering* extension $\prec_T$ as:

$$\mathbf{X}^{\mathbf{a}} \prec_T \mathbf{X}^{\mathbf{b}} \iff \begin{cases} \deg(\mathbf{X}^{\mathbf{a}}) < \deg(\mathbf{X}^{\mathbf{b}}) \text{ or} \\ \deg(\mathbf{X}^{\mathbf{a}}) = \deg(\mathbf{X}^{\mathbf{b}}) \text{ and } \mathbf{X}^{\mathbf{a}} \prec \mathbf{X}^{\mathbf{b}} \end{cases}$$

Now we fix a term order $\prec$ on $\mathbb{K}[\mathbf{X}]$. Let $f = \sum_{t \in [\mathbf{X}]} c_t t$ be a nonzero polynomial in $\mathbb{K}[\mathbf{X}]$.

- The *support* of f is the set of terms which occurs with nonzero coefficient in the expansion of f, i.e. $\mathrm{supp}(f) = \{t \in [\mathbf{X}] \mid c_t \neq 0\}$.
- The *leading term* of f, denoted by $\mathrm{LT}_\prec(f)$, is the term t in f such that $t \succ t'$ for all $t' \in \mathrm{supp}(f)$.
- The *leading coefficient* of f is $\mathrm{LC}_\prec(f) = c_{\mathrm{LT}_\prec(f)}$.
- The *leading monomial* of f is $\mathrm{LM}_\prec(f) = \mathrm{LC}_\prec(f) \cdot \mathrm{LT}_\prec(f)$.

Let F be a finite set of polynomials in $\mathbb{K}[\mathbf{X}]$. Then $\mathrm{LT}_\prec\{F\} = \{\mathrm{LT}_\prec(f) \mid f \in F\}$ denotes the set of leading terms of all non-zero polynomials in F and $\mathrm{LT}_\prec(F)$ denotes the semigroup ideal generated by $\mathrm{LT}_\prec\{F\}$, i.e. $\mathrm{LT}_\prec(F) = \{t \cdot \mathrm{LT}_\prec(f) \mid t \in [\mathbf{X}], f \in F\}$.

Remark 1.33. A *semigroup ideal* S (resp. Σ) is a subset of $\mathbb{K}[\mathbf{X}]$ (resp. $\mathbb{N}^n$) such that if $s \in S$ and $t \in [\mathbf{X}]$, then $ts \in S$ (resp. if $a \in \Sigma$ and $b \in \mathbb{N}^n$ such that $a \leq b$, then $b \in \Sigma$).

Algorithm 2 describes a division algorithm for multivariate polynomials. This algorithm is taken from [35].

Algorithm 2 terminates since a term order on $\mathbb{K}[\mathbf{X}]$ is a well ordering.

Remark 1.34. Dividing f by $(f_1, \ldots, f_s)$ w.r.t. $\prec$ gives a unique expression of the form

$$f = a_1 f_1 + \ldots + a_s f_s + r \text{ where } \deg(f) \geq \deg(a_i f_i) \text{ for } i \in \{1, \ldots, s\}.$$

However, $a_1, \ldots, a_s, r \in \mathbb{K}[\mathbf{X}]$ depend on the chosen term order $\prec$.

Example 1.35. Let $f = xy^2 - x$, $f_1 = xy + 1$ and $f_2 = y^2 - 1$ and take $\prec$ to be the `lex` ordering with $x > y$. Then:

- Dividing f by (f_1, f_2) yields: $xy^2 - x = y(xy + 1) + (-x - y)$.
- Dividing f by (f_2, f_1) yields: $xy^2 - x = x(y^2 - 1) + 0$.

Definition 1.36. A non empty subset $I \subset \mathbb{K}[\mathbf{X}]$ is called a *polynomial ideal* of $\mathbb{K}[\mathbf{X}]$ if I is closed under:

- Addition, i.e. for all $f,\ g \in I$, $f + g \in I$.
- Multiplication by elements of $\mathbb{K}[\mathbf{X}]$, i.e. $h \in \mathbb{K}[\mathbf{X}]$ and $f \in I$, implies that $hf \in I$.

By the well-known Hilbert basis theorem [35, Section2.5] every polynomial ideal I has a finite generating set. That is, $I = \langle f_1, \ldots, f_s \rangle$ for some $f_1, \ldots, f_s \in I$.

Let I be a nonzero ideal in $\mathbb{K}[\mathbf{X}]$. Then its *initial ideal* w.r.t. a term ordering $\prec$, denoted by $\mathrm{in}_\prec(I)$, is the ideal generated by the leading terms of all the polynomials in I:

$$\mathrm{in}_\prec(I) = \langle \mathrm{LT}_\prec(f) \mid f \in I \rangle .$$

The terms which do not lie in the ideal $\mathrm{in}_\prec(I)$ are called *standard monomials* .

Let F be a finite generating set for I. Note that $\langle \mathrm{LT}_\prec\{F\} \rangle \subset \mathrm{in}_\prec(I)$. However $\langle \mathrm{LT}_\prec\{F\} \rangle$ and $\mathrm{in}_\prec(I)$ may be different ideals, as we see by the following counterexample.

Algorithm 2: A division algorithm in $\mathbb{K}[\mathbf{X}]$

Data: A dividen $f \in \mathbb{K}[\mathbf{X}]$, an ordered set of divisors $\{f_1, \ldots, f_s\} \subseteq \mathbb{K}[\mathbf{X}]$ and a term order $\prec$.

Result: The quotients $a_1, \ldots, a_s$ and the remainder r in $\mathbb{K}[\mathbf{X}]$ such that $f = a_1 f_1 + \ldots + a_s f_s + r$ where no term in r is a multiple of $\mathrm{LT}_\prec(f_i)$ for $i \in \{1, \ldots, s\}$.

1 **Initialization**
2 $a_1 := 0, \ldots, a_s := 0, r := 0$;
// Each of the quotient registers and the remainder is set to zero;
3 $p := f$;
// The dividend is f;
4 $G_2 \longleftarrow \emptyset$; `List` $\longleftarrow \{1\}$; $N \longleftarrow \emptyset$; $r \longleftarrow 0$;
5 **while** $p \neq 0$ **do**
6 $i := 1$;
7 `divisionoccurred` := `false`;
8 **while** $i \leq s$ *and* `divisionoccurred` = `false` **do**
// Find the smallest value $i \in \{1, \ldots, s\}$ (if any) for which $\mathrm{LT}_\prec(p)$ is a multiple of $\mathrm{LT}_\prec(f_i)$;
9 **if** $\mathrm{LT}_\prec(f_i)$ *divides* $\mathrm{LT}_\prec(p)$ **then**
10 $a_i := a_i + \frac{\mathrm{LT}_\prec(p)}{\mathrm{LT}(f_i)}$;
11 $p := p - \frac{\mathrm{LT}_\prec(p)}{\mathrm{LT}(f_i)} f_i$;
12 `divisionoccurred` := `true`;
13 **else**
14 $i := i + 1$;
15 **endif**
16 **if** `divisionoccurred` = `false` **then**
// If there is no such i subtract $\mathrm{LT}_\prec(p)$ from the dividend and add it to the remainder $r := r + \mathrm{LT}(p)$;
17 $p := p - \mathrm{LT}(p)$;
18 **endif**
19 **endw**
20 **endw**

Example 1.37. Let $f = xy^2 + xy$ and $g = x^2y + x^2 - y$ be polynomials in $\mathbb{K}[x, y]$ and take $\prec$ to be the `degrevlex` ordering with $x > y$. Note that $h = y^2$ is an element of the ideal I generated by f and g since $h = xf - yg$. However, $h \notin \langle \mathrm{LT}_\prec(f), \mathrm{LT}_\prec(g) \rangle$.

1.2.1 Gröbner Bases

Definition 1.38. A finite subset $G_\prec = \{g_1, \ldots, g_s\}$ of an ideal I is a Gröbner basis w.r.t. the term order $\prec$ if

$$\mathrm{in}_\prec(I) = \langle \mathrm{LT}_\prec(g_1), \ldots, \mathrm{LT}_\prec(g_s) \rangle = \langle \mathrm{LT}_\prec\{G\} \rangle$$

Note that if $G_\prec$ is a Gröbner basis for I, then $G_\prec$ is a basis of I, and any finite subset of I that contains $G_\prec$ is also a Gröbner basis. To remedy the non-minimality, we say that $G_\prec$ is a *reduced Gröbner basis* if

1. g_i are monic for all $i \in \{1, \ldots, s\}$.
2. For all distinct pairs $i, j \in \{1, \ldots, s\}$. None of the monomials appearing in the expansion of g_j is divisible by $\mathrm{LT}_\prec(g_i)$.

Remark 1.39. A note on existence and uniqueness of Gröbner Bases. Fix a term order $\prec$:

1. Every ideal $I \subseteq K[\mathbf{X}]$ has a Gröbner basis w.r.t. $\prec$ (Corollary of Hilbert basis theorem).
2. Any Gröbner basis for an ideal I is a system of generators of I.
3. Every non-zero ideal has a unique reduced Gröbner basis w.r.t. $\prec$, see for instance [35, Section 2.7].

Definition 1.40. We write $\mathrm{Red}_\prec(f, F)$ for the remainder on the division of f by the list of polynomials $F = \{f_1, \ldots, f_s\}$ w.r.t. the term order $\prec$.

Let $\mathscr{G}$ be a Gröbner basis for an ideal I, then $\mathrm{Red}_\prec(f, \mathscr{G})$ is called the *normal form* of f in I w.r.t. the term order $\prec$.

Given an ideal $I = \langle f_1, \ldots, f_s \rangle$ and a polynomial $f \in \mathbb{K}[\mathbf{X}]$, how can we test whether f lies in I or not? This problem is called the *ideal membership problem* and can be solved using Gröbner basis techniques. The idea is simple, fix a term ordering $\prec$ and let $G_\prec$ be a Gröbner basis for I w.r.t. $\prec$ then $f \in I$ if and only if $\mathrm{Red}_\prec(f, G_\prec) = 0$.

Moreover, Gröbner Bases provide a uniform approach to *solve systems of polynomials in several variables*. Let $F = \{f_1, \ldots, f_s\}$ be a finite set of polynomials in $\mathbb{K}[\mathbf{X}]$. Finding all common solutions in $\mathbb{K}^n$ of the system

$$f_1(X_1, \ldots, X_n) \;=\; \ldots \;=\; f_s(X_1, \ldots, X_n) \;=\; 0$$

is the same problem as searching for all the points in the affine variety $V(f_1, \ldots, f_s) = V(F)$. Since the definition of $V(F)$ is independent of the chosen generating set, then the reduced Gröbner basis $G_\prec$ for the ideal $I = \langle F \rangle$ w.r.t. a fix term ordering $\prec$ specifies the same variety, i.e. $V(F) = V(G)$. The advantage of G is that it reveals some geometric properties of the variety that are not visible from F. Observe that:

- By Hilbert's Nullstellensatz, $V(F)$ is empty if and only if $G_\prec$ is equal $\{1\}$.

- $V(F)$ is finite if and only if the set of standard monomials is finite.

Moreover, the number of zeros counted with multiplicity of $V(F)$ is equal to the number of standard monomials. Note that for $n = 1$, this yields to the *Fundamental Theorem of Algebra*.

Another problem where Gröbner Bases is very useful is the *implicitization problem*, i.e. compute the implicit representations of varieties given by rational parametrizations. Suppose that the rational parametrization, which define a subset of an algebraic variety V in $\mathbb{K}^n$, is given by a polynomial parametrization, i.e.

$$\begin{cases} X_1 = f_1(t_1, \ldots, t_m) \\ \vdots \\ X_n = f_n(t_1, \ldots, t_m) \end{cases}$$

How can we find polynomial equations in the X_i that define V? One way to solve this problem is to use elimination theory, that is, to compute a Gröbner basis of the ideal

$$I = \langle X_1 - f_1, \ldots, X_n - f_n \rangle \cap \mathbb{K}[\mathbf{X}]$$

using an elimination ordering for the variables $\mathbf{t}$, e.g. a lexicographic ordering $t_1 > \ldots > t_m > X_1 > \ldots > X_n$.

A Gröbner basis $\mathcal{G}$ for I can be computed from any generating set of I by a method that was introduced by Bruno Buchberger in [24] Buchberger was the first to give an algorithm for computing Gröbner Bases.

Algorithm 3: Buchberger's Algorithm

Data: The generating set $F = \{f_1, \ldots, f_s\}$ of a nonzero ideal I.
Result: A Gröbner basis $G = \{g_1, \ldots, g_t\}$ for I.

```
1  G := F;
2  repeat
       // Repeat the process until all S-polynomials of G have remainder 0 after
       division by G;
3      G := G';
4      for each pair (p, q), p ≠ q in G'
5          S := Red_≺ (S(p, q), G');
6          if S ≠ 0 then
               // If S ≠ 0 add S to G and start again;
7              G := G ∪ {S};
8          endif
9      endfor
10 until G = G' ;
```

Definition 1.41. Let $f, g \in \mathbb{K}[\mathbf{X}]$ be nonzero polynomials and $\prec$ be any term ordering. The S-polynomial of f and g w.r.t. $\prec$ is the combination:

$$S_{\prec}(f,g) = \frac{\mathbf{X}^{\gamma}}{\mathrm{LT}_{\prec}(f)} f - \frac{\mathbf{X}^{\gamma}}{\mathrm{LT}_{\prec}(g)} g$$

where $\mathbf{X}^{\gamma}$ is the least common multiple of $\mathrm{LT}_{\prec}(f)$ and $\mathrm{LT}_{\prec}(g)$.

We can now describe Buchberger's algorithm. This algorithm hinges on the fact that $\mathscr{G}$ is a Gröbner basis w.r.t $\prec$ if and only if for each pair $f, g \in \mathscr{G}$ we have that $\mathrm{Red}_{\prec}\left(S(f,g), \mathscr{G}_{\prec}\right) = 0$. We reproduce the algorithm from [35].

FGLM algorithm

The FGLM algorithm [42] is a method for changing a Gröbner basis from one term order to another by means of linear algebra techniques. This method applies to Gröbner Bases of zero-dimensional ideals and is based on the following key ideas:

- The quotient ring with respect to an ideal may be viewed as a vector space.
- For zero-dimensional ideals I, the quotient ring $\mathbb{K}[\mathbf{X}]/I$ is finitely generated and any Gröbner basis provides a canonical vector-space basis.
- The elements that are not in the vector-space are linearly dependent from the elements in the basis. Moreover, the dependency relations yields to polynomials of the Gröbner basis.

We require the following subroutines:

- `NormalForm`(f) returns the normal form of f w.r.t. the old basis G_1.
- `NextTerm[List]` removes the first element of the list `List` and returns it.
- `InsertNext`$[w,$`List`$]$ inserts to the list `List` the products wx for $x \in [\mathbf{X}]_1$, sorts the list `List` by increasing ordering for $\prec_1$ and removes duplicates.
- `LinearDependency`$[v, \{v_1, \ldots, v_r\}]$ returns `false` if v is not a linear combination of $\{v_1, \ldots, v_r\}$ and the set $\{\lambda_1, \ldots, \lambda_r\} \subset \mathbb{K}$ if $v = \sum_{i=1}^{r} \lambda_i v_i$.

A proof of its correctness may be found in [42].

Let I be a zero-dimensional ideal with codimension $D(I)$ in $\mathbb{K}[\mathbf{X}]$ i.e. $D(I)$ is the dimension of the $\mathbb{K}$-vector space $\mathbb{K}[\mathbf{X}]/I$. Let n be the number of variables of the polynomial ring $\mathbb{K}[\mathbf{X}]$.

Theorem 1.42. *Let G_1 be a Gröbner basis of I w.r.t. a term order $\prec_1$. Given a different term order $\prec_2$, then a reduced Gröbner basis G_2 of I w.r.t. $\prec_2$ can be calculated in $\mathcal{O}\left(nD(I)^3\right)$ arithmetic operations.*

Proof. See for instance [42, Theorem 5.1]. □

Algorithm 4: Algorithm FGLM

Data: A Gröbner basis G_1 of a zero-dimensional ideal I w.r.t. a term order $\prec_1$ and a new admissible order $\prec_2$.
Result: The reduced Gröbner basis G_2 of I w.r.t. $\prec_2$

```
G_2 ⟵ ∅; List ⟵ {1}; N ⟵ ∅; r ⟵ 0;
while List ≠ ∅ do
    w ⟵ NextTerm(List);
    if w ∉ LT_{≺_2}(G_2) then
        v = NormalForm(w);
        if LinearDependency[v, {v_1, ..., v_r}] ≠ false then
            G_2 = G_2 ∪ {w − Σ_{i=1}^{r} λ_i w_i};
        else
            r ⟵ r + 1;
            v_r = v;
            w_r = w;
            N = N ∪ {w_r};
            List = InsertNexts[w_r, List]
        endif
    endif
endw
```

1.2.2 Border Bases

For every $d \geq 0$, we let $[\mathbf{X}]_d$ be the set of terms of $\mathbb{K}[\mathbf{X}]$ of degree d.

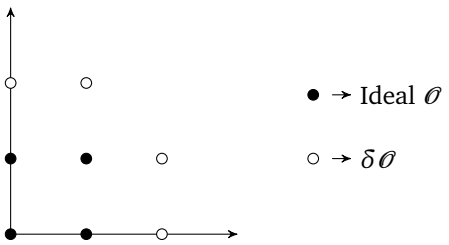
A non-empty set of terms $\mathcal{O} \subseteq [\mathbf{X}]$ is called an *order ideal* if every term dividing a term in $\mathcal{O}$ is contained in $\mathcal{O}$. That is, if $t \in \mathcal{O}$ implies $t' \in \mathcal{O}$ for every term t' dividing t. The set of terms $\delta\mathcal{O} = [\mathbf{X}]_1\mathcal{O} \setminus \mathcal{O} = \left(X_1\mathcal{O} \cup \ldots \cup X_n\mathcal{O}\right) \setminus \mathcal{O}$ forms the *Border* of $\mathcal{O}$.

Let $\mathcal{O} = \{t_1, \ldots, t_s\}$ be an order ideal with Border $\delta\mathcal{O} = \{b_1, \ldots, b_r\}$. A set of polynomials $\{g_1, \ldots, g_r\} \subseteq \mathbb{K}[\mathbf{X}]$ of the form

$$g_j = b_j - \sum_{i=1}^{s} c_{ij} t_i \text{ with } c_{ij} \in \mathbb{K}$$

is called an $\mathcal{O}$-Border prebasis.

Example 1.43. Consider the order ideal $\mathcal{O} = \{1, x, y, xy\}$ in $\mathbb{K}[x, y]$. Its Border is $\delta\mathcal{O} = \{x^2, x^2y, xy^2, y^2\}$.



Therefore the following set of polynomials

$$g_1 = x^2 - x, \quad g_2 = x^2y - xy, \quad g_3 = xy^2 - xy \quad \text{and} \quad g_4 = y^2 - y$$

form an $\mathcal{O}$- Border basis of I.

Let

$$\overline{\mathcal{O}_0} = \mathcal{O} \quad \text{and} \quad \overline{\mathcal{O}_i} = \overline{\mathcal{O}_{i-1}} \cup \delta\overline{\mathcal{O}_{i-1}} \quad \text{for } i \geq 1.$$

For every term $t \in [\mathbf{X}]$, there is a unique integer $i = \operatorname{ind}_{\mathcal{O}}(t) \geq 0$ such that $t \in \overline{\mathcal{O}_i} \setminus \overline{\mathcal{O}_{i-1}}$ which is called the $\mathcal{O}$-*index* of t. In other words, let $\mathcal{O}$ be an order ideal and $t \in [\mathbf{X}]$, $\operatorname{ind}_{\mathcal{O}}(t)$ is the smallest integer k such that $t = t't_1$ with $t_1 \in \mathcal{O}$ and $t' \in [\mathbf{X}]_k$. More generally, the index of a polynomial f is defined as

$$\operatorname{ind}_{\mathcal{O}}(f) = \max\left\{k \geq 0 \mid k = \operatorname{ind}_{\mathcal{O}}(t_i) \text{ with } t_i \in \operatorname{supp}(f)\right\}.$$

The index possesses properties resembling those of term orderings but index inequalities do not need to be preserved under multiplication.

Definition 1.44. Let $\mathcal{O} = \{t_1, \ldots, t_s\}$ be an order ideal with Border $\delta\mathcal{O} = \{b_1, \ldots, b_r\}$, $\mathcal{G} = \{g_1, \ldots, g_r\}$ be an $\mathcal{O}$-Border prebasis and I be a zero-dimensional ideal of $\mathbb{K}[\mathbf{X}]$ containing $\mathcal{G}$. Then $\mathcal{G}$ is a *Border basis* if and only if the residue classes of $t_1, \ldots, t_s$ modulo I are $\mathbb{K}$-linearly independent.

Let $b_i,\ b_j \in \delta\mathcal{O}$ be two distinct Border terms, b_i and b_j are *neigbours* if either $b_i = X_k b_j$ for some $k \in \{1, \ldots, n\}$ or $X_k b_i = X_l b_j$ for some $l, k \in \{1, \ldots, n\}$.

Let $g_i = b_i - \sum_{l=1}^{s} c_{li} b_l$ and $g_j = b_j - \sum_{l=1}^{s} c_{lj} b_l$ be two distinct Border prebasis polynomials. Then the polynomial

$$S(g_i, g_j) = \frac{\operatorname{lcm}(b_i, b_j)}{b_i} g_i - \frac{\operatorname{lcm}(b_i, b_j)}{b_j} g_j$$

is called the S-polynomial of g_i and g_j.

Proposition 1.45 (Characterizations of Border-Bases)**.** *Let $\mathcal{O}$ be an order ideal and $\mathcal{G} = \{g_1, \ldots, g_r\}$ be an $\mathcal{O}$-Border prebasis. $\mathcal{G}$ is an $\mathcal{O}$-Border basis of I if and only if for every nonzero polynomial f in I, there exists polynomials $f_1, \ldots, f_r$ in $\mathbb{K}[\mathbf{X}]$ such that*

$$f = f_1 g_1 + \ldots + f_r g_r \text{ with } \deg(f_i) \leq \operatorname{ind}_{\mathcal{O}}(f) - 1 \text{ whenever } f_i g_i \neq 0.$$

Proof. See [66, Proposition 9]. □

Remark 1.46. For Border Bases we have a similar remark to what we did for Gröbner Bases (Remark 1.39). Let $\mathcal{O} = \{t_1, \ldots, t_s\}$ be an order ideal with Border $\delta\mathcal{O} = \{b_1, \ldots, b_m\}$ and $\mathcal{G} = \{g_1, \ldots, g_m\}$ be an $\mathcal{O}$-Border prebasis then:

1. Every zero dimensional ideal I in $\mathbb{K}[\mathbf{X}]$ do not have an $\mathcal{O}$-Border basis, i.e. the existence is not guaranteed.
2. Any $\mathcal{O}$-Border basis $\mathcal{G}$ for a zero dimensional ideal I is a system of generators of I. Moreover $\mathcal{G}$ is uniquely determined by $\mathcal{O}$ and $\delta\mathcal{O}$.

Remark 1.47. If we fix the $\mathcal{O}$-border prebasis $\mathcal{G} = \{g_1, \ldots, g_r\}$ then the result of Algorithm 5 is uniquely determined.

Example 1.48. Let $\mathcal{O} = \{t_1 = 1, t_2 = x\} \subseteq \mathbb{K}[x, y]$, $\delta\mathcal{O} = \{b_1 = y, b_2 = xy, b_3 = x^2\}$ and $\mathcal{G} = \{g_1 = y, g_2 = xy - 1, g_3 = x^2 - 1\}$. We apply Algorithm 5 to divide the polynomial $f = x^2y + x^2 + 2xy$ by $\mathcal{G}$.

1. We initialize the algorithm with: $f_1 = f_2 = f_3 = 0$; $c_1 = c_2 = 0$ and $h = f$.
2. Since $\mathrm{ind}_{\mathcal{O}}(h) = 2$ and $h = h_1 + x^2 + 2xy$ with $h_1 = x^2y = x^2b_1$. Then, we replace:
$$h \longleftarrow h - x^2y = x^2 + 2xy \quad \text{and} \quad f_1 \longleftarrow x^2.$$
3. Now, $\mathrm{ind}_{\mathcal{O}}(h) = 1$ and $h = h_1 + 2xy$ with $h_1 = x^2 = b_3$. Thus,
$$h \longleftarrow h - x^2 + 1 = 2xy + 1 \quad \text{and} \quad f_3 \longleftarrow 1.$$
4. Note that $\mathrm{ind}_{\mathcal{O}}(h) = 1$ and $h = 2h_1 + 1$ with $h_1 = xy = b_2$. Thus,
$$h \longleftarrow h - 2(xy - 1) = 3 \quad \text{and} \quad f_2 \longleftarrow 2.$$
5. Finally $\mathrm{ind}_{\mathcal{O}}(h) = 0$ and $h = 3 \cdot 1 + 0 \cdot x$. Hence, $c_1 \longleftarrow 3$ and $c_2 \longleftarrow 0$.

Therefore the algorithm returns $(x^2, 2, 1, 3, 0)$ such that $f = x^2g_1 + 2g_2 + g_3 + 3$.

Let $f = f_1g_1 + \ldots + f_rg_r + c_1t_1 + \ldots + c_st_s$ be a representation of f computed by Algorithm 5. Then
$$\mathrm{NR}_{\mathcal{O},\mathcal{G}}(f) = c_1t_1 + \ldots + c_st_s$$
is called the *normal $\mathcal{O}$-remainder of f*.

Theorem 1.49 (Buchberger criterion for Border Bases)**.** *An $\mathcal{O}$-Border prebasis $\mathcal{G}$ is an $\mathcal{O}$-Border basis if and only if for all pair of neighbours (b_i, b_j) the normal $\mathcal{O}$-remainder of the S-polynomial $S(g_i, g_j)$ is zero.*

Proof. See a proof in [66, Proposition 18] □

Algorithm 5: The Border division algorithm

Data: An order ideal $\mathcal{O} = \{t_1, \ldots, t_s\}$, its border $\delta\mathcal{O} = \{b_1, \ldots, b_r\}$, an $\mathcal{O}$-border prebasis $\mathcal{G} = \{g_1, \ldots, g_r\}$ and a polynomial $f \in \mathbb{K}[\mathbf{X}]$.
Result: The tuple $(f_1, \ldots, f_r, c_1, \ldots, c_s) \in \mathbb{K}[\mathbf{X}]^r \times \mathbb{K}^s$ such that

$$f = f_1 g_1 + \ldots + f_r g_r + c_1 t_1 + \ldots + c_s t_s$$

where $\deg(f_i) \leq \mathrm{ind}_{\mathcal{O}}(f) - 1$ for all $i \in \{1, \ldots, r\}$ with $f_i g_i \neq 0$.

Initialization
$f_1 = \ldots = f_r = 0$; $c_1 = \ldots = c_s = 0$; $h = f$;
while $h \neq 0$ **do**
  if $\mathrm{ind}_{\mathcal{O}}(h) = 0$ **then**
    Find $c_1, \ldots, c_s \in \mathbb{K}$ such that $h = c_1 t_1 + \ldots + c_s t_s$;
    $h := 0$;
  else
    Find $a_1, \ldots, a_m \in \mathbb{K} \setminus \{0\}$ and $h_1, \ldots, h_m \in [\mathbf{X}]_n$ with $\mathrm{ind}_{\mathcal{O}}(h_1) = \mathrm{ind}_{\mathcal{O}}(h)$ such that $h = a_1 h_1 + \ldots + a_m h_m$;
    // The algorithm does not depend on the choice of the term h_1
  endif
  // Determine the smallest $i \in \{1, \ldots, r\}$: $h_1 = \mathbf{t}' b_i$ with

$$\deg(\mathbf{t}') = \mathrm{ind}_{\mathcal{O}}(h) - 1.$$

  // Note that this factorization always exists. `boolean := false`;
  while $i \leq i$ *and* `boolean` $=$ `false` **do**
    if $h_1 = \mathbf{t}' b_i$ *with* $\deg(\mathbf{t}') = \mathrm{ind}_{\mathcal{O}}(h) - 1$ **then**
      $h := h - a_1 \mathbf{t}' g_i$;
      $f_i := f_i + a_1 \mathbf{t}'$;
      `boolean := true`;
    else
      $i := i + 1$;
    endif
  endw
endw

1.2.3 The relation between Gröbner Bases and Border Bases

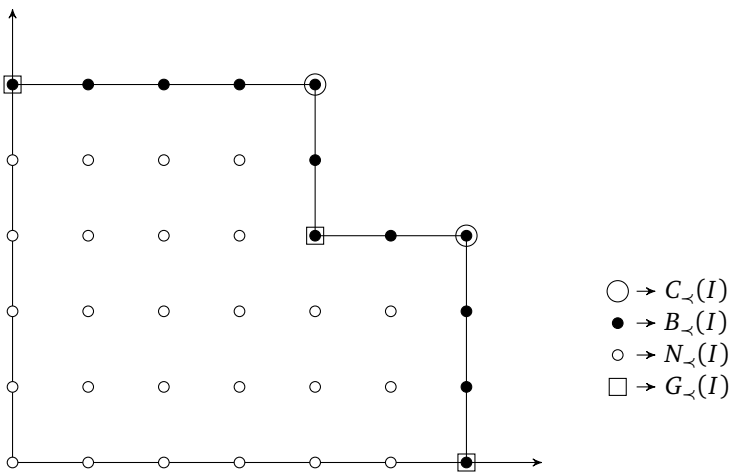
Let $\prec$ be a term ordering and I be an ideal of $\mathbb{K}[\mathbf{X}]$ generated by the finite set of polynomials $F = \{f_1, \ldots, f_s\}$. Then $\mathcal{O} = [\mathbf{X}] \setminus \mathrm{LT}_{\prec}(F)$ is an order ideal of terms such that I has an $\mathcal{O}$-Border basis.

Remark 1.50. Note that the elements of the reduced Gröbner basis of I w.r.t. $\prec$ are exactly the Border basis polynomials corresponding to the corners of $\delta\mathcal{O}$, i.e. to the minimal generators of the Border basis.

We will fix some notation and terminology (similar to those introduced in [98]).

1. $N_\prec(I)$ stands for the *set of standard monomials* of I w.r.t. $\prec$.
2. $G_\prec(I)$ denotes the unique minimal basis of $\mathrm{in}_\prec(I)$.
3. $B_\prec(I) = \{X_h t \mid 1 \leq h \leq n \text{ and } t \in N_\prec(I)\}$, the Border of the order ideal $N_\prec(I)$, for short the *Border set* of I w.r.t. $\prec$.
4. $C_\prec(I) = \{t \in B_\prec(I) \mid t \in \mathrm{LT}_\prec(\mathrm{F})\}$ means the *corner set* of I.

Example 1.51. Consider the ideal $I = \left\langle X_1^6, X_1^4 X_2^3, X_2^5 \right\rangle$, then we have the following figure:



Proposition 1.52 (Properties)**.** *The following properties are trivially satisfied.*

1. *$N_\prec(I)$, $N_\prec(I) \cup G_\prec(I)$ and $N_\prec(I) \cup B_\prec(I)$ are order ideals.*
2. *For every $w \in \mathrm{LT}_\prec\{F\}$, there exists $u \in [\mathbf{X}]$ and $v \in B_\prec(I)$ such that $w = vu$.*
3. *$\mathbb{K}[\mathbf{X}] = I \oplus \mathrm{Span}_{\mathbb{K}}\left(N_\prec(I)\right)$, where $\mathrm{Span}_{\mathbb{K}}\left(N_\prec(I)\right)$ stands for the $\mathbb{K}$-vector space whose basis is $\left(N_\prec(I)\right)$.*
4. *For each polynomial $f \in \mathbb{K}[\mathbf{X}]$ there is a unique remainder on the division of f by F w.r.t. $\prec$, such that $f - \mathrm{Red}_\prec(f, F) \in I$. The normal form also satisfies:*
   (a) $\mathrm{Red}_\prec(f_1, F) = \mathrm{Red}_\prec(f_2, F)$ *if and only if* $f_1 - f_2 \in I$.
   (b) $\mathrm{Red}_\prec(f, F) = 0$ *if and only if* $f \in I$.
5. *The reduced Gröbner basis of I w.r.t. $\prec$ is the set $\mathscr{G}_\prec = \{t - \mathrm{Red}(t, F) \mid t \in G_\prec(I)\}$.*
6. *The Border basis of I w.r.t. $\prec$ is the set $\mathscr{B}_\prec = \{t - \mathrm{Red}(t, F) \mid t \in B_\prec(I)\}$.*

Proof. These properties can be found in [98]. □

Remark 1.53. I is a zero-dimensional ideal if the dimension of the quotient ring $\mathbb{K}[\mathbf{X}]/I$ is finite. In such case, $\dim_{\mathbb{K}}(\mathbb{K}[\mathbf{X}]/I)$ is equal to the cardinality of the set $N_{\prec}(I)$.

1.3 Matroids

We briefly state the connections between coding and matroid theory.

Definition 1.54. A *matroid* M is a pair (E, I) consisting of a finite set E called ground set and a collection I of subsets of E called independent sets, satisfying the following conditions:

1. the empty set is independent, i.e. $\emptyset \in I$.
2. if $A \in I$ and $B \subset A$, then $A \in I$.
3. If $A,\ B \in I$ and $|A| < |B|$, then there exists $e \in B \setminus A$ such that $A \cup \{e\} \in I$.

A maximal independent subset of E is called a *basis* of M. A direct consequence of the previous definition is that all bases of M have the same cardinality. Therefore we define the *rank* of the matroid M as the cardinality of any basis of M, denoted by $\text{rank}(M)$. A subset of E that does not belong to I is called *dependent set*. Minimal dependent subsets of E are known as *circuits* of M. A set is said to be a *cycle* if it is a disjoint union of circuits. The collection of all cycles of M is denoted by $\mathfrak{C}(M)$.

Let us consider an $m \times n$ matrix A in $\mathbb{F}_q$ whose columns are indexed by $E = \{1, \ldots, n\}$ and take I to be the collection of subsets J of E for which the column vectors $\left\{A_j \mid j \in J\right\}$ are linearly independent over $\mathbb{F}_q$. Then (E, I) defines a matroid denoted by $M[A]$. A matroid $M = (E, I)$ is $\mathbb{F}_q$-*representable* if it is isomorphic to $M[A]$ for some $A \in \mathbb{F}_q^{m \times n}$. Then A is called the *representation matrix* of M.

The following well known result describes the relation between the collection of all cycles of a matroid M and its representation matrix.

Proposition 1.55. *Let $M = (E, I)$ be a $\mathbb{F}_q$-representable matroid. Then $\mathfrak{C}(M)$ is the null space of a representation matrix of M. Furthermore, the dimension of $\mathfrak{C}(M)$ is $|E| - rank(M)$.*

Given an $m \times n$ matrix H in $\mathbb{F}_q$ then H can be seen not only as the representation matrix of the $\mathbb{F}_q$-representable matroid $M[H]$ but also as a parity check matrix of an $[n, k]$-code $\mathscr{C}$. Furthermore there exists a one to one correspondence between $\mathbb{F}_q$-representable matroids and linear codes since for any $H,\ H' \in \mathbb{F}_q^{m \times n}$, $M[H] = M[H']$ if and only if H and H' are parity check matrices of the same code $\mathscr{C}$. This association enables us to work with $\mathbb{F}_q$-representable matroids and linear codes as if they were the same object and thus we can deduce some properties of linear codes using tools from matroid theory and vice-versa.

$\mathbb{F}_q$-representable matroid $M[H]=(E,I)$	$[n,k]$-code $\mathscr{C}$
$\lvert E\rvert$	n
$\operatorname{rank}(M[H])$	$\dim(\mathscr{C})=k$
A cycle of $M[H]$	The support of a codeword of $\mathscr{C}$
A circuit of $M[H]$	$\operatorname{supp}(\mathbf{c})\ :\ \mathbf{c}\in\mathscr{M}_{\mathscr{C}}$
The weight of a minimal circuit	The minimum distance

Table 1.1: Codes and their relation to matroids

Table 1.1 provides some similarities between both theories. For example, by definition of parity check matrix of a code and making use of Proposition 1.55, we can relate the support of any codeword of $\mathscr{C}$ with a cycle in $M[H]$. In addition, we have that $\mathbf{c}\in\mathscr{C}$ is a minimal support codeword if and only if $\operatorname{supp}(\mathbf{c})$ is a circuit of the matroid $M[H]$.

The decomposition theory of matroids when applied to binary matrices provides a powerful decomposition theory for binary linear codes with applications in maximum-likelihood decoding, see [63] and the references therein. For some relations between matroids and the Gröbner basis associated to binary codes we refer the reader to [17, 21].

2

Binary codes: Gröbner representation and related structures

Contents

Throughout this chapter $\mathscr{C}$ will be a binary linear code of length n and dimension k, i.e. a k-dimensional linear subspace of $\mathbb{F}_2^n$ where $\mathbb{F}_2$ is the field of two elements.

The first two sections essentially follow [12] while the last section presents the results in [18]. Both are joint works with M. Borges-Quintana and M.A. Borges-Trenard from the University of Oriente (Santiago de Cuba), and E. Martínez-Moro from University of Valladolid (Spain). The purpose of this chapter is to extend the previous work on Gröbner representation of binary codes to get a better understanding of the underlying coset enumeration procedure as well as relating it to the problem of gradient descent decoding procedures known in the literature. Indeed, our view point sheds some new light on the latter problem and unifies some existing approaches.

The outline of the Chapter is as follows. First, in Section 2.2 we present an algorithm that not only provides a Gröbner representation of a binary linear code $\mathscr{C}$

but also the whole set of all coset leaders denoted by CL($\mathscr{C}$). Its efficiency stands on the fact that its complexity is linear on the number of elements of CL($\mathscr{C}$) which is smaller than exhaustive search in $\mathbb{F}_2^n$. There are a few applications for CL($\mathscr{C}$): the set of coset leaders in linear codes has been related to the set of minimal support codewords which have been used in maximum likelihood decoding analysis [3, 74], and also to secret sharing schemes since they describe the minimal access structure [84]. The example presented in Section 2.2 with 64 coset and 118 coset leaders suggests some extra applications of the algorithm such as obtaining the weight distribution of the coset leaders or the Newton and Covering radius of the code. We shall remark that these applications do not pose a large additional cost to the proposed algorithm.

Section 2.3 is devoted to show how the algorithm presented in the previous section can be adapted to compute a test-set for the code which we refer to as *leader codewords*. Not only do we prove that the elements of this set are zero neighbours but also that the knowledge of this set can be used to compute the subset of coset leaders corresponding to the coset of the received word.

Finally, in Section 2.4 we show a unified approach, via the Gröbner representation of a code, to the two gradient descent decoding algorithm known for binary codes: the one where the search is done changing the coset representative (l-GDDA) due to Liebler [71] and the one given by descending within the same coset (ts-GDDA) due to Ashikhmin and Barg [2]. Note that they were claimed to be of different nature. However, we will show that both come from two ways of computing the reduction of a monomial modulo the binomial ideal $I_2(\mathscr{C})$ associated to the binary code.

2.1 Gröbner representation of a binary code

Let $\{\mathbf{e}_i \mid i \in \{1,\dots,n\}\}$ be the canonical basis of $\mathbb{F}_2^n$.

Let us consider the integer $q \geq 2$. We define the following characteristic crossing functions:

$$\blacktriangledown : \ \mathbb{Z}^s \longrightarrow \mathbb{Z}_q^s \qquad \text{and} \qquad \blacktriangle : \ \mathbb{Z}_q^s \longrightarrow \mathbb{Z}^s$$

where s is determined by context, note that the space may be also the matrix space. The map $\blacktriangledown$ is reduction modulo q while the map $\blacktriangle$ replace the class of $0, 1, \dots, q-1$ by the same symbols regarded as integers. Both maps act coordinate-wise.

Let $\mathbf{a} = (a_1, \dots, a_n)$ be an element in $\mathbb{Z}_q^n$ then $\mathbf{X}^{\blacktriangle \mathbf{a}} = x_1^{\blacktriangle a_1} \cdots x_n^{\blacktriangle a_n} \in \mathbb{K}[\mathbf{X}]$. Therefore, these functions enable us to go back to the usual definition of terms in $\mathbb{K}[\mathbf{X}]$.

Definition 2.1. A *Gröbner representation* of an $[n,k]$ binary linear code $\mathscr{C}$ is a pair $(\mathscr{N}, \phi)$ where:

- $\mathscr{N}$ is a transversal of the cosets in $\mathbb{F}_2^n/\mathscr{C}$ (i.e. one element of each coset) verifying that $\mathbf{0} \in \mathscr{N}$ and for each $\mathbf{n} \in \mathscr{N} \setminus \{\mathbf{0}\}$ there exists an $\mathbf{e}_i$ with $i \in \{1,\dots,n\}$ such that $\mathbf{n} = \mathbf{n}' + \mathbf{e}_i$ with $\mathbf{n}' \in \mathscr{N}$.

- $\phi : \ \mathscr{N} \times \{\mathbf{e}_i\}_{i=1}^n \longrightarrow \mathscr{N}$ is a function called *Matphi function* that maps each pair $(\mathbf{n}, \mathbf{e}_i)$ to the element of $\mathscr{N}$ representing the coset that contains $\mathbf{n}+\mathbf{e}_i$.

The word Gröbner is not casual. Indeed, if we consider the binomial ideal

$$I_2(\mathscr{C}) = \left\langle \mathbf{X}^{\blacktriangle \mathbf{w}_1} - \mathbf{X}^{\blacktriangle \mathbf{w}_2} \mid \mathbf{w}_1 - \mathbf{w}_2 \in \mathscr{C} \right\rangle \subseteq \mathbb{K}[\mathbf{X}]$$

and a total degree ordering $\prec$ and we compute the reduced Gröbner basis $\mathscr{G}$ of $I_2(\mathscr{C})$ w.r.t. $\prec$. Then we can take $\mathscr{N}$ as the vectors $\mathbf{w}$ such that $\mathbf{X}^{\blacktriangle \mathbf{w}}$ is a standard monomial in $\mathscr{G}$ w.r.t. $\prec$. Moreover, the function *Matphi* can be seen as the multiplication tables of the standard monomials times the variables $x_i \in [\mathbf{X}]_1$ modulo the ideal $I_2(\mathscr{C})$. Note that the *Matphi* structure is independent of the particular chosen set $\mathscr{N}$ of representative elements of the quotient ring $\mathbb{F}_2^n/\mathscr{C}$. See [11, 13, 14, 15, 16] for a more general treatment of these concepts.

Remark 2.2. It is important to remark that $\mathbb{K}$ does not need to be the same field as the one where the code is defined. In fact, any field will work since the code is "encoded" in the exponents. Thus we will use $\mathbb{K} = \mathbb{F}_2$ in the computation which is the easiest field to tackle with.

The Gröbner representation of a code can be computed with a slight modification of the FGLM algorithm [42], since we are dealing with a zero-dimensional ideal. This extension algorithm can be found in [15]. Let $\prec_T$ be a total degree ordering, there are three functions needed to understand the algorithm:

- `InsertNext[`$\mathbf{w}$`,List]` adds to `List` all the sums $\mathbf{w} + \mathbf{e}_k$ with $k \notin \operatorname{supp}(\mathbf{w})$, keeps the increasing order of the list `List` w.r.t. $\prec_T$ and removes duplicates.
- `NextTerm[List]` returns the first element of `List` and deletes it from `List`.
- `Member[`$\mathbf{v}, \{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$`]` returns j if $\mathbf{v} = \mathbf{v}_j$ and `false` otherwise.

The following theorem showns the importance of using a total degree ordering (for example `degrevlex`).

Theorem 2.3. *Let $\mathscr{C}$ be an $[n,k]$ binary linear code with error correcting capability t and G_T be the reduced Gröbner basis of $I_2(\mathscr{C})$ w.r.t. a total degree ordering $\prec_T$. If*

$$\deg\left(\operatorname{Red}_{\prec_T}(\mathbf{X}^{\blacktriangle \mathbf{w}}, G_T)\right) \leq t$$

then the exponent of $\operatorname{Red}_{\prec_T}(\mathbf{X}^{\blacktriangle \mathbf{w}}, G_T)$ is the error vector corresponding to the received word $\mathbf{w} \in \mathbb{F}_2^n$. In other words, by subtracting the exponent of $\operatorname{Red}_{\prec_T}(\mathbf{X}^{\blacktriangle \mathbf{w}}, G_T)$ from $\mathbf{w}$, we obtain the closest codeword to $\mathbf{w}$.

On the other hand, if $\deg\left(\operatorname{Red}_{\prec_T}(\mathbf{X}^{\blacktriangle \mathbf{w}}, G_T)\right) > t$, then $\mathbf{w}$ contains more than t errors.

Proof. The following proof can be found in [11, Theorem2]. However, we expose it here to familiarize the reader with the new notation in this section and render the chapter as self-contained as possible.

The uniqueness of the normal form is guaranteed by its definition. Thus, it suffices to show that the monomial associated to the error vector $\mathbf{e}$ corresponding to the received word $\mathbf{w} \in \mathbb{F}_2^n$ is the normal form of $\mathbf{X}^{\blacktriangle \mathbf{w}}$ w.r.t. G_T.

Algorithm 6: Computing a Gröbner representation of a binary linear code $\mathscr{C}$

Data: A total degree ordering $\prec_T$ and the parity check matrix H of a binary code $\mathscr{C}$

Result: $(\mathscr{N}, \phi)$ a Gröbner representation for $\mathscr{C}$

```
List ⟵ [0]; S ⟵ ∅; 𝒩 ⟵ ∅; r ⟵ 0;
while List ≠ ∅ do
    w ⟵ NextTerm[List];
    s ⟵ wH^T;
    j ⟵ Member[s, S];
    if j ≠ false then
        for k ∈ supp(w) : w = w' + e_k with w' ∈ 𝒩
            φ(w', e_k) ⟵ w_j;
        endfor
    else
        r ⟵ r + 1;
        w_r ⟵ w;
        𝒩 ⟵ 𝒩 ∪ {w_r};
        S ⟵ S ∪ {s};
        List = InsertNexts[w, List];
        for k ∈ supp(w) : w = w' + e_k with w' ∈ 𝒩
            φ(w', e_k) ⟵ w;
            φ(w, e_k) ⟵ w';
        endfor
    endif
endw
```

Suppose that $\mathbf{w} \in B(\mathscr{C}, t) = \left\{\mathbf{y} \in \mathbb{F}_2^n \mid \exists \mathbf{c} \in \mathscr{C} \ : \ \mathrm{d}_H(\mathbf{c}, \mathbf{y}) \leq t\right\}$, or equivalently, that there exists a unique codeword $\mathbf{c} \in \mathscr{C}$ such that $\mathbf{w} = \mathbf{c} + \mathbf{e}$ with $\mathrm{w}_H(\mathbf{e}) \leq t$. Note that $\deg(\mathbf{X}^{\blacktriangle \mathbf{e}}) = \mathrm{w}_H(\mathbf{e})$. This equality implies that there cannot be another monomial $\mathbf{X}^{\mathbf{b}}$ with $\deg(\mathbf{X}^{\mathbf{b}}) \leq t$ such that $\blacktriangledown \mathbf{b}$ has the same syndrome as the received word $\mathbf{w}$, since this would contradict the definition of the error correcting capacity t of $\mathscr{C}$.

On the other hand, if $\deg\left(\mathrm{Red}_{\prec_T}(\mathbf{X}^{\blacktriangle \mathbf{w}}, G_T)\right)$ is greater that t then the previous arguments means that the minimum weight representative of the coset $\mathbf{w} + \mathscr{C}$ has weight greater than t, i.e. $\mathbf{w}$ contains more than t errors.

□

Note that Algorithm 6 provides us a representation of the algebra $\mathbb{K}[X]/I_2(\mathscr{C})$ given a set of representatives and how they behave under multiplication by the variables $\{x_i\}_{i=1,\ldots,n}$.

Example 2.4. Let $\mathscr{C}$ be a $[6,3,3]$ binary code with generator matrix G and parity

check matrix H given by:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

We will use the `degrevlex` order with $x_1 < x_2 < \ldots < x_6$. Then Algorithm 6 computes a Gröbner representation $(\mathscr{N}, \phi)$ of $\mathscr{C}$ which corresponds to:

$$\mathscr{N} = \left\{ \begin{array}{cccc} n_1 = \mathbf{0}, & n_2 = \mathbf{e}_1, & n_3 = \mathbf{e}_2, & n_4 = \mathbf{e}_3, \\ n_5 = \mathbf{e}_4, & n_6 = \mathbf{e}_5, & n_7 = \mathbf{e}_6, & n_8 = \mathbf{e}_1 + \mathbf{e}_6 \end{array} \right\}$$

and the following representation of ϕ where the first entry corresponds to the elements $n_i \in \mathscr{N}$ and the second points to the values $\phi(n_i, \mathbf{e}_j)$ for $j = 1, \ldots, 6$ and $i = 1, \ldots, 8$.

$$\left[\begin{array}{lll} [\mathbf{0},\ [2,3,4,5,6,7]], & [\mathbf{e}_1,\ [1,5,6,3,4,8]], & [\mathbf{e}_2,\ [5,1,8,2,7,6]], \\ [\mathbf{e}_3,\ [6,8,1,7,2,5]], & [\mathbf{e}_4,\ [3,2,7,1,2,5]], & [\mathbf{e}_5,\ [4,7,2,8,1,3]], \\ [\mathbf{e}_6,\ [8,6,5,4,3,1]], & [\mathbf{e}_1 + \mathbf{e}_6,\ [7,4,3,6,5,2]] & \end{array} \right].$$

Therefore, $\phi(\mathbf{e}_3, \mathbf{e}_1) = \mathbf{e}_5$ or $\phi(\mathbf{e}_2, \mathbf{e}_3) = \mathbf{e}_1 + \mathbf{e}_6$ and so on with the other cases.

By [11, Theorem 1], associated to the code $\mathscr{C}$ we can define the following binomial ideal:

$$I_2(\mathscr{C}) = \left\langle \{x_1x_4x_5x_6 - 1, x_2x_5x_6 - 1, x_3x_4x_6 - 1\} \cup \left\{x_i^2 - 1\right\}_{i=1,\ldots,6} \right\rangle.$$

The reduced Gröbner basis of this ideal w.r.t. `degrevlex` order with $x_1 < x_2 < \ldots < x_6$ has 20 elements given by the following set:

$$\left\{ \begin{array}{l} x_6x_5 - x_3,\ \ x_6x_4 - x_2,\ \ x_6x_3 - x_5,\ \ x_6x_2 - x_4, \\ x_5x_4 - x_6x_1,\ \ x_5x_3 - x_6,\ \ x_5x_2 - x_1,\ \ x_5x_1 - x_2, \\ x_4x_3 - x_1,\ \ x_4x_2 - x_6,\ \ x_4x_1 - x_3, \\ x_3x_2 - x_6x_1,\ \ x_3x_1 - x_4, \\ x_2x_1 - x_5 \end{array} \right\} \cup \left\{x_i^2 - 1\right\}_{i=1,\ldots,6}$$

i.e. it corresponds to the set $\left\{\mathbf{X}^{\mathbf{n}_i}\mathbf{X}^{\mathbf{e}_l} - \mathbf{X}^{\mathbf{n}_j} \mid \mathbf{n}_i, \mathbf{n}_j \in \mathscr{N} \text{ and } \phi(\mathbf{n}_i, \mathbf{e}_l) = \mathbf{n}_j\right\}$ modulo the ideal $\left\langle \left\{x_i^2 - 1\right\}_{i=1,\ldots,6} \right\rangle$.

2.2 Computing coset leaders

Definition 2.5. An ordering $\prec$ on $\mathbb{F}_2^n$ is a *weight compatible ordering* if for any $\mathbf{a},\ \mathbf{b} \in \mathbb{F}_2^n$ we say $\mathbf{a} \prec \mathbf{b}$ if

$$\mathrm{w}_H(\mathbf{a}) < \mathrm{w}_H(\mathbf{b}) \quad \text{or} \quad \mathrm{w}_H(\mathbf{a}) = \mathrm{w}_H(\mathbf{b}) \text{ and } \blacktriangle\mathbf{a} \prec_1 \blacktriangle\mathbf{b}$$

where $\prec_1$ is any admissible order on $\mathbb{N}^n$.

Remark 2.6. A weight compatible ordering $\prec$ is in general not admissible on $\mathbb{F}_2^n$. Note that $\mathbf{a} \prec \mathbf{b}$ does not always imply that $\mathbf{ac} \prec \mathbf{bc}$ for all $\mathbf{c} \in \mathbb{F}_2^n$. However $\prec$ is a Noetherian order on $\mathbb{F}_2^n$ since $\mathbb{F}_2^n$ is finite; and for all $\mathbf{v},\ \mathbf{w} \in \mathbb{F}_2^n$, if $\mathrm{supp}(\mathbf{v}) \subset \mathrm{supp}(\mathbf{w})$ then $\mathbf{v} \prec \mathbf{w}$. Moreover, for all $\mathbf{a} \in \mathbb{F}_2^n$ we have that $\deg(\mathbf{X}^{\blacktriangle \mathbf{a}}) = \mathrm{w}_H(\mathbf{a})$, that is, a weight compatible ordering on $\mathbb{F}_2^n$ can be viewed as total degree ordering on $\mathbb{K}[\mathbf{X}]$.

Definition 2.7. We define the object `List` as an ordered set of elements in F_2^n w.r.t. a weight compatible order $\prec$ verifying the following properties:

1. $\mathbf{0} \in$ `List`.
2. If $\mathbf{v} \in$ `List` and $\mathrm{w}_H(\mathbf{v}) = \mathrm{w}_H(N(\mathbf{v}))$ then $\{\mathbf{v} + \mathbf{e}_i \mid i \notin \mathrm{supp}(\mathbf{v})\} \subset$ `List`, where $N(\mathbf{v}) = \min_{\prec}\{\mathbf{w} \mid \mathbf{w} \in$ `List` $\cap\, (\mathscr{C} + \mathbf{v})\}$.

Remark 2.8. If the second condition holds for $\mathbf{v} \in \mathbb{F}_2^n$ then $\mathbf{v}$ is a coset leader for the coset $\mathscr{C} + \mathbf{v}$. Note as well the resemblances with Definition 2.1 of Gröbner representation.

Definition 2.9. For each $\mathbf{v} \in$ `List` we have defined $N(\mathbf{v})$ as the minimal element w.r.t. a fixed weight compatible order $\prec$ which belongs to the intersection `List` $\cap$ $(\mathbf{v} + \mathscr{C})$. We will denote by $\mathscr{N}$ the set of distinct $N(\mathbf{v})$ with $\mathbf{v} \in$ `List`.

The next theorem states that the object `List` includes the set of coset leaders of a given binary linear code.

Theorem 2.10. *Let* $\mathbf{w} \in \mathbb{F}_2^n$. *If* $\mathbf{w} \in \mathrm{CL}(\mathscr{C})$ *then* $\mathbf{w} \in$ `List`.

Proof. We will proceed by Noetherian induction on $\mathbb{F}_2^n$ with a weight compatible ordering $\prec$.

The statement is true for $\mathbf{0} \in \mathbb{F}_2^n$ by definition. Now assume that the desired property is true for any word $\mathbf{u}$ smaller than an arbitrary but fixed $\mathbf{w} \in \mathrm{CL}(\mathscr{C}) \setminus \{\mathbf{0}\}$ w.r.t. $\prec$, i.e.

$$\text{if } \mathbf{u} \in \mathrm{CL}(\mathscr{C}) \text{ and } \mathbf{u} \prec \mathbf{w} \text{ then } \mathbf{u} \in \texttt{List}.$$

Let $i \in \mathrm{supp}(\mathbf{w})$, then we can write $\mathbf{w} = \mathbf{u} + \mathbf{e}_i$ with $i \notin \mathrm{supp}(\mathbf{u})$, or equivalently, $\mathrm{supp}(\mathbf{u}) \subset \mathrm{supp}(\mathbf{w})$ and hence $\mathbf{u} \prec \mathbf{w}$. In addition, since $\mathbf{w} \in \mathrm{CL}(\mathscr{C})$, then by Theorem 1.18 $\mathbf{u}$ also belongs to $\mathrm{CL}(\mathscr{C})$. So if we invoke the induction hypothesis we have that $\mathbf{u} \in$ `List`. Moreover, $\mathrm{w}_H(\mathbf{u}) = \mathrm{w}_H(N(\mathbf{u}))$ which is clear from the fact that $\mathbf{u} \in \mathrm{CL}(\mathscr{C})$. We now apply property 2 of Definition 2.7 which gives, as claimed, that $\mathbf{w} = \mathbf{u} + \mathbf{e}_i \in$ `List` with $i \notin \mathrm{supp}(\mathbf{u})$. □

Theorem 2.10 and its proof suggest an algorithm for computing all the coset leaders of a given binary code $\mathscr{C}$. Note that in [16] a similar algorithm was proposed as described above in Algorithm 6. Indeed, the output for those cosets with only one coset leader is the same. We shall also keep track of the *Matphi* function in the new algorithm but it is not strictly necessary if we only want to compute the set $\mathrm{CL}(\mathscr{C})$.

The subfunctions used in the new algorithm are:

- `InsertNext[`$\mathbf{t}$`,List]`, adds to `List` all the sums $\mathbf{t} + \mathbf{e}_k$ with $k \notin \mathrm{supp}(\mathbf{t})$, removes duplicates and keeps `List` in increasing order w.r.t. the ordering $\prec$.

Algorithm 7: CLBC Algorithm

Data: A weight compatible ordering $\prec$ and a parity check matrix H of a binary code $\mathscr{C}$.

Result: The set of coset leaders $\mathrm{CL}(\mathscr{C})$ and $(\mathscr{N}, \phi)$ a Gröbner representation for $\mathscr{C}$.

```
1  List ⟵ [0]; 𝒩 ⟵ ∅; r ⟵ 0; CL(𝒞) ⟵ ∅; S ⟵ ∅;
2  while List ≠ ∅ do
3      t ⟵ NextTerm[List]; s ⟵ tH^T;
4      j ⟵ Member[s, S];
5      if j ≠ false then
6          for k ∈ supp(t) : t = t' + e_k with t' ∈ 𝒩
7              φ(t', e_k) ⟵ t_j
8          endfor
9          if w_H(t) = w_H(t_j) then
10             CL(𝒞)[t_j] ⟵ CL𝒞[t_j] ∪ {t};
11             List ⟵ InsertNext[t, List];
12         endif
13     else
14         r ⟵ r + 1; t_r ⟵ t; 𝒩 ⟵ 𝒩 ∪ {t_r};
15         CL(𝒞)[t_r] ⟵ {t_r}; S ⟵ S ∪ {s};
16         List = InsertNext[t_r, List];
17         for k ∈ supp(t_r) : t_r = t' + e_k with t' ∈ 𝒩
18             φ(t', e_k) ⟵ t_r;
19             φ(t_r, e_k) ⟵ t';
20         endfor
21     endif
22 endw
```

- `NextTerm[List]`, returns the first element from `List` and deletes it. If `List` is empty returns $\emptyset$.
- `Member[obj, G]`, returns the position j of obj in G if obj $\in G$ and `false` otherwise.

Remark 2.11. First we perform subroutine $\mathbf{t} = \texttt{NextTerm[List]}$ where the element $\mathbf{t}$ is deleted from the set `List`. Then subroutine `InsertNext[t, List]` is carried out which inserts in `List` all the elements of the form:

$$\mathbf{t}' = \mathbf{t} + \mathbf{e}_k \text{ with } k \notin \mathrm{supp}(\mathbf{t}), \text{ i.e. } \mathbf{t}' \succ \mathbf{t}.$$

Therefore all the new elements inserted in `List` are greater than those that have already been deleted from it with respect to $\prec$.

Theorem 2.12. *Algorithm 7 computes the set of coset leaders of a given binary code $\mathscr{C}$ and its corresponding* Matphi *function.*

Proof. Let us first prove that the set `List` is well defined according to Definition 2.7. It is clear that $\mathbf{0} \in$ `List` verifying property 1, by **Step 1**. In **Step 3** the syndrome of $\mathbf{t} =$ `NextTerm[List]` is computed, then we have two possible cases based on the outcome of **Step 4**:

1. If $j =$ `false` then the coset $\mathscr{C} + \mathbf{t}$ has not yet been considered. Thus, according to Remark 2.11, we have that $N(\mathbf{t}) = \mathbf{t}$ and **Step 16** guarantees property 2.

2. On the other hand, if $j \neq$ `false`, then the element $N(\mathbf{t}) = \mathbf{t}_j$ has already been computed. However, if $\mathbf{t} \in \mathrm{CL}(\mathscr{C})$, or equivalently, $\mathrm{w}_H(\mathbf{t}) = \mathrm{w}_H(\mathbf{t}_j)$, then **Step 11** certified property 2.

Therefore, Algorithm 7 constructs the object `List` in accordance with Definition 2.7.

Furthermore, on one hand **Step 10** and **Step 15** assure the set of coset leaders of the given code; and on the other hand **Step 7**, **Step 18** and **Step 19** compute the *Matphi* function. Note that **Step 17** is necessary since the first case above ensures that $N(\mathbf{t}_r) = \mathbf{t}_r$ so by Theorem 1.18 $\mathbf{t}' \in \mathrm{CL}(\mathscr{C})$. But $\mathbf{t}_r = \mathbf{t}' + \mathbf{e}_k$ with $k \in \mathrm{supp}(\mathbf{t}_r)$ so by Remark 2.11 $\mathbf{t}' \prec \mathbf{t}_r$ has already been considered on the algorithm. Thus, we have actually proved that Algorithm 7 guarantees the desired outputs.

Finally, notice that the cardinality of the set `List` is bounded by n times the cardinality of $\mathrm{CL}(\mathscr{C})$, and **Step 11** and **Step 16** guarantee that when the complete set of coset leaders is computed no more elements are inserted in `List` while **Step 3** continues deleting elements from it. Thus, after a finite number of steps the set `List` is empty. Consequently, **Step 2** provides the end of the algorithm. □

Remark 2.13. Note that Algorithm 7 returns $(\mathscr{N}, \phi)$ that fulfill Definition 2.1, for correctness we refer the reader to [16, Theorem 1]. Furthermore, by Definition 2.9, those representative of the cosets given by $\mathscr{N}$ are the smallest terms in `List` w.r.t. $\prec$.

The next theorem states an upper bound for the number of iterations that Algorithm 7 will perform.

Theorem 2.14. *Algorithm 7 computes the set of coset leaders of a given binary code $\mathscr{C}$ of length n after at most $n|\mathrm{CL}(\mathscr{C})|$ iterations.*

Proof. Notice that by looking how Algorithm 7 is constructed, the number of iterations is exactly the size of `List`. Moreover, note that we can write `List` as the following set

$$\texttt{List} = \{\mathbf{w} + \mathbf{e}_i \mid \mathbf{w} \in \mathrm{CL}(\mathscr{C}) \text{ and } i \in \{1, \ldots, n\}\}.$$

Therefore, it is clear that the size of List is bounded by $n|\mathrm{CL}(\mathscr{C})|$. □

Remark 2.15.

1. We can proceed analogously to the previous proof to estimate the required memory space which is $\mathscr{O}(n|\mathrm{CL}(\mathscr{C})|)$. In the best case, $\mathscr{O}(|\mathrm{CL}(\mathscr{C})|)$ of memory space is needed, thus Algorithm 7 is near the optimal case when considering memory requirements.

2. Algorithm 7 generates at most $n|\mathrm{CL}(\mathscr{C})|$ words from $\mathbb{F}_2^n$ to compute the set of all coset leaders. Therefore, the proposed algorithm has near-optimal performance and significantly reduced complexity.

For a detailed complexity analysis and some useful considerations from the computational point of view of Algorithm 6 we refer the reader to [16]. In Algorithm 7, the main difference with respect to the previous relatives (see [14, 16] and the references therein) is that we do not only provide a set of canonical forms $\mathscr{N}$ but also the set of all coset leaders.

Remark 2.16. The collection of programs and procedures **GBLA_LC** (*Gröbner Basis by Linear Algebra and Linear Codes*) has already been presented in previous works (see for instance [11, 13, 16]). This frameworks consists of various files written in the GAP [48] language and included in GAP's package GUAVA 3.10.

We have implemented Algorithm 7 and added to the collection **GBLA_LC**. The so-called function **GBLA**, has the same input of the algorithm and returns a list with three components given by the set of coset leaders with respect to any total degree compatible ordering, the function *Matphi* and the error correction capacity of a given binary code.

Example 2.17. Consider the $[n = 10, k = 4, d = 4]$ binary code $\mathscr{C}$ defined by the following parity check matrix:

$$H_{\mathscr{C}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{F}_2^{6\times 10}.$$

The CLBC function returns the list showed in Table 2.1 of cosets leaders with 64 components where each component corresponds to a coset with its coset leaders ordered w.r.t. the `degrevlex` ordering $\prec$ with $x_1 \prec \ldots \prec x_{10}$. Thus, the set $\mathscr{N}$ is built up by the first element of each component. Recall that this function also returns the map *Matphi* and the error correction capacity of $\mathscr{C}$, in this case $t = 1$. We denote by $\mathrm{CL}(\mathscr{C})_i^j$ the j-th element of the set of coset leaders of weight i.

Note that no subword of two elements of $\mathbf{y} = \mathbf{e}_4 + \mathbf{e}_5 + \mathbf{e}_6 \in \mathrm{CL}(\mathscr{C})_3^2$ is part of $\mathscr{N}$, i.e. $\mathbf{e}_4 + \mathbf{e}_5 \in \mathrm{CL}(\mathscr{C})_2^7$, $\mathbf{e}_4 + \mathbf{e}_6 \in \mathrm{CL}(\mathscr{C})_2^{13}$ and $\mathbf{e}_5 + \mathbf{e}_6 \in \mathrm{CL}(\mathscr{C})_2^1$ do not lie in $\mathscr{N}$. Therefore, the importance of the second property of the Definition 2.7 to obtain the complete set of coset leaders.

The algorithm could be adapted without incrementing the complexity to get more information such as:

- The *Newton radius* $\nu(\mathscr{C})$ of a binary code $\mathscr{C}$ is the largest weight of any vector that can be uniquely corrected, or equivalently, $\nu(\mathscr{C})$ is the largest value among the cosets with only one coset leader since an error is uniquely correctable if and only if it is the unique coset leader in its coset. Recall that every coset

Coset Leaders $\mathrm{CL}(\mathscr{C})$	
$\mathrm{CL}(\mathscr{C})_0$	$[\mathbf{0}]$
$\mathrm{CL}(\mathscr{C})_1$	$[\mathbf{e}_1]$, $[\mathbf{e}_2]$, $[\mathbf{e}_3]$, $[\mathbf{e}_4]$, $[\mathbf{e}_5]$, $[\mathbf{e}_6]$, $[\mathbf{e}_7]$, $[\mathbf{e}_8]$, $[\mathbf{e}_9]$, $[\mathbf{e}_{10}]$,
$\mathrm{CL}(\mathscr{C})_2$	$[\mathbf{e}_1+\mathbf{e}_2, \mathbf{e}_5+\mathbf{e}_6]$, $[\mathbf{e}_1+\mathbf{e}_3, \mathbf{e}_5+\mathbf{e}_7]$, $[\mathbf{e}_1+\mathbf{e}_4, \mathbf{e}_5+\mathbf{e}_8]$, $[\mathbf{e}_1+\mathbf{e}_5, \mathbf{e}_2+\mathbf{e}_6, \mathbf{e}_3+\mathbf{e}_7, \mathbf{e}_4+\mathbf{e}_8]$, $[\mathbf{e}_1+\mathbf{e}_6, \mathbf{e}_2+\mathbf{e}_5]$, $[\mathbf{e}_1+\mathbf{e}_7, \mathbf{e}_3+\mathbf{e}_5]$, $[\mathbf{e}_1+\mathbf{e}_8, \mathbf{e}_4+\mathbf{e}_5]$, $[\mathbf{e}_1+\mathbf{e}_9]$, $[\mathbf{e}_1+\mathbf{e}_{10}]$, $[\mathbf{e}_2+\mathbf{e}_3, \mathbf{e}_6+\mathbf{e}_7]$, $[\mathbf{e}_2+\mathbf{e}_4, \mathbf{e}_6+\mathbf{e}_8]$, $[\mathbf{e}_2+\mathbf{e}_7, \mathbf{e}_3+\mathbf{e}_6]$, $[\mathbf{e}_2+\mathbf{e}_8, \mathbf{e}_4+\mathbf{e}_6]$, $[\mathbf{e}_2+\mathbf{e}_9]$, $[\mathbf{e}_2+\mathbf{e}_{10}]$, $[\mathbf{e}_3+\mathbf{e}_4, \mathbf{e}_7+\mathbf{e}_8]$, $[\mathbf{e}_3+\mathbf{e}_8, \mathbf{e}_4+\mathbf{e}_7]$, $[\mathbf{e}_3+\mathbf{e}_9]$, $[\mathbf{e}_3+\mathbf{e}_{10}]$, $[\mathbf{e}_4+\mathbf{e}_9]$, $[\mathbf{e}_4+\mathbf{e}_{10}]$, $[\mathbf{e}_5+\mathbf{e}_9]$, $[\mathbf{e}_5+\mathbf{e}_{10}]$, $[\mathbf{e}_6+\mathbf{e}_9]$, $[\mathbf{e}_6+\mathbf{e}_{10}]$, $[\mathbf{e}_7+\mathbf{e}_9]$, $[\mathbf{e}_7+\mathbf{e}_{10}]$, $[\mathbf{e}_8+\mathbf{e}_9]$, $[\mathbf{e}_8+\mathbf{e}_{10}]$, $[\mathbf{e}_9+\mathbf{e}_{10}]$,
$\mathrm{CL}(\mathscr{C})_3$	$[\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_3, \mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_7, \mathbf{e}_2+\mathbf{e}_5+\mathbf{e}_7, \mathbf{e}_3+\mathbf{e}_5+\mathbf{e}_6]$, $[\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_4, \mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_8, \mathbf{e}_2+\mathbf{e}_5+\mathbf{e}_8, \mathbf{e}_4+\mathbf{e}_5+\mathbf{e}_6]$, $[\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_7, \mathbf{e}_1+\mathbf{e}_3+\mathbf{e}_6, \mathbf{e}_2+\mathbf{e}_3+\mathbf{e}_5, \mathbf{e}_5+\mathbf{e}_6+\mathbf{e}_7]$, $[\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_8, \mathbf{e}_1+\mathbf{e}_4+\mathbf{e}_6, \mathbf{e}_2+\mathbf{e}_4+\mathbf{e}_5, \mathbf{e}_5+\mathbf{e}_6+\mathbf{e}_8]$, $[\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_9, \mathbf{e}_5+\mathbf{e}_6+\mathbf{e}_9]$, $[\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_{10}, \mathbf{e}_5+\mathbf{e}_6+\mathbf{e}_{10}]$, $[\mathbf{e}_1+\mathbf{e}_3+\mathbf{e}_4, \mathbf{e}_1+\mathbf{e}_7+\mathbf{e}_8, \mathbf{e}_3+\mathbf{e}_5+\mathbf{e}_8, \mathbf{e}_4+\mathbf{e}_5+\mathbf{e}_7]$, $[\mathbf{e}_1+\mathbf{e}_3+\mathbf{e}_8, \mathbf{e}_1+\mathbf{e}_4+\mathbf{e}_7, \mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_5, \mathbf{e}_5+\mathbf{e}_7+\mathbf{e}_8]$, $[\mathbf{e}_1+\mathbf{e}_3+\mathbf{e}_9, \mathbf{e}_5+\mathbf{e}_7+\mathbf{e}_9]$, $[\mathbf{e}_1+\mathbf{e}_3+\mathbf{e}_{10}, \mathbf{e}_5+\mathbf{e}_7+\mathbf{e}_{10}]$, $[\mathbf{e}_1+\mathbf{e}_4+\mathbf{e}_9, \mathbf{e}_5+\mathbf{e}_8+\mathbf{e}_9]$, $[\mathbf{e}_1+\mathbf{e}_4+\mathbf{e}_{10}, \mathbf{e}_5+\mathbf{e}_8+\mathbf{e}_{10}]$, $[\mathbf{e}_1+\mathbf{e}_5+\mathbf{e}_9, \mathbf{e}_2+\mathbf{e}_6+\mathbf{e}_9, \mathbf{e}_3+\mathbf{e}_7+\mathbf{e}_9, \mathbf{e}_4+\mathbf{e}_8+\mathbf{e}_9]$, $[\mathbf{e}_1+\mathbf{e}_5+\mathbf{e}_{10}, \mathbf{e}_2+\mathbf{e}_6+\mathbf{e}_{10}, \mathbf{e}_3+\mathbf{e}_7+\mathbf{e}_{10}, \mathbf{e}_4+\mathbf{e}_8+\mathbf{e}_{10}]$, $[\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_9, \mathbf{e}_2+\mathbf{e}_5+\mathbf{e}_9]$, $[\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_{10}, \mathbf{e}_2+\mathbf{e}_5+\mathbf{e}_{10}]$, $[\mathbf{e}_1+\mathbf{e}_7+\mathbf{e}_9, \mathbf{e}_3+\mathbf{e}_5+\mathbf{e}_9]$, $[\mathbf{e}_1+\mathbf{e}_7+\mathbf{e}_{10}, \mathbf{e}_3+\mathbf{e}_5+\mathbf{e}_{10}]$, $[\mathbf{e}_1+\mathbf{e}_8+\mathbf{e}_9, \mathbf{e}_4+\mathbf{e}_5+\mathbf{e}_9]$, $[\mathbf{e}_1+\mathbf{e}_8+\mathbf{e}_{10}, \mathbf{e}_4+\mathbf{e}_5+\mathbf{e}_{10}]$, $[\mathbf{e}_1+\mathbf{e}_9+\mathbf{e}_{10}]$, $[\mathbf{e}_2+\mathbf{e}_3+\mathbf{e}_8, \mathbf{e}_2+\mathbf{e}_4+\mathbf{e}_7, \mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_6, \mathbf{e}_6+\mathbf{e}_7+\mathbf{e}_8]$, $[\mathbf{e}_5+\mathbf{e}_9+\mathbf{e}_{10}]$

Table 2.1: List of coset leaders in Example 2.17

of weight at most $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ has a unique coset leader. However, there are errors of weight greater than t which are also uniquely correctable. Therefore, the study of the Newton radius is important when we want to study decoding beyond half the minimum distance. See [47, 54] for a deeper discussion of this parameter.

In our example it suffices to analyze the last element of the list $\mathrm{CL}(\mathscr{C})$ to obtain the coset of highest weight which contains only one leader, i.e. $\nu(\mathscr{C}) = 3$ since $\mathrm{CL}(\mathscr{C})_3^{23} = [\mathbf{e}_5 + \mathbf{e}_9 + \mathbf{e}_{10}]$.

- The *Covering radius* $\rho(\mathscr{C})$ of a binary code $\mathscr{C}$ is the smallest integer s such that $\mathbb{F}_2^n$ is the union of the spheres of radius s centered at the codewords of $\mathscr{C}$, i.e. $\rho(\mathscr{C}) = \max_{\mathbf{y}\in\mathbb{F}_2^n} \min_{\mathbf{c}\in\mathscr{C}} d_H(\mathbf{y},\mathbf{c})$. It is well known that $\rho(\mathscr{C})$ is the weight of the coset of largest weight. Likewise, in our example $\rho(\mathscr{C}) = 3$ since $\mathrm{CL}(\mathscr{C})_3^{23} = [\mathbf{e}_5 + \mathbf{e}_9 + \mathbf{e}_{10}]$ is the coset of highest weight.

- The *Weight Distribution of the Coset Leaders* of a binary code $\mathscr{C}$ is the list

$$\mathrm{WDCL} = (\alpha_0, \ldots, \alpha_n) \text{ where } \alpha_i \text{ with } 1 \leq i \leq n$$

  is the number of cosets with coset leaders of weight i, i.e. $|\mathrm{CL}(\mathscr{C})_i|$. Note that the set $\mathscr{N}$ is enough to compute this parameter. It is clear that

$$\mathrm{WDCL} = \left[\begin{array}{cccccccccc} 1, & 10, & 30, & 23, & 0, & 0, & 0, & 0, & 0, & 0 \end{array} \right].$$

- The number of coset leaders in each coset :

$$\sharp(\mathrm{CL}) = \left[\begin{array}{l} 1, \\ 1,1,1,1,1,1,1,1,1,1, \\ 2,2,2,4,2,2,2,1,1,2,2,2,2,1,1,2,2,1,1,1,1,1,1,1,1,1,1,1, \\ 4,4,4,4,2,2,4,4,2,2,2,2,4,4,2,2,2,2,2,2,1,4,1 \end{array} \right]$$

  Note that there are 30 of the 64 cosets where the Complete Decoding Problem (CDP) has a unique solution. It is also interesting to note that amongst the cosets with one leaders there are more cosets exceeding the error correction capacity (19) than achieving such capacity (11).

Example 2.18. Continuing with Example 2.4 and now using Algorithm 7 we obtain the list of cosets leaders given by table 2.2.

Moreover in the same manner as Example 2.17 we can obtain the following additional information: the *Newton radius* of $\mathscr{C}$ is $\nu(\mathscr{C}) = 1$, the *Covering radius* of $\mathscr{C}$ is $\rho(\mathscr{C}) = 2$, the *Weight Distribution of the Coset Leaders* of $\mathscr{C}$ is given by

$$\mathrm{WDCL} = \left[\begin{array}{cccccc} 1, & 6, & 1, & 0, & 0, & 0 \end{array} \right].$$

and the number of coset leaders in each coset by

$$\sharp(\mathrm{CL}) = \left[\begin{array}{l} 1, \\ 1,1,1,1,1,1 \\ 3 \end{array} \right].$$

Coset Leaders $\mathrm{CL}(\mathscr{C})$	
$\mathrm{CL}(\mathscr{C})_0$	$[\mathbf{0}]$
$\mathrm{CL}(\mathscr{C})_1$	$[\mathbf{e}_1]$, $[\mathbf{e}_2]$, $[\mathbf{e}_3]$, $[\mathbf{e}_4]$, $[\mathbf{e}_5]$, $[\mathbf{e}_6]$
$\mathrm{CL}(\mathscr{C})_2$	$[\mathbf{e}_1+\mathbf{e}_6,\ \mathbf{e}_2+\mathbf{e}_3,\ \mathbf{e}_4+\mathbf{e}_5]$

Table 2.2: List of coset leaders in Example 2.18

For all $e \in \mathbb{Z}_{\geq 0}$ and $\mathbf{v} \in \mathbb{F}_q^n$ the set $B(\mathbf{v}, e) := \left\{\mathbf{w} \in \mathbb{F}_q^n \mid d_H(\mathbf{v}, \mathbf{w}) \leq e\right\}$ is called *sphere* around $\mathbf{v}$ with radius e respect to the Hamming metric. Note that its cardinality is $|B(\mathbf{v}, e)| = \sum_{i=0}^{e} \binom{n}{i}(q-1)^i$.

As we have already defined, the *covering radius* of an $[n, k]$ linear code over $\mathbb{F}_q$ is defined as the smallest integer $\rho(\mathscr{C})$ such that the spheres of radius $\rho(\mathscr{C})$ around the codewords of $\mathscr{C}$ completely cover the space $\mathbb{F}_q^n$. Or equivalently,

$$\rho(\mathscr{C}) = \max_{\mathbf{y} \in \mathbb{F}_q^n} \min_{\mathbf{c} \in \mathscr{C}} d_H(\mathbf{y}, \mathbf{c}).$$

Note that the statement $B(\mathbf{c}, e) \cap B(\hat{\mathbf{c}}, e) = \emptyset$ holds true for all $\mathbf{c}, \hat{\mathbf{c}} \in \mathscr{C}$ with $\mathbf{c} \neq \hat{\mathbf{c}}$ if and only if $2e + 1 \leq d(\mathscr{C})$ is valid. Moreover, $\mathbb{F}_q^n = \cup_{\mathbf{c} \in \mathscr{C}} B(\mathbf{c}, e)$ holds true if and only if $\rho(\mathscr{C}) \leq e$. Therefore the minimum distance and the covering radius of any code are related by $d(\mathscr{C}) \leq 2\rho(\mathscr{C}) + 1$.

Lemma 2.19. *For any $[n, k]$ binary code $\mathscr{C}$ the following inequality holds:*

$$\sum_{i=0}^{t} \binom{n}{i} \leq |\mathrm{CL}(\mathscr{C})| \leq \sum_{j=0}^{\rho(\mathscr{C})} \binom{n}{j},$$

where t denotes the error-correcting capacity of $\mathscr{C}$ and $\rho(\mathscr{C})$ its covering radius.

Proof. Let us first prove that every vector $\mathbf{e} \in \mathbb{F}_2^n$ with $\mathrm{w}_H(\mathbf{e}) \leq t$ is a coset leader. Assume to the contrary that there exists a vector $\mathbf{e} \in \mathbb{F}_2^n$ with $\mathrm{w}_H(\mathbf{e}) \leq t$ and $\mathbf{e} \notin \mathrm{CL}(\mathscr{C})$. Hence there is another vector $\hat{\mathbf{e}} \in \mathbb{F}_2^n$ with $S(\mathbf{e}) = S(\hat{\mathbf{e}})$ and $\mathrm{w}_H(\hat{\mathbf{e}}) < \mathrm{w}_H(\mathbf{e})$. Or equivalently, there exists a codeword $\mathbf{e} - \hat{\mathbf{e}} \in \mathscr{C}$ with

$$\mathrm{w}_H(\mathbf{e} - \hat{\mathbf{e}}) \leq \mathrm{w}_H(\mathbf{e}) + \mathrm{w}_H(\hat{\mathbf{e}}) \leq 2t \leq d(\mathscr{C}) - 1$$

which is a contradiction to the definition of the minimum distance of $\mathscr{C}$. For the previous inequalities, recall that if $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$ then

$$\mathrm{w}_H(\mathbf{x} + \mathbf{y}) = \mathrm{w}_H(\mathbf{x}) + \mathrm{w}_H(\mathbf{y}) - 2|\mathrm{supp}(\mathbf{x}) \cap \mathrm{supp}(\mathbf{y})|.$$

Moreover the error-correcting capacity of a linear code $\mathscr{C}$ is defined as $t = \left\lfloor \frac{d(\mathscr{C})-1}{2} \right\rfloor$ where $\lfloor \cdot \rfloor$ denotes the greatest integer function. Hence, we have actually proved that the number of vectors of weight up to t is a lower bound for the cardinality of the set $\mathrm{CL}(\mathscr{C})$, i.e.

$$\sum_{i=0}^{t} \binom{n}{i} \leq |\mathrm{CL}(\mathscr{C})|.$$

Furthermore, by the definition of the covering radius of $\mathscr{C}$, we have that for all $\mathbf{y} \in \mathbb{F}_2^n$ there exists a codeword $\mathbf{c} \in \mathscr{C}$ such that $\mathrm{d}_H(\mathbf{c},\mathbf{y}) \leq \rho(\mathscr{C})$. In other words, there exists a vector $\mathbf{e} \in \mathbb{F}_2^n$ such that $\mathrm{w}_H(\mathbf{e}) \leq \rho(\mathscr{C})$ and $S(\mathbf{e}) = S(\mathbf{y})$. Thus, $\mathrm{w}_H(\mathrm{CL}(\mathbf{y})) \leq \rho(\mathscr{C})$ and the lemma holds. □

Remark 2.20. If the above lemma holds with equality then $\mathscr{C}$ is called a *perfect* code. That is to say, let $\mathscr{C}$ be a linear code with more than one codeword, then $\mathscr{C}$ is a perfect code if and only if $\rho(\mathscr{C}) = t$.

Remark 2.21. As presented in [58, Section 11.7], there is a natural relationship between the cosets of an $[n,k]$ binary code $\mathscr{C}$ based on the following partial ordering $\leq$ on the set of cosets of $\mathscr{C}$

$$\mathbf{x}+\mathscr{C} \leq \mathbf{y}+\mathscr{C} \iff \mathrm{supp}(\overline{\mathbf{x}}) \subseteq \mathrm{supp}(\overline{\mathbf{y}}) \quad \text{with } \overline{\mathbf{x}} \in \mathrm{CL}(\mathbf{x}) \text{ and } \overline{\mathbf{y}} \in \mathrm{CL}(\mathbf{y}).$$

Under this partial ordering we can develop a hierarchy on the cosets:

- If $\mathbf{x}+\mathscr{C} < \mathbf{y}+\mathscr{C}$ then $\mathbf{x}+\mathscr{C}$ is called a *descendant* of $\mathbf{y}+\mathscr{C}$. While $\mathbf{y}+\mathscr{C}$ turn to be an *ancestor* of $\mathbf{x}+\mathscr{C}$.

  Note that if $\mathbf{x}+\mathscr{C} < \mathbf{y}+\mathscr{C}$ then $\mathrm{w}_H(\mathbf{x}+\mathscr{C}) \leq \mathrm{w}_H(\mathbf{y}+\mathscr{C}) - 1$.

- $\mathbf{x}+\mathscr{C}$ is a *child* of $\mathbf{y}+\mathscr{C}$ if $\mathbf{x}+\mathscr{C} < \mathbf{y}+\mathscr{C}$ and $\mathrm{w}_H(\mathbf{x}+\mathscr{C}) = \mathrm{w}_H(\mathbf{y}+\mathscr{C}) - 1$. In such case, $\mathbf{y}+\mathscr{C}$ is viewed as a *parent* of $\mathbf{x}+\mathscr{C}$.

- A coset is an *orphan* if its has no parents.

Therefore, all cosets of maximum weight (that is of weight $\rho(\mathscr{C})$, the covering radius of $\mathscr{C}$), is an orphan. Moreover, The code $\mathscr{C}$ is the unique minimal element of the set of cosets.

Diagram 2.1 shows an scheme of this hierarchy using the code of Example 2.4, where the different cosets of $\mathscr{C}$ are represented by a coset leader, i.e.

$$\begin{array}{lll} \mathbf{n}_0 = [\mathbf{0}]+\mathscr{C}, & \mathbf{n}_3 = [\mathbf{e}_3]+\mathscr{C}, & \mathbf{n}_6 = [\mathbf{e}_6]+\mathscr{C}, \\ \mathbf{n}_1 = [\mathbf{e}_1]+\mathscr{C}, & \mathbf{n}_4 = [\mathbf{e}_4]+\mathscr{C}, & \mathbf{n}_7 = [\mathbf{e}_1+\mathbf{e}_6]+\mathscr{C}. \\ \mathbf{n}_2 = [\mathbf{e}_2]+\mathscr{C}, & \mathbf{n}_5 = [\mathbf{e}_5]+\mathscr{C}, & \end{array}$$

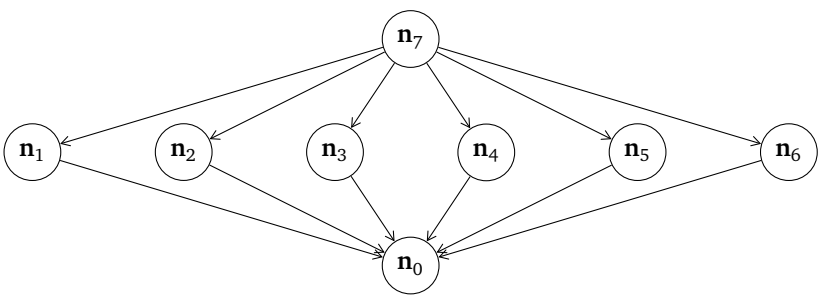


Figure 2.1: Hierarchy relationship amongst the cosets of a binary code.

Remark 2.22. A Maximum-likelihood List Decoding (MLLD) Algorithm takes an arbitrary received vector as input and produces as output a list of all error patterns in the same coset which have minimal weight within the coset, i.e. the set of coset leaders.

2.3 Computing leader codewords

The following Algorithm is just an adaptation of Algorithm 7 for computing a test-set.

Definition 2.23. The *set of leader codewords* of a given code $\mathscr{C}$ is defined as:

$$\mathrm{L}(\mathscr{C}) = \left\{ \mathbf{n}_1 + \mathbf{n}_2 + \mathbf{e}_i \;\middle|\; \begin{array}{c} i \notin \mathrm{supp}(\mathbf{n}_1),\ \mathbf{n}_1 + \mathbf{e}_i \neq \mathbf{n}_2, \\ \mathbf{n}_1, \mathbf{n}_2 \in \mathrm{CL}(\mathscr{C}) \text{ and } S(\mathbf{n}_1 + \mathbf{e}_i) = S(\mathbf{n}_2) \end{array} \right\}.$$

For efficiency reasons we are just interested in a particular case of the above objects, when $\mathrm{supp}(\mathbf{n}_1 + \mathbf{e}_i) \cap \mathrm{supp}(\mathbf{n}_2) = \emptyset$.

Algorithm 8: CLBC2 Algorithm

Data: A weight compatible ordering $\prec$ and a parity check matrix H of a binary code $\mathscr{C}$.
Result: The set of coset leaders $\mathrm{CL}(\mathscr{C})$ and the set of leader codewords $\mathrm{L}(\mathscr{C})$ for $\mathscr{C}$.

1 $\texttt{List} \longleftarrow [0]$; $\mathscr{N} \longleftarrow \emptyset$; $r \longleftarrow 0$; $\mathrm{CL}(\mathscr{C}) \longleftarrow \emptyset$; $S \longleftarrow \emptyset$; $\mathrm{L}(\mathscr{C}) \longleftarrow \emptyset$;
2 **while** $\texttt{List} \neq \emptyset$ **do**
3 $\quad \mathbf{t} \longleftarrow \texttt{NextTerm[List]}$; $\mathbf{s} \longleftarrow \mathbf{t}H^T$;
4 $\quad k \longleftarrow \texttt{Member}[s, S]$;
5 $\quad$ **if** $k \neq \texttt{false}$ **then**
6 $\quad\quad$ **if** $\mathrm{w}_H(\mathbf{t}) = \mathrm{w}_H(\mathbf{t}_k)$ **then**
7 $\quad\quad\quad \mathrm{CL}(\mathscr{C})[\mathbf{t}_k] \longleftarrow \mathrm{CL}(\mathscr{C})[\mathbf{t}_k] \cup \{\mathbf{t}\}$;
8 $\quad\quad\quad \texttt{List} \longleftarrow \texttt{InsertNext}[\mathbf{t}, \texttt{List}]$;
9 $\quad\quad$ **endif**
10 $\quad\quad$ **if** $\exists i \in \mathrm{supp}(\mathbf{t}) \ :\ \mathbf{t} = \mathbf{t}' + \mathbf{e}_i$ *with* $\mathbf{t}' \in \mathrm{CL}(\mathscr{C})$ *and* $i \notin \mathrm{supp}(\mathbf{t}')$ **then**
11 $\quad\quad\quad \mathrm{L}(\mathscr{C}) \longleftarrow$
$\quad\quad\quad \mathrm{L}(\mathscr{C}) \cup \left\{ \mathbf{t} + \mathbf{t}_j \mid \mathbf{t}_j \in \mathrm{CL}(\mathscr{C})[\mathbf{t}_k] \text{ and } \mathrm{supp}(\mathbf{t}) \cap \mathrm{supp}(\mathbf{t}_j) = \emptyset \right\}$
12 $\quad\quad$ **endif**
13 $\quad$ **else**
14 $\quad\quad r \longleftarrow r + 1$; $\mathbf{t}_r \longleftarrow \mathbf{t}$; $\mathscr{N} \longleftarrow \mathscr{N} \cup \{\mathbf{t}_r\}$;
15 $\quad\quad \mathrm{CL}(\mathscr{C})[\mathbf{t}_r] \longleftarrow \{\mathbf{t}_r\}$; $S \longleftarrow S \cup \{\mathbf{s}\}$;
16 $\quad\quad \texttt{List} = \texttt{InsertNext}[\mathbf{t}_r, \texttt{List}]$;
17 $\quad$ **endif**
18 **endw**

Remark 2.24. The difference between Algorithm 7 and Algorithm 8 are Steps **10-12**.

Theorem 2.25. *Algorithm 8 computes the set of coset leaders and the set of leader codewords of a given binary code $\mathscr{C}$.*

Proof. Taking into account Remark 2.24 and Theorem 2.12 we only need to prove that Algorithm 8 computes the set of leader codewords. We first observe that all the words of the set $\mathrm{L}(\mathscr{C})$ are leader codewords since $\mathbf{t} = \mathbf{t}' + \mathbf{e}_i$ and $\mathbf{t}_j$ are in the same coset, in particular $\mathbf{t}, \mathbf{t}_j \in \mathrm{CL}(\mathscr{C})[\mathbf{t}_k]$, and by definition we have

$$\mathbf{t}, \mathbf{t}_j \in \mathrm{CL}(\mathscr{C}), \quad \mathrm{supp}(\mathbf{t}) \cap \mathrm{supp}(\mathbf{t}_j) = \emptyset \quad \text{and} \quad i \notin \mathrm{supp}(\mathbf{t}').$$

Furthermore, Algorithm 8 ratifies that we are considering all the leader codewords, since the elements of `List` are ordered by $\prec$, then in each loop we study all leader codewords of the form $\mathbf{n}_1 + \mathbf{e}_i + \mathbf{n}_2$ with $\mathbf{n}_1, \mathbf{n}_2 \preceq \mathbf{t}$ with $\mathbf{t} \in \mathrm{CL}(\mathscr{C})$. □

Remark 2.26. By its construction, Algorithm 8 has the same time complexity as Algorithm 7. The advantage of computing the set of leader codewords is that it helps in solving the same problems as the function *Matphi* does but with a structure which is considerately smaller.

Example 2.27. Using Algorithm 8 with the $[6,3,3]$ binary code $\mathscr{C}$ of Example 2.4 we obtain the following set of leader codewords:

$$\mathrm{L}(\mathscr{C}) = \left\{ \begin{array}{cc} \mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_4, & \mathbf{e}_1+\mathbf{e}_3+\mathbf{e}_5, \\ \mathbf{e}_2+\mathbf{e}_5+\mathbf{e}_6, & \mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_6, \\ \mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_3+\mathbf{e}_6, & \mathbf{e}_1+\mathbf{e}_4+\mathbf{e}_5+\mathbf{e}_6, \\ \multicolumn{2}{c}{\mathbf{e}_2+\mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_5} \end{array} \right\}.$$

Note that all nonzero codewords of $\mathscr{C}$ are in $\mathrm{L}(\mathscr{C})$.

Definition 2.28. We define the subset $\mathrm{L}^1(\mathscr{C})$ of $\mathrm{L}(\mathscr{C})$ as

$$\mathrm{L}^1(\mathscr{C}) = \left\{ \mathbf{n}_1 + \mathbf{n}_2 + \mathbf{e}_i \;\middle|\; \begin{array}{c} i \notin \mathrm{supp}(\mathbf{n}_1),\ \mathbf{n}_1 \in \mathrm{CL}(\mathscr{C}),\ \mathbf{n}_2 \in \mathscr{N}, \\ \mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_i) > \mathrm{w}_H(\mathbf{n}_2) \text{ and } S(\mathbf{n}_1 + \mathbf{e}_i) = S(\mathbf{n}_2) \end{array} \right\}.$$

Following Definition 2.9, $\mathscr{N}$ denotes the set of minimal elements w.r.t. a weight compatible ordering $\prec$ of each coset.

Remark 2.29. Note that the condition $\mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_i) > \mathrm{w}_H(\mathbf{n}_2)$ is imposed just to improve the efficiency of computing this set. Therefore $\mathrm{L}(\mathscr{C})$ can be rewritten as

$$\mathrm{L}(\mathscr{C}) = \left\{ \mathbf{n}_1 + \mathbf{n}_2 + \mathbf{e}_i \;\middle|\; \begin{array}{c} i \notin \mathrm{supp}(\mathbf{n}_1),\ \mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_i) > \mathrm{w}_H(\mathbf{n}_2) \\ \mathbf{n}_1, \mathbf{n}_2 \in \mathrm{CL}(\mathscr{C}) \text{ and } S(\mathbf{n}_1 + \mathbf{e}_i) = S(\mathbf{n}_2) \end{array} \right\}.$$

Thus, the only difference between the sets $\mathrm{L}^1(\mathscr{C})$ and $\mathrm{L}(\mathscr{C})$ is that $\mathbf{n}_2 \in \mathscr{N}$ instead of $\mathbf{n}_2 \in \mathrm{CL}(\mathscr{C})$. In other words, the element $\mathbf{n}_2$ in $\mathrm{L}^1(\mathscr{C})$ is required not only to belong to the set of the coset leaders but also to be the smallest element in its coset according to a fix weight compatible ordering $\prec$.

Example 2.30. From Example 2.18, note that the only nonzero codeword of $\mathrm{L}(\mathscr{C})$ that is missing in $\mathrm{L}^1(\mathscr{C})$ is $\mathbf{e}_2+\mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_5$.

Theorem 2.31. *The subset* $\mathrm{L}^1(\mathscr{C}) \subseteq \mathrm{L}(\mathscr{C})$ *is a test-set for* $\mathscr{C}$.

Proof. Let us consider a word $\mathbf{y} \in \mathbb{F}_2^n$ with $\mathrm{supp}(\mathbf{y}) = \{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ such that $\mathbf{y} \notin \mathrm{CL}(\mathscr{C})$. Thus, there must exist an integer $1 \leq l < m$ such that

$$\mathbf{n}_1 := \mathbf{e}_{i_1} + \ldots + \mathbf{e}_{i_l} \in \mathrm{CL}(\mathscr{C}) \quad \text{and} \quad \mathbf{n}_1 + \mathbf{e}_{i_{l+1}} \notin \mathrm{CL}(\mathscr{C}).$$

We define $\mathbf{n}_2 = \mathscr{N}\left(\mathbf{n}_1 + \mathbf{e}_{i_{l+1}}\right)$, i.e. $\mathbf{n}_2$ is the smallest element in the coset of $\mathbf{n}_1 + \mathbf{e}_{i_{l+1}}$ according to a fix compatible weight ordering $\succ$. Since $\mathbf{n}_1 + \mathbf{e}_{i_{l+1}} \notin \mathrm{CL}(\mathscr{C})$ we have that $\mathrm{w}_H(\mathbf{n}_2) < \mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_{i_{l+1}})$. Thus $\mathbf{t} = \mathbf{n}_1 + \mathbf{n}_2 + \mathbf{e}_{i_{l+1}} \in \mathrm{L}^1(\mathscr{C})$.

Without loss of generality we may assume that $\mathrm{supp}(\mathbf{n}_1 + \mathbf{e}_{i_{l+1}}) \cap \mathrm{supp}(\mathbf{n}_2) = \emptyset$. Indeed,

- if $i_{l+1} \in \mathrm{supp}(\mathbf{n}_2)$ then, by Theorem 1.18, $\mathbf{n}_2 + \mathbf{e}_{i_{l+1}} \in \mathrm{CL}(\mathscr{C})$. Moreover

  $$S(\mathbf{n}_2 + \mathbf{e}_{i_{l+1}}) = S(\mathbf{n}_1) \quad \text{and} \quad \mathrm{w}_H(\mathbf{n}_2 + \mathbf{e}_{i_{l+1}}) < \mathrm{w}_H(\mathbf{n}_1) \ ,$$

  which contradicts the fact that $\mathbf{n}_1 \in \mathrm{CL}(\mathscr{C})$.

- Otherwise, if there exists $j \in \mathrm{supp}(\mathbf{n}_1) \cap \mathrm{supp}(\mathbf{n}_2)$. Then, we may define

  $$\overline{\mathbf{n}_1} = \mathbf{n}_1 + \mathbf{e}_j \quad \text{and} \quad \overline{\mathbf{n}_2} = \mathbf{n}_2 + \mathbf{e}_j \ .$$

  Note that, by Theorem 1.18, $\overline{\mathbf{n}_1}, \overline{\mathbf{n}_2} \in \mathrm{CL}(\mathscr{C})$. Furthermore,

  $$\mathrm{w}_H(\overline{\mathbf{n}_1} + \mathbf{e}_{i_{l+1}}) < \mathrm{w}_H(\overline{\mathbf{n}_2}) \quad \text{and} \quad S\left(\overline{\mathbf{n}_1} + \mathbf{e}_{i_{l+1}}\right) = S(\overline{\mathbf{n}_2}).$$

  Thus, $\mathbf{t} = \overline{\mathbf{n}_1} + \overline{\mathbf{n}_2} + \mathbf{e}_{i_{l+1}} \in \mathrm{L}^1(\mathscr{C})$.

Therefore,

$$|\mathrm{supp}(\mathbf{t}) \cap \mathrm{supp}(\mathbf{y})| \geq \mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_{i_{l+1}}) > \mathrm{w}_H(\mathbf{n}_2) \geq |\mathrm{supp}(\mathbf{t}) \cap \mathrm{supp}(\overline{\mathbf{y}})|$$

where $\overline{\mathbf{y}}$ denotes the relative complement of $\mathbf{y}$ in $\mathbb{F}_2^n$, and in consequence, $\mathrm{w}_H(\mathbf{y}-\mathbf{t}) < \mathrm{w}_H(\mathbf{y})$ which completes the proof. □

Remark 2.32. Since $\mathrm{L}^1(\mathscr{C}) \subseteq \mathrm{L}(\mathscr{C})$ and by Theorem 2.31 the subset $\mathrm{L}^1(\mathscr{C})$ is a test-set for $\mathscr{C}$, then so is the set $\mathrm{L}(\mathscr{C})$.

The following theorem gives a bound for the weight of a leader codeword of a given binary code $\mathscr{C}$.

Theorem 2.33. *Let* $\mathbf{c} \in \mathrm{L}(\mathscr{C})$ *then* $\mathrm{w}_H(\mathbf{c}) \leq 2\rho(\mathscr{C}) + 1$ *where* $\rho(\mathscr{C})$ *is the covering radius of* $\mathscr{C}$.

Proof. Let $\mathbf{c} \in \mathrm{L}(\mathscr{C})$ then there exists $\mathbf{n}_1,\ \mathbf{n}_2 \in \mathrm{CL}(\mathscr{C})$ and $i \notin \mathrm{supp}(\mathbf{n}_1)$ such that $\mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_i) > \mathrm{w}_H(\mathbf{n}_2)$ and $\mathbf{c} = \mathbf{n}_1 + \mathbf{e}_i + \mathbf{n}_2$. Applying the definition of covering radius we have that $\mathrm{w}_H(\mathbf{n}_1), \mathrm{w}_H(\mathbf{n}_2) \leq \rho(\mathscr{C})$, thus $\mathrm{w}_H(\mathbf{c}) \leq 2\rho(\mathscr{C}) + 1$. □

Algorithm 9: Computing the set $\mathrm{CL}(\mathbf{y})$

Data: A received vector $\mathbf{y} \in \mathbb{F}_2^n$ and the set of leader codewords $\mathrm{L}(\mathscr{C})$ of $\mathscr{C}$.
Result: The subset $\mathrm{CL}(\mathbf{y})$ of coset leaders corresponding to the coset $\mathbf{y}+\mathscr{C}$.

```
Compute N(y) by gradient-like decoding using the test-set L(C); ; // We refer
  the reader to Section 1.1.3
y ⟵ N(y); S ⟵ {y}; L ⟵ L(C);
while there exists c ∈ L : w_H(y − c) = w_H(y) do
    y ⟵ y − c; S ⟵ S ∪ {y};
    L ⟵ L − {c};
endw
Return S
```

Theorem 2.34. *Algorithm 9 computes, from the set* $\mathrm{L}(\mathscr{C})$*, the subset* $\mathrm{CL}(\mathbf{y})$ *of coset leaders corresponding to the coset* $\mathbf{y}+\mathscr{C}$ *for a given received vector* $\mathbf{y} \in \mathbb{F}_2^n$.

Proof. Let us first prove that every $\mathbf{z} \in \mathrm{CL}(\mathbf{y})$ can be rewritten as $\mathbf{z} = N(\mathbf{y}) - \mathbf{c}$ with $\mathbf{c} \in \mathrm{L}(\mathscr{C})$. Let $i \in \mathrm{supp}(\mathbf{z})$ then $\mathbf{z} = \mathbf{n}_1 + \mathbf{e}_i$ with $i \notin \mathrm{supp}(\mathbf{n}_1)$. Hence, by Theorem 1.18, $\mathbf{n}_1 \in \mathrm{CL}(\mathscr{C})$. Furthermore we have that

$$S(N(\mathbf{y})) = S(\mathbf{z}) \quad \text{and} \quad \mathrm{w}_H(\mathbf{n}_1) < \mathrm{w}_H(\mathbf{z}) = \mathrm{w}_H(N(\mathbf{y})).$$

Thus, from the definition of leader codewords, $\mathbf{c} = N(\mathbf{y}) + (\mathbf{n}_1 + \mathbf{e}_i) \in \mathrm{L}(\mathscr{C}) \subseteq \mathscr{C}$, or equivalently, $\mathbf{z} = \mathbf{n}_1 + \mathbf{e}_i = \mathbf{c} - N(\mathbf{y})$ with $\mathbf{y} \in L(\mathscr{C})$.

The proof is completed by noting that Theorem 2.31 guarantees **Step 1**. □

To show Algorithm 9, in the following lines a toy example will be fully discussed.

Example 2.35. Given the $[6,3,3]$ binary code $\mathscr{C}$ in Example 2.4 which corrects up to $t = 1$ error. Suppose we have a received word $\mathbf{y} = (1,1,1,1,1,0)$ with error vector $\mathbf{e} = \mathbf{e}_1$.

- **Initialization:** First we compute a coset leader of $\mathbf{y}+\mathscr{C}$ by gradient-like decoding using the test-set $\mathrm{L}(\mathscr{C})$. We obtain that $\mathbf{e}_1 \in \mathrm{CL}(\mathbf{y})$ and we check using Algorithm 9 if there are more coset leaders corresponding to the coset of the received word. The initial data is

  $$\mathbf{y} = \mathbf{e}_1, \quad S = \{\mathbf{y}\} \quad \text{and} \quad \mathrm{L} = \mathrm{L}(\mathscr{C})$$

  See Example 2.18 for a complete description of the set $\mathrm{L}(\mathscr{C})$.

- **Step 1:** Since there is no codeword $\mathbf{c} \in \mathrm{L}$ such that $\mathrm{w}_H(\mathbf{y}-\mathbf{c}) = \mathrm{w}_H(\mathbf{y})$ then the algorithm terminates and produces $\mathrm{CL}(\mathbf{y}) = \{\mathbf{e}_1\}$.

Now suppose we received the word $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_5$ with error vector $\mathbf{e} = \mathbf{e}_1 + \mathbf{e}_6$.

- **Initialization:** First we compute $N(\mathbf{y}) = \mathbf{e}_1 + \mathbf{e}_6$. Then we have as initial data:

  $$\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_6, \quad S = \{\mathbf{y}\} \quad \text{and} \quad \mathrm{L} = \mathrm{L}(\mathscr{C})$$

- **Step 1:** There exists $\mathbf{c}_1 = \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_5 + \mathbf{e}_6 \in \mathrm{L}$ such that $\mathrm{w}_H(\mathbf{y} - \mathbf{c}) = \mathbf{w}_H(\mathbf{y})$. So $\mathbf{y} - \mathbf{c}_1 = \mathbf{e}_4 + \mathbf{e}_5$ is inserted into the list S, $\mathbf{c}_1$ is removed from L and $\mathbf{y}$ is replaced by $\mathbf{y} - \mathbf{c}_1$.
- **Step 2:** There exists $\mathbf{c}_2 = \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5 \in \mathrm{L}$ verifying the required property. Thus, $\mathbf{y} - \mathbf{c}_2 = \mathbf{e}_2 + \mathbf{e}_3$ is added to the list S, $\mathbf{c}_2$ is subtracted from L and $\mathbf{y}$ is substituted by $\mathbf{y} - \mathbf{c}_2$.
- **Step 3:** There exists $\mathbf{c}_3 = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_6$ with the desired property. Thus $\mathbf{y} - \mathbf{c}_3 = \mathbf{e}_1 + \mathbf{e}_6$ is included into the list S (although it already exists in the list S), $\mathbf{c}_3$ is removed from L and $\mathbf{y}$ is substituted by $\mathbf{y} - \mathbf{c}_3$.
- **Step 4:** Since there is no other codeword $\mathbf{c} \in \mathrm{L}$ with the requested property then the algorithm terminates and returns

$$\mathrm{CL}(\mathbf{y}) = \left\{ \ \mathbf{e}_1 + \mathbf{e}_6, \quad \mathbf{e}_2 + \mathbf{e}_3, \quad \mathbf{e}_4 + \mathbf{e}_5 \ \right\}.$$

2.3.1 Leader codewords and zero neighbours

In this section we will give a brief review of basic concepts from [3, Section 3] and thus establish the relation between Zero-Neighbours and leader codewords of a binary code $\mathscr{C}$.

Definition 2.36. For any subset $A \subset \mathbb{F}_2^n$ we define $\mathscr{X}(A)$ as the set of words at Hamming distance 1 from A, i.e.

$$\mathscr{X}(A) = \left\{ \mathbf{y} \in \mathbb{F}_2^n \mid \min\left\{ d_H(\mathbf{y}, \mathbf{a}) \ : \ \mathbf{a} \in A \right\} = 1 \right\}.$$

We define the *boundary* of A as $\delta(A) = \mathscr{X}(A) \cup \mathscr{X}(\mathbb{F}_2^n \setminus A)$.

Definition 2.37. A nonzero codeword $\mathbf{c} \in \mathscr{C}$ is called *Zero-Neighbour* if its Voronoi region shares a common boundary with the set of coset leaders, i.e.

$$\delta(\mathrm{D}(\mathbf{c})) \cap \delta(\mathrm{D}(\mathbf{0})) \neq \emptyset.$$

We will denote by $\mathscr{Z}(\mathscr{C})$ the set of all Zero-Neighbours of $\mathscr{C}$, that is to say:

$$\mathscr{Z}(\mathscr{C}) = \{ \mathbf{z} \in \mathscr{C} \setminus \{\mathbf{0}\} \ : \ \delta(\mathrm{D}(\mathbf{z})) \cap \delta(\mathrm{D}(\mathbf{0})) \neq \emptyset \}.$$

Note that if $\mathbf{z} \in \mathscr{C} \setminus \{\mathbf{0}\}$ satisfies that $\mathscr{X}(\mathrm{D}(\mathbf{0})) \cap \mathrm{D}(\mathbf{z}) \neq \emptyset$, then $\mathbf{z} \in \mathscr{Z}(\mathscr{C})$. Furthermore $\mathscr{Z}(\mathscr{C})$ is a test-set for $\mathscr{C}$ (see for instance [3, Theorem 3.16]). However, the only property of the set $\mathscr{Z}(\mathscr{C})$ that is essential for successful decoding is

$$\mathscr{X}(\mathrm{D}(\mathbf{0})) \subseteq \bigcup_{\mathbf{z} \in \mathscr{Z}(\mathscr{C})} \mathrm{D}(\mathbf{z}).$$

Thus, if we restrict the set $\mathscr{Z}(\mathscr{C})$ to a smallest subset verifying the previous property we still have a test-set for $\mathscr{C}$. We will denote such subset of $\mathscr{Z}(\mathscr{C})$ by $\mathscr{Z}_{\min}(\mathscr{C})$. Note that the set $\mathscr{Z}_{\min}(\mathscr{C})$ may be not unique, however its size is well defined.

Theorem 2.38. *Let $\mathscr{C}$ be a binary code and $\mathbf{z} \in \mathscr{C} \setminus \{\mathbf{0}\}$. Then the following are equivalent:*

- $\mathscr{X}(\mathrm{D}(\mathbf{0})) \cap \mathrm{D}(\mathbf{z}) \neq \emptyset$.
- $\mathbf{z} \in \mathrm{L}(\mathscr{C})$.

Proof. If $\mathscr{X}(\mathrm{D}(\mathbf{0})) \cap \mathrm{D}(\mathbf{z}) \neq \emptyset$ then there exists $\mathbf{n}_1 \in \mathrm{D}(\mathbf{0}) = \mathrm{CL}(\mathscr{C})$ and $i \notin \mathrm{supp}(\mathbf{n}_1)$ such that $\mathbf{n}_1 + \mathbf{e}_i \in \mathscr{X}(\mathrm{D}(\mathbf{0}))$ and $\mathbf{n}_1 + \mathbf{e}_i \in \mathrm{D}(\mathbf{z})$. In other words,

$$\mathrm{w}_H(\mathbf{z} - (\mathbf{n}_1 + \mathbf{e}_i)) \leq \mathrm{w}_H(\mathbf{c} - (\mathbf{n}_1 + \mathbf{e}_i)) \text{ for all } \mathbf{c} \in \mathscr{C} \setminus \{\mathbf{z}\}, \tag{2.1}$$

or equivalently, $\mathbf{n}_2 = \mathbf{z} - (\mathbf{n}_1 + \mathbf{e}_i) \in \mathrm{CL}(\mathscr{C})$ with $\mathbf{z} \in \mathscr{C}$, thus $S(\mathbf{n}_2) = S(\mathbf{n}_1 + \mathbf{e}_i)$. Furthermore, the special case of $\mathbf{c} = \mathbf{0} \in \mathscr{C} \setminus \{\mathbf{z}\}$ of Equation 2.1 implies that $\mathrm{w}_H(\mathbf{n}_2) \leq \mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_i)$. Therefore, all conditions in Definition 2.23 are verified, i.e. $\mathbf{z} = \mathbf{n}_1 + \mathbf{n}_2 + \mathbf{e}_i \in \mathrm{L}(\mathscr{C})$.

Conversely, if $\mathbf{z} \in \mathrm{L}(\mathscr{C})$, then $\mathbf{z}$ can be rewritten as $\mathbf{z} = \mathbf{n}_1 + \mathbf{n}_2 + \mathbf{e}_i$ where

(1)	$\mathbf{n}_1,\ \mathbf{n}_2 \in \mathrm{CL}(\mathscr{C})$.	(3)	$\mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_i) > \mathrm{w}_H(\mathbf{n}_2)$.
(2)	$i \notin \mathrm{supp}(\mathbf{n}_1)$.	(4)	$S(\mathbf{n}_2) = S(\mathbf{n}_1 + \mathbf{e}_i)$.

Now (1) and (2) gives that $\mathbf{n}_1 + \mathbf{e}_i \in \mathscr{X}(\mathrm{D}(\mathbf{0}))$, whereas (1), (3) and (4) clearly force that $\mathrm{CL}(\mathbf{n}_1 + \mathbf{e}_i) = \mathbf{n}_2$, i.e. $\mathrm{w}_H(\mathbf{n}_2) \leq \mathrm{w}_H(\mathbf{n}_1 + \mathbf{e}_i + \mathbf{c})$ for all $\mathbf{c} \in \mathscr{C}$, or equivalently, $\mathbf{n}_1 + \mathbf{e}_i \in \mathrm{D}(\mathbf{z})$. Therefore, $\mathscr{X}(\mathrm{D}(\mathbf{0})) \cap \mathrm{D}(\mathbf{z}) \neq \emptyset$. □

Corollary 2.39. *Let $\mathscr{C}$ be a binary code then $\mathscr{Z}_{\min}(\mathscr{C}) \subseteq \mathrm{L}(\mathscr{C})$, for any minimal test-set $\mathscr{Z}_{\min}(\mathscr{C})$ obtained from $\mathscr{Z}(\mathscr{C})$.*

Proof. Let $\mathscr{Z}_{\min}(\mathscr{C})$ be a minimal test-set of $\mathscr{C}$ obtained from $\mathscr{Z}(\mathscr{C})$, then every $\mathbf{z} \in \mathscr{Z}_{\min}(\mathscr{C})$ satisfies that $\mathscr{X}(\mathrm{D}(\mathbf{0})) \cap \mathrm{D}(\mathbf{z}) \neq \emptyset$. Thus, by Theorem 2.38, we obtained the required result. □

Algorithm 8 gives the set of leader codewords $\mathrm{L}(\mathscr{C})$ of a binary code $\mathscr{C}$. Furthermore, any minimal test-set $\mathscr{Z}_{\min}$ is a subset of $\mathrm{L}(\mathscr{C})$. Thus, after performing redundancy elimination to $\mathrm{L}(\mathscr{C})$, a minimal test-set $\mathscr{Z}_{\min}$ can also be obtained.

Remark 2.40. There is a built in function CLBC2 in **GBLA_LC** which performs Algorithm 8 in order to compute $\mathrm{L}(\mathscr{C})$ and a variant of the same function which computes $\mathrm{L}^1(\mathscr{C})$.

Example 2.41. We use the same code of Example 2.17. Algorithm 8 returns $\mathrm{L}(\mathscr{C})$ and $\mathrm{L}^1(\mathscr{C})$, in this case we obtained that both sets coincide. We describe below the set of leader codewords with 14 elements of the given binary code $\mathscr{C}$.

$$\mathrm{L}(\mathscr{C}) = \mathrm{L}^1(\mathscr{C}) = \left\{ \begin{array}{cc} \mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_7+\mathbf{e}_8, & \mathbf{e}_2+\mathbf{e}_4+\mathbf{e}_6+\mathbf{e}_8, \\ \mathbf{e}_2+\mathbf{e}_3+\mathbf{e}_6+\mathbf{e}_7, & \mathbf{e}_1+\mathbf{e}_4+\mathbf{e}_5+\mathbf{e}_8, \\ \mathbf{e}_1+\mathbf{e}_3+\mathbf{e}_5+\mathbf{e}_7, & \mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_5+\mathbf{e}_6, \\ \mathbf{e}_4+\mathbf{e}_6+\mathbf{e}_7+\mathbf{e}_9+\mathbf{e}_{10}, & \mathbf{e}_3+\mathbf{e}_6+\mathbf{e}_8+\mathbf{e}_9+\mathbf{e}_{10}, \\ \mathbf{e}_2+\mathbf{e}_7+\mathbf{e}_8+\mathbf{e}_9+\mathbf{e}_{10}, & \mathbf{e}_2+\mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_9+\mathbf{e}_{10}, \\ \multicolumn{2}{c}{\mathbf{e}_1+\mathbf{e}_5+\mathbf{e}_6+\mathbf{e}_7+\mathbf{e}_8+\mathbf{e}_9+\mathbf{e}_{10},} \\ \multicolumn{2}{c}{\mathbf{e}_1+\mathbf{e}_3+\mathbf{e}_4+\mathbf{e}_5+\mathbf{e}_6+\mathbf{e}_9+\mathbf{e}_{10},} \\ \multicolumn{2}{c}{\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_4+\mathbf{e}_5+\mathbf{e}_7+\mathbf{e}_9+\mathbf{e}_{10},} \\ \multicolumn{2}{c}{\mathbf{e}_1+\mathbf{e}_2+\mathbf{e}_3+\mathbf{e}_5+\mathbf{e}_8+\mathbf{e}_9+\mathbf{e}_{10}} \end{array} \right\}.$$

Note that the only nonzero codeword of $\mathscr{C}$ that is missing in $\mathrm{L}(\mathscr{C})$ is the codeword $\mathbf{y} = (1,1,1,1,1,1,1,0,0)$ of weight 8. This result is consistent with the fact that the covering radius of $\mathscr{C}$ is $\rho(\mathscr{C}) = 3$, as shown in Example 2.17, and the statement of Theorem 2.33, where we proved that the weight of a leader codeword is always less or equal to $2\rho(\mathscr{C}) + 1 = 7$.

$$\mathscr{C} = \left\{ \begin{array}{ccc} (0,0,0,0,0,0,0,0,0,0), & (1,0,0,0,1,1,1,1,1,1), & (0,1,0,0,0,0,1,1,1,1), \\ (1,1,0,0,1,1,0,0,0,0), & (0,0,1,0,0,1,0,1,1,1), & (1,0,1,0,1,0,1,0,0,0), \\ (0,1,1,0,0,1,1,0,0,0), & (1,1,1,0,1,0,0,1,1,1), & (0,0,0,1,0,1,1,0,1,1), \\ (1,0,0,1,1,0,0,1,0,0), & (0,1,0,1,0,1,0,1,0,0), & (1,1,0,1,1,0,1,0,1,1), \\ (0,0,1,1,0,0,1,1,0,0), & (1,0,1,1,1,1,0,0,1,1), & (0,1,1,1,0,0,0,0,1,1), \\ \multicolumn{3}{c}{\mathbf{(1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 0,\ 0)}} \end{array} \right\}$$

In the following table we present the computational results for a binary Golay code and a binary BCH code.

	$[23,12]$ Golay code	$[21,12]$ BCH code
Codewords (2^k)	4096	4096
Cosets (2^{n-k})	2048	512
Leader codewords ($\lvert\mathrm{L}(\mathscr{C})\rvert$)	253	478
$\lvert\mathrm{L}^1(\mathscr{C})\rvert$	253	449

Table 2.3: Number of codewords, number of cosets, number of leader codewords and the cardinality of $|\mathrm{L}^1(\mathscr{C})|$ of the $[23,12,7]$ binary Golay code and the $[21,12,5]$ binary BCH code computed by GBLA-LC.

Therefore, we show an example where the subsets $\mathrm{L}(\mathscr{C})$ and $\mathrm{L}^1(\mathscr{C})$ agree (this is not a surprise since the Golay code is a perfect code) and an example where the set $\mathrm{L}^1(\mathscr{C})$ is smaller than $\mathrm{L}(\mathscr{C})$. Also, note that both codes have the same number of codewords but the Golay code has four times the number of cosets of the BCH code. On the other hand, the number of leader codewords is less in the Golay code.

Lemma 2.42. *If $\mathscr{C}$ is a perfect code, then $|\mathrm{L}(\mathscr{C})| = |\mathrm{L}^1(\mathscr{C})|$.*

Proof. If $\mathscr{C}$ is a perfect code then every coset of $\mathscr{C}$ has a unique coset leader. That is, $\mathscr{N} = \mathrm{CL}(\mathscr{C})$. Recall that the only difference between the sets $\mathrm{L}^1(\mathscr{C})$ and $\mathrm{L}(\mathscr{C})$ is that the component $\mathbf{n}_2$ of any element $\mathbf{b} = \mathbf{n}_1 + \mathbf{n}_2 + \mathbf{e}_i$ from $\mathrm{L}^1(\mathscr{C})$ is required to belong to $\mathscr{N} \subseteq \mathrm{CL}(\mathscr{C})$. But in this case this difference doesn't exists. □

2.4 Gradient descent decoding

As it was explained in Section 1.1.3, the classical syndrome decoding basically stands as follows:

- First construct the *syndrome lookup table* (i.e. enumerate the cosets of $\mathscr{C}$ in $\mathbb{F}_2^n$, choose a coset leader for each coset and compute its syndrome).
- Then if $\mathbf{y}$ is the received word, determine from the table which coset leader $\mathbf{e}$ satisfies that $S(\mathbf{y}) = S(\mathbf{e})$.
- Finally decode $\mathbf{y}$ as $\mathbf{y} - \mathbf{e} \in \mathscr{C}$.

Unfortunately, for a fixed rate $\frac{k}{n}$ the precomputation of this method grows exponentially with the length of the code. The main advantage of gradient descent from other decoding methods is that this task is broken into smaller steps. Gradient Descent Decoding Algorithm (GDD) have gained renewed interest over the past 20 years thanks to the efficient implementation achieved with turbo codes and low-density parity check (LDPC) codes. Unfortunately, there is no theoretical reason for the success of these algorithms. This is why Calderbank [25] said that *"GDD has moved coding towards an experimental science"*. This leads to another question for which there is not a clear answer: *Which parameters of a code contribute to have an efficient gradient descent decoding method?*

In the literature there are two gradient descent decoding algorithms for binary codes proposed independently by Liebler [71] and Ashikhmin and Barg [2]. In Liebler's own words:

"The gradient-like decoding algorithm of Ashikhmin and Barg is similar to ours in some way but is actually quite different".

In this section we will show that both algorithms can be seen as two ways of understanding the reduction associated to the Gröbner representation of the code.

Leader Gradient Descent Decoding Algorithm: This method requires a gradient function $\gamma : \mathbb{F}_2^n/\mathscr{C} \longleftarrow \mathbb{Z}$ which is a strictly increasing function with respect to the Hamming weight. Liebler [71] presented the first construction of this type of function for any linear code on a binary symmetric channel (BSC). However, this construction has no complexity advantage over classical syndrome decoding for an arbitrary code.

Given a received word $\mathbf{y} \in \mathbb{F}_2^n$, let us briefly give the outline of the algorithm. First find a Hamming neighbor $\mathbf{y}'$ of $\mathbf{y}$ (i.e. $\mathbf{y}' \in \mathscr{X}(\mathbf{y})$, or equivalently, $\mathrm{d}_H(\mathbf{y}, \mathbf{y}') = 1$) such that

$$\mathrm{w}_H(\mathrm{CL}(\mathbf{y}')) < \mathrm{w}_H(\mathrm{CL}(\mathbf{y})).$$

Then replace $\mathbf{y}$ with $\mathbf{y}'$ and iterates until $\mathrm{w}_H(\mathrm{CL}(\mathbf{y})) = 0$.

Algorithm 10: Leader GDDA

Data: The received word $\mathbf{y} \in \mathbb{F}_2^n$.
Result: A codeword $\mathbf{c} \in \mathscr{C}$ that minimized the Hamming distance $\mathrm{d}_H(\mathbf{c}, \mathbf{y})$.

```
while w_H(CL(y)) ≠ 0 do
    Look for y' ∈ X(y) such that w_H(CL(y')) < w_H(CL(y));
    y ⟵ y'
endw
Return c = y.
```

Example 2.43. Let $\mathscr{C}$ be the code in Example 2.4. Suppose we received the word $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5$ with error vector $\mathbf{e} = \mathbf{e}_1$. We will apply Algorithm 10 to obtain the closest codeword $\mathbf{c}$ to $\mathbf{y}$.

- **Step 1:** There exists $\mathbf{y}' = \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5 \in \mathscr{X}(\mathbf{y})$ such that

$$0 = \mathrm{w}_H(\mathrm{CL}(\mathbf{y}')) < \mathrm{w}_H(\mathrm{CL}(\mathbf{y})) = \mathrm{w}_H(\mathbf{e}_1) = 1.$$

  Thus, we replace $\mathbf{y}$ by $\mathbf{y}'$. Since $\mathrm{w}_H(\mathrm{CL}(\mathbf{y})) = 0$ then the algorithm terminates and returns $\mathbf{c} = \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5$.

If the received vector is $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_5$ and the transmitted codeword was $\mathbf{c} = \mathbf{e}_2 + \mathbf{e}_5 + \mathbf{e}_6$, then $\mathbf{e} = \mathbf{e}_1 + \mathbf{e}_6$ is the error vector. Applying Algorithm 10 yields to the following lines:

- **Step 1:** We found $\mathbf{y}' = \mathbf{e}_2 + \mathbf{e}_5 \in \mathscr{X}(\mathbf{y})$ with the desired property. So we substituted $\mathbf{y}$ by $\mathbf{y}'$.
- **Step 2:** There exists $\mathbf{y}' = \mathbf{e}_2 + \mathbf{e}_5 + \mathbf{e}_6$ such that $\mathrm{w}_H(\mathrm{CL}(\mathbf{y}')) = 0$. Hence we replace $\mathbf{y}$ by $\mathbf{y}'$ and the algorithm terminates giving the right solution.

Test-set Gradient Descent Decoding Algorithm: Given a received word $\mathbf{y} \in \mathbb{F}_2^n$ and suppose that a test-set $\mathscr{T}_{\mathscr{C}}$ of codewords has been precomputed and stored in the memory. This algorithm recursively search an element $\mathbf{t} \in \mathscr{T}_{\mathscr{C}}$ such that

$$\mathrm{w}_H(\mathbf{y} - \mathbf{t}) < \mathrm{w}_H(\mathbf{y})$$

and replace $\mathbf{y}$ by $\mathbf{y}' = \mathbf{y} - \mathbf{t}$. Of course $\mathbf{y} - \mathbf{t}$ belongs to the same coset of $\mathbf{y}$, since $\mathbf{t} \in \mathscr{T}_{\mathscr{C}} \subset \mathscr{C}$. The algorithm terminates when we arrive to a coset leader corresponding to the coset $\mathbf{y} + \mathscr{C}$, that is when $\mathbf{y}' \in \mathrm{CL}(\mathbf{y})$. See [2] for further details and correctness of Algorithm 11.

Note that this algorithm requires a nontrivial preprocessing for the construction of a test-set of codewords. Moreover, the main difference from the previous algorithm is that it stays entirely in one coset of the code and systematically seeks out a coset

Algorithm 11: Test-set GDDA

Data: The received word $\mathbf{y} \in \mathbb{F}_2^n$ and a test-set $\mathscr{T}_{\mathscr{C}}$ for $\mathscr{C}$.
Result: A codeword $\mathbf{c} \in \mathscr{C}$ that minimized the Hamming distance $\mathrm{d}_H(\mathbf{c}, \mathbf{y})$.

```
c ⟵ 0;
while there exists t ∈ T_C such that w_H(y + t) < w_H(y) do
    c ⟵ c + t;
    y ⟵ y + t;
endw
Return c.
```

leader while Algorithm 10 changes between different cosets of $\mathbb{F}_2^n/\mathscr{C}$ until it arrives to the $\mathbf{0}$-coset, i.e. the code itself.

Note that in section 2.3 we already propose an algorithm to obtain a test-set (called leader codewords) that allows gradient descent decoding.

It is pointed in [3, Theorem 3.13] that setting $\mathscr{T}_{\mathscr{C}} = \mathscr{M}_{\mathscr{C}}$ in Algorithm 11, then the so-called *minimal vector algorithm* performs complete minimum distance decoding. Moreover, the set of all Zero-Neighbours of $\mathscr{C}$ is also a test-set, i.e. we can set $\mathscr{T}_{\mathscr{C}} = \mathscr{Z}(\mathscr{C})$. Then this version of Algorithm 11 is called *Zero-Neighbours decoding* which also performs complete minimum distance decoding (see for instance [3, Theorem 3.16]).

Example 2.44. Following with Example 2.43 we shall decode using Algorithm 11. Consider the test-set $\mathrm{L}(\mathscr{C})$ given in Example 2.30. Suppose we receive the vector $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5$. First, the word $\mathbf{c}$ is initialized to zero.

- **Step 1:** There exists $\mathbf{t} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_4 \in \mathrm{L}(\mathscr{C})$ such that $\mathrm{w}_H(\mathbf{y}+\mathbf{t}) < \mathrm{w}_H(\mathbf{t})$. Thus $\mathbf{t}$ is added to the vector $\mathbf{c}$ and $\mathbf{y}$ is replaced by $\mathbf{y} + \mathbf{t} = \mathbf{e}_3 + \mathbf{e}_5$.
- **Step 2:** We found $\mathbf{t} = \mathbf{e}_1 + \mathbf{e}_3 + \mathbf{e}_5$ in $\mathrm{L}(\mathscr{C})$ with the desired property. So $\mathbf{c}$ is replaced by $\mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5$ and $\mathbf{y}$ by $\mathbf{y} + \mathbf{t} = \mathbf{e}_1$.
- **Step 3:** There is no other codeword $\mathbf{c} \in \mathrm{L}(\mathscr{C})$ with the required property so the algorithm terminates and returns $\mathbf{c}$ which was the transmitted codeword.

If $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_5$ is the received word with error vector $\mathbf{e} = \mathbf{e}_1 + \mathbf{e}_6$. Then, we detect but cannot correct the errors since there are at least two choices:

1. Suppose that our first option is to choose $\mathbf{t} = \mathbf{e}_1 + \mathbf{e}_3 + \mathbf{e}_5$ in $\mathrm{L}(\mathscr{C})$ which verifies the required property. Then the algorithm returns $\mathbf{y} = \mathbf{y} + \mathbf{t} = \mathbf{e}_2 + \mathbf{e}_3$.
2. Otherwise, suppose that our first option is to choose $\mathbf{t} = \mathbf{e}_2 + \mathbf{e}_5 + \mathbf{e}_6$ from the test-set $\mathrm{L}(\mathscr{C})$ which also verifies the desired property. In this case the algorithm returns $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_6$.

Note that this last fact happens because we have exceeded the error correction capacity of the code.

2.4.1 An algebraic view to gradient descent decoding

Given a binary code $\mathscr{C}$ and its corresponding Gröbner representation $(\mathscr{N}, \phi)$ we can accomplish two types of reduction that are associated to Algorithm 10 and Algorithm 11. Therefore, both GDD algorithms obey to the same algebraic structure.

Reduction by ϕ: We shall define the reduction of an element $\mathbf{n} \in \mathscr{N}$ relative to $\mathbf{e}_i$ with $i \in \{1, \ldots, n\}$ as the element $\mathbf{n}' = \phi(\mathbf{n}, \mathbf{e}_i) \in \mathscr{N}$, denoted by $\mathbf{n} \longrightarrow_i \mathbf{n}'$.

If we write each element $\mathbf{y} \in \mathbb{F}_2^n$ as

$$\mathbf{y} = \sum_{j=1}^{s} \mathbf{e}_{i_j} \quad \text{where} \quad \operatorname{supp}(\mathbf{y}) = \{i_1, \ldots, i_s\} \subseteq \{1, \ldots, n\},$$

we can conclude that iterating a finite number of reduction yields to a representative of the coset $\mathbf{y} + \mathscr{C}$, that is, an element of the subset $\mathrm{CL}(\mathbf{y})$. By Remark 2.13, if a total degree ordering $\prec$ is chosen, Algorithm 7 returns a Gröbner representation $(\mathscr{N}, \phi)$ such that the representative given by $\mathscr{N}$ corresponds to the set of coset leaders of $\mathscr{C}$. We will consider that this is the case from now on.

These ideas give us another gradient descent decoding algorithm described below as Algorithm 12.

Algorithm 12: GDDA using the reduction by ϕ

Data: $(\mathscr{N}, \phi)$ a Gröbner representation for $\mathscr{C}$ w.r.t. a total degree ordering and the received word $\mathbf{y} \in \mathbb{F}_2^n$.
Result: A codeword $\mathbf{c} \in \mathscr{C}$ that minimized the Hamming distance $\mathrm{d}_H(\mathbf{c}, \mathbf{y})$.
$\mathbf{y} = \sum_{j=1}^{s} \mathbf{e}_{i_j}$ i.e. $\operatorname{supp}(\mathbf{y}) = \{i_1, \ldots, i_s\}$

Forward Step: // Compute $n \in \mathscr{N}$ corresponding to the coset $\mathbf{y} + \mathscr{C}$, i.e. $\mathbf{n} \in \mathrm{CL}(\mathbf{y})$
$\mathbf{n} \longleftarrow \mathbf{0}$
for $j \leftarrow 1$ **to** s
  $\mathbf{n} \longrightarrow_{i_j} \mathbf{n}'$ // i.e. $\mathbf{n}' = \phi(\mathbf{n}, \mathbf{e}_{i_j})$
  $\mathbf{n} \longleftarrow \mathbf{n}'$
endfor

Backward Step:
while $\mathbf{n} \neq \mathbf{0}$ **do**
  Find $i \in \{1, \ldots, n\}$ such that $\mathrm{w}_H(\mathbf{n}) > \mathrm{w}_H(\phi(\mathbf{n}, \mathbf{e}_i))$
  $\mathbf{y} \longleftarrow \mathbf{y} + \mathbf{e}_i$
  $\mathbf{n} \longleftarrow \phi(\mathbf{n}, \mathbf{e}_i)$
endw
Return $\mathbf{c} = \mathbf{y}$

Example 2.45. This example compares Algorithm 12 with Algorithm 10. Consider the same data as in Example 2.43 and take the pair $(\mathscr{N}, \phi)$ in Example 2.4. We initialize the vector $\mathbf{n}$ to zero.

If $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5$ is the received word with error vector $\mathbf{e} = \mathbf{e}_1$, then:

- The **Forward Step** computes $\mathbf{n} \longleftarrow_i \mathbf{n}'$ for i taking values on the support of the received vector $\mathbf{y}$ and replacing after each step $\mathbf{n}$ by $\mathbf{n}'$. The following table summarizes the result of this step:

i	n	$\mathbf{n}'$
1	$\mathbf{n} = \mathbf{0}$	$\phi(\mathbf{0}, \mathbf{e}_1) = \mathbf{e}_1$
2	$\mathbf{n} \longleftarrow \mathbf{e}_1$	$\phi(\mathbf{n}, \mathbf{e}_2) = \mathbf{e}_4$
3	$\mathbf{n} \longleftarrow \mathbf{e}_4$	$\phi(\mathbf{n}, \mathbf{e}_3) = \mathbf{e}_6$
4	$\mathbf{n} \longleftarrow \mathbf{e}_6$	$\phi(\mathbf{n}, \mathbf{e}_4) = \mathbf{e}_3$
5	$\mathbf{n} \longleftarrow \mathbf{e}_3$	$\phi(\mathbf{n}, \mathbf{e}_5) = \mathbf{e}_1$

  Thus, **Forward Step** computes the coset leader $\mathbf{n} = \mathbf{e}_1$ of the coset $\mathbf{y} + \mathscr{C}$.

- The **Backward Step** looks for $i \in \{1, \ldots, 6\}$ such that $\mathrm{w}_H(\mathbf{n}) > \mathrm{w}_H(\phi(\mathbf{n}, \mathbf{e}_i))$. Note that $\phi(\mathbf{n}, \mathbf{e}_1) = \mathbf{0}$, thus in one iteration the algorithm terminates and returns $\mathbf{c} = \mathbf{y} + \mathbf{e}_1$.

Now take $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_5$ as the received word with error vector $\mathbf{e} = \mathbf{e}_1 + \mathbf{e}_6$. In this case the **Forward Step** gives $\mathbf{n} = \mathbf{e}_1 + \mathbf{e}_6$. With the following example we will see that if the coset of the received word does not have a unique coset leader (i.e. the weight of its coset leader exceeds the error correcting capacity of the code) then the backward step could give different solutions.

1. Suppose that our first option is choosing $i = 1$ such that $\phi(\mathbf{n}, \mathbf{e}_i) = \mathbf{e}_6$ verifies the required property. Then the algorithm returns $\mathbf{y} = \mathbf{e}_2 + \mathbf{e}_5 + \mathbf{e}_6$.

2. On the other hand, assume that our first option is choosing $i = 2$ (note that $\phi(\mathbf{n}, \mathbf{e}_2) = \mathbf{e}_3$ which verifies that $\mathrm{w}_H(\mathbf{n}) > \mathrm{w}_H(\phi(\mathbf{n}, \mathbf{e}_2))$). In this case the algorithm returns $\mathbf{y} = \mathbf{e}_1 + \mathbf{e}_3 + \mathbf{e}_5$, i.e. the error vector is assumed to take the value $\mathbf{e} = \mathbf{e}_2 + \mathbf{e}_3$.

Note that Algorithm 12 is somehow redundant, since at the end of the forward step we end with a coset leader $\mathbf{n}$ of the coset $\mathbf{y} + \mathscr{C}$. Thus we can decode without performing the backward step. We have expressed the algorithm in this way to see the resemblance to Algorithm 10. We can modify the previous algorithm to make it more effective with the following definition.

Definition 2.46. Let $(\mathscr{N}, \phi)$ be a Gröbner representation of $\mathbb{F}_2^n/\mathscr{C}$. Take an ordering on N with $\mathbf{n}_1 = 0$, say $N = \{\mathbf{n}_i\}_{i=1,\ldots,2^{n-k}}$. We define $(\mathscr{N}^*, \phi^*)$ as the pair where:

- We replace each element of $\mathscr{N}$ with vectors of type (i, w_i) where w_i represents the weight of $\mathrm{CL}(\mathbf{n}_i)$, i.e.

$$\mathscr{N}^* = \left\{(i, w_i) \mid w_i = \mathrm{w}_H(\mathbf{n}_i) \text{ for } i = 1, \ldots, 2^{n-k}\right\}.$$

- $\phi^* : \mathscr{N}^* \times \{\mathbf{e}_i\}_{i=1}^n \longleftarrow \mathscr{N}^*$ is a function that maps each triple $(i, w_i, \mathbf{e}_j)$ to the element $(i_j, w_{i_j}) \in \mathscr{N}^*$ such that $\mathbf{n}_{i_j} = \phi(\mathbf{n}_i, \mathbf{e}_j)$ and $w_{i_j} = \mathrm{w}_H(\mathbf{n}_{i_j})$.

In other words, we keep track only on the ordering of the elements of $\mathscr{N}$ and the weight of its coset leaders. Thus, we reduce the memory necessary to execute Algorithm 12. Let $(\mathscr{N},\phi)$ be the Gröbner representation of $\mathscr{C}$ obtained from Algorithm 6. Then, $(\mathscr{N}^*,\phi^*)$ can be easily obtained from $(\mathscr{N},\phi)$, since in this case $\mathscr{N}$ coincides with the set of coset leaders of $\mathscr{C}$. Moreover, Algorithm 6 gives us an incremental construction of $\mathscr{N}^*$ ordered non-decreasingly on the second component i.e. if (i,w_i) and (j,w_j) are elements of $\mathscr{N}^*$ with $i<j$, then $w_i \leq w_j$. Therefore, we can decode using the pair $(\mathscr{N}^*,\phi^*)$.

Algorithm 13: GDDA using the reduction by $(\mathscr{N}^*,\phi^*)$

Data: The pair $(\mathscr{N}^*,\phi^*)$ associated to $\mathscr{C}$ and the received word $\mathbf{y}\in\mathbb{F}_2^n$.
Result: A codeword $\mathbf{c}\in\mathscr{C}$ that minimized the Hamming distance $\mathrm{d}_H(\mathbf{c},\mathbf{y})$.
$\mathbf{y}=\sum_{j=1}^{s}\mathbf{e}_{i_j}$ i.e. $\mathrm{supp}(\mathbf{y})=\{i_1,\ldots,i_s\}$

Forward Step: // Returns $i\in\{1,\ldots,2^{n-k}\}$ which corresponds to the position of the element $\mathrm{CL}(\mathbf{y})$ in $\mathscr{N}^*$
$(i,w_i)\longleftarrow(1,0)$
for $j\leftarrow 1$ **to** s
  $(i',w_{i'})\longleftarrow\phi*\left((i,w_i),e_{i_j}\right)$
  $(i,w_i)\longleftarrow(i',w_{i'})$
endfor

Backward Step:
while $\mathbf{i}\neq 1$ **do**
  Find $j\in\{1,\ldots,n\}$ such that $w_i>w_{i'}$
  // where $w_{i'}$ is the 2^{nd} component of $\phi^*((i,w_i),\mathbf{e}_j)$
  $\mathbf{y}\longleftarrow\mathbf{y}+\mathbf{e}_j$
  $(i,w_i)\longleftarrow\phi^*((i,w_i),e_j)$
endw

After performing the forward step we can decide whether the coset of the received word $\mathbf{y}+\mathscr{C}$ has a unique coset $\mathbf{e}$, i.e. $w_l\leq t=\left\lfloor\frac{d-1}{2}\right\rfloor$. If so, the backward step gives the unique solution $\mathbf{c}=\mathbf{y}-\mathbf{e}\in\mathscr{C}$. Otherwise, if $w_l>t$, then there are as many solutions given by the backward step as number of coset leaders has the l-th coset, i.e. $|\mathrm{CL}(\mathbf{y})|$.

Remark 2.47. Again, the backward step is exactly Algorithm 10. As pointed by Liebler [71], in each step of the backtracking procedure we change of coset untill we arrive to the $\mathbf{0}$-coset.

Border reduction: Associated to the Gröbner representation $(\mathscr{N},\phi)$ for the binary code $\mathscr{C}$ obtained by Algorithm 6, we can define the *Border of a code* as follows:

$$B(\mathscr{C})=\left\{\ (\mathbf{n}_1+\mathbf{e}_i,\mathbf{n}_2)\ \middle|\ \begin{array}{ll}\mathbf{n}_1+\mathbf{e}_i\neq\mathbf{n}_2, & \mathbf{n}_1,\mathbf{n}_2\in\mathscr{N},\\ S(\mathbf{n}_1+\mathbf{e}_i)=S(\mathbf{n}_2), & i\in\{1,\ldots,n\}\end{array}\right\}$$

By definition, both components of an element of the set $B(\mathscr{C})$ belong to the same coset, that is, their sum is a codeword of the code. We can also describe the Border as the following set:

$$B(\mathscr{C}) = \left\{ \left(\mathbf{n}+\mathbf{e}_i, \phi(\mathbf{n},\mathbf{e}_i)\right) \quad | \quad \begin{array}{c} \mathbf{n}+\mathbf{e}_i \neq \phi(\mathbf{n},\mathbf{e}_i), \\ \text{and} \end{array} \quad \begin{array}{c} \mathbf{n} \in \mathscr{N} \\ i \in \{1,\dots,n\} \end{array} \right\}$$

The Border of a code $B(\mathscr{C})$ is associated to the $\mathscr{O}$-Border basis of the ideal $I_2(\mathscr{C})$ where $\mathscr{O} = [\mathbf{X}] \setminus \mathrm{LT}_{\prec}\left(I_2(\mathscr{C})\right)$ (see Definition 1.44). By Remark 1.50, every Gröbner basis with respect to a term order can be extended to a Border basis but the opposite is not true in general.

Let $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2) \in B(\mathscr{C})$, we define the *head* and the *tail* of $\mathbf{b}$ as

$$\mathrm{head}(\mathbf{b}) = \mathbf{b}_1 \in \mathbb{F}_2^n \quad \text{and} \quad \mathrm{tail}(\mathbf{b}) = \mathbf{b}_2 \in \mathbf{F}_2^n.$$

As pointed above $\mathrm{head}(\mathbf{b}) + \mathrm{tail}(\mathbf{b})$ is a codeword of $\mathscr{C}$ for all $\mathbf{b} \in B(\mathscr{C})$.

Example 2.48. Let us consider the code $\mathscr{C}$ defined by Example 2.4. The Border of $\mathscr{C}$ is given by the following set:

$$B(\mathscr{C}) = \left\{ \begin{array}{l} (\mathbf{e}_1+\mathbf{e}_2, \mathbf{e}_4),\ (\mathbf{e}_1+\mathbf{e}_3, \mathbf{e}_5),\ (\mathbf{e}_1+\mathbf{e}_4, \mathbf{e}_2),\ (\mathbf{e}_1+\mathbf{e}_5, \mathbf{e}_3), \\ (\mathbf{e}_2+\mathbf{e}_3, \mathbf{e}_1+\mathbf{e}_6),\ (\mathbf{e}_2+\mathbf{e}_4, \mathbf{e}_1),\ (\mathbf{e}_2+\mathbf{e}_5, \mathbf{e}_6),\ (\mathbf{e}_2+\mathbf{e}_6, \mathbf{e}_5), \\ (\mathbf{e}_3+\mathbf{e}_4, \mathbf{e}_6),\ (\mathbf{e}_3+\mathbf{e}_5, \mathbf{e}_1),\ (\mathbf{e}_3+\mathbf{e}_6, \mathbf{e}_4), \\ (\mathbf{e}_4+\mathbf{e}_5, \mathbf{e}_1+\mathbf{e}_6),\ (\mathbf{e}_4+\mathbf{e}_6, \mathbf{e}_3), \\ (\mathbf{e}_5+\mathbf{e}_6, \mathbf{e}_2), \\ (\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_2, \mathbf{e}_3),\ (\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_3, \mathbf{e}_2),\ (\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_4, \mathbf{e}_5), \\ (\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_5, \mathbf{e}_4) \end{array} \right\}$$

The information in the Border is somehow redundant, we can reduce the number of codewords in it by defining the following structure.

Definition 2.49. Let $\prec$ be a term ordering. A subset $R(\mathscr{C}) \subseteq B(\mathscr{C})$ is called the *reduced Border of the code* $\mathscr{C}$ w.r.t. $\prec$ if it fulfills the following conditions:

- For each element in the Border $\mathbf{b} \in B(\mathscr{C})$ there exists an element $\mathbf{h}$ in $R(\mathscr{C})$ such that $\mathrm{supp}(\mathrm{head}(\mathbf{h})) \subseteq \mathrm{supp}(\mathrm{head}(\mathbf{b}))$.
- For every two different elements $\mathbf{h}_1$ and $\mathbf{h}_2$ in $R(\mathscr{C})$ neither $\mathrm{supp}(\mathrm{head}(\mathbf{h}_1)) \subseteq \mathrm{supp}\left(\mathrm{head}(\mathbf{h}_2)\right)$ nor $\mathrm{supp}(\mathrm{head}(\mathbf{h}_2)) \subseteq \mathrm{supp}\left(\mathrm{head}(\mathbf{h}_1)\right)$ is verified.

Example 2.50. Again if we consider the code $\mathscr{C}$ defined by Example 2.4. The reduced Border of $\mathscr{C}$ is given by the following set:

$$\mathscr{R}(\mathscr{C}) = B(\mathscr{C}) \setminus \left\{ \begin{array}{l} (\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_2, \mathbf{e}_3),\ (\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_3, \mathbf{e}_2), \\ (\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_4, \mathbf{e}_5),\ (\mathbf{e}_1+\mathbf{e}_6+\mathbf{e}_5, \mathbf{e}_4) \end{array} \right\}.$$

See Example 2.48 for a complete description of the set $B(\mathscr{C})$.

Proposition 2.51. *Let us consider the set of codewords in $\mathscr{C}$ given by*

$$M_{\mathrm{Red}_\prec}(\mathscr{C}) = \{\mathrm{head}(\mathbf{b}) + \mathrm{tail}(\mathbf{b}) \mid \mathbf{b} \in R(\mathscr{C})\}$$

Then $M_{\mathrm{Red}_\prec}(\mathscr{C})$ corresponds to a subset of codewords of minimal support of $\mathscr{C}$, $\mathscr{M}_{\mathscr{C}}$.

Proof. Let $\mathbf{c}$ be an element of the set $M_{\mathrm{Red}_\prec}(\mathscr{C})$. We can rewrite $\mathbf{c}$ as

$$\mathbf{c} = \mathrm{head}(\mathbf{b}) + \mathrm{tail}(\mathbf{b}) \in \mathscr{C} \text{ with } \mathbf{b} \in R(\mathscr{C}).$$

Let us assume that $\mathbf{c}$ is not an element of $\mathscr{M}_{\mathscr{C}}$, then there exists a different codeword $\mathbf{c}' \in \mathscr{C} \setminus \{\mathbf{0}\}$ such that $\mathrm{supp}(\mathbf{c}') \subset \mathrm{supp}(\mathbf{c})$. Let $\mathbf{c}_1 \in \mathbb{F}_2^n$ be a vector with the property that $\mathrm{supp}(\mathbf{c}_1) = \mathrm{supp}(\mathbf{c}') \cap \mathrm{supp}(\mathrm{head}(\mathbf{b}))$. Hence the vector $\mathbf{c}_2 = \mathbf{c} + \mathbf{c}_1$ satisfies the condition $\mathrm{supp}(\mathbf{c}_2) \subset \mathrm{supp}(\mathrm{tail}(\mathbf{b}))$. Let us define $\mathbf{m}$ as a vector of $\mathbb{F}_2^{2n}$ by imposing the conditions that $\mathrm{head}(\mathbf{m}) = \mathbf{c}_1$ and $\mathrm{tail}(\mathbf{m}) = \mathbf{c}_2$. We claim that $\mathbf{m} \in B(\mathscr{C})$. Indeed,

- $\mathrm{head}(\mathbf{m}) \neq \mathrm{tail}(\mathbf{m})$, otherwise we would have that $\mathbf{c}_1 = \mathbf{c}_2$ and thus $\mathbf{c}' = \mathbf{c}_1 + \mathbf{c}_2 = 0$.
- Both components of $\mathbf{m}$ lie in the same coset.

Note that we have actually found an element $\mathbf{m} \in B(\mathscr{C})$ such that $\mathrm{supp}(\mathrm{head}(\mathbf{m})) \subset \mathrm{supp}(\mathrm{head}(\mathbf{b}))$, which contradicts the definition of $R(\mathscr{C})$. □

Therefore, $M_{\mathrm{Red}_\prec}(\mathscr{C})$ is a minimal test-set that allows GDDA as stated in Algorithm 11.

Remark 2.52. In Chapter 3 we will prove that the above set can also be seen as a test-set for the modular integer program associated.

Example 2.53. Continuing with Example 2.4, the set $M_{\mathrm{Red}_\prec}(\mathscr{C})$ is given by $\mathrm{L}(\mathscr{C})$ which was defined in Example 2.30.

Remark 2.54. Previously, similar ideas of the ones in this Chapter have been used. The idea behind Loop transversal codes is near to a Gröbner representation of the quotient $\mathbb{K}[X]/I_2(\mathscr{C})$. *Loop transversal codes* (LP codes) were introduced by J.D.H. Smith in [102] with the aim of giving a new approach to linear codes. Instead of constructing a code and then study its correction capability, with this theory first we look to specify this parameter and then construct the code. This type of codes have been shown to be suitable for burst-error correction [30, 57].

Let us briefly describe a general loop transversal code construction. Given a subgroup $\mathscr{C}$ of a (not necessarily abelian) group $(V, +, \cdot)$. A transversal E to $\mathscr{C}$ is a subset of V such that V may be represented as the disjoint union $V = \bigcup_{\mathbf{e}\in E}(\mathscr{C} + \mathbf{e})$. Therefore each element $\mathbf{v} \in V$ can be expressed uniquely as $\mathbf{v} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in \mathscr{C}$ and $\mathbf{e} \in E$.

We construct a *syndrome function* $\varphi : V \longrightarrow V$ such that φ is linear and $\varphi_{|E}$ is injective, that is no two different elements in the set E should get the same syndrome assigned. Then the kernel of φ is the subgroup $\mathscr{C}$ that corrects E, i.e. the

translation to Coding theory may be that a received word $\mathbf{v} = \mathbf{c} + \mathbf{e}$ is decoded to a codeword $\mathbf{c}$ with error $\mathbf{e} = \varphi(\mathbf{v}) \in E$.

We define a binary operation on E by $\mathbf{t}_1 * \mathbf{t}_2 = \varphi\left(\mathbf{t}_1 + \mathbf{t}_2\right)$ whose domain is $E \times E$. We can generalize such operation by recursion, i.e.

$$\prod_{i=1}^{0} \mathbf{u}_i = \varphi(\mathbf{0}) \quad \text{and} \quad \prod_{i=1}^{r} \mathbf{u}_i = \left[\prod_{i=1}^{r-1} \mathbf{u}_i\right] * \mathbf{u}_r \text{ for } r > 0.$$

For any $\mathbf{t}_1, \mathbf{t}_2 \in E$ note that the equation $\mathbf{x} * \mathbf{t}_1 = \mathbf{t}_2$ has a unique solution $\mathbf{x}$. Moreover, if the equation $\mathbf{x} * \mathbf{t}_1 = \mathbf{t}_2$ has also a unique solution, then E is called a *loop transversal*. Note that if V is abelian then each transversal is a loop transversal.

In the particular case of linear codes, V is a finite dimensional vector space over $\mathbb{F}_q$. We define $\lambda \times \mathbf{t} = \varphi(\lambda \mathbf{t})$ for $\lambda \in \mathbb{F}_q$ and $\mathbf{t} \in E$. Induction extends the operation $\times$ to:

$$\prod_{i=1}^{r} (\lambda_i \times \mathbf{t}_i) = \varphi\left(\sum_{i=1}^{r} \lambda_i \mathbf{t}_i\right) \text{ for } \mathbf{t}_i \in E \text{ and } \lambda_i \in \mathbb{F}_q.$$

That is $(E, *, \times)$ is a vector space over $\mathbb{F}_q$ and its is reasonable to require E to contain a basis $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ for $V = \mathbb{F}_q^n$ where $\mathbf{e}_i$ denotes the canonical basis of $\mathbb{F}_q^n$.

Then the knowledge of the vector space $(T, *, \times)$ is sufficient to determine the code $\mathscr{C}$. Indeed,

$$\mathscr{C} = \{\mathbf{v} - \varphi(\mathbf{v}) \mid \mathbf{v} \in V\} = \left\{\sum_{i=1}^{k} \lambda_i \mathbf{e}_i - \prod_{i=1}^{k} (\lambda_i \times \mathbf{e}_i) \text{ with } \lambda_i \in \mathbb{F}_q\right\}$$

Here is another way of passing between the code $\mathscr{C}$ and the transversal E. Taking account of the fact that

$$\varphi(\mathbf{x} + \mathbf{y}) = \varphi(\mathbf{x}) * \varphi(\mathbf{y}) \quad \text{and} \quad \varphi(\lambda \mathbf{x}) = \lambda \times \varphi(\mathbf{x})$$

for $\mathbf{x}, \mathbf{y} \in V$ and $\lambda \in \mathbb{F}_q$, we define a linear transformation $(V, +, \cdot) \longrightarrow (E, *, \times)$ known as *parity map*. Note that parity check matrices of $\mathscr{C}$ can be given by matrices of φ with respect to appropriate bases. In other words, a set of coset representatives for $\mathscr{C}$ (e.g. a complete set of coset leaders) defines a loop transversal E for $\mathscr{C}$.

3

Modular codes and the set of codewords of minimal support

Contents

Throughout this chapter $\mathscr{C}$ will be a modular code defined over $\mathbb{Z}_q$ where $\mathbb{Z}_q$ denotes the ring of integers modulo q. A *modular code* $\mathscr{C}$ over $\mathbb{Z}_q$ of length n is an additive subgroup of $\left(\mathbb{Z}_q^n, +\right)$. The notion of elementary row operations on a matrix and its consequences are carried over $\mathbb{Z}_q$ with the understanding that only multiplication of a row by a unit in $\mathbb{Z}_q$ is allowed, as opposed to multiplication by a

non-zero element in the case of a field. Note that if q is prime, $\mathscr{C}$ is just a linear code in the usual sense. From now on we simple write an $[n,k]$-code for a modular code of length n and rank k.

See Section 1.1.2 for the definition of a generator matrix of a modular code. Recall that r modular independent codewords in a modular code of rank r do not necessarily form a basis. However, for the results presented along this chapter it would be enough to require a set of generators of the modular code, without insisting in the fact of its independence over $\mathbb{Z}_q$.

It has been widely known that complete decoding for binary linear codes can be regarded as a linear integer programming problem with binary arithmetic conditions. This approach which leads our study, aimed to give an algebraic description to the set of codewords of minimal support of a modular code. Note that for q prime our goal is equivalently to find the collection of all cycles of an $\mathbb{F}_q$-representable matroid. Tutte [111] proved that the set of codewords of minimal support of $\mathscr{C}$, when $\mathscr{C}$ is defined over a finite field $\mathbb{F}_q$, is precisely the set of cocircuits of its associated vector matroid (see also [90, Theorem 9.2.4]). The vocabulary necessary to translate the language of matroid theory into that of coding theory can be found, for example, in [64].

Conti and Traverso in 1991 [33] proposed an efficient algorithm which uses Gröbner Bases to solve integer programming with ordinary integer arithmetic conditions. This idea was later extended by Ikegami and Kaji [60] to solve modular integer problems. Therefore, we have a method for computing a test-set (called the Gröbner test-set) for a modular code which only works in the binary case. Furthermore, the complete decoding scheme allowed by the Gröbner test-set is equivalent to the gradient descent decoding given by Ashikhmin and Barg [2], which has been proven in [74] to be equivalent to Liebler's approach [71]. See Chapter 2 for a further account on this topic.

Thus, it is natural to consider for those problems the Graver basis associated to them that provides a universal test-set, which turns out to be a set containing the set of codewords of minimal support of codes defined over $\mathbb{Z}_q$.

We will like to emphasize that the modular integer programming approach presented in this chapter does not allow us to perform complete decoding for $q > 2$, but the description of the Graver basis of the modular problem provides a description of the set of codewords of minimal support for any modular code.

The organization of this chapter is as follows. In Section 3.1 we start by briefly reviewing the algorithm of Conti and Traverso [33]. The main idea of this algorithm is to compute a Gröbner basis of the binomial ideal associated to the integral kernel of the coefficient matrix of the problem with respect to a term order induced by the corresponding cost vector, and so obtain a solution which minimizes the indicated linear function using the Gröbner basis reduction procedure. Then we discuss the extension of Conti-Traverso algorithm given by Ikegami and Kaji [60] to solve linear integer programming with modulo arithmetic conditions. If we fix an $m \times n$ coefficient matrix of the modular problem, the main disadvantage of Ikegami-Kaji's algorithm is that it involves $m \times n$ variables. Moreover they do not show any specific system for improving the complexity of the classical Buchberger algorithm for

computing Gröbner Bases. However, note that in our case the coefficient growth is not a problem since all the necessary information is encoded in the exponents of the binomials of our defined ideals. The solutions proposed in this section to improve the efficiency of the above algorithm using the philosophy presented by Di Biase - Urbanke in [38] looks to reduce the number of variables. We can also see this fact as a generalization of the ideas in the paper [11] to the non-binary modular case. In Subsection 3.1.1 we recall how in the binary case the maximum likelihood decoding can be regarded as a modular integer problem.

All these ideas will allow us to introduce in Section 3.2 the binomial ideal associated to a modular code $I_m(\mathscr{C})$ which we prove that it can be seen as the binomial ideal associated to a modular integer program. Finally, we describe the set of codewords of minimal support of modular codes using the Graver basis associated to the defined ideal. This Graver basis is again the $\mathbb{Z}_q$-kernel associated to a Lawrence lifted matrix, therefore we can apply the techniques previously used by Sturmfels in [107].

In Section 3.3 we discuss an alternative for the computation of the Gröbner basis of $I_m(\mathscr{C})$ relative to a degree compatible ordering. For this section, we will use the "change of basis" or FGLM technique proposed in [11]. In that paper the algorithm is stated for the binary case, but the generalization to the modulo q is straightforward.

Finally, in Section 3.4 we try to reduce the complexity of the above algorithm taking advantage of the powerful decomposition theory for linear codes provided by the decomposition theory of representable matroids over finite fields. In [63, 65] Kashyap gives an overview of an example of such decomposition theory. In this way we identify the codes that can be written as "gluing" of codes of shorter length. The concept of "glue" was first used in the context of semigroups representation by A. Thomas [109] and Rosales [97]. Recently, some generalisations have been made of this concept in [49], article that has motivated our work. If this decomposition verifies certain properties, then computing the set of codewords of minimal support in each code appearing in the decomposition is equivalent to computing the set of codewords of minimal support for the original code. Moreover, these computations are independent of each other, thus they can be carried out in parallel for each component, thereby not only obtaining a reduction of the complexity of the algorithm but also decreasing the time needed to process it.

The results in this chapter are joint works with E. Martínez-Moro from University of Valladolid (Spain) and appeared in [74, 75].

3.1 Relationship to integer linear programming

Given an integral matrix $A \in \mathbb{Z}^{m\times n}$, which is known as the *matrix of coefficients*, the vector $\mathbf{b} \in \mathbb{Z}^m$ and the *cost vector* $\mathbf{w} \in \mathbb{R}^n$, the *integer linear programming (LP) problem*, denoted by $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ consists of three parts: a *linear function to be minimized*, a *problem constraint* and *non-negative variables*. Therefore, if we express the problem

in matrix form it becomes:

$$\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b}) = \begin{cases} \text{minimize} & \mathbf{w}\cdot\mathbf{u} \\ \text{subject to} & \begin{cases} A\mathbf{u}^t = \mathbf{b} \\ \mathbf{u} \in \mathbb{Z}^n_{\geq 0} \end{cases} \end{cases} \tag{3.1}$$

A solution $\mathbf{u} \in \mathbb{Z}^n_{\geq 0}$ which satisfies $A\mathbf{u}^t = \mathbf{b}$ is called *optimal* if $\mathbf{u}$ minimizes the inner product $\mathbf{w}\cdot\mathbf{u}$. Although an integer program differs from a linear program only in the requirement that solutions are integral instead of real, the general integer program is NP-complete while linear programs can be solved in polynomial time, see for instance [20, 91]. The first general algorithm to solve an integer program was *Gomory's cutting plane method*. Then, further methods around *branching and bounding* integer linear programs were designed. For other algorithms and further reading on both linear and integer programming see [100].

In [33] Conti and Traverso introduced a Gröbner basis based algorithm to solve the problem $\mathrm{IP}_{A,\mathbf{w}}$ as follows: for a given integer linear programming problem $\mathrm{IP}_{A,\mathbf{w}}$ stated as in Equation (3.1), the algorithm computes a reduced Gröbner basis $\mathscr{G}_{\succ_\mathbf{w}}$ of the *toric ideal*

$$I = \left\langle \{\mathbf{X}^{\mathbf{u}^+} - \mathbf{X}^{\mathbf{u}^-} \mid \mathbf{u} = \mathbf{u}^+ - \mathbf{u}^- \in \ker_{\mathbb{Z}}(A)\} \right\rangle$$

where $\mathbf{u}^+,\ \mathbf{u}^- \in \mathbb{Z}^n_{\geq 0}$ and have disjoint supports w.r.t the term order $\succ_\mathbf{w}$ induced by the cost vector $\mathbf{w} \in \mathbb{R}^n$ defined as:

$$\alpha \succ_\mathbf{w} \beta \Leftrightarrow \begin{cases} \text{either } \mathbf{w}\cdot\alpha \succ \mathbf{w}\cdot\beta \text{ or} \\ \mathbf{w}\cdot\alpha = \mathbf{w}\cdot\beta, \quad \alpha \succ \beta, \text{ for } \succ \text{ a fixed admissible ordering.} \end{cases}$$

Note that the reduced Gröbner basis of the binomial ideal I is given also by binomials (see for instance [107, Corollary 4.4], for a proof). Finally, for any non-optimal solution $\mathbf{u}$ of $IP_{A,\mathbf{w}}(b)$, we compute the normal form of the monomial $\mathbf{X}^\mathbf{u}$ w.r.t. $\mathscr{G}_{\succ_\mathbf{w}}$ namely $\mathrm{Red}_{\succ_\mathbf{w}}(\mathbf{X}^\mathbf{u}, \mathscr{G}_{\succ_\mathbf{w}}) = \mathbf{X}^{\mathbf{u}'}$, and then $\mathbf{u}'$ is the optimal solution. The interested reader can refer to [107] for an account on Gröbner basis material related to integer linear programming.

Algorithm 14: Conti-Traverso Algorithm

Data: The matrix $A \in \mathbb{Z}^{m\times n}$ and the vectors $\mathbf{b} \in \mathbb{Z}^m$, $\mathbf{w} \in \mathbb{R}^n$.
Result: An optimal solution of $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$

1 Define the ideal I related with the constraint equations and a monomial order $\succ_\mathbf{w}$ induced by the cost vector $\mathbf{w}$;
2 Compute the reduced Gröbner basis $\mathscr{G}_{\succ_\mathbf{w}}$ of I w.r.t. $\succ_\mathbf{w}$;
3 Compute $\mathrm{Red}_{\succ_\mathbf{w}}(\mathbf{X}^\mathbf{b}, \mathscr{G}_{\succ_\mathbf{w}}) = \mathbf{X}^\mathbf{u}$;
4 Return $\mathbf{u} \in \mathbb{Z}^n$;

Definition 3.1. We denote by $\mathrm{IP}_{A,\mathbf{w}}$ the family of problems of type $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ where the coefficient matrix $A \in \mathbb{Z}^{m\times n}$ and the cost vector $\mathbf{w} \in \mathbb{R}^n$ are fixed whereas the

vector $\mathbf{b} \in \mathbb{Z}^m$ runs over all possible integer vectors of size m. A *test-set* for the family of problems $\mathrm{IP}_{A,\mathbf{w}}$ is a subset $\mathscr{T}_{\succ_{\mathbf{w}}} \subseteq \ker_{\mathbb{Z}}(A)$ if, for each non-optimal solution $\mathbf{u}$ to a program $\mathrm{IP}_{A,\mathbf{w}}(b)$, there exists $\mathbf{t} \in \mathscr{T}_{\succ_{\mathbf{w}}}$ such that $\mathbf{u} - \mathbf{t}$ is also a solution and $\mathbf{t} \succ_{\mathbf{w}} 0$.

A set $\mathscr{U}_A \subseteq \ker_{\mathbb{Z}}(A)$ is a *universal test-set* for IP_A if $\mathscr{U}_A$ contains a test-set for the family of integer programs $\mathrm{IP}_{A,\mathbf{w}}$ for every generic $\mathbf{w}$.

The exponents of the binomials involved in the reduced Gröbner basis $\mathscr{G}_{\succ_{\mathbf{w}}}$ induce a (uniquely defined) test-set for $\mathrm{IP}_{A,\mathbf{w}}$ called the *Gröbner test-set*. The existence of a finite test-set $\mathscr{T}_{\succ_{\mathbf{w}}}$ gives a trivial gradient descent method for finding the optimal solution of the problem $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$. Indeed, starting at a solution $\mathbf{u}$ of $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ we can successively move to an improved solution of the program by subtracting an appropriate element of $\mathscr{T}_{\succ_{\mathbf{w}}}$, then the solution $\mathbf{u}'$ is optimal if and only if there does not exists $\mathbf{t} \in \mathscr{T}_{\succ_{\mathbf{w}}}$ such that $\mathbf{u}' - \mathbf{t}$ is feasible for $\mathrm{IP}_{A,\mathbf{w}}(b)$. Test-sets for integer programming were introduced by Graver [51] and provide a way of telling if a feasible solution is optimal or not by checking for each element in the test-set whether adding it to the solution yields an improved value of the objective function. The same holds when the test-set is replaced by the Gröbner basis and addition is replaced by the reduction induced by the basis.

Ikegami and Kaji in [60] adapted the ideas of the Conti-Traverso algorithm to solve the modular integer programming problem.

Definition 3.2. Consider the matrix $A \in \mathbb{Z}_q^{m\times n}$ and the vectors $\mathbf{b} \in \mathbb{Z}_q^m$, $\mathbf{w} \in \mathbb{R}^n$, we define a *modular integer program*, denoted by $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ as the problem of finding a vector $\mathbf{u} \in \mathbb{Z}_q^n$ that minimizes the inner product $\mathbf{w} \cdot \blacktriangle\mathbf{u}$ subject to $A\mathbf{u}^t \equiv b \mod q$. If we express the problem in matrix form it becomes:

$$\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b}) = \begin{cases} \text{minimize} & \mathbf{w} \cdot \blacktriangle\mathbf{u} \\ \text{subject to} & \begin{cases} A\mathbf{u}^t \equiv \mathbf{b} \mod q \\ \mathbf{u} \in \mathbb{Z}_q^n \end{cases} \end{cases} \tag{3.2}$$

Note that the constrain conditions are modular ones but the weight minimizing condition is over the reals.

Let $\mathbf{X}$ denotes n variables $x_1, \ldots, x_n$ and $\mathbf{Y}$ denotes m variables $y_1, \ldots, y_m$. Consider the ring homomorphism

$$\Theta : \quad \mathbb{K}[\mathbf{X}] \quad \longrightarrow \quad \mathbb{K}[\mathbf{Y}]$$

defined by $\Theta(\mathbf{X}^{\mathbf{v}}) = \mathbf{Y}^{(\blacktriangle A)\mathbf{v}^T}$.

Let J_q be a binomial ideal defined by $J_q = \langle \{y_i^q - 1\}_{i=1}^m \rangle$ in the polynomial ring $\mathbb{K}[\mathbf{Y}]$. We have the following result.

Lemma 3.3. *$A\mathbf{u}^t \equiv \mathbf{b} \mod q$ if and only if $\Theta(\mathbf{X}^{\blacktriangle\mathbf{u}}) \equiv \mathbf{Y}^{\blacktriangle\mathbf{b}} \mod J_q$.*

Proof. See a proof of this result in [60, Lemma 1]. □

In [60] they showed that the ideal generated by binomials associated to the vectors belonging to the $\mathbb{Z}_q$-kernel of a matrix $A \in \mathbb{Z}_q^{m\times n}$, which is the kernel of the ring homomorphism Θ, is given by the elimination ideal $I = I_A \cap \mathbb{K}[\mathbf{X}]$ where

$$I_A = \left\langle \ \{\Theta(x_i) - x_i\}_{i=1}^{n} \quad \cup \quad \left\{y_j^q - 1\right\}_{j=1}^{m} \ \right\rangle \subseteq \mathbb{K}[\mathbf{X},\mathbf{Y}]. \tag{3.3}$$

Remark 3.4. In other words, we can see the ideal related to the $\mathbb{Z}_q$-kernel of a matrix $A \in \mathbb{Z}_q^{m\times n}$ as an elimination ideal of the $\mathbb{Z}$-kernel of the matrix

$$\begin{pmatrix} \blacktriangle A & q \cdot \mathrm{Id}_m \end{pmatrix} \in \mathbb{Z}^{m\times(m+n)},$$

where Id_m denotes the identity matrix of size m.

The following lemma gives the characterization of the polynomials that belong to the ideal I_A. Necessary condition was already stated in [60], we show it here again for completeness.

Lemma 3.5. *$f \in I_A \cap \mathbb{K}[\mathbf{X}]$ if and only if $f \in \mathbb{K}[\mathbf{X}]$ and $\Theta(f) \equiv 0 \mod J_q$.*

Proof. Let $f \in I_A$. Thus we have

$$f(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{n} \lambda_i \left(\Theta(x_i) - x_i\right) + \sum_{j=1}^{m} \beta_j (y_j^q - 1) \text{ where } \lambda_1,\ldots,\lambda_n,\beta_1,\ldots,\beta_m \in \mathbb{K}[\mathbf{X},\mathbf{Y}].$$

Then

$$\begin{aligned} \Theta(f) &= f(\Theta(x_1),\ldots,\Theta(x_n),y_1,\ldots,y_m) \\ &= \sum_{i=1}^{n} \Theta(\lambda_i)\left(\Theta(x_i) - \Theta(x_i)\right) + \sum_{j=1}^{m} \Theta(\beta_j)(y_j^q - 1) \\ &= \sum_{j=1}^{m} \Theta(\beta_j)(y_j^q - 1) \equiv 0 \mod J_q. \end{aligned}$$

To prove the converse, first note that given $\mathbf{u} \in \mathbb{Z}_q^n$ a monomial $\mathbf{X}^{\blacktriangle\mathbf{u}} = x_1^{u_1}\cdots x_n^{u_n}$ can be written, for some $B_1,\ldots,B_n \in \mathbb{K}[\mathbf{X},\mathbf{Y}]$, as:

$$\begin{aligned} x_1^{u_1}\cdots x_n^{u_n} &= \left(\Theta(x_1) + (x_1 - \Theta(x_1))\right)^{u_1}\cdots\left(\Theta(x_n) + (x_n - \Theta(x_n))\right)^{u_n} \\ &= \Theta(x_1)^{u_1}\cdots\Theta(x_n)^{u_n} + B_1(x_1 - \Theta(x_1)) + \ldots + B_n(x_n - \Theta(x_n)). \end{aligned}$$

Thus for all polynomial $f \in \mathbb{K}[\mathbf{X}]$ there exists $C_1,\ldots,C_n \in \mathbb{K}[\mathbf{X},\mathbf{Y}]$ such that

$$f(\mathbf{X}) = f(\Theta(x_1),\ldots,\Theta(x_n)) + \sum_{i=1}^{n} C_i(x_i - \Theta(x_i)) = \Theta(f) + \sum_{i=1}^{n} C_i(x_i - \Theta(x_i)).$$

From the initial assumption we have $\Theta(f) \equiv 0 \mod J_q$, i.e. $\Theta(f) \in J_q \subseteq I_A$. Therefore

$$f(x_1,\ldots,x_n) = \Theta(f) + \sum_{i=1}^{n} C_i(x_i - \Theta(x_i)) \in I_A \cap \mathbb{K}[\mathbf{X}].$$

□

Definition 3.6. A term order $\succ_{\mathbf{w}}$ on $\mathbb{K}[\mathbf{X},\mathbf{Y}]$ *is adapted to the problem* $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ if it is an elimination order for $\mathbb{K}[\mathbf{X}]$ and it is compatible with $\mathbf{w}$. In other words, for any $\mathbf{u},\ \mathbf{v}\in\mathbb{Z}_q^n$ such that $\Theta(\mathbf{X}^{\blacktriangle\mathbf{u}})\equiv\Theta(\mathbf{X}^{\blacktriangle\mathbf{v}})\mod J_q$ if $\mathbf{w}\cdot\mathbf{u}\succ\mathbf{w}\cdot\mathbf{v}$ then $\mathbf{X}^{\mathbf{u}}\succ_{\mathbf{w}}\mathbf{X}^{\mathbf{v}}$.

Consider $\mathscr{G}_{\succ_{\mathbf{w}}}$ a Gröbner basis of the ideal I_A w.r.t. an adapted monomial order $\succ_{\mathbf{w}}$ then the Conti-Traverso algorithm is extended to the modular case as follows:

Theorem 3.7. *Given the monomial* $\mathbf{X}^{\blacktriangle\mathbf{b}}$ *and let* $\mathrm{Red}_{\succ_{\mathbf{w}}}(\mathbf{X}^{\blacktriangle\mathbf{b}},\mathscr{G}_{\succ_{\mathbf{w}}})=\mathbf{X}^{\mathbf{u}'}$*, then* $\blacktriangledown\mathbf{u}'$ *will give an optimal solution of the problem* $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$.

Proof. See [60, Theorem 6]. □

Algorithm 15: Extended Conti-Traverso Algorithm

Data: The matrix $A\in\mathbb{Z}_q^{m\times n}$ and the vectors $\mathbf{b}\in\mathbb{Z}_q^m$, $\mathbf{w}\in\mathbb{R}^n$.
Result: An optimal solution of $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$

1 Define the ideal I_A related with the constraint equations and a term order $\succ_{\mathbf{w}}$ on $\mathbb{K}[\mathbf{X},\mathbf{Y}]$ which is adapted to the problem $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$;
2 Compute the reduced Gröbner basis $\mathscr{G}_{\succ_{\mathbf{w}}}$ of I_A w.r.t. $\succ_{\mathbf{w}}$;
3 Compute $\mathrm{Red}_{\succ_{\mathbf{w}}}(\mathbf{X}^{\mathbf{b}},\mathscr{G}_{\succ_{\mathbf{w}}})=\mathbf{X}^{\mathbf{u}}$;
4 Return $\mathbf{u}\in\mathbb{Z}^n$;

Note that in the previous description one requires $m\times n$ variables to describe the elimination ideal $I_A\cap\mathbb{K}[\mathbf{X}]$ which is in the ambient space $\mathbb{K}[\mathbf{X}]$ involving only n variables. Now we present a natural description of the ideal associated to a modular integer program so that we do not require the use of the extra variables in $\mathbf{Y}$. Thus we will restrict the whole computation inside $\mathbb{K}[\mathbf{X}]$ and therefore this will allow us to use the technique described in Section 3.3 for computing the Gröbner basis. Di Biase and Urbanke [38] developed some ideas concerning this problem for the integer $\mathbb{Z}$-kernel related to the problem $\mathrm{IP}_{A,\mathbf{w}}(\mathbf{b})$ in Equation (3.1). We will describe in this section how to compute the $\mathbb{Z}_q$-kernel of the problem $\mathrm{IP}_{A,\mathbf{w},q}(\mathbf{b})$ in Equation (3.2) considering only computations on the polynomial ring $\mathbb{K}[\mathbf{X}]$.

Let $B\in\mathbb{Z}_q^{k\times n}$ be a matrix and $\{\mathbf{w}_1,\dots,\mathbf{w}_k\}\subseteq\mathbb{Z}_q^n$ be a set of generators of its row space. We will define the following ideal

$$I(B)=\left\langle\{\mathbf{X}^{\mathbf{u}_1}-\mathbf{X}^{\mathbf{u}_2}\mid B\cdot\blacktriangledown(\mathbf{u}_1-\mathbf{u}_2)^T\equiv\mathbf{0}\mod q\}\right\rangle \tag{3.4}$$

i.e. all the binomials such that $\blacktriangledown(\mathbf{u}_1-\mathbf{u}_2)$ is in the modular kernel of B, i.e. $\ker_{\mathbb{Z}_q}(B)$. Let us consider the linear subspace $\left\{\mathbf{u}\in\mathbb{Z}_q^n\mid\mathbf{u}\cdot\mathbf{a}=0,\ \forall\,\mathbf{a}\text{ row of the matrix }B\right\}$ and $B^{\perp}$ be a matrix whose rows generate such linear subspace.

The following proposition is a generalization of [11, Proposition 1] for the non-binary case.

Proposition 3.8. *The following conditions are equivalent:*

1. $\mathbf{X}^{\mathbf{a}}-\mathbf{X}^{\mathbf{b}}\in I(B^{\perp})$.

2. *There exist* $\mathbf{t}_1,\ \mathbf{t}_2$ *in* $\mathbb{K}[\mathbf{X}]$ *and* $\lambda_1,\dots,\lambda_k \in \mathbb{Z}$ *such that*

$$\mathbf{X}^{\mathbf{a}+(q-1)\mathbf{b}}\mathbf{t}_1^q = \mathbf{t}_2^q \prod_{i=1}^{k} \mathbf{X}^{\lambda_i \blacktriangle \mathbf{w}_i}.$$

Proof. Consider the homomorphism

$$\phi : \quad \mathbb{K}[\mathbf{X}] \quad \longrightarrow \quad \mathbb{Z}_q^n$$

defined by $\phi(\mathbf{X}^{\mathbf{a}}) = \blacktriangledown\mathbf{a} \in \mathbb{Z}_q^n$. One can easily check that the following two properties hold:

$$\phi(\mathbf{X}^{\mathbf{a}}) = \phi(\mathbf{X}^{\mathbf{b}}) \Longleftrightarrow \exists \mathbf{t}_1, \mathbf{t}_2 \in \mathbb{K}[\mathbf{X}] \text{ such that } \mathbf{t}_1^q \mathbf{X}^{\mathbf{a}} = \mathbf{t}_2^q \mathbf{X}^{\mathbf{b}}, \tag{3.5}$$

$$\phi(\mathbf{X}^{\mathbf{a}}) - \phi(\mathbf{X}^{\mathbf{b}}) = \phi(\mathbf{X}^{\mathbf{a}}) + (q-1)\phi(\mathbf{X}^{\mathbf{b}}) = \phi(\mathbf{X}^{\mathbf{a}+(q-1)\mathbf{b}}). \tag{3.6}$$

Furthermore, $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \in I(B^{\perp})$ if and only if $\blacktriangledown(\mathbf{a}-\mathbf{b})$ is a linear combination of $\{\mathbf{w}_1,\dots,\mathbf{w}_k\}$, i.e.

$$\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \in I(B^{\perp}) \Longleftrightarrow \phi(\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}}) = \sum_{i=1}^{k} \lambda_{i,q}\mathbf{w}_i = \phi\left(\prod_{i=1}^{k} \mathbf{X}^{\lambda_i \blacktriangle \mathbf{w}_i}\right), \tag{3.7}$$

where $\lambda_{i,q} \in \mathbb{Z}_q$ and $\blacktriangle\lambda_{i,q} = \lambda_i$ for $i = 1,\dots,k$. We are now in a position to show the result. Assume that statement 1 holds, i.e. $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \in I(B^{\perp})$. Equations 3.7 and 3.6 imply that

$$\phi(\mathbf{X}^{\mathbf{a}+(q-1)\mathbf{b}}) = \phi(\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}}) = \phi\left(\prod_{i=1}^{k} \mathbf{X}^{\lambda_i \blacktriangle \mathbf{w}_i}\right).$$

Hence, from 3.5 we deduce statement 2.

The converse inclusion is proved by reading the above backwards.

□

Again, let $\{\mathbf{w}_1,\dots,\mathbf{w}_k\} \subseteq \mathbb{Z}_q^n$ be a set of generators of the row space of a matrix $B \in \mathbb{Z}_q^{k\times n}$.

Let us define the following binomial ideal:

$$\blacktriangle I = \left\langle \ \left\{\mathbf{X}^{\blacktriangle \mathbf{w}_i} - 1\right\}_{i=1,\dots,k} \quad \cup \quad \left\{x_i^q - 1\right\}_{i=1,\dots,n} \ \right\rangle. \tag{3.8}$$

Theorem 3.9. $\blacktriangle I = I(B^{\perp}) = I_{B^{\perp}} \cap \mathbb{K}[\mathbf{X}]$.

Proof. It is clear that $\blacktriangle I \subseteq I(B^{\perp})$ since all binomials in the generating set of $\blacktriangle I$ belong to $I(B^{\perp})$.

To show the converse, it suffices to use Proposition 3.8 together with the observation that

$$\mathbf{z}_1 - 1,\ \mathbf{z}_2 - 1 \in \blacktriangle I \Longleftrightarrow \mathbf{z}_1\mathbf{z}_2 - 1 \in \blacktriangle I \tag{3.9}$$

since $\mathbf{z}_1\mathbf{z}_2 - 1 = (\mathbf{z}_1 - 1)\mathbf{z}_2 + (\mathbf{z}_2 - 1)$. For detail, consider any binomial $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}}$ in $I(B^{\perp})$. By Proposition 3.8, there exist $\mathbf{t}_1,\ \mathbf{t}_2 \in \mathbb{K}[\mathbf{X}]$ and $\lambda_1,\dots,\lambda_k \in \mathbb{Z}$ such that

$\mathbf{X}^{\mathbf{a}+(q-1)\mathbf{b}}\mathbf{t}_1^q = \mathbf{t}_2^q \prod_{i=1}^k \mathbf{X}^{\lambda_i \blacktriangle \mathbf{w}_i}$. According to the Equation 3.9, we have $\prod_{i=1}^k \mathbf{X}^{\blacktriangle \mathbf{w}_i} - 1 \in \blacktriangle I$. Consequently, repeated application of the above equation enables us to write

$$\prod_{j=1}^k \mathbf{X}^{\lambda_j \blacktriangle \mathbf{w}_j} - 1 = \underbrace{\left(\prod_{j=1}^k \mathbf{X}^{\blacktriangle \mathbf{w}_j} - 1\right)\left(\prod_{j|\lambda_j>1} \mathbf{X}^{(\lambda_j-1)\blacktriangle \mathbf{w}_j}\right)}_{B_1} + \left(\prod_{j|\lambda_j>1} \mathbf{X}^{(\lambda_j-1)\blacktriangle \mathbf{w}_j} - 1\right) =$$

$$= B_1 + \underbrace{\left(\prod_{j|\lambda_j>1} \mathbf{X}^{\blacktriangle \mathbf{w}_j} - 1\right)\left(\prod_{j|\lambda_j>2} \mathbf{X}^{(\lambda_j-2)\blacktriangle \mathbf{w}_j}\right)}_{B_2} + \left(\prod_{j|\lambda_j>2} \mathbf{X}^{(\lambda_j-2)\blacktriangle \mathbf{w}_j} - 1\right)$$

$$= \ldots = B_1 + B_2 + \ldots + \left(\prod_{j|\lambda_j>s-1} \mathbf{X}^{\blacktriangle \mathbf{w}_j} - 1\right) \in \blacktriangle I,$$

where $s = \max\left\{\lambda_j \mid j = 1, \ldots, k\right\}$. In the same manner we can see that $\mathbf{X}^{q\mathbf{b}} - 1 \in \blacktriangle I$. Now $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \in \blacktriangle I$ which is due to the fact that

$$\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} = \mathbf{X}^{\mathbf{b}}\left(\mathbf{X}^{\mathbf{a}+(q-1)\mathbf{b}} - 1\right) - \mathbf{X}^{\mathbf{a}}\left(\mathbf{X}^{q\mathbf{b}} - 1\right).$$

Our next goal is to determine the equality $\blacktriangle I = I_{B^\perp} \cap \mathbb{K}[\mathbf{X}]$.

Let a_{ij} denote the entry in the i-th row and j-th column of $B^\perp \in \mathbb{Z}_q^{m\times n}$ and $\mathbf{a}_j$ denote the j-th column of $B^\perp$ which is an m-vector. First notice that:

- $\Theta\left(x_i^q - 1\right) = \Theta\left(\mathbf{X}^{q\mathbf{e}_i} - 1\right) = \mathbf{Y}^{q\mathbf{v}} - 1$ with $\mathbf{v} = (a_{1j}, \ldots, a_{mj}) = \mathbf{a}_j^T \in \mathbb{Z}_q^m$. Similarly to the previous case we can show that $\mathbf{Y}^{q\mathbf{v}} - 1 \in J_q$.
- $\Theta\left(\mathbf{X}^{\blacktriangle \mathbf{w}_i} - 1\right) = 0$ since $BB^\perp = \mathbf{0}$.

Therefore, Lemma 3.5 leads to $\blacktriangle I \subseteq I_{B^\perp} \cap \mathbb{K}[\mathbf{X}]$.

On the other hand, let us consider any binomial $f = \mathbf{X}^{\mathbf{u}} - \mathbf{X}^{\mathbf{v}} \in I_{B^\perp} \cap \mathbb{K}[\mathbf{X}]$. By Lemma 3.5 we have that $\Theta(f) = \mathbf{Y}^{\blacktriangle B^\perp \mathbf{u}^T} - \mathbf{Y}^{\blacktriangle B^\perp \mathbf{v}^T} \equiv \mathbf{0} \mod J_q$. This gives that $f \in I(B^\perp) = \blacktriangle I$ when combined with Lemma 3.3. □

Remark 3.10. Note that the matrix $B^\perp$ plays the role of the non negative matrix that Di Biase and Urbanke look for in their paper, thus the previous theorem can be seen as a generalization of the setting in [38] for getting rid of the variables concerning $\mathbf{Y}$ in $I_{B^\perp}$.

3.1.1 Integer linear programming approach to decoding binary codes

Let $q = 2$ and H be the parity check matrix of a binary linear code. Then solving the modular program $\mathrm{IP}_{H,\mathbf{1},2}(\mathbf{b})$ where $\mathbf{1} = (1, 1, \ldots, 1)$ is equivalent to complete decoding $\mathbf{b}$.

This is the approach in [60] and we have just shown by Theorem 3.9 that (only in the binary case) it is equivalent to the approach in [11]. Therefore, in the binary

case, this decoding scheme is equivalent to the gradient descent decoding given by Ashikhmin and Barg [2] which has been proven to be equivalent to Liebler approach [71] (see Section 2.4.1 for a proof). Unfortunately, Hamming metric can not be stated as a linear programming objective for $q > 2$, i.e. $\min\{\mathbf{w}\cdot\mathbf{u}\} \neq \min\{\mathrm{w}_H(\mathbf{u})\}$ where w_H denotes the Hamming weight. In Chapter 4 we will see how this approach can be generalized to any linear code.

In Section 3.2.1 we will prove that for any value of q, the Graver basis associated to these problems provides us the set of codewords of minimal support for the corresponding modular code.

3.1.2 A note on Graver basis

Given an integer matrix $A \in \mathbb{Z}^{m\times n}$. The lattice ideal of A, denoted by $I(A)$, is spanned as a $\mathbb{K}$-vector space by the set of binomials:

$$I(A) = \left\langle \left\{ \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \mathbf{a},\mathbf{b} \in \mathbb{N}^n \text{ and } A\mathbf{a}^T = A\mathbf{b}^T \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}].$$

This construction is adapted from [107].

Every vector $\mathbf{u} \in \mathbb{Z}^n$ can be written uniquely as $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$ where $\mathbf{u}^+$ and $\mathbf{u}^-$ are non-negative and have disjoint support. Therefore the above ideal can be rewritten as

$$I(A) = \left\langle \left\{ \mathbf{X}^{\mathbf{u}^+} - \mathbf{X}^{\mathbf{u}^-} \mid \mathbf{u} \in \ker_{\mathbb{Z}}(A) \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}].$$

We define the *Universal Gröbner basis* of A, denoted by UGB_A, as the union of all reduced Gröbner basis $\mathscr{G}_{\succ}$ of the ideal $I(A)$ as $\succ$ runs over all the term ordering.

Definition 3.11. A binomial $\mathbf{X}^{\mathbf{u}^+} - \mathbf{X}^{\mathbf{u}^-}$ in $I(A)$ is *primitive* if there exists no other binomial $\mathbf{X}^{\mathbf{v}^+} - \mathbf{X}^{\mathbf{v}^-}$ in $I(A)$ such that $\mathbf{X}^{\mathbf{v}^+}$ divides $\mathbf{X}^{\mathbf{u}^+}$ and $\mathbf{X}^{\mathbf{v}^-}$ divides $\mathbf{X}^{\mathbf{u}^-}$.

In other words, an integral vector $\mathbf{u} \in \ker_{\mathbb{Z}}(A)$ is said to be *primitive* if the greatest common divisor of the components of the corresponding binomials (+'ve and −'ve) is one.

We call the set of primitive binomials the *Graver basis* of A and denoted by Gr_A. A circuit of A is a non-zero primitive vector $\mathbf{u} \in \ker_{\mathbb{Z}}(A)$ such that its support $\mathrm{supp}(\mathbf{u})$ is minimal with respect to inclusion. Or equivalently, a circuit is an irreducible binomial $\mathbf{X}^{\mathbf{u}^+} - \mathbf{X}^{\mathbf{u}^-}$ in $I(A)$ which has minimal support. The set of circuits in $I(A)$ is denoted by $\mathscr{C}_A$.

Proposition 3.12. *For every matrix $A \in \mathbb{Z}^{m\times n}$ we have $\mathscr{C}_A \subseteq \mathrm{UGB}_A \subseteq \mathrm{Gr}_A$.*

Proof. See for instance [107, Proposition 4.11] □

Definition 3.13. Let $\mathbf{u},\ \mathbf{v} \in \mathbb{Z}^n$. We say that $\mathbf{u}$ is conformal to $\mathbf{v}$ and denoted by $\mathbf{u} \sqsubset \mathbf{v}$, if $|\mathbf{u}_i| \leq |\mathbf{v}_i|$ and $\mathbf{u}_i \cdot \mathbf{v}_i \geq 0$ for all $i = 1,\ldots,n$.

That is, $\mathbf{u}$ and $\mathbf{v}$ lie in the same orthant of $\mathbb{R}^n$ and each component of $\mathbf{u}$ is bounded by the corresponding component of $\mathbf{v}$ in absolute value.

Note that the binomial $\mathbf{X}^{\mathbf{u}^+} - \mathbf{X}^{\mathbf{u}^-}$ in I_A is primitive if the vector $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^- \in \mathbb{Z}^n$ is minimal with respect to the order $\sqsubset$. Thus, the Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A.

Remark 3.14. The *Graver basis* Gr_A and the *universal Gröbner basis* UGB_A of A are universal test-set for the family of integer programs $\mathrm{IP}_{A,\mathbf{w}}$ for every generic cost vector $\mathbf{w}$.

Definition 3.15. The Lawrence lifting of $A \in \mathbb{Z}^{m\times n}$ is defined as the enlarged matrix

$$\Lambda(A) = \begin{pmatrix} A & 0_{m\times n} \\ \mathrm{Id}_n & \mathrm{Id}_n \end{pmatrix} \in \mathbb{Z}^{(m+n)\times 2n}$$

where $\mathrm{Id}_n \in \mathbb{Z}^{n\times n}$ is the n-identity matrix and $0_{m\times n} \in \mathbb{Z}^{m\times n}$ is the all zero matrix.

The matrices A and $\Lambda(A)$ have isomorphic kernels. Indeed,

$$\ker_{\mathbb{Z}}(\Lambda(A)) = \left\{(\mathbf{u}, -\mathbf{u}) \mid \mathbf{u} \in \ker_{\mathbb{Z}}(A)\right\}.$$

The toric ideal $I(\Lambda(A))$ is an homogeneous prime ideal defined as

$$I(\Lambda(A)) = \left\langle \left\{ \mathbf{X}^{\mathbf{u}^+}\mathbf{Z}^{\mathbf{u}^-} - \mathbf{X}^{\mathbf{u}^-}\mathbf{Z}^{\mathbf{u}^+} \mid \mathbf{u} \in \ker_{\mathbb{Z}}(A) \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}, \mathbf{Z}]$$

where $\mathbf{Z}$ represents the variables $z_1, z_2, \ldots, z_n$.

Theorem 3.16. *For the Lawrence type matrix* $\Lambda(A)$ *the following sets coincide:*

1. *The Graver basis of* $\Lambda(A)$.
2. *The universal Gröbner basis of* $\Lambda(A)$.
3. *Any reduced Gröbner basis of* $\Lambda(A)$.
4. *Any minimal generating set of* $\Lambda(A)$ *(up to scalar multiples)*.

Proof. See [107, Theorem 7.1]. □

Theorem 3.16 suggests the following algorithm for computing a Graver basis of an integer matrix A. Choose any term order on the polynomial ring $\mathbb{K}[\mathbf{X}, \mathbf{Z}]$ and compute a reduced Gröbner basis of $\Lambda(A)$. By Theorem 3.16, any reduced Gröbner basis of $\Lambda(A)$ is also a Graver basis of $\Lambda(A)$. Thus for each element in the Graver basis $\mathbf{X}^{\alpha}\mathbf{Z}^{\beta} - \mathbf{X}^{\beta}\mathbf{Z}^{\alpha}$, the element $\mathbf{X}^{\alpha} - \mathbf{X}^{\beta}$ belongs to the Graver basis of A.

Algorithm 16: Algorithm for computing the Graver basis of A

Data: An integer matrix $A \in \mathbb{Z}^{m\times n}$.
Result: The Graver basis of A, Gr_A.
1 Choose any term order on $\mathbb{K}[\mathbf{X}, \mathbf{Z}]$;
2 Defined the Lawrence lifting of the matrix $\Lambda(A)$;
3 Compute a reduced Gröbner basis of $I(\Lambda(A))$;
4 Substitute the variable $\mathbf{Z}$ by $\mathbf{1}$;

Now let us consider any modular matrix $A \in \mathbb{Z}_q^{m\times n}$, in a similar fashion we can define the Lawrence lifting for the modulo q of A as

$$\Lambda(A)_q = \begin{pmatrix} A & 0_{q,m\times n} \\ \mathrm{Id}_{q,n} & \mathrm{Id}_{q,n} \end{pmatrix} \in \mathbb{Z}_q^{(m+n)\times 2n}$$

where $\mathrm{Id}_{q,n} \in \mathbb{Z}_q^{n\times n}$ is the n-identity matrix and $0_{q,m\times n} \in \mathbb{Z}_q^{m\times n}$ is the all zero matrix over the ring $\mathbb{Z}_q$.

Remark 3.17. As pointed in Remark 3.4, we can see the ideal related to the $\mathbb{Z}_q$-kernel of $\Lambda(A)_q$ as an elimination ideal of the $\mathbb{Z}$-kernel of the matrix

$$\begin{pmatrix} \blacktriangle A & 0_{m\times n} & q\cdot \mathrm{Id}_m \\ \mathrm{Id}_n & \mathrm{Id}_n & 0_{n\times m} \end{pmatrix} \in \mathbb{Z}^{(m+n)\times(2n+m)}.$$

We have a similar result to Theorem 3.16 relating the $\mathbb{Z}_q$-kernel of A and the $\mathbb{Z}_q$-kernel of $\Lambda(A)_q$ (see [60, Theorem 8]).

3.2 The lattice ideal associated with a modular code

Given an $[n,k]$ code $\mathscr{C}$ we define the lattice ideal $I(\mathscr{C})$ associated with $\mathscr{C}$ as the ideal:

$$I(\mathscr{C}) = \left\langle \left\{ \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \blacktriangledown(\mathbf{a}-\mathbf{b}) \in \mathscr{C} \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}].$$

Given a generator matrix $G \in \mathbb{Z}_q^{k\times n}$ of $\mathscr{C}$ and let label its rows by $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\} \subseteq \mathbb{Z}_q^n$. We have proved, see Theorem 3.9, that the ideal generated by the set of binomials:

$$I_m(\mathscr{C}) = \left\langle \ \{\mathbf{X}^{\mathbf{w}_j} - 1\}_{j=1,\ldots,k} \quad \cup \quad \left\{x_i^q - 1\right\}_{i=1,\ldots,n} \ \right\rangle \subseteq \mathbb{K}[\mathbf{X}] \tag{3.10}$$

match the ideal $I(\mathscr{C})$.

Remark 3.18. If the vectors $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\} \subseteq \mathbb{Z}_q^n$ are regarded as a set of $\mathbb{Z}_q$-generators of the row space of a matrix $A \in \mathbb{Z}_q^{k\times n}$, then the above ideal defines also the *ideal associated to the family of modular problems* $\mathrm{IP}_{A,q}$ in Equation 3.2. In this particular case, the above ideal will be denoted by $I(A^\perp)$. Theorem 3.9 expresses the equivalence between the above ideals.

3.2.1 Minimal support codewords

The interest in the set of codewords of minimal support is due to its relationship with the complete decoding problem which is an NP-problem [6], even if preprocessing is allowed (see [23]). Moreover, its relationship with the so-called gradient-like decoding algorithms [3, 11, 59]. Furthermore, it has applications in cryptography; in particular, in secret sharing schemes based on linear codes where the codewords of minimal support describe the minimal access structure (see [84] for further details).

Lemma 3.19. *Two minimal support codewords in $\mathscr{C} \lhd \mathbb{Z}_q^n$ with the same support should be one scalar multiple of the other.*

Proof. Given any $\mathbf{m} = (m_1, \ldots, m_n) \in \mathscr{M}_{\mathscr{C}}$. Suppose that there is a zero divisor element m_i in $\mathbf{m}$. Then m_i is a non-zero element, i.e. $i \in \operatorname{supp}(\mathbf{m})$, and there exists $\alpha \in \mathbb{Z}_q \setminus \{0\}$ satisfying $\alpha m_i = 0$. Then $\alpha\mathbf{m} \in \mathscr{C}$ and $\operatorname{supp}(\alpha\mathbf{m}) \subset \operatorname{supp}(\mathbf{m})$, which contradicts the minimality of $\mathbf{m}$ unless $\alpha\mathbf{m} = 0$. Thus either all the non-zero entries of $\mathbf{m}$ are zero divisors or all of them are units in $\mathbb{Z}_q$. Moreover, an analysis similar to the one above shows that if all the non-zero entries of $\mathbf{m}$ are zero divisors they must be equal.

Now consider the case that all the non-zero entries of $\mathbf{m}$ are units in $\mathbb{Z}_q$ and suppose the lemma were false. Then we could find another codeword $\mathbf{m}'$ of minimal support of $\mathscr{C}$ such that $\operatorname{supp}(\mathbf{m}) = \operatorname{supp}(\mathbf{m}')$ but $\mathbf{m} \neq \lambda\mathbf{m}'$ for any $\lambda \in \mathbb{Z}_q$. Let us choose $\beta \in \mathbb{Z}_q \setminus \{0\}$ such that $m_i = \beta m_i'$ for at least one index $i \in \operatorname{supp}(\mathbf{m})$, then

$$\mathbf{m} - \beta\mathbf{m}' \in \mathscr{C} \setminus \{\mathbf{0}\} \quad \text{and} \quad \operatorname{supp}(\mathbf{m} - \beta\mathbf{m}') \subset \operatorname{supp}(\mathbf{m})$$

which contradicts the minimality of $\mathbf{m}$. □

Theorem 3.20. *Let $H \in \mathbb{Z}_q^{(n-k)\times n}$ be a parity check matrix for $\mathscr{C}$. The set of codewords of minimal support of the code $\mathscr{C}$ is a subset of the Graver basis of H.*

Proof. Let $\mathbf{m} \in \mathscr{M}_{\mathscr{C}}$ and consider $\mathbf{u}$ the minimal element w.r.t. the order $\sqsubset$ in the set of codewords of minimal support of the code $\mathscr{C}$ having the same support as $\mathbf{m}$, i.e.

$$\mathbf{u} = \min_{\sqsubset} \left\{ \blacktriangle\mathbf{m}' \mid \mathbf{m}' \in \mathscr{M}_{\mathscr{C}} \text{ and } \operatorname{supp}(\mathbf{m}') = \operatorname{supp}(\mathbf{m}) \right\}.$$

Suppose the Theorem were false. Then the binomial $\mathbf{X}^{\mathbf{u}^+} - \mathbf{X}^{\mathbf{u}^-}$ in $I(H)$ is not primitive. Thus there exists another binomial $\mathbf{X}^{\mathbf{v}^+} - \mathbf{X}^{\mathbf{v}^-}$ in $I(H)$ such that $\mathbf{X}^{\mathbf{v}^+}$ divides $\mathbf{X}^{\mathbf{u}^+}$ and $\mathbf{X}^{\mathbf{v}^-}$ divides $\mathbf{X}^{\mathbf{u}^-}$. Or equivalently there exists a non-zero codeword $\blacktriangledown\mathbf{v} \in \mathscr{C}$, such that $\mathbf{v} \sqsubset \mathbf{u}$ contradicting the fact that $\mathbf{m}$ has minimal support. □

The previous result gives us a procedure to compute the set of codewords of minimal support for codes defined on $\mathbb{Z}_q$. In particular for linear codes over $\mathbb{F}_p$ with p prime, but not for the case p^r since $\mathbb{F}_{p^r} \ncong \mathbb{Z}_{p^r}$. In the binary case, since the codewords of minimal support are also minimal codewords, we have $\mathscr{C}_H = \mathrm{UGB}_H = \mathrm{Gr}_H$. The cases in which this equality holds and the obstructions for this property in the non modular case are studied by Bogart, Jensen and Thomas [10].

Now the two following corollaries are straight forward.

Corollary 3.21. *The set of codewords of minimal support of $\mathscr{C}$ can be computed from the ideal*

$$\left\langle \{\mathbf{X}^{\blacktriangle\mathbf{w}_1}\mathbf{Z}^{\blacktriangle\mathbf{w}_1(q-1)} - 1, \ldots, \mathbf{X}^{\blacktriangle\mathbf{w}_k}\mathbf{Z}^{\blacktriangle\mathbf{w}_k(q-1)} - 1\} \cup \{x_i^q - 1\}_{i=1}^n \cup \{z_i^q - 1\}_{i=1}^n \right\rangle \subseteq \mathbb{K}[\mathbf{X}, \mathbf{Z}]$$

where $\mathbf{w}_i$ for $i = 1, \ldots, k$ are the rows of any generator matrix of $\mathscr{C}$.

Proof. By Theorem 3.20 we know that the exponents of the binomials involved in the Graver basis of a parity check matrix H of the code $\mathscr{C}$ give us the set of codewords of minimal support of $\mathscr{C}$. Moreover, as we have already seen, Theorem 3.16 suggests an algorithm to compute the Graver basis of the ideal $I(H)$ by just computing a Gröbner basis of the ideal associated to the Lawrence lifting modulo q of the matrix H, which by Theorem 3.9 is equal to the ideal proposed. □

Corollary 3.22. *Let $\mathscr{C}$ be an $[n,k]$ modular code defined over $\mathbb{Z}_q^n$ and $H \in \mathbb{Z}_q^{(n-k)\times n}$ be a parity check matrix for $\mathscr{C}$, then the set of codewords of minimal support of $\mathscr{C}$ is contained on the projection of the first n coordinates of the set $\{\mathbf{w} \mid \mathbf{w} = (\mathbf{v}, (q-1)\cdot\mathbf{v}) \in \mathscr{T}\}$ where $\mathscr{T}$ is a universal test-set of the modular linear integer program with coefficient matrix*

$$\Lambda(H)_q = \begin{pmatrix} H & 0_{(n-k)\times n} \\ \mathrm{Id}_n & \mathrm{Id}_n \end{pmatrix} \in \mathbb{Z}_q^{(2n-k)\times 2n},$$

where Id_n denotes the identity matrix of size n and $0_{m\times n}$ the zero-matrix of size $m \times n$, both matrices defined over $\mathbb{Z}_q$.

Proof. This result follows directly from Corollary 3.21 and the fact that the ideal related to the $\mathbb{Z}_q$-kernel of the Lawrence lifting for the modulo q of the matrix H, denoted by $\Lambda(H)_q$, can be seen as an elimination ideal of the $\mathbb{Z}$-kernel of the following matrix as pointed in 3.17.

$$\begin{pmatrix} \blacktriangle H & 0_{(n-k)\times n} & q\cdot \mathrm{Id}_{n-k} \\ \mathrm{Id}_n & \mathrm{Id}_n & 0_{n\times(n-k)} \end{pmatrix} \in \mathbb{Z}^{(2n-k)\times(3n-k)}.$$

□

Example 3.23. Consider $\mathscr{C}$ the $[7,4,3]$ Hamming code over $\mathbb{Z}_2$ with generator matrix

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{Z}_2^{4\times 7}.$$

We have 16 codewords of weights 0, 3, 4 or 7. All the 14 codewords of weight 3 and 4 are codewords of minimal support while the only non-minimal support codewords are **0** and **1**. By Remark 3.18, the ideal associated to $\mathscr{C}$ is defined as the following binomial ideal:

$$\begin{aligned} I(\mathscr{C}) &= \left\langle \begin{Bmatrix} \mathbf{X}^{(1,0,1,0,0,0,1)} - 1, & \mathbf{X}^{(1,0,0,0,1,1,0)} - 1, \\ \mathbf{X}^{(0,0,1,1,0,1,0)} - 1, & \mathbf{X}^{(0,1,0,0,0,1,1)} - 1 \end{Bmatrix} \cup \left\{x_i^2 - 1\right\}_{i=1,\ldots,7} \right\rangle \\ &= \left\langle \begin{Bmatrix} x_1x_3x_7 - 1, & x_1x_5x_6 - 1, \\ x_3x_4x_6 - 1, & x_2x_6x_7 - 1 \end{Bmatrix} \cup \left\{x_i^2 - 1\right\}_{i=1,\ldots,7} \right\rangle \subseteq \mathbb{K}[\mathbf{X}] \end{aligned}$$

If we compute a Gröbner basis of $I(\mathscr{C})$ w.r.t. a degree reverse lexicographic ordering we get the following Gröbner basis:

$$\left\{ \begin{array}{ccccc} x_3x_7 + x_1, & x_1x_7 + x_3, & x_5x_6 + x_1, & x_4x_6 + x_3, & x_3x_6 + x_4, \\ x_2x_6 + x_7, & x_1x_6 + x_5, & x_4x_5 + x_7, & x_3x_5 + x_2, & x_2x_5 + x_3, \\ x_1x_5 + x_6, & x_3x_4 + x_6, & x_2x_4 + x_1, & x_1x_4 + x_2, & x_2x_3 + x_5, \\ & x_1x_3 + x_7, & x_1x_2 + x_4 & & \end{array} \right\} \cup \{x_i^2 - 1\}_{i=1}^7,$$

which gives us the 7 minimal codewords of weight 3 of $\mathscr{C}$:

$$(0,1,0,0,0,1,1),\ (1,0,1,0,0,0,1),\ (1,1,0,1,0,0,0),\ (0,1,1,0,1,0,0),$$
$$(0,0,1,1,0,1,0),\ (1,0,0,0,1,1,0),\ (0,0,0,1,1,0,1).$$

The degree reverse lexicographic Gröbner basis of the Lawrence lifting used for computing the Graver basis has 155 elements representing the 7 words already in the Gröbner test-set plus 7 minimal codewords of weight 4. Thus

$$\mathscr{M}_{\mathscr{C}} = \left\{ \begin{array}{c} (0,1,0,1,1,1,0),\ (1,0,1,1,1,0,0),\ (1,1,1,0,0,1,0),\ (0,1,1,1,0,0,1), \\ (1,1,0,0,1,0,1),\ (1,0,0,1,0,1,1),\ (0,0,1,0,1,1,1) \end{array} \right\}.$$

Example 3.24. Let us consider the $[6,3,2]$-code $\mathscr{C}$ defined over $\mathbb{Z}_3$ with generator matrix

$$\begin{pmatrix} 1 & 0 & 0 & 2 & 2 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 2 & 1 \end{pmatrix} \in \mathbb{Z}_3^{3\times 6}.$$

We have 27 codewords. By Remark 3.18, the ideal associated to $\mathscr{C}$ is defined as the following binomial ideal:

$$I(\mathscr{C}) = \left\langle \left\{ \begin{array}{c} x_1x_4^2x_5^2 - 1, x_2x_4x_5 - 1, \\ x_3x_4x_5^2x_6 - 1 \end{array} \right\} \cup \left\{x_i^3 - 1\right\}_{i=1,\ldots,6} \right\rangle.$$

The Graver basis of this ideal w.r.t. the degree lexicographical ordering has 318 elements representing 22 codewords:

$$\left\{ \begin{array}{lll} (0,0,2,2,1,2),\ (0,0,1,1,2,1), & (0,1,0,1,1,0),\ (0,2,0,2,2,0), \\ (0,1,1,2,0,1),\ (0,2,2,1,0,2), & (0,2,1,0,1,1),\ (0,1,2,0,2,2) \\ (1,0,0,2,2,0),\ (2,0,0,1,1,0), & (1,0,1,0,1,1),\ (2,0,2,0,2,2), \\ (1,0,2,1,0,2),\ (2,0,1,2,0,1), & (1,1,0,0,0,0),\ (2,2,0,0,0,0), \\ (1,2,0,1,1,0),\ (2,1,0,2,2,0), & (1,2,1,2,0,1),\ (2,1,2,1,0,2), \\ (1,2,2,0,2,2),\ (2,1,1,0,1,1) & \end{array} \right\}$$

This set represents all the codewords of $\mathscr{C}$ except the 4 codewords of weight 6 of $\mathscr{C}$ which are not codewords of minimal support:

$$\left\{ \ (1,1,1,1,2,1),\ (2,2,2,2,1,2) \quad (1,1,2,2,1,2),\ (2,2,1,1,2,1) \ \right\}$$

However, note that the codeword $(1,2,0,1,1,0)$ is an element of the Graver basis of $I(\mathscr{C})$ but is not a codeword of minimal support of $\mathscr{C}$, since

$$\operatorname{supp}(1,1,0,0,0,0) \subset \operatorname{supp}(1,2,0,1,1,0) \quad \text{where } (1,1,0,0,0,0) \in \mathscr{C}.$$

Therefore, the inclusion of Theorem 3.20 may or may not be strict for the non-binary case.

Example 3.25. Consider $\mathscr{C}$ the $[5,3,1]$ modular code defined over $\mathbb{Z}_4$ with generator matrix

$$G = \begin{pmatrix} 2 & 1 & 0 & 1 & 1 \\ 1 & 2 & 3 & 1 & 0 \\ 2 & 3 & 0 & 0 & 3 \end{pmatrix} \in \mathbb{Z}_4^{3\times 5}.$$

By Remark 3.18, the ideal associated to $\mathscr{C}$ is defined as the following binomial ideal:

$$I(\mathscr{C}) = \left\langle \left\{ \begin{array}{c} x_1^2x_2x_4x_5 - 1, x_1x_2^2x_3^3x_4 - 1, \\ x_1^2x_2^3x_5^3 - 1 \end{array} \right\} \cup \left\{x_i^4 - 1\right\}_{i=1,\ldots,5} \right\rangle$$

This code has 64 codewords but only three of them are codewords of minimal support:

$$\mathscr{M}_{\mathscr{C}} = \left\{ (0,2,0,0,2), \quad (0,0,0,1,0), \quad (2,0,2,0,0) \right\}.$$

The Gröbner basis of $I(\mathscr{C})$ w.r.t. the degree lexicographical ordering has 10 elements representing among others the codewords $\left\{ (0,2,0,0,2), \quad (0,0,0,1,0) \right\}$.

However, the Graver basis of $I(\mathscr{C})$ has 100 elements representing among others, the 2 minimal support codewords already in the Gröbner test-set plus $(2,0,2,0,0)$. Therefore, the elements of the Graver basis represents a set of codewords that contains the set of codewords of minimal support.

3.3 Computation of the Gröbner basis

In order to compute the set of codewords of minimal support of any modular code $\mathscr{C}$, we must compute a reduced Gröbner basis of the ideal related to the $\mathbb{Z}_q$-kernel of the Lawrence type matrix $\Lambda(H)_q$ where H is a parity check-matrix for $\mathscr{C}$ (i.e. a Graver basis of H). Note that we know a set of generators of such ideal given by

$$I = \left\langle \left\{\mathbf{X}^{\blacktriangle \mathbf{w}_i}\mathbf{Z}^{\blacktriangle \mathbf{w}_i(q-1)} - 1\right\}_{i=1,\ldots,k} \cup \left\{x_i^q - 1\right\}_{i=1,\ldots,n} \cup \left\{z_i^q - 1\right\}_{i=1,\ldots,n} \right\rangle \subseteq \mathbb{K}[\mathbf{X},\mathbf{Z}],$$

where $\left\{\mathbf{w}_1,\ldots,\mathbf{w}_k\right\}$ label the rows of a generator matrix for $\mathscr{C}$.

Hence, we can use the FGLM-techniques presented in [11]. On that paper the algorithm was stated only for the binary case, but the generalization to the modulo q is straight forward. We add here an study of this generalization.

Throughout this section we require some theory of Gröbner Bases for submodules $M \subseteq \mathbb{K}[\mathbf{X}]^m$. We define a term $\mathbf{t}$ in $\mathbb{K}[\mathbf{X}]^m$ as an element of the form $\mathbf{t} = \mathbf{X}^{\mathbf{v}}\mathbf{e}_i$ where $\mathbf{e}_i$ denotes a standard basis of $\mathbb{K}^m$. A term ordering $\prec$ on $\mathbb{K}[\mathbf{X}]^m$ is a total well-ordering such that if $\mathbf{t}_1 \prec \mathbf{t}_2$ then $\mathbf{X}^{\mathbf{u}}\mathbf{t}_1 \prec \mathbf{X}^{\mathbf{u}}\mathbf{t}_2$ for every pair of terms $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{K}[\mathbf{X}]^m$ and every monomial $\mathbf{X}^{\mathbf{u}} \in \mathbb{K}[\mathbf{X}]$. Let $\prec$ be any monomial order on $\mathbb{K}[\mathbf{X}]$ the following term orderings are natural extensions of $\prec$ on $\mathbb{K}[\mathbf{X}]^m$:

- *Term-over-position* order (TOP order) first compares the monomials by $\prec$ and then the position within the vectors in $\mathbb{K}[\mathbf{X}]^m$. That is to say,

$$\mathbf{X}^{\alpha}\mathbf{e}_i \prec_{\text{TOP}} \mathbf{X}^{\beta}\mathbf{e}_j \iff \mathbf{X}^{\alpha} \prec \mathbf{X}^{\beta} \quad \text{or} \quad \mathbf{X}^{\alpha} \prec \mathbf{X}^{\beta} \text{ and } i < j \ .$$

- *Position-over-term* order (POT order) which gives priority to the position of the vector in $\mathbb{K}[\mathbf{X}]^m$. In other words,

$$\mathbf{X}^{\alpha}\mathbf{e}_i \prec_{\text{POT}} \mathbf{X}^{\beta}\mathbf{e}_j \iff i < j \quad \text{or} \quad i = j \text{ and } \mathbf{X}^{\alpha} \prec \mathbf{X}^{\beta} \ .$$

The main ideas are as follows:

1. Fix a generator matrix $G \in \mathbb{Z}_q^{k\times n}$ of the $\mathbb{Z}_q$-modular code $\mathscr{C}$ whose rows are labelled by $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\}$.

2. Consider the set of binomials $F = \{f_1,\ldots,f_k\}$ defined as

$$f_i = \mathbf{X}^{\blacktriangle \mathbf{w}_i}\mathbf{Z}^{\blacktriangle \mathbf{w}_i(q-1)} - 1 \in \mathbb{K}[\mathbf{X},\mathbf{Z}] \quad \text{for} \quad i = 1,\ldots,k.$$

Hence, the set $F \cup \left\{x_i^q - 1\right\}_{i=1,\ldots,n} \cup \left\{z_i^q - 1\right\}_{i=1,\ldots,n}$ generates the ideal I. Moreover, not only the following set

$$\mathscr{G}_1 = \begin{cases} \hat{f}_1 &= (f_1, 1, 0, \ldots, 0) \in \mathbb{K}[\mathbf{X},\mathbf{Z}]^{k+1} \\ &\vdots \\ \hat{f}_k &= (f_k, 0, \ldots, 0, 1) \in \mathbb{K}[\mathbf{X},\mathbf{Z}]^{k+1} \end{cases}$$

is a basis for the syzygy module M with generating set $\hat{F} = \{-1, f_1,\ldots,f_k\}$ in $\mathbb{K}[\mathbf{X},\mathbf{Z}]^{k+1}$, where the binomials

$$J_{\mathbf{X},\mathbf{Z}} = \left\{ \left\{x_i^q - 1\right\}_{i=1,\ldots,n} \cup \left\{z_i^q - 1\right\}_{i=1,\ldots,n} \right\}$$

are considered to be implicit rules for operations on the module M. But also $\mathscr{G}_1$ is a Gröbner basis of M relative to a POT ordering $\prec_{\mathbf{w},\mathrm{POT}}$ induced by an ordering $\prec$ in $\mathbb{K}[\mathbf{X},\mathbf{Z}]$ and the weight vector

$$\mathbf{w} = \left(1, \mathrm{LT}_\prec(f_1), \ldots, \mathrm{LT}_\prec(f_k)\right).$$

In other words, $\mathbf{X}^\alpha\mathbf{Z}^\gamma\mathbf{e}_i \prec_{\mathbf{w},\mathrm{POT}} \mathbf{X}^\beta\mathbf{Z}^\delta\mathbf{e}_j$ if and only if

$$(i<j) \quad \text{or} \quad \left(i = j \quad \text{and} \quad \begin{cases} \mathbf{X}^\alpha\mathbf{Z}^\gamma \prec \mathbf{X}^\beta\mathbf{Z}^\delta, \text{ if } i = 1 \\ \mathbf{X}^\alpha\mathbf{Z}^\gamma\mathrm{LT}_\prec(f_i) \prec \mathbf{X}^\beta\mathbf{Z}^\delta\mathrm{LT}_\prec(f_i), \text{ otherwise.} \end{cases} \right)$$

Observe that the leading term of $\hat{f}_i$ w.r.t. $\prec_{\mathbf{w},\mathrm{POT}}$ is $\mathbf{e}_{i+1}$ where $\mathbf{e}_j$ denotes the unit vector of length $k+1$ whith a one in the j-th position.

Note that each syzygy corresponds to a solution of the following equation:

$$-\beta_0 + \textstyle\sum_{i=1}^k \beta_i f_i = 0 \quad \text{with } \beta_i \in \mathbb{K}[\mathbf{X},\mathbf{Z}] \quad \text{for } i = 0,\ldots,k.$$

Hence the first component of any syzygy of the module M indicates an element of the ideal generated by F.

It is easy to check that $\mathscr{G}_1$ is indeed a basis for M since any element $\mathbf{g} \in M$ can be written as $\mathbf{g} = (g_0, g_1, \ldots, g_k)$ with $g_0 = \sum_{i=1}^k g_i f_i$. Or equivalently,

$$\mathbf{g} = g_1\hat{f}_1 + \ldots + g_k\hat{f}_k \mod \langle J_{\mathbf{X},\mathbf{Z}}\rangle \in \langle \mathscr{G}_1\rangle.$$

Moreover $\mathscr{G}_1$ is a Gröbner basis for M relative to the order $\prec_{\mathbf{w},\mathrm{POT}}$. Note that, $\mathrm{LT}_{\prec_{\mathbf{w},\mathrm{POT}}}(\hat{f}_i) = \mathbf{e}_{i+1}$. Thus, for any $\mathbf{f} = (\lambda_0, \lambda_1, \ldots, \lambda_k) \in M \setminus \{\mathbf{0}\}$ we have that

$$\mathrm{LT}_{\prec_{\mathbf{w},\mathrm{POT}}}(\mathbf{f}) = \lambda_{i_m}\mathbf{e}_{m+1} \quad \text{where} \quad \mathrm{supp}(\mathbf{f}) = \{i_0, i_1, \ldots, i_m\} \subseteq \{0, 1, \ldots, k\}$$

and $m \neq 0$ since

$$\left(\lambda_0 = \sum_{i=1}^{k} \lambda_i f_i \neq 0, 0, \ldots, 0\right) \notin M.$$

It is clear that the remainder of the division of an element $f \in \mathbb{K}[\mathbf{X}, \mathbf{Z}]$ by $\mathscr{G}_1$ or, equivalently, the normal form of f respect to $\mathscr{G}_1$, is zero except in the first component. In other words, the linear combination that Fitzpatrick's algorithm [45] look for take place in the first component.

3. Use the generalized FGLM algorithm [42] running through the terms of $\mathbb{K}[\mathbf{X}, \mathbf{Z}]^{k+1}$ using a TOP ordering, to obtain a Gröbner basis $\mathscr{G}_2$ relative to the new order. The first component of each element of $\mathscr{G}_2$ gives a corresponding element of the desired Gröbner basis of I. To prove the above claim, let $\mathscr{G}_2 = \{g_1, \ldots, g_s\}$ be a Gröbner basis for M relative to the order $\prec_{TOP}$ where

$$g_i = (g_{i0}, \ldots, g_{ik}) \in \mathbb{K}[\mathbf{X}, \mathbf{Z}]^{k+1} \quad \text{and} \quad g_{i0} = \textstyle\sum_{j=1}^{k} g_{ij} f_j \ .$$

Note that, for any $\mathbf{h} = (h_0, h_1, \ldots, h_k) \in M$, since $\mathscr{G}_2$ is a Gröbner basis for M, then

$$\mathbf{h} = \lambda_1 g_1 + \ldots + \lambda_s g_s \quad \text{with} \quad \lambda_1, \ldots, \lambda_s \in \mathbb{K}[\mathbf{X}, \mathbf{Z}].$$

Hence,

$$\mathrm{LT}_{\prec_{\mathrm{TOP}}}(\mathbf{h}) = \max_{\prec_{\mathrm{TOP}}} \{\mathrm{LT}_{\prec}(h_0), \ldots, \mathrm{LT}_{\prec}(h_k)\} = \mathrm{LT}_{\prec}(h_0) \in \left\langle \{\mathrm{LT}_{\prec}(g_{i0})\}_{i=1,\ldots,k} \right\rangle.$$

Three structures are used in the algorithm:

- The list `List` whose elements has the form $\mathbf{v} = (\mathbf{v}[1], \mathbf{v}[2])$ where $\mathbf{v}[2]$ represents the polynomial in $\mathbb{K}[\mathbf{X}, \mathbf{Z}]$ which can be uniquely written as

$$\mathbf{v}[2] = \lambda + \sum_{i=1}^{k} \lambda_i f_i \text{ with } \lambda, \lambda_1, \ldots, \lambda_k \in \mathbb{K}[\mathbf{X}, \mathbf{Z}].$$

That is, the vector $(\lambda, \lambda_1, \ldots, \lambda_k) \in \mathbb{K}[\mathbf{X}, \mathbf{Z}]^{k+1}$ corresponds to the coordinates of $\mathbf{v}[2]$ on the module M. While $\mathbf{v}[1]$ corresponds to the first component of such vector, i.e. following with the notation of the above lines: $\mathbf{v}[1] = \lambda$.

- The list G_T which turns out being a reduced Gröbner basis of I w.r.t. a degree compatible ordering $\prec_T$.

- The list $\mathscr{N}$ which correspond to the set of standard monomials of I w.r.t. $\prec_T$.

We also require the following subroutines:

- `InsertNexts`($\mathbf{w}$, `List`): inserts the products $\mathbf{wt}$ for $\mathbf{t} \in [\mathbf{X}, \mathbf{Z}]$ in `List` and removes duplicates. Take account that the binomials $J_{\mathbf{X},\mathbf{Z}}$ are considered as implicit in the computation. Then the elements of `List` are sorted by increasing order w.r.t. $\prec_T$ in the first component and in case of equality by comparing the second component.

- `NextTerm`(`List`): removes the first element from the list `List` and returns it.

- `Member`($\mathbf{v}$, $[\mathbf{v}_1, \ldots, \mathbf{v}_r]$): returns j if $\mathbf{v} = \mathbf{v}_j$ and `false`, otherwise.

Remark 3.26. The ideal $\langle J_{\mathbf{X},\mathbf{Z}} \rangle$ is generated by the binomials expressing all modular relations of $\mathbb{Z}_q$. The operation $\mathbf{X}^{\blacktriangle \mathbf{a}} \mathbf{Z}^{\blacktriangle \mathbf{b}} \mathbf{t}$ with $\mathbf{t} \in [\mathbf{X}, \mathbf{Z}]$ and $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_q^n$ modulo the ideal $\langle J_{\mathbf{X},\mathbf{Z}} \rangle$, is similar to compute $(\mathbf{a}, \mathbf{b}) + \mathbf{u} \mod q$ where $\mathbf{u}$ is a unit vector defined as

$$\mathbf{u} = \begin{cases} (\mathbf{e}_j, \mathbf{0}) \in \mathbb{Z}_q^{2n} & \text{, if } \mathbf{t} = x_j \text{ with } j = 1, \ldots, n \\ (\mathbf{0}, \mathbf{e}_j) \in \mathbb{Z}_q^{2n} & \text{, if } \mathbf{t} = z_j \text{ with } j = 1, \ldots, n \end{cases}$$

Note that $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ denote the standard basis of $\mathbb{Z}_q^n$.

See Theorem 4.40 for a complete proof for the correctness of Algorithm 17.

This algorithm for computing the reduced Gröbner basis of the ideal I is specially well suited in our setting since it has the following advantages:

1. All the steps can be carried out as Gaussian elimination steps.

2. We can encode all the information of the problem in the exponents, thus we can always take the base field as $\mathbb{K} = \mathbb{F}_2$.

3. The total degree of the binomials involved is bounded by $2n \times q$ since the degree of each variable is at most q. Just take into account that the binomials J belongs to our ideal I.

The last two items above (the growth of degrees and coefficients) are the most common drawbacks of the usual Gröbner basis computation but in this case they should not be considered. The complexity of the algorithm was stated in [11, 15] and it is $\mathcal{O}\left(N^2 q^{N-k}\right)$ where k is the dimension of the code and $N = 2n$ is the number of variables involved in the algorithm.

Moreover, a slight modification of the Algorithm 17 allow us to compute just the set of codewords of minimal support, see Algorithm 22.

On the next lines, we study a very simple code to see how Algorithm 17 works. We present below just the most important steps. To see a complete example of this algorithm we refer the reader to Example 4.43 where we show these techniques for a linear code. Note that, on that case, the difference is mainly the set of generators of the ideal associated to the code.

Algorithm 17: Adapted FGLM algorithm for computing a Graver basis of an $[n,k]$-modular code $\mathscr{C}$

Data: The rows $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\}\subseteq\mathbb{Z}_q^n$ of a generator matrix of an $[n,k]$ modular code $\mathscr{C}$ defined and a degree compatible ordering $<_T$ on $\mathbb{K}[\mathbf{X},\mathbf{Z}]$.
Result: A Graver basis of $\mathscr{C}$ w.r.t. $\prec_T$.

```
1  r ⟵ 0;
2  List ⟵ [(1,1), {(1, (▲w_i, ▲w_i(q−1)))}_{i=1,...,k}];
3  G_T ⟵ ∅;
4  𝒩 ⟵ ∅;
5  while List ≠ ∅ do
6      w ⟵ NextTerm(List);
7      if w[1] ∉ LT_{<_T}(G_T) then
8          j = Member(w[2], [v_1[2], ..., v_r[2]]);
9          if j ≠ false then
10             G_T ⟵ G_T ∪ {w[1] − v_j[1]};
11             for i = 1 to r
12                 if v_i[1] is a multiple of w[1] then
13                     Removes v_i[1] from 𝒩
14                 endif
15             endfor
16         else
17             r ⟵ r + 1;
18             v_r ⟵ w;
19             𝒩 ⟵ 𝒩 ∪ {v_r[1]};
20             List = InsertNexts(w, List);
21         endif
22     endif
23 endw
```

Example 3.27 (Toy example)**.** Let $\mathscr{C}$ be the $[3,2,1]$ modular code defined over $\mathbb{Z}_3$ with generator matrix:

$$G=\begin{pmatrix}1&0&1\\0&1&0\end{pmatrix}\in\mathbb{Z}_3^{2\times 3}.$$

We find that

$$I=\left\langle\ \left\{\begin{array}{c}x_1x_3z_1^2z_3^2-1,\\ x_2z_2^2-1\end{array}\right\}\ \bigcup\ \left\{x_i^3-1\right\}_{i=1,2,3}\ \bigcup\ \left\{z_i^3-1\right\}_{i=1,2,3}\ \right\rangle.$$

We use Algorithm 17 to compute a reduced Gröbner basis G_T of I w.r.t. the `degrevlex` order with $x_3>x_2>x_1>z_3>z_2>z_1$.

The algorithm is initialize with the following data:

	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1	1							1
1		1		1	2		2	$x_1x_3z_1^2z_3^2$
1			1			2		$x_2z_2^2$

We introduce the variables z_2 and x_2 obtaining the following tables:

Introduce z_2	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
z_2						1		z_2
z_2		1		1	2	1	2	$x_1x_3z_1^2z_2z_3^2$
z_2			1			0		x_2

Introduce x_2	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
x_2			1					x_2
x_2		1	1	1	2		2	$x_1x_2x_3z_1^2z_3^2$
x_2			2			2		$x_2^2z_2^2$

Hence $x_2 - z_2$ is the first element of the reduced Gröbner basis G_T.

Passing through all the terms of degree one of $[\mathbf{X}, \mathbf{Z}]$ on increasing ordering w.r.t. $\prec$, we introduce then the terms of degree 2. In particular we introduce the monomials z_1z_3 and x_1x_3.

Introduce z_1z_3	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
z_1z_3					1		1	z_1z_3
z_1z_3		1		1				x_1x_3
z_1z_3			1		1	2	1	$x_2z_1z_2^2z_3$

Introduce x_1x_3	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
x_1x_3		1		1				x_1x_3
x_1x_3		2		2	2		2	$x_1^2x_3^2z_1^2z_3^2$
x_1x_3		1	1	1		2		$x_1x_2x_3z_2^2$

Thus, $x_1x_3 - z_1z_3$ is the second basis element in G_T.

We continue on increasing ordering throughout the terms of degree 3 of $[\mathbf{X}, \mathbf{Z}]$. In the following lines, we show how the terms $x_1z_1^2$, $x_3z_1^2$, $x_1^2z_1$, $x_3^2z_1$, $x_1z_3^2$, $x_3z_3^2$, $x_1^2z_3$, $x_3^2z_3$ are inserted on the algorithm.

Introduce $x_1z_1^2$	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1		1			2			$x_1z_1^2$
1		2		1	1		2	$x_1^2x_3z_1z_3^2$
1		1	1		2	2		$x_1x_2z_1^2z_2^2$

Introduce $x_3z_1^2$	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1				1	2			$x_3z_1^2$
1		1		2	1		2	$x_1x_3^2z_1z_3^2$
1			1	1	2	2		$x_2x_3z_1^2z_2^2$

Introduce $x_1^2z_1$	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1		2			1			$x_1^2z_1$
1		0		1	0		2	$x_3z_3^2$
1		2	1		1	2		$x_1^2x_2z_1z_2^2$

Introduce $x_3^2z_1$	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1				2	1			$x_3^2z_1$
1		1		0	0		2	$x_1z_3^2$
1			1	2	1	2		$x_2x_3^2z_1z_2^2$

Introduce $x_1z_3^2$	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1		1			2			$x_1z_3^2$
1		2		1	2		1	$x_1^2x_3z_1^2z_3$
1		1	1			2	2	$x_1x_2z_2^2z_3^2$

Introduce $x_3z_3^2$	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1				1			2	$x_3z_3^2$
1		1		2	2		1	$x_1x_3^2z_1^2z_3$
1			1	1		2	2	$x_2x_3z_2^2z_3^2$

Introduce $x_1^2z_3$	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1		2					1	$x_1^2z_3$
1		0		1	2		0	$x_3z_1^2$
1		2	1			2	1	$x_1^2x_2z_2^2z_3$

Introduce $x_3^2z_3$	$\mathbb{Z}_3^{2n}$-Representation of the exponent of $\mathbf{v}[2]$							
$\mathbf{v}[1]$	1	x_1	x_2	x_3	z_1	z_2	z_3	$\mathbf{v}[2]$
1				2			1	$x_3^2z_3$
1		1		0	2		0	$x_1z_1^2$
1			1	2		2	1	$x_2x_3^2z_2^2z_3$

After reduction, it is easily seen that the binomials $x_1z_3^2 - x_3^2z_1$, $x_3z_3^2 - x_1^2z_1$, $x_1^2z_3 - x_3z_1^2$ and $x_1z_1^2 - x_3z_3$ are elements in G_T.

Moreover, the inclusion of the terms x_i^3 and z_i^3 with $i = 1, 2, 3$ gives the generators of the ideal

$$\langle J_{\mathbf{X},\mathbf{Z}} \rangle = \left\langle \ \left\{x_i^3 - 1\right\}_{i=1,\ldots,3} \ \bigcup \ \left\{z_i^3 - 1\right\}_{i=1,\ldots,3} \ \right\rangle.$$

This ends the algorithm.

3.4 Decomposition of modular codes

In order to obtain a Gröbner test-set for the binary case or the set of codewords of minimal support of a modular code $\mathscr{C}$, we must compute a reduced Gröbner basis of an ideal from which we know a generating set. Thus we can use the FGLM-based trick described in Section 3.3. Note that the complexity of this algorithm is shown in [11, 16] to be $\mathcal{O}\left(n^2q^{n-k}\right)$ where k is the number of rows of a generator matrix of the code and n is the number of variables involved in our ideal, i.e. the length of the code.

The main task of this section is to reduce the complexity of the previous algorithms by using the decomposition of a given code as a "gluing" of smaller ones. That is to say, our aim is to explicitly define a procedure that:

1. (Decomposition) Find a decomposition of an $[n, k]$-code $\mathscr{C}$ into the m-gluing of two (or more) smaller codes, denoted by $\left\{\mathscr{C}_\alpha\right\}_{\alpha\in A}$.

2. Compute $\mathscr{M}_{\mathscr{C}_\alpha}$, the set of codewords of minimal support of $\mathscr{C}_\alpha$ for each $\alpha \in A$.

3. (Gluing) Compute $\mathscr{M}_{\mathscr{C}}$ from $\left\{\mathscr{M}_{\mathscr{C}_\alpha}\right\}_{\alpha\in A}$.

In the binary case, a similar process can be defined to compute the Gröbner test-set for $\mathscr{C}$ via the Gröbner test-set of the codes appearing in the decomposition of $\mathscr{C}$ as the m-gluing of smaller codes. Furthermore, we would like to know under which hypothesis this procedure turns out to be effective, i.e. when this algorithm is faster than computing a reduced Gröbner basis of the original code $\mathscr{C}$. Note that parallel computing is implicitly well suited for step 2, since the computation can be carried out in parallel for each component of the gluing.

We start by describing the decomposition of a code as the m-gluing of several smaller codes where m is a positive integer. The concept of "glue" was first used by A. Thomas [109] and Rosales [97] in the context of semigroups presentations. Recently some generalizations have been made of this concept in [49].

The connection between linear codes and matroids will turn out to be fundamental for the development of the subsequent results. We will relate the m-gluing operation with the m-sum operation described by Kashyap in [63, 65], which for some particular cases turns out to be the same operation.

In the following three subsections we then study some individual cases. We start by studying the simplest case, that is to say when a code is a direct sum or the 0-gluing of two codes (see Subsection 3.4.1), then we study the 1-gluing operation in Subsection 3.4.2 and the 3-gluing of codes in detail in Section 3.4.3. Note that the 2-gluing operation does not exist according to the definition proposed in this section.

Finally, we generalize the previous results from which we draw conclusions about the effectiveness of using the decomposition in our goal of reducing the complexity of computing the set of codewords of minimal support of an arbitrary code over $\mathbb{Z}_q$ and we discuss some lines of future work.

For any vector $\mathbf{x} \in \mathbb{Z}_q^n$ and a subset $J = \{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ consisting of m ordered integers, we denote by $\mathbf{x}_J = \left(x_{i_1}, \ldots, x_{i_m}\right) \in \mathbb{F}_q^m$ the restriction of $\mathbf{x}$ to the coordinates indexed by J and by $\overline{J}$ the relative complement of J in $\{1, \ldots, n\}$.

We can *puncture* an $[n, k]$-code $\mathscr{C}$ by deleting columns from a generator matrix of $\mathscr{C}$ or equivalently by deleting the same set of coordinates in each codeword. Let us consider a subset J of $\{1, \ldots, n\}$ the *punctured code* $\mathscr{C}_J$ is the set of codewords of $\mathscr{C}$ restricted to the positions of $\overline{J}$, i.e.

$$\mathscr{C}_J = \left\{\mathbf{c}_{\overline{J}} \mid \mathbf{c} \in \mathscr{C}\right\}.$$

In the field case, if J consist of m elements then the punctured code $\mathscr{C}_J$ has parameters $[n - m, k', d']$ with $d - m \leq d' \leq d$ and $k - m \leq k' \leq k$ where d and d' are the minimal distances of $\mathscr{C}$ and $\mathscr{C}_J$ respectively. Therefore, in puncturing a code normally we keep its dimension fixed but we vary its length and redundancy.

We can *shorten* an $[n, k]$-code $\mathscr{C}$ by deleting columns from a parity check matrix of $\mathscr{C}$. The *shortened code* $\mathscr{C}_{.J}$ is obtained by puncturing at J the set of codewords

that have a zero in the J-locations i.e.

$$\mathscr{C}_{\cdot J} = \left\{ \mathbf{c}_{\overline{J}} \mid \mathbf{c} \in \mathscr{C} \text{ and } \mathbf{c}_J = 0 \right\}.$$

To compute a generator matrix of $\mathscr{C}_{\cdot\{j\}}$, we first find a generator matrix of the original code that has a unique row in which the j-column is nonzero. Then delete that row and the j-column leaving the sought matrix. This process can be generalized also to subsets with more than one element.

Again in the field case, if J consist of m elements then the shortened code $\mathscr{C}_{\cdot J}$ has parameters $[n-m, k', d']$ and minimal distance d' with $k - m \leq k' \leq k$ and $d \leq d'$.

Hence, via the operation of *puncturing* and the method of *shortening* we can obtain codes of shorter length from $\mathscr{C}$. Take notice of some properties of these operations such as the shortened code $\mathscr{C}_{\cdot J}$ is a subcode of the punctured code $\mathscr{C}_J$ (i.e. $\mathscr{C}_{\cdot J} \subseteq \mathscr{C}_J$), the following statement on the dimension: $\dim(\mathscr{C}_{\cdot \overline{J}}) + \dim(\mathscr{C}_J) = \dim(\mathscr{C})$ or the fact that shortening a code is dual to puncturing, to be precise,

$$\left(\mathscr{C}_J\right)^{\perp} = \left(\mathscr{C}^{\perp}\right)_{\cdot J} \quad \text{and} \quad \left(\mathscr{C}_{\cdot J}\right)^{\perp} = \left(\mathscr{C}^{\perp}\right)_J .$$

From now on, for any positive integer i, let $\mathscr{C}_i$ be an $[n_i, k_i]$-code over $\mathbb{Z}_q$ whose coordinates are indexed by the set I_i (note that the sets of indices do not need to be disjoint). We will use $c^{(i)}$ to denote the i-th coordinate of a codeword $\mathbf{c}$.

We define the *sum code* of two modular codes $\mathscr{C}_1$ and $\mathscr{C}_2$ as the code

$$\mathscr{C}_1 \oplus \mathscr{C}_2 = \left\{ \mathbf{c} = \left(\mathbf{c}_1 \parallel \mathbf{c}_2\right) \mid \mathbf{c}_1 \in \mathscr{C}_1 \text{ and } \mathbf{c}_2 \in \mathscr{C}_2 \right\}.$$

where $\mathbf{c} = \left(\mathbf{c}_1 \parallel \mathbf{c}_2\right) = \left(c_i \mid i \in I_1 \cup I_2\right)$ is defined as follows:

$$c_i = \begin{cases} c_1^{(i)} & \text{for } i \in I_1 \setminus I_2 \\ c_2^{(i)} & \text{for } i \in I_2 \setminus I_1 \\ c_1^{(i)} + c_2^{(i)} & \text{for } i \in I_1 \cap I_2 \end{cases}$$

for all $\mathbf{c}_1 = (c_1^{(i)} \mid i \in I_1) \in \mathscr{C}_1$ and $\mathbf{c}_2 = (c_2^{(i)} \mid i \in I_2) \in \mathscr{C}_2$.

Given two vectors $\mathbf{a} \in \mathbb{Z}_q^{m_1}$ and $\mathbf{b} \in \mathbb{Z}_q^{m_2}$ whose coordinates are indexed by the sets I_1 and I_2 respectively, we denote their concatenation by $(\mathbf{a} \mid \mathbf{b}) \in \mathbb{Z}_q^{m_1+m_2}$. Note that $(\mathbf{a} \parallel \mathbf{b}) = (\mathbf{a} \mid \mathbf{b})$ if and only if $I_1 \cap I_2 = \emptyset$.

Over finite fields, it is straightforward to prove that $\mathscr{C}_1 \oplus \mathscr{C}_2$ is a modular code of length $n_1 + n_2 - |I_1 \cap I_2|$ and dimension

$$\dim(\mathscr{C}_1 \oplus \mathscr{C}_2) = k_1 + k_2 - \dim(\mathscr{C}_{1.\overline{(I_1 \cap I_2)}} \cap \mathscr{C}_{2.\overline{(I_2 \cap I_1)}}). \tag{3.11}$$

To prove this, note that the kernel of the following homomorphism

$$\phi : \quad \left(\mathscr{C}_1 \mid \mathscr{C}_2\right) \quad \longrightarrow \quad \mathscr{C}_1 \oplus \mathscr{C}_2$$

defined by $\phi(\mathbf{c}_1, \mathbf{c}_2) = \mathbf{c}_1 \oplus \mathbf{c}_2$ is isomorphic to the code $\left(\mathscr{C}_{1.\overline{(I_1 \cap I_2)}} \cap \mathscr{C}_{2.\overline{I_1 \cap I_2}}\right)$. Indeed,

$$\begin{aligned} \ker\phi &= \left\{(\mathbf{x} \mid \mathbf{y}) \in \left(\mathscr{C}_1 \mid \mathscr{C}_2\right) \,:\, (\mathbf{x} \parallel \mathbf{y}) = \mathbf{0}\right\} \\ &= \left\{(\mathbf{x} \mid \mathbf{y}) \in \left(\mathscr{C}_1 \mid \mathscr{C}_2\right) \,:\, \mathbf{x}_{I_1 \setminus I_2} = \mathbf{y}_{I_2 \setminus I_1} = \mathbf{0} \quad \text{and} \quad \mathbf{x}_{I_1 \cap I_2} + \mathbf{y}_{I_1 \cap I_2} = \mathbf{0} \right\} \\ &= \left\{\mathbf{z} \in \mathbb{Z}_q^{n_1+n_2} \,:\, \mathbf{z} \in \left(\mathscr{C}_{1.\overline{(I_1 \cap I_2)}} \cap \mathscr{C}_{2.\overline{I_1 \cap I_2}}\right)\right\}. \end{aligned}$$

Hence, $\dim(\mathscr{C}_1 \oplus \mathscr{C}_2) = \dim(\mathscr{C}_1 \mid \mathscr{C}_2) - \dim(\ker \phi)$ as desired.

From $\mathscr{C}_1 \oplus \mathscr{C}_2$ we can obtain a new code $S(\mathscr{C}_1, \mathscr{C}_2)$ by shortening at the $m = |I_1 \cap I_2|$ positions where $\mathscr{C}_1$ and $\mathscr{C}_2$ overlap. The codewords of this code will be denoted by $\mathbf{c}_1 \parallel_m \mathbf{c}_2$ where $\mathbf{c}_1 \in \mathscr{C}_1$ and $\mathbf{c}_2 \in \mathscr{C}_2$. When the defined alphabet is a finite field, Kashyap [65, Proposition 4.1] proves that $S(\mathscr{C}_1, \mathscr{C}_2)$ is a linear code of length $n_1 + n_2 - 2|I_1 \cap I_2|$ and dimension:

$$\begin{aligned}\dim(S(\mathscr{C}_1, \mathscr{C}_2)) &= \dim(\mathscr{C}_1 \oplus \mathscr{C}_2) - \dim(\mathscr{C}_{1_{\overline{I_1 \cap I_2}}} \oplus \mathscr{C}_{2_{\overline{I_1 \cap I_2}}}) \\ &= k_1 + k_2 - \dim(\mathscr{C}_{1_{\cdot(I_1 \cap I_2)}} \cap \mathscr{C}_{2_{\cdot(I_2 \cap I_1)}}) - \dim(\mathscr{C}_{1_{\overline{I_1 \cap I_2}}} \oplus \mathscr{C}_{2_{\overline{I_1 \cap I_2}}}).\end{aligned}$$

To show the above claim, note that for any code $\mathscr{C}$ and a subset J of its index set, the kernel of the projection map $\pi : \mathscr{C} \longrightarrow \mathscr{C}_{\overline{J}}$ is isomorphic to $\mathscr{C}_{\cdot J}$. Thus, $\dim(\mathscr{C}_{\cdot J}) = \dim(\mathscr{C}) - \dim(\mathscr{C}_{\overline{J}})$. Now, taking $\mathscr{C} = \mathscr{C}_1 \oplus \mathscr{C}_2$ and $J = I_1 \cap I_2$, we have that

$$\dim\left(S(\mathscr{C}_1, \mathscr{C}_2)\right) = \dim\left(\mathscr{C}_1 \oplus \mathscr{C}_2\right) - \dim\left(\left(\mathscr{C}_1 \oplus \mathscr{C}_2\right)_{\overline{I_1 \cap I_2}}\right).$$

This result together with Equation 3.11 gives the sought result.

We are interested in a particular case of the above construction, explained below.

Definition 3.28. Let m be a positive integer other than two and let $\mathscr{C}_1$ and $\mathscr{C}_2$ be linear codes over $\mathbb{Z}_q$ of length at least $2m+1$ such that $|I_1 \cap I_2| = m$. If the following conditions are satisfied:

1. $0 \ldots 0 \underbrace{1 \ldots 1}_{m} \in \mathscr{M}_{\mathscr{C}_1}$ and all possible m-bit words appear in the last m coordinates of $\mathscr{C}_1$,
2. $\underbrace{1 \ldots 1}_{m} 0 \ldots 0 \in \mathscr{M}_{\mathscr{C}_2}$ and all possible m-bit words appear in the first m coordinates of $\mathscr{C}_2$,

then the code $S(\mathscr{C}_1, \mathscr{C}_2)$ is called the *m-gluing* modular code $\mathscr{C}_1 \textcircled{g}_m \mathscr{C}_2$.

Remark 3.29. The main reason for imposing the above condition on the lengths of the codes $\mathscr{C}_1$ and $\mathscr{C}_2$ is to ensure that the linear code $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_m \mathscr{C}_2$ has greater length than the maximum of the length of the original codes; i.e. $n \geq \max\{n_1, n_2\}$.

Remark 3.30. In the binary case, the above construction was studied in [63], where the case $m = 1$ was called "1-sum" and the case $m = 2$ was called "$\overline{3}$-sum". To avoid any confusion we remark that this operation is not the same as the m-sum as defined by Kashyap in [65], except when $m = 1$. Moreover, recall that Kashyap studied codes defined over a finite field.

Let $G_1 = \left(g_1^{(1)}, \ldots, g_{n_1}^{(1)}\right) \in \mathbb{Z}_q^{k_1 \times n_1}$ and $G_2 = \left(g_1^{(2)}, \ldots, g_{n_2}^{(2)}\right) \in \mathbb{Z}_q^{k_2 \times n_2}$ be generator matrices for any two linear codes $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively, satisfying the conditions of Definition 3.28, and let m be an integer such that $0 \leq 2m < \min\{n_1, n_2\}$. These two generator matrices can be composed to form a generator matrix of the code $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_m \mathscr{C}_2$ by:

1. First, constructing the $(k_1+k_2)\times(n_1+n_2-m)$-size matrix

$$\hat{G}=\begin{pmatrix} g_1^{(1)} & \cdots & g_{n_1-m}^{(1)} & g_{n_1-m+1}^{(1)} & \cdots & g_{n_1}^{(1)} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & g_1^{(2)} & \cdots & g_m^{(2)} & g_{m+1}^{(2)} & \cdots & g_{n_2}^{(2)} \end{pmatrix}$$

2. Then, shortening the code generated by the rows in $\hat{G}$ at the m positions where $\mathscr{C}_1$ and $\mathscr{C}_2$ overlap, resulting a generator matrix of the code $\mathscr{C}$ of length n_1+n_2-2m.

3.4.1 Direct sum of modular codes

Definition 3.31. Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be $[n_i,k_i]$-codes over $\mathbb{Z}_q$ for $i=1,2$ defined on mutually disjoint index set I_1 and I_2, respectively, (i.e. $I_1\cap I_2=\emptyset$). We can construct the direct sum (or 0-gluing) code $\mathscr{C}_1 \textcircled{g} \mathscr{C}_2$ with $I_1\cup I_2$ as its index set such that any codeword $\mathbf{c}$ of $\mathscr{C}$ is defined by the concatenation $\mathbf{c}=(\mathbf{c}_1 \parallel \mathbf{c}_2)$, that is

$$c_i=\begin{cases} c_1^{(i)} & \text{for } i\in I_1 \\ c_2^{(i)} & \text{for } i\in I_2 \end{cases} \quad \text{where} \quad \begin{cases} \mathbf{c}_1=\left(c_1^{(i)} \mid i\in I_1\right)\in\mathscr{C}_1 \\ \mathbf{c}_2=\left(c_2^{(i)} \mid i\in I_2\right)\in\mathscr{C}_2 \end{cases}$$

The following proposition states the connection between the direct sum of two codes and the sum of its associated ideals.

Let $\mathbf{X}$ denotes the n_1 variables $\{x_i \mid i\in I_1\}$ and $\mathbf{Y}$ denotes the n_2 variables $\{y_j \mid j\in I_2\}$ then the binomial ideals associated to the codes $\mathscr{C}_1 \textcircled{g} \mathscr{C}_2$, $\mathscr{C}_1$ and $\mathscr{C}_2$ are defined as follows:

- $I(\mathscr{C}_1)=\left\langle \left\{ \mathbf{X}^{\blacktriangle\alpha}-\mathbf{X}^{\blacktriangle\beta} \mid \alpha-\beta\in\mathscr{C}_1 \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}]$.
- $I(\mathscr{C}_2)=\left\langle \left\{ \mathbf{Y}^{\blacktriangle\gamma}-\mathbf{Y}^{\blacktriangle\delta} \mid \gamma-\delta\in\mathscr{C}_2 \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{Y}]$.
- $I(\mathscr{C}_1 \textcircled{g} \mathscr{C}_2)=\left\langle \left\{ \mathbf{X}^{\blacktriangle\alpha}\mathbf{Y}^{\blacktriangle\gamma}-\mathbf{X}^{\blacktriangle\beta}\mathbf{Y}^{\blacktriangle\delta} \mid (\alpha-\beta \parallel \gamma-\delta)\in\mathscr{C}_1 \textcircled{g} \mathscr{C}_2 \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X},\mathbf{Y}]$.

In [74, Theorem 3.2] or equivalently in Theorem 3.9 we proved the equality between the above ideals and the ideal associated to a code as defined in Equation 3.10.

Proposition 3.32. *$\mathscr{C}=\mathscr{C}_1 \textcircled{g} \mathscr{C}_2$ if and only if $I(\mathscr{C})=I(\mathscr{C}_1)+I(\mathscr{C}_2)$.*

Proof. First, let us assume that $\mathscr{C}=\mathscr{C}_1 \textcircled{g} \mathscr{C}_2$. Let $\mathbf{X}^\alpha-\mathbf{X}^\beta$ be a binomial of $I(\mathscr{C}_1)$ then $\blacktriangledown(\alpha-\beta)\in\mathscr{C}_1$ or equivalently $(\blacktriangledown\alpha-\blacktriangledown\beta \parallel \mathbf{0})\in\mathscr{C}$, i.e. $\mathbf{X}^\alpha-\mathbf{X}^\beta\in I(\mathscr{C})$. The case $I(\mathscr{C}_2)\subseteq I(\mathscr{C})$ can be solved likeliwise. Thus $I(\mathscr{C}_1)+I(\mathscr{C}_2)\subseteq I(\mathscr{C})$.

Conversely, let $\mathbf{X}^\alpha\mathbf{Y}^\gamma-\mathbf{X}^\beta\mathbf{Y}^\delta\in I(\mathscr{C})$ then $(\blacktriangledown\alpha-\blacktriangledown\beta \parallel \blacktriangledown\gamma-\blacktriangledown\delta)$ is a codeword of $\mathscr{C}$, that is to say that $\blacktriangledown\alpha-\blacktriangledown\beta\in\mathscr{C}_1$ and $\blacktriangledown\gamma-\blacktriangledown\delta\in\mathscr{C}_2$. Thus,

$$\begin{aligned} \mathbf{X}^\alpha\mathbf{Y}^\gamma-\mathbf{X}^\beta\mathbf{Y}^\delta &= \mathbf{X}^\alpha\mathbf{Y}^\gamma-\mathbf{X}^\beta\mathbf{Y}^\gamma+\mathbf{X}^\beta\mathbf{Y}^\gamma-\mathbf{X}^\beta\mathbf{Y}^\delta \\ &= (\mathbf{X}^\alpha-\mathbf{X}^\beta)\mathbf{Y}^\gamma+\mathbf{x}^\beta(\mathbf{Y}^\gamma-\mathbf{Y}^\delta)\in I(\mathscr{C}_1)+I(\mathscr{C}_2). \end{aligned}$$

Therefore we have that $I(\mathscr{C}) = I(\mathscr{C}_1) + I(\mathscr{C}_2)$.

On the other hand, assume that $I(\mathscr{C}) = I(\mathscr{C}_1) + I(\mathscr{C}_2)$. Specifically

$$\ker(H) = \langle \ker(H_1) \times \ker(H_2) \rangle$$

where H, H_1 and H_2 are parity check matrices for $\mathscr{C}$, $\mathscr{C}_1$ and $\mathscr{C}_2$ respectively. Given $\mathbf{c} \in \mathscr{C}$ by hypothesis $H\mathbf{c}^T = 0$ and it can be expressed as $\mathbf{c} = (\mathbf{c}_{I_1} \parallel \mathbf{c}_{I_2})$, or equivalently

$$\mathbf{c}_{I_1} H_1^T = \mathbf{c}_{I_2} H_2^T = 0; \quad \text{i.e. } \mathbf{c} = (\mathbf{c}_{I_1} \parallel \mathbf{c}_{I_2}) \quad \text{with} \quad \mathbf{c}_{I_1} \in \mathscr{C}_1 \quad \text{and} \quad \mathbf{c}_{I_2} \in \mathscr{C}_2 \ .$$

Thus $\mathscr{C} \subseteq \mathscr{C}_1 \textcircled{g} \mathscr{C}_2$.

Contrariwise, let $\mathbf{c}_1 \in \mathscr{C}_1$ and $\mathbf{c}_2 \in \mathscr{C}_2$ then $(\mathbf{c}_1 \parallel \mathbf{c}_2) \in \ker(H_1) \times \ker(H_2) \subseteq \ker(H)$; i.e. $(\mathbf{c}_1 \parallel \mathbf{c}_2) \in \mathscr{C}$ and we deduce the required result. □

Corollary 3.33. *Let $\mathscr{C} = \mathscr{C}_1 \textcircled{g} \mathscr{C}_2$ then $I(\mathscr{C})$ is generated by the disjoint union of the generators of each $I(\mathscr{C}_i)$ with $i = 1, 2$.*

Proof. Trivial since the generators for each of the ideals $I(\mathscr{C}_i)$ with $i = 1, 2$ do not have common variables. □

The above result is also true for a Gröbner basis and Graver basis. Thus, in the binary case, we can study the test-sets for $\mathscr{C}$ by using the test-set for each $\mathscr{C}_i$ with $i = 1, 2$. More generally, for every modular code $\mathscr{C} = \mathscr{C}_1 \textcircled{g} \mathscr{C}_2$ defined over $\mathbb{Z}_q$ we can obtain the set of codewords of minimal support of $\mathscr{C}$ via the set of codewords of minimal support of $\mathscr{C}_1$ and $\mathscr{C}_2$.

Note also that the definition of direct sum can be easily extended to a finite family of linear codes. Indeed, given a set of linear codes $\{\mathscr{C}_\alpha\}_{\alpha \in A}$ defined over $\mathbb{Z}_q$ of parameters $[n_\alpha, k_\alpha]$ and defined on mutually disjoint index sets I_α with $\alpha \in A$, then we can define the code $\textcircled{g} \{\mathscr{C}_\alpha\}_{\alpha \in A}$ and a similar study of this code can be done.

Example 3.34. Consider $\mathscr{C}_H$ the $[7, 4, 3]$ Hamming code over $\mathbb{Z}_2$. See Example 3.23 for a complete description of its Gröbner test-set and the set of codewords of minimal support. Let us construct the code $\mathscr{C} = \mathscr{C}_H \textcircled{g} \mathscr{C}_H$ which turns out to be the binary linear $[14, 8, 3]$-code with generator matrix:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{Z}_2^{8 \times 14}.$$

Associated to the code $\mathscr{C}$ we can define the following binomial ideal:

$$I(\mathscr{C}) = \left\langle \left\{ \begin{array}{c} x_1x_6x_7 - 1, x_2x_5x_7 - 1, x_3x_5x_6 - 1, \\ x_4x_5x_6x_7 - 1, x_8x_{13}x_{14} - 1, x_9x_{12}x_{14} - 1, \\ x_{10}x_{12}x_{13} - 1, x_{11}x_{12}x_{13}x_{14} - 1 \end{array} \right\} \cup \left\{ x_i^2 - 1 \right\}_{i=1}^{14} \right\rangle.$$

The Graver basis of this ideal has 310 elements representing the set of codewords of minimal support for the code $\mathscr{C}$. Note that the 28 codewords of this set are formed by the concatenation of a codeword of minimal support of the code $\mathscr{C}_H$ and the vector $\mathbf{0} \in \mathbb{Z}_2^7$, i.e. they are of the form $(\mathbf{c} \parallel \mathbf{0})$ or $(\mathbf{0} \parallel \mathbf{c})$ with $\mathbf{c}$ belonging to $\mathscr{M}_{\mathscr{C}_H}$.

$$\mathscr{M}_{\mathscr{C}} = \left\{ \begin{array}{ll} (0,1,1,1,1,0,0,0,0,0,0,0,0,0), & (1,0,1,1,0,1,0,0,0,0,0,0,0,0), \\ (1,1,0,0,1,1,0,0,0,0,0,0,0,0), & (1,1,0,1,0,0,1,0,0,0,0,0,0,0), \\ (1,0,1,0,1,0,1,0,0,0,0,0,0,0), & (0,1,1,0,0,1,1,0,0,0,0,0,0,0), \\ (0,0,0,1,1,1,1,0,0,0,0,0,0,0), & (0,0,0,0,0,0,0,0,1,1,1,1,0,0), \\ (0,0,0,0,0,0,0,1,0,1,1,0,1,0), & (0,0,0,0,0,0,0,1,1,0,0,1,1,0), \\ (0,0,0,0,0,0,0,1,1,0,1,0,0,1), & (0,0,0,0,0,0,0,1,0,1,0,1,0,1), \\ (0,0,0,0,0,0,0,0,1,1,0,0,1,1), & (0,0,0,0,0,0,0,0,0,0,1,1,1,1), \\ (1,1,1,0,0,0,0,0,0,0,0,0,0,0), & (1,0,0,1,1,0,0,0,0,0,0,0,0,0), \\ (0,1,0,1,0,1,0,0,0,0,0,0,0,0), & (0,0,1,0,1,1,0,0,0,0,0,0,0,0), \\ (0,0,1,1,0,0,1,0,0,0,0,0,0,0), & (0,1,0,0,1,0,1,0,0,0,0,0,0,0), \\ (1,0,0,0,0,1,1,0,0,0,0,0,0,0), & (0,0,0,0,0,0,0,1,1,1,0,0,0,0), \\ (0,0,0,0,0,0,0,1,0,0,1,1,0,0), & (0,0,0,0,0,0,0,0,1,0,1,0,1,0), \\ (0,0,0,0,0,0,0,0,0,0,1,0,1,1,0), & (0,0,0,0,0,0,0,0,0,0,1,1,0,0,1), \\ (0,0,0,0,0,0,0,0,0,1,0,0,1,0,1), & (0,0,0,0,0,0,0,1,0,0,0,0,1,1) \end{array} \right\}$$

Similarly, for the Gröbner basis of $I(\mathscr{C})$ we obtain 56 elements representing 14 codewords of weight 3 form by the concatenation of a codeword of the Gröbner basis of $\mathscr{C}_H$ and the vector $\mathbf{0} \in \mathbb{Z}_2^7$.

Therefore, if we end with a decomposition of the code $\mathscr{C}$ as a direct sum of several smaller codes $\mathscr{C}_i$ with $i \in A$, the cost of computing the set of codewords of minimal support of $\mathscr{C}$ is reduced to the cost of computing the Graver basis for every code $\mathscr{C}_i$ that appears on its decomposition. Furthermore, since this procedure can be parallelized, we can reduce the time required for computing the set of codewords of minimal support of $\mathscr{C}$ to the time needed to compute the Universal test-set of the largest length code $\mathscr{C}_i$ with $i \in A$ that appears on its decomposition. In the binary case we have a similar procedure to reduce the time required for computing the Gröbner test-set for $\mathscr{C}$.

Remark 3.35. In Example 3.34 Sage [104] on a Mac Os X it took 0.02 CPU time to compute the set of codewords of minimal support of $\mathscr{C} = \mathscr{C}_H \textcircled{g} \mathscr{C}_H$ while just 0.01 CPU time were enough to compute the set of codewords of minimal support of $\mathscr{C}_H$. Therefore, in this toy example we have reduced the time by a half.

Remark 3.36. The test to determine if a linear code $\mathscr{C}$ over $\mathbb{Z}_q$ is expressible as a direct sum of a family of linear codes $\{\mathscr{C}_\alpha\}_{\alpha \in A}$ is given by the following steps.

1. Take any generator or parity check matrix of $\mathscr{C}$, say $M = (m_{ij}) \in \mathbb{Z}_q^{m \times n}$.

2. Bring M into systematic form $M = \begin{pmatrix} I_m & A \end{pmatrix}$ using row operations and column permutations.

3. Associate a bipartite graph $T = (R \cup C; E_T)$ to the matrix M by setting the variables $C = \{c_1, \ldots, c_n\}$ to the columns of M, the variables $R = \{r_1, \ldots, r_m\}$ to the rows of M and defining any edge of T as $e = \{c_i, r_j\}$ with $c_i \in C$ and $r_j \in R$ whenever $m_{ij} \neq 0$. The connected components of T then induces the required structure of the codes $\mathscr{C}_\alpha$. If T is connected then $\mathscr{C}$ cannot be expressed as the direct sum of two or more linear codes.

Observe that if we consider a parity-check matrix of $\mathscr{C}$ in systematic form, then the graph T is the corresponding Tanner Graph of $\mathscr{C}$. This test has previously been used by Kashyap in [63].

Example 3.37. Let us consider the $[7,4]$-code $\mathscr{C}$ defined over $\mathbb{Z}_5$ with parity check matrix

$$H_{\mathscr{C}_2} = \begin{pmatrix} 1 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 4 & 4 \end{pmatrix} \in \mathbb{Z}_5^{3\times 7}.$$

Figure 3.1 represents the Tanner graph of the parity check matrix H.

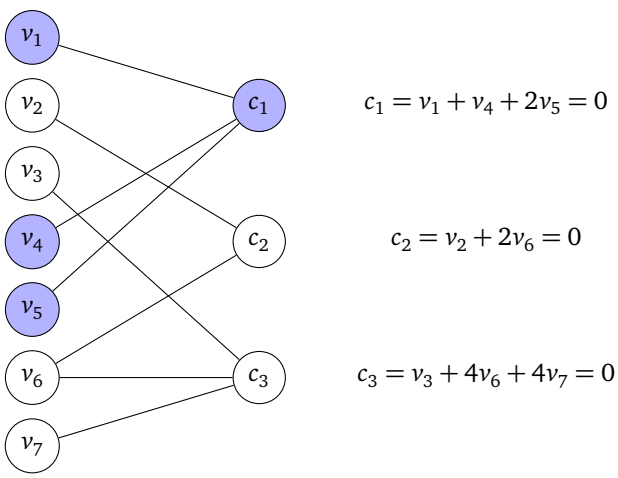


Figure 3.1: Tanner graph of the parity check matrix H.

Note that this graph has two connected components which describe two codes $\mathscr{C}_1$ and $\mathscr{C}_2$ defined over $\mathbb{Z}_5$ with parity check matrices

$$H_{\mathscr{C}_1} = \begin{pmatrix} 1 & 1 & 2 \end{pmatrix} \in \mathbb{Z}_5^{1\times 3} \quad \text{and} \quad H_{\mathscr{C}_2} = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 4 & 4 \end{pmatrix} \in \mathbb{Z}_5^{2\times 4}$$

and index sets $I_1 = \{1,4,5\}$ and $I_2 = \{2,3,6,7\}$, respectively. These two codes verified that $\mathscr{C} = \mathscr{C}_1 \oplus \mathscr{C}_2$.

The ideals associated to these codes are:

$$I(\mathscr{C}_1) = \left\langle \left\{ x_1x_4x_5^2 - 1 \right\} \cup \left\{ x_i^5 - 1 \right\}_{i=1,4,5} \right\rangle \subseteq \mathbb{F}_2[x_1, x_4, x_5]$$

$$I(\mathscr{C}_2) = \left\langle \left\{ x_2x_6^2 - 1, x_3x_6^4x_7^4 \right\} \cup \left\{ x_i^5 - 1 \right\}_{i=2,3,6,7} \right\rangle \subseteq \mathbb{F}_2[x_2, x_3, x_6, x_7].$$

If we compute a Graver basis of the above ideals we get the set of codewords of minimal support of each code. That is:

- All codewords of $\mathscr{C}_1$ has minimal support. Thus,

$$\mathscr{M}_{\mathscr{C}_1} = \left\{ \begin{array}{cccc} (1,1,2) & (2,2,4) & (3,3,1) & (4,4,3) \end{array} \right\}$$

- The code $\mathscr{C}_2$ has 12 minimal support codewords given by

$$\mathscr{M}_{\mathscr{C}_2} = \left\{ \begin{array}{cccc} (1,0,2,0) & (2,0,4,0) & (3,0,1,0) & (4,0,3,0) \\ (0,1,4,4) & (0,2,3,3) & (0,3,2,2) & (0,4,1,1) \\ (1,2,0,3) & (2,4,0,1) & (3,1,0,4) & (4,3,0,2) \end{array} \right\}.$$

Similarly we can compute the set of 16 codewords of minimal support of $\mathscr{C}$ defined by

$$\mathscr{M}_{\mathscr{C}} = \left\{ \begin{array}{cccc} (1,0,0,1,2,0,0) & (2,0,0,2,4,0,0) & (3,0,0,3,1,0,0) & (4,0,0,4,3,0,0) \\ (0,1,0,0,0,2,0) & (0,2,0,0,0,4,0) & (0,3,0,0,0,1,0) & (0,4,0,0,0,3,0) \\ (0,0,1,0,0,4,4) & (0,0,2,0,0,3,3) & (0,0,3,0,0,2,2) & (0,0,4,0,0,1,1) \\ (0,1,2,0,0,0,3) & (0,2,4,0,0,0,1) & (0,3,1,0,0,0,4) & (0,4,3,0,0,0,2) \end{array} \right\}.$$

Note that all codewords of $\mathscr{M}_{\mathscr{C}}$ are of the form:

$$\left(\mathbf{c}_1 \,||\, \mathbf{0}\right) \text{ with } \mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_1} \quad \text{or} \quad \left(\mathbf{0} \,||\, \mathbf{c}_2\right) \text{ with } \mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_2} \ .$$

3.4.2 1-gluing of modular codes

Definition 3.38. Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be modular codes over $\mathbb{Z}_q$ of length at least 3 such that $|I_1 \cap I_2| = 1$. We will assume w.l.o.g. that the common index corresponds to the last coordinate of $\mathscr{C}_1$ and the first coordinate of $\mathscr{C}_2$. Moreover, if the following conditions are satisfied:

1. $(0, \ldots, 0, 1)$ is not a codeword of $\mathscr{C}_1$ and the last coordinate of $\mathscr{C}_1$ is neither identically zero nor a zero divisor.
2. $(1, 0, \ldots, 0)$ is not a codeword of $\mathscr{C}_2$ and the first coordinate of $\mathscr{C}_2$ is neither identically zero nor a zero divisor.

Then the 1-gluing modular code $\mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ can be defined as the modular code $S(\mathscr{C}_1, \mathscr{C}_2)$ over $\mathbb{Z}_q$.

The following lemma allows us to characterize the set of codewords of the code $\mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$.

Lemma 3.39. *$\mathbf{c} \in \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ if and only if there exist two codewords $\mathbf{c}_1 \in \mathscr{C}_1$ and $\mathbf{c}_2 \in \mathscr{C}_2$ such that $c_1^{(n_1)} + c_2^{(1)} = 0$ and $\mathbf{c} = \mathbf{c}_1 \parallel_1 \mathbf{c}_2$.*

Proof. Any vector $\mathbf{c} \in \mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ can be extended to a vector $\hat{\mathbf{c}}$ of $\mathscr{C}_1 \oplus \mathscr{C}_2$ by inserting a zero in the n_1-th position. In other words, we found two codewords $\mathbf{c}_1 \in \mathscr{C}_1$ and $\mathbf{c}_2 \in \mathscr{C}_2$ such that

$$\hat{\mathbf{c}} = (\hat{c}_i \mid i \in I_1 \cup I_2) \quad \text{with} \quad \hat{c}_i = \begin{cases} c_1^{(i)} & 1 \leq i < n_1 \\ 0 = c_1^{(n_1)} + c_2^{(1)} & i = n_1 \\ c_2^{(i-n_1+1)} & n_1 < i \leq n_1 + n_2 - 1 \end{cases}$$

Therefore $\mathbf{c} = \mathbf{c}_1 \parallel_1 \mathbf{c}_2$ with $c_1^{(n_1)} + c_2^{(1)} = 0$.

Conversely, let $\mathbf{c}_1$ and $\mathbf{c}_2$ be codewords of $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively, such that $c_1^{(n_1)} + c_2^{(1)} = 0$. Then the vector $(\mathbf{c}_1 \parallel \mathbf{c}_2)$ shortened on the n_1-th coordinate is by definition a codeword of $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$. □

Let $\mathbf{X}$ denote the $n_1 - 1$ variables $\left\{x_1, \ldots, x_{n_1-1}\right\}$, $\mathbf{Y}$ denote the $n_2 - 1$ variables $\left\{y_2, \ldots, y_{n_2}\right\}$ and $\mathbf{Z}$ denote one extra variable. Then the binomials ideal associated to each of the following codes $\mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$, $\mathscr{C}_{1_{\cdot\{n_1\}}}$ and $\mathscr{C}_{2_{\cdot\{1\}}}$ are defined as follows:

- $I(\mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2) = \left\langle \left\{ \mathbf{X}^{\blacktriangle\alpha} \mathbf{Y}^{\blacktriangle\beta} - \mathbf{X}^{\blacktriangle\gamma} \mathbf{Y}^{\blacktriangle\delta} \mid (\alpha - \gamma \mid \beta - \delta) \in \mathscr{C} \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}, \mathbf{Y}]$.
- $I(\mathscr{C}_1) = \left\langle \left\{ \mathbf{X}^{\blacktriangle\alpha} \mathbf{Z}^{\blacktriangle\alpha_{n_1}} - \mathbf{X}^{\blacktriangle\beta} \mathbf{Z}^{\blacktriangle\beta_{n_1}} \mid \left(\alpha - \beta \mid \alpha_{n_1} - \beta_{n_1}\right) \in \mathscr{C}_1 \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}, \mathbf{Z}]$ thus

$$I(\mathscr{C}_{1_{\cdot\{n_1\}}}) = \left\langle \left\{ \mathbf{X}^{\blacktriangle\alpha} - \mathbf{X}^{\blacktriangle\beta} \mid \alpha - \beta \in \mathscr{C}_{1_{\cdot\{n_1\}}} \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}].$$

- $I(\mathscr{C}_2) = \left\langle \left\{ \mathbf{Z}^{\blacktriangle\gamma_1} \mathbf{Y}^{\blacktriangle\gamma} - \mathbf{Z}^{\blacktriangle\delta_1} \mathbf{Y}^{\blacktriangle\delta} \mid (\gamma_1 - \delta_1 \mid \gamma - \delta) \in \mathscr{C}_2 \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{Z}, \mathbf{Y}]$ thus

$$I(\mathscr{C}_{2_{\cdot\{1\}}}) = \left\langle \left\{ \mathbf{Y}^{\blacktriangle\gamma} - \mathbf{Y}^{\blacktriangle\delta} \mid \gamma - \delta \in \mathscr{C}_{2_{\cdot\{1\}}} \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{Y}].$$

The following proposition states the connection between the 1-gluing of two codes and the sum of its associated ideals.

Proposition 3.40. *$\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ if and only if*

$$I(\mathscr{C}) = I(\mathscr{C}_{1_{\cdot\{n_1\}}}) + I(\mathscr{C}_{2_{\cdot\{1\}}}) + \left\langle \mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y} \right\rangle,$$

where:

- *$\gamma_x \in \mathbb{Z}_q^{(n_1-1)}$, $\mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Z} \in I(\mathscr{C}_1)$ and $\lambda_1 \gamma_x + \lambda_2 \mathbf{c}_1 \neq 0$ for all $\lambda_1,\ \lambda_2 \in \mathbb{Z}_q \setminus \{0\}$ and $\mathbf{c}_1 \in \mathscr{C}_{1_{\cdot\{n_1\}}}$.*

- $\gamma_y \in \mathbb{Z}_q^{(n_2-1)}$, $\mathbf{Z} - \mathbf{Y}^{\blacktriangle\gamma_y} \in I(\mathscr{C}_2)$ *and* $\lambda_1\gamma_y + \lambda_2\mathbf{c}_2 \neq 0$ *for all* $\lambda_1,\ \lambda_2 \in \mathbb{Z}_q \setminus \{0\}$ *and* $\mathbf{c}_2 \in \mathscr{C}_{2_{\cdot\{1\}}}$.

Proof. Let us assume that $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$. We claim that

$$I(\mathscr{C}) = I(\mathscr{C}_{1_{\cdot\{n_1\}}}) + I(\mathscr{C}_{2_{\cdot\{1\}}}) + \left\langle \mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y} \right\rangle.$$

Let $\mathbf{X}^\alpha - \mathbf{X}^\beta$ be a binomial of $I(\mathscr{C}_{1_{\cdot\{n_1\}}})$, then $\blacktriangledown\left(\alpha-\beta\right) \in \mathscr{C}_{1_{\cdot\{n_1\}}}$; or equivalently, $\left(\blacktriangledown(\alpha-\beta) \mid 0\right) \in \mathscr{C}_1$. Therefore, by Lemma 3.39, $\left(\blacktriangledown\left(\alpha-\beta\right), 0\right) \|_1 \mathbf{0}$ is a codeword of $\mathscr{C}$ with $\mathbf{0} \in \mathbb{Z}_q^{n_2}$. Hence $\mathbf{X}^\alpha - \mathbf{X}^\beta \in I(\mathscr{C})$. The case $I(\mathscr{C}_{2_{\cdot\{1\}}}) \subseteq I(\mathscr{C})$ can be solved likewise.

Furthermore, since $\mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Z}$ and $\mathbf{Z} - \mathbf{Y}^{\blacktriangle\gamma_y}$ belongs to $I(\mathscr{C}_1)$ and $I(\mathscr{C}_2)$, respectively; by definition we have that $\left(\gamma_x \mid -1\right) \in \mathscr{C}_1$ and $\left(1 \mid -\gamma_y\right) \in \mathscr{C}_2$. Thus, by Lemma 3.39, $\left(\gamma_x \mid -\gamma_y\right) \in \mathscr{C}$ i.e. $\mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y} \in I(\mathscr{C})$.
We have actually proved that $I(\mathscr{C}_{1_{\cdot\{n_1\}}}) + I(\mathscr{C}_{2_{\cdot\{1\}}}) + \left\langle \mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y} \right\rangle \subseteq I(\mathscr{C})$.

Conversely, let $\mathbf{X}^\alpha\mathbf{Y}^\beta - \mathbf{X}^\gamma\mathbf{Y}^\delta$ be a binomial of $I(\mathscr{C})$ then $\left(\blacktriangledown(\alpha-\gamma) \mid \blacktriangledown(\beta-\delta)\right)$ is a codeword of $\mathscr{C}$. By Lemma 3.39 there exist two codewords $\left(\blacktriangledown(\alpha-\gamma) \mid *_1\right) \in \mathscr{C}_1$ and $\left(*_2 \mid \blacktriangledown(\beta-\delta)\right) \in \mathscr{C}_2$ with $*_1, *_2 \in \mathbb{Z}_q$ and $*_1 + *_2 = 0$. We distinguish two cases:

1. If $*_1 = *_2 = 0$, that is to say, $\left(\blacktriangledown(\alpha-\gamma) \mid 0\right) \in \mathscr{C}_1$ and $\left(0 \mid \blacktriangledown(\beta-\delta)\right) \in \mathscr{C}_2$ or, equivalently, $\mathbf{X}^\alpha - \mathbf{X}^\gamma \in I(\mathscr{C}_{1_{\cdot\{n_1\}}})$ and $\mathbf{Y}^\beta - \mathbf{Y}^\delta \in I(\mathscr{C}_{2_{\cdot\{1\}}})$. Thus

$$\begin{aligned}\mathbf{X}^\alpha\mathbf{Y}^\beta - \mathbf{X}^\gamma\mathbf{Y}^\delta &= \mathbf{X}^\alpha\mathbf{Y}^\beta - \mathbf{X}^\gamma\mathbf{Y}^\beta + \mathbf{X}^\gamma\mathbf{Y}^\beta - \mathbf{X}^\gamma\mathbf{Y}^\delta \\ &= (\mathbf{X}^\alpha - \mathbf{X}^\gamma)\mathbf{Y}^\beta + \mathbf{X}^\gamma(\mathbf{Y}^\beta - \mathbf{Y}^\delta) \in I(\mathscr{C}_{1_{\cdot\{n_1\}}}) + I(\mathscr{C}_{2_{\cdot\{1\}}}).\end{aligned}$$

2. If $*_1 \neq 0 \neq *_2$, that is to say, $\left(\blacktriangledown(\alpha-\gamma) \mid *_1\right) \in \mathscr{C}_1$ and $\left(*_2 \mid \blacktriangledown(\beta-\delta)\right) \in \mathscr{C}_2$ or, equivalently, $\mathbf{X}^\alpha\mathbf{Z}^{\blacktriangle *_1} - \mathbf{X}^\gamma \in I(\mathscr{C}_1)$ and $\mathbf{Z}^{\blacktriangle *_2}\mathbf{Y}^\beta - \mathbf{Y}^\delta \in I(\mathscr{C}_2)$. Since $\left(\gamma_x \mid -1\right) \in \mathscr{C}_1$ then

$$\left(\blacktriangledown(\alpha-\gamma) \mid *_1\right) + *_1\left(\gamma_x \mid -1\right) = \left(\blacktriangledown(\alpha-\gamma) + *_1\gamma_x \mid 0\right) \in \mathscr{C}_1.$$

So if we remove the n_1-th coordinate we have that $(\blacktriangledown(\alpha-\gamma) + *_1\gamma_x) \in \mathscr{C}_{1_{\cdot\{n_1\}}}$ or, equivalently, $\mathbf{X}^\alpha\mathbf{X}^{\blacktriangle(*_1\gamma_x)} - \mathbf{X}^\gamma \in I(\mathscr{C}_{1_{\cdot\{n_1\}}})$. Similarly we prove that

$$(0 \mid \blacktriangledown(\beta-\delta) + *_2\gamma_y) = (0 \mid \blacktriangledown\beta - *_1\gamma_y - \blacktriangledown\delta) \in \mathscr{C}_2,$$

i.e. $\mathbf{Y}^\beta - \mathbf{Y}^\delta\mathbf{Y}^{\blacktriangle(*_1\gamma_y)} \in I(\mathscr{C}_{2_{\cdot\{1\}}})$.

Thus $\mathbf{X}^\alpha\mathbf{Y}^\beta - \mathbf{X}^\gamma\mathbf{Y}^\delta$ can be written as:

$$\begin{aligned}&\mathbf{X}^\alpha\mathbf{Y}^\beta - \mathbf{X}^\alpha\mathbf{Y}^\delta\mathbf{Y}^{\blacktriangle(*_1\gamma_y)} + \mathbf{X}^\alpha\mathbf{Y}^\delta\mathbf{Y}^{\blacktriangle(*_1\gamma_y)} - \mathbf{X}^\alpha\mathbf{X}^{\blacktriangle(*_1\gamma_x)}\mathbf{Y}^\delta + \mathbf{X}^\alpha\mathbf{X}^{\blacktriangle(*_1\gamma_x)}\mathbf{Y}^\delta - \mathbf{X}^\gamma\mathbf{Y}^\delta \\ &= \mathbf{X}^\alpha\left(\mathbf{Y}^\beta - \mathbf{Y}^\delta\mathbf{Y}^{\blacktriangle(*_1\gamma_y)}\right) - \mathbf{X}^\alpha\mathbf{Y}^\delta\left(\mathbf{X}^{\blacktriangle(*_1\gamma_x)} - \mathbf{Y}^{\blacktriangle(*_1\gamma_y)}\right) + \mathbf{Y}^\delta\left(\mathbf{X}^\alpha\mathbf{X}^{\blacktriangle(*_1\gamma_x)} - \mathbf{X}^\gamma\right)\end{aligned}$$

which gives $\mathbf{X}^\alpha\mathbf{Y}^\beta - \mathbf{X}^\gamma\mathbf{Y}^\delta \in I(\mathscr{C}_{1_{\cdot\{n_1\}}}) + I(\mathscr{C}_{2_{\cdot\{1\}}}) + \left\langle \mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y} \right\rangle$ since

$$\mathbf{X}^{\blacktriangle(*_1\gamma_x)} - \mathbf{Y}^{\blacktriangle(*_1\gamma_y)} = \left(\mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y}\right)\left(\sum_{i=1}^{\blacktriangle *_1} \mathbf{X}^{(\blacktriangle *_1 - i)\blacktriangle\gamma_x}\mathbf{Y}^{(i-1)\blacktriangle\gamma_y}\right).$$

On the other hand, let H, H_1 and H_2 be parity check matrices of $\mathscr{C}$, $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively. By definition the matrices $\hat{H}_1$ obtained by deleting the last column of H_1 and $\hat{H}_2$ by deleting the first column of H_2, are parity check matrices of $\mathscr{C}_{1_{\cdot\{n_1\}}}$ and $\mathscr{C}_{2_{\cdot\{1\}}}$, respectively.
The following assumption is straightfoward:

$$\text{If } I(\mathscr{C}) = I(\mathscr{C}_{1_{\cdot\{n_1\}}}) + I(\mathscr{C}_{2_{\cdot\{1\}}}) + \left\langle \mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y} \right\rangle, \text{ then}$$

$$\ker(H) = \left\langle \left\{ \ker(\hat{H}_1) \times \ker(\hat{H}_2) \right\} \cup \left\{ \left(\gamma_x, -\gamma_y \right) \right\} \right\rangle. \tag{3.12}$$

Let us assume that $I(\mathscr{C}) = I(\mathscr{C}_{1_{\cdot\{n_1\}}}) + I(\mathscr{C}_{2_{\cdot\{1\}}}) + \left\langle \mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y} \right\rangle$ and let $\mathbf{c}$ be a codeword of $\mathscr{C}$ then $\mathbf{H}\mathbf{c}^T = 0$. By Equation 3.12, there exist $\lambda_1,\ \lambda_2 \in \mathbb{Z}_q$ such that $\mathbf{c} = \lambda_1 \left(\hat{\mathbf{c}}_1 \mid \hat{\mathbf{c}}_2 \right) + \lambda_2 \left(\gamma_x \mid -\gamma_y \right)$ with $\left(\hat{\mathbf{c}}_1 \mid \hat{\mathbf{c}}_2 \right) \in \ker(\hat{H}_1) \times \ker(\hat{H}_2)$; that is to say, $\hat{\mathbf{c}}_1 \in \mathscr{C}_{1_{\cdot\{n_1\}}}$ and $\hat{\mathbf{c}}_2 \in \mathscr{C}_{2_{\cdot\{1\}}}$ or, equivalently, $(\hat{\mathbf{c}}_1 \mid 0) \in \mathscr{C}_1$ and $(0 \mid \hat{\mathbf{c}}_2) \in \mathscr{C}_2$. Therefore, $\mathbf{c} = \mathbf{c}_1 \parallel_1 \mathbf{c}_2 \in \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ where

$$\mathbf{c}_1 = \left(\lambda_1 \hat{\mathbf{c}}_1 + \lambda_2 \gamma_x \mid 1 \right) \in \mathscr{C}_1 \quad \text{and} \quad \mathbf{c}_2 = \left(-1 \mid \lambda_1 \hat{\mathbf{c}}_2 - \lambda_2 \gamma_y \right) \in \mathscr{C}_2.$$

Hence $\mathscr{C} \subseteq \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$.

Conversely, let $\mathbf{c}_1 \in \mathscr{C}_1$ and $\mathbf{c}_2 \in \mathscr{C}_2$ such that $c_1^{(n_1)} + c_2^{(1)} = 0$. We can define the vector $\mathbf{c}_1 \parallel_1 \mathbf{c}_2 = (\hat{\mathbf{c}}_1 \mid \hat{\mathbf{c}}_2)$ where $\hat{\mathbf{c}}_1$ denotes the restriction of $\mathbf{c}_1$ to the coordinates indexed by $\{1, \ldots, n_1 - 1\}$ and $\hat{\mathbf{c}}_2$ denotes the restriction of $\mathbf{c}_2$ to the coordinates indexed by $\{2, \ldots, n_2\}$.
If $c_1^{(n_1)} = 0$ then $c_2^{(1)} = 0$, or equivalently $\hat{\mathbf{c}}_1 \in \mathscr{C}_{1_{\cdot\{n_1\}}}$ and $\hat{\mathbf{c}}_2 \in \mathscr{C}_{2_{\cdot\{1\}}}$ i.e.

$$\mathbf{c}_1 \parallel_1 \mathbf{c}_2 \in \ker(\hat{H}_1) \times \ker(\hat{H}_2) \subseteq \ker(H).$$

Otherwise, since $\left(\gamma_x \mid -1 \right) \in \mathscr{C}_1$, then $\hat{H}_1 \gamma_x^T = H_1^{(n_1)}$ where $H_1^{(n_1)}$ denotes the last column of the matrix $H_1 \in \mathbb{Z}_q^{(n_1 - k_1) \times n_1}$. Therefore,

$$0 = H_1 \mathbf{c}_1^T = \hat{H}_1 \hat{\mathbf{c}}_1^T + c_1^{(n_1)} H_1^{(n_1)} = \hat{H}_1 \hat{\mathbf{c}}_1^T + c_1^{(n_1)} \hat{H}_1 \gamma_x^T.$$

That is, $\hat{\mathbf{c}}_1 + c_1^{(n_1)} \gamma_x \in \ker(\hat{H}_1)$. Similarly we have that $\hat{\mathbf{c}}_2 + c_2^{(1)} \gamma_y \in \ker(\hat{H}_2)$. Thus

$$\hat{\mathbf{c}} = \left(\hat{\mathbf{c}}_1 + c_1^{(n_1)} \gamma_x \mid \hat{\mathbf{c}}_2 - c_1^{(n_1)} \gamma_y \right) \in \ker(\hat{H}_1) \times \ker(\hat{H}_2)$$

Or equivalently, $\mathbf{c} = \mathbf{c}_1 \parallel_1 \mathbf{c}_2 = \hat{\mathbf{c}} + c_1^{(n_1)} (\gamma_x \mid -\gamma_y) \in \ker(H)$. Hence, the equality $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ is proved. □

Remark 3.41. By Definition 3.38 we can always find the binomials $\mathbf{X}^{\gamma_x} - \mathbf{Z}$ and $\mathbf{Z} - \mathbf{Y}^{\gamma_y}$ belonging to $I(\mathscr{C}_1)$ and $I(\mathscr{C}_2)$, respectively, verifying the required properties.

In the binary case, the following proposition allows us to compute the Gröbner test-set for $I(\mathscr{C})$ when $\mathscr{C} = \mathscr{C}_1 \textcircled{g} \mathscr{C}_2$ by using the Gröbner test-set of each $\mathscr{C}_i$ with $i = 1, 2$ w.r.t. the same degree compatible ordering.

Proposition 3.42. *Let $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$. Compute a reduced Gröbner basis of $I(\mathscr{C})$ w.r.t. a degree compatible ordering $\prec$ in $\mathbb{K}[\mathbf{X},\mathbf{Y}]$ induced by the order*

$$x_1 > \ldots > x_{n_1} > y_2 > \ldots > y_{n_2}$$

is equivalent to compute a reduced Gröbner basis of $I(\mathscr{C}_1)$ w.r.t. $\prec$ in $\mathbb{K}[\mathbf{X},\mathbf{Z}]$ induced by the order $x_1 > \ldots > x_{n_1-1} > \mathbf{Z}$, then a reduced Gröbner basis of $I(\mathscr{C}_2)$ w.r.t. $\prec$ in $\mathbb{K}[\mathbf{Z},\mathbf{Y}]$ induced by the order $\mathbf{Z} > y_2 > \ldots > y_{n_2}$ and making all possible combinations between the binomials of the two sets. Finally, if it were necessary, reduce it.

Proof. Let us fix an arbitrary degree compatible ordering $\prec$. Let $\mathscr{G}_1 \subseteq \mathbb{K}[\mathbf{X},\mathbf{Z}]$ be a reduced Gröbner basis of $I(\mathscr{C}_1)$ w.r.t. $\prec$ induced by the variable orderring $x_1 > \ldots > x_{n_1-1} > \mathbf{Z}$. Then this basis can be decomposed into two disjoints sets of binomials:

- $\mathscr{G}_1^{\mathbf{Z}}$ which contains all the binomials of $\mathscr{G}_1$ involving the variable $\mathbf{Z}$.
- $\mathscr{G}_1^{\overline{\mathbf{Z}}}$ which contains all the binomials of $\mathscr{G}_1$ that do not use the variable $\mathbf{Z}$.

Since $\mathscr{G}_1$ is a reduced Gröbner basis it is easy to see that $\mathscr{G}_1^{\overline{\mathbf{Z}}}$ is a reduced Gröbner basis of $I(\mathscr{C}_{1_{\cdot\{n_1\}}})$.

Similarly, we can decompose the reduced Gröbner basis $\mathscr{G}_2 \subseteq \mathbb{K}[\mathbf{Z},\mathbf{Y}]$ of $I(\mathscr{C}_2)$ w.r.t. $\prec$ induced the variable ordering $\mathbf{Z} > y_2 > \ldots > y_{n_2}$ into to disjoints binomials sets $\mathscr{G}_2^{\mathbf{Z}}$ and $\mathscr{G}_2^{\overline{\mathbf{Z}}}$.

It suffices to prove that $\left\{ \mathscr{G}_1^{\overline{\mathbf{Z}}} \|_1 \mathbf{0}, \quad \mathbf{0} \|_1 \mathscr{G}_1^{\overline{\mathbf{Z}}}, \quad \mathscr{G}_1^{\mathbf{Z}} \|_1 \mathscr{G}_2^{\mathbf{Z}} \right\}$ is a Gröbner basis of $I(\mathscr{C})$ w.r.t. $\prec$ induced by the order $x_1 > \ldots > x_{n_1-1} > y_2 > \ldots > y_{n_2}$. This statement is trivial, otherwise we could find a codeword $\mathbf{c}$ in $\mathscr{C}$ which is not of the form $\mathbf{c}_1 \|_1 \mathbf{c}_2$ with $c_1^{(n_1)} + c_2^{(1)} = 0$ which contradicts Lemma 3.39. □

The next proposition describes the set of codewords of minimal support of a 1-gluing code. In its proof we use some notions from $\mathbb{F}_q$-representable matroid theory, thus our results are restricted to linear codes defined over $\mathbb{Z}_q$ with q prime.

Proposition 3.43. *Let $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ be a linear code defined over $\mathbb{Z}_q$ with q prime, then the following statements are equivalent:*

1. $\mathbf{c} \in \mathscr{M}_{\mathscr{C}}$.
2. $\mathbf{c}$ *belongs to one of the following sets:*
   - $A = \left\{ (\mathbf{c}_1 \mid \mathbf{0}) \ : \ \mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_{1_{\cdot\{n_1\}}}} \quad \text{and} \quad \mathbf{0} \in \mathbb{Z}_q^{n_2-1} \right\}$.
   - $B = \left\{ (\mathbf{0} \mid \mathbf{c}_2) \ : \ \mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_{2_{\cdot\{1\}}}} \quad \text{and} \quad \mathbf{0} \in \mathbb{Z}_q^{n_1-1} \right\}$.
   - $C = \left\{ \mathbf{c}_1 \|_1 \mathbf{c}_2 \ : \ \mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_1}, \quad \mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_2} \quad \text{and} \quad c_1^{(n_1)} \neq 0 \neq c_2^{(1)} \right\}$.

Proof. Let $G_i \in \mathbb{Z}_q^{k_i \times n_i}$ and $H_i \in \mathbb{Z}_q^{(n_i-k_i)\times n_i}$ be a generator and a parity check matrix, respectively of the code $\mathscr{C}_i$ with $i = 1,2$. Using Proposition 3.40 it is easy to check that

$$H = \left(\begin{array}{c|c} \hat{H}_1 & A \\ \hline \mathbf{0} & B \end{array} \right) \in \mathbb{Z}_q^{(n-k)\times n}$$

is a parity check matrix of $\mathscr{C}$, where:

- $\hat{H}_1$ is a submatrix of H_1 obtained by deleting its last column, that is to say it is a parity check matrix of the code $\mathscr{C}_{1_{\cdot\{n_1\}}}$.
- Similarly, $\hat{H}_2$ is a submatrix of H_2 obtained by deleting its first column, i.e. $\hat{H}_2$ is a parity check matrix of the code $\mathscr{C}_{2_{\cdot\{1\}}}$.
- Now assume that B_1 denotes the matrix formed by the rows of H_2 where the first element is zero, then B is obtained from B_1 by deleting the first column.
- Furthermore, let A_1 be an $(n_1-k_1)\times n_2$ matrix whose rows are linear combinations of the rows of H_2 in such a way that $h^{(1)}_{i,n_1} = -a_{i,1}$ in $\mathbb{Z}_q$, where $a_{i,j}$ denotes the element of matrix A_1 in row i and column j and in the same manner $h^{(1)}_{i,j}$ indicates the (i,j)-th element of the matrix H_1. Then A is obtained from A_1 by deleting the first column.

Note that all rows of $\hat{H}_2$ are present in some way in the matrix $\left(\frac{A}{B}\right)$.

Let E_1 and E_2 be the index set of the columns of $\hat{H}_1$ and $H \setminus \hat{H}_1$ respectively. We consider any codeword $\mathbf{c}$ from the set $\mathscr{M}_{\mathscr{C}}$:

- If $\mathrm{supp}(\mathbf{c}) \cap E_2 = \emptyset$ and taking into account that $(\alpha 0 \dots 0) \notin \mathscr{C}_2$, then the only possibility remaining (that does not contradict the minimality of $\mathbf{c}$) is that there exists $\mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_{1_{\cdot\{n_1\}}}}$ such that $\mathbf{c} = (\mathbf{c}_1 \mid \mathbf{0})$.
- Similarly, if $\mathrm{supp}(\mathbf{c}) \cap E_1 = \emptyset$, there must exist a codeword $\mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_{2_{\cdot\{1\}}}}$ such that $\mathbf{c} = (\mathbf{0} \mid \mathbf{c}_2)$.
- If $\mathrm{supp}(\mathbf{c})$ intersects both E_1 and E_2, say $\hat{E}_1$ and $\hat{E}_2$, then we know by matroid theory that the set of columns indexed by $\hat{E}_1 \cup \hat{E}_2$ is a minimally dependent set. Therefore the set of columns of $\hat{H}_1$ indexed by $\hat{E}_1$ are linearly independent, so any codeword of $\mathscr{C}_1$ with support $\hat{E}_1 \cup \{n_1\}$ is a codeword of the set $\mathscr{M}_{\mathscr{C}_1}$. Likewise, any codeword of $\mathscr{C}_2$ with support $\{1\} \cap \hat{E}_2$ is a codeword of the set $\mathscr{M}_{\mathscr{C}_2}$.

  In other words, if $\mathbf{c} = \mathbf{c}_1 \parallel_1 \mathbf{c}_2$ with $c_1^{(n_1)} \neq 0$ then $\mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_1}$ and $\mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_2}$.

Conversely, let $\mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_1}$ and $\mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_2}$ with $c_1^{(n_1)} + c_2^{(1)} = 0$ then we distinguish two cases:

1. If $c_1^{(n_1)} = c_2^{(1)} = 0$, then $(\hat{\mathbf{c}}_1 \mid \mathbf{0}) \in \mathscr{M}_{\mathscr{C}}$ and $(\mathbf{0} \mid \hat{\mathbf{c}}_2) \in \mathscr{M}_{\mathscr{C}}$ where $\hat{\mathbf{c}}_1$ denotes the restriction of $\mathbf{c}_1$ to the coordinates indexed by $\{1, \dots, n_1 - 1\}$ and $\hat{\mathbf{c}}_2$ denotes the restriction of $\mathbf{c}_2$ to the coordinates indexed by $\{2, \dots, n_2\}$. Or equivalently, $\hat{\mathbf{c}}_1 \in \mathscr{C}_{1_{\cdot\{n_1\}}}$ and $\hat{\mathbf{c}}_2 \in \mathscr{C}_{2_{\cdot\{1\}}}$. Otherwise we obtain a contradiction to the minimality of $\mathbf{c}_1$ in $\mathscr{C}_1$ and similarly with the minimality of $\mathbf{c}_2$ in $\mathscr{C}_2$.

2. If $c_1^{(n_1)} \neq 0$. Assume to the contrary that $\mathbf{c} = \mathbf{c}_1 \,||_1\, \mathbf{c}_2 \notin \mathscr{M}_{\mathscr{C}}$. Then, there exists $\hat{\mathbf{c}} \in \mathscr{C}$ such that $\mathrm{supp}(\hat{\mathbf{c}}) \subsetneq \mathrm{supp}(\mathbf{c})$. Furthermore, since $\hat{\mathbf{c}} \in \mathscr{C}$, by Lemma 3.39, there exists $\hat{\mathbf{c}}_1 \in \mathscr{C}_1$ and $\hat{\mathbf{c}}_2 \in \mathscr{C}_2$ such that $\hat{\mathbf{c}} = \hat{\mathbf{c}}_1 \,||_1\, \hat{\mathbf{c}}_2$. Thus at least one of the following expressions must hold:

$$\mathrm{supp}\left(\hat{\mathbf{c}}_{1_{|\{1,\ldots,n_1-1\}}}\right) \subsetneq \mathrm{supp}\left(\mathbf{c}_{1_{|\{1,\ldots,n_1-1\}}}\right) \quad \text{or} \quad \mathrm{supp}\left(\hat{\mathbf{c}}_{2_{|\{2,\ldots,n_2\}}}\right) \subsetneq \mathrm{supp}\left(\mathbf{c}_{2_{|\{2,\ldots,n_2\}}}\right)$$

As $c_1^{(n_1)} \neq 0 \neq c_2^{(1)}$ then we deduce that

$$\mathrm{supp}\left(\hat{\mathbf{c}}_1\right) \subsetneq \mathrm{supp}\left(\mathbf{c}_1\right) \quad \text{and/or} \quad \mathrm{supp}\left(\hat{\mathbf{c}}_2\right) \subsetneq \mathrm{supp}\left(\mathbf{c}_2\right)$$

which contradicts the minimality of $\mathbf{c}_1$ in $\mathscr{C}_1$ and / or the minimality of $\mathbf{c}_2$ in $\mathscr{C}_2$.

□

Let us consider first an easy example showing the ideas stated above.

Example 3.44. Let us consider the $[3,2,1]$-code $\mathscr{C}_1$ and the $[4,3,1]$-code $\mathscr{C}_2$ defined over $\mathbb{Z}_5$ with generator matrices:

$$G_{\mathscr{C}_1} = \begin{pmatrix} 0 & 1 & 0 \\ 2 & 3 & 4 \end{pmatrix} \in \mathbb{Z}_5^{2\times 3} \quad \text{and} \quad G_{\mathscr{C}_2} = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \in \mathbb{Z}_5^{3\times 4} \ ,$$

respectively.

If we compute the Graver basis of the ideals associated to these codes we get the set of codewords of minimal support of each code, i.e. $\mathscr{M}_{\mathscr{C}_1}$ and $\mathscr{M}_{\mathscr{C}_2}$, respectively.

$$\mathscr{M}_{\mathscr{C}_1} = \left\{ \begin{array}{ll} (403), & (301), \\ (204), & (102), \\ (010) & \end{array} \right\} \quad \text{and} \quad \mathscr{M}_{\mathscr{C}_2} = \left\{ \begin{array}{llll} (0101), & (0202), & (0303), & (0404), \\ (0010), & & & \\ (4003), & (2004), & (3001), & (1002), \\ (3400), & (1300), & (2100), & (4200) \end{array} \right\}$$

Since $\mathscr{C}_1$ and $\mathscr{C}_2$ satisfy the statement of Definition 3.38, then $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ is well defined and it works out to be the $[5,4,1]$-code with generator matrix

$$G_{\mathscr{C}} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \in \mathbb{Z}_5^{4\times 5}.$$

A Graver basis of the ideal associated to this code gives us the set of 14 codewords of minimal support of $\mathscr{C}$. Following the notation of Proposition 3.43, this set consists of:

- 1 codeword belonging to the set A: $\{(01000)\}$.

- 5 codewords belonging to the set B :

$$\left\{ \begin{array}{ccccc} (00404), & (00303), & (00202), & (00101), & (00010) \end{array} \right\}.$$

- And 8 codewords belonging to the set C:

$$\left\{ \begin{array}{cccc} (40100), & (30200), & (20300), & (10400), \\ (40004), & (30003), & (20002), & (10001) \end{array} \right\}$$

Example 3.45. Take $\mathscr{C}_1$ to be the $[5,3,1]$-code defined over $\mathbb{Z}_3$ with corresponding generator and parity check matrices

$$G_{\mathscr{C}_1} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 2 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{Z}_3^{3\times 5} \quad \text{and} \quad H_{\mathscr{C}_1} = \begin{pmatrix} 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 2 \end{pmatrix} \in \mathbb{Z}_3^{2\times 5},$$

respectively. Therefore, the ideal associated to $\mathscr{C}_1$ is defined as the following binomial ideal:

$$I(\mathscr{C}_1) = \left\langle \left\{ x_2x_4 - 1, x_1x_2^2x_4^2 - 1, x_1x_3x_4x_5 - 1 \right\} \cup \left\{ x_i^3 - 1 \right\}_{i=1}^{5} \right\rangle.$$

The Graver basis of this ideal turns out to be the set of codewords of minimal support of the code $\mathscr{C}_1$ which consists of 18 codewords. Sage [104] on a Mac Os X takes under 3 seconds to compute this algorithm.

Let $\mathscr{C}_2$ be the $[6,3,2]$-code over $\mathbb{Z}_3$ with corresponding generator and parity check matrices

$$G_{\mathscr{C}_2} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 2 & 1 \end{pmatrix} \in \mathbb{Z}_3^{3\times 6} \text{ and } H_{\mathscr{C}_2} = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{Z}_3^{3\times 6},$$

respectively. Hence, the binomial ideal associated to $\mathscr{C}_2$ is defined as:

$$I(\mathscr{C}_2) = \left\langle \left\{ x_1x_5x_6^2 - 1, x_2x_4 - 1, x_3x_4^2x_5^2x_6 - 1 \right\} \cup \left\{ x_i^3 - 1 \right\}_{i=1}^{6} \right\rangle.$$

Sage takes about 0.23 seconds to determine a Graver basis of this ideal which gives the set of 21 codewords of minimal support of the code $\mathscr{C}_2$.

These codes satisfy the statement of Definition 3.38, thus their 1-gluing code $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$ is the $[9,5,1]$-code with generator matrix

$$G_{\mathscr{C}} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 1 \end{pmatrix} \in \mathbb{Z}_3^{5\times 9}.$$

Associated to the code $\mathscr{C}$ we can define the following binomial ideal:

$$I(\mathscr{C}) = \left\langle \left\{ \begin{array}{c} x_2x_4 - 1, x_1x_2^2x_4^2 - 1, x_1x_3x_4x_8^2x_9 - 1, \\ x_5x_7 - 1, x_6x_7^2x_8^2x_9 - 1 \end{array} \right\} \cup \left\{ x_i^3 - 1 \right\}_{i=1}^{9} \right\rangle.$$

In this case Sage takes 21.13 seconds to determine the set of 50 codewords of minimal support of $\mathscr{C}$.

Therefore, we conclude that if we know a decomposition of our original code $\mathscr{C}$ as $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$, then computing the set of codewords of minimal support of the codes $\mathscr{C}_1$ and $\mathscr{C}_2$ and running parallel computations is 7 times faster than computing a complete Graver basis for $\mathscr{C}$.

Remark 3.46. Suppose that, after row operations and column permutations (denoted by π), the generator matrix of an $[n,k]$-code $\mathscr{C}$ is expressible as a matrix of the following form:

$$M = \left(\begin{array}{c|c} A_1 & \mathbf{0} \\ \hline \mathbf{v}_1 & \mathbf{v}_2 \\ \hline \mathbf{0} & A_2 \end{array}\right) \in \mathbb{Z}_q^{k\times n},$$

where the matrix A_i has size $(k_i - 1) \times (n_i - 1)$ and the vector $\mathbf{v}_i$ is an $(n_i - 1)$-tuple of $\mathbb{Z}_q$-elements, for $i = 1, 2$. Then $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_1 \mathscr{C}_2$, where $\mathscr{C}_i$ are the linear codes of parameters $[n_i, k_i]$ defined over the index set $I_i = \{j \in \{1,\dots,n\} \mid \pi(j) \in \{1,\dots,n_i\}\}$ for $i = 1, 2$ and with corresponding generator matrices:

$$G_{\mathscr{C}_1} = \left(\begin{array}{c|c} A_1 & 0 \\ \hline \mathbf{v}_1 & *_1 \end{array}\right) \in \mathbb{Z}_q^{k_1\times n_1} \quad \text{and} \quad G_{\mathscr{C}_2} = \left(\begin{array}{c|c} 0 & A_2 \\ \hline *_2 & \mathbf{v}_2 \end{array}\right) \in \mathbb{Z}_q^{k_2\times n_2},$$

such that $*_1 + *_2 = 0$.

3.4.3 3-gluing of modular codes

Definition 3.47. Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be linear codes over $\mathbb{Z}_q$ of length at least 7 such that $|I_1 \cap I_2| = 3$. We will assume w.l.o.g. that the common indexes corresponds to the last three coordinates of $\mathscr{C}_1$ and the last three coordinates of $\mathscr{C}_2$

If the following conditions are satisfied:

1. $(0,\dots,0,1,1,1) \in \mathscr{M}_{\mathscr{C}_1}$ and all possible 3-bit words appear in the last 3 coordinates of $\mathscr{C}_1$.

2. $(1,1,1,0,\dots,0) \in \mathscr{M}_{\mathscr{C}_2}$ and all possible 3-bit words appear in the first 3 coordinates of $\mathscr{C}_2$.

Then the 3-gluing modular code $\mathscr{C}_1 \textcircled{g}_3 \mathscr{C}_2$ can be defined as the modular code $S(\mathscr{C}_1, \mathscr{C}_2)$ over $\mathbb{Z}_q$.

Analogous to the 1-gluing case we have the following characterization for the set of codewords of the code $\mathscr{C}_1 \textcircled{g}_3 \mathscr{C}_2$.

Lemma 3.48. *$\mathbf{c} \in \mathscr{C}_1 \textcircled{g}_3 \mathscr{C}_2$ if and only if there exist two codewords $\mathbf{c}_1 \in \mathscr{C}_1$ and $\mathbf{c}_2 \in \mathscr{C}_2$ such that $c_1^{(n_1)} + c_2^{(3)} = c_1^{(n_1-1)} + c_2^{(2)} = c_1^{(n_1-2)} + c_2^{(1)} = 0$ and $\mathbf{c} = \mathbf{c}_1 \,||_3\, \mathbf{c}_2$.*

Proof. This proposition is a straight forward generalization of Lemma 3.39. □

In this subsection let $\mathbf{X}$ denote the $n_1 - 3$ variables $\{x_1, \ldots, x_{n_1-3}\}$ and $\mathbf{Y}$ denote the $n_2 - 3$ variables $\{y_4, \ldots, y_{n_2}\}$. We have the following result which states the connection between the 3-gluing of two codes and the sum of its associated ideals.

Proposition 3.49. *$\mathscr{C} = \mathscr{C}_1 \textcircled{g}_3 \mathscr{C}_2$ if and only if*

$$I(\mathscr{C}) = I(\mathscr{C}_{1_{\cdot\{n_1-2,n_1-1,n_1\}}}) + I(\mathscr{C}_{2_{\cdot\{1,2,3\}}}) + \left\langle \mathbf{X}^{\blacktriangle\gamma_x} - \mathbf{Y}^{\blacktriangle\gamma_y}, \mathbf{X}^{\blacktriangle\alpha_x} - \mathbf{Y}^{\blacktriangle\alpha_y}, \mathbf{X}^{\blacktriangle\beta_x} - \mathbf{Y}^{\blacktriangle\beta_y} \right\rangle,$$

where:

- $(\alpha_x \mid 100)$, $(\beta_x \mid 010)$, $(\gamma_x \mid 001) \in \mathscr{C}_1$ *and* $\alpha_x + \beta_x + \gamma_x = 0$.
- $(100 \mid \alpha_y)$, $(010 \mid \beta_y)$, $(001 \mid \gamma_y) \in \mathscr{C}_2$ *and* $\alpha_y + \beta_y + \gamma_y = 0$.

Proof. Note that this Proposition is a natural extension of Proposition 3.40. □

Next proposition is related to the set of codewords of minimal support of a 3-gluing. Similar to the 1-gluing case we restrict our result to linear codes defined over $\mathbb{F}_q$ with q prime, or equivalently, for any code defined on $\mathbb{Z}_q$ with q prime.

Proposition 3.50. *Let $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_3 \mathscr{C}_2$ be a linear code defined over $\mathbb{Z}_q$ with q prime. If $\mathbf{c} \in \mathscr{M}_{\mathscr{C}}$ then $\mathbf{c}$ belongs to one of the following sets:*

- $A = \left\{ (\mathbf{c}_1 \mid \mathbf{0}) \ : \ \mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_{1_{\cdot\{n_1-2,n_1-1,n_1\}}}} \quad \text{and} \quad \mathbf{0} \in \mathbb{Z}_q^{n_2-3} \right\}$.
- $B = \left\{ (\mathbf{0} \mid \mathbf{c}_2) \ : \ \mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_{2_{\cdot\{1,2,3\}}}} \quad \text{and} \quad \mathbf{0} \in \mathbb{Z}_q^{n_1-3} \right\}$.
- $C = \left\{ \mathbf{c}_1 \mid\mid_3 \mathbf{c}_2 \ : \ \mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_1}, \quad \mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_2} \quad \text{and } c_2^{(1)}, c_2^{(2)}, c_2^{(3)} \in \mathbb{Z}_q \setminus \{0\} \right\}$.

Proof. Similarly to the proof in Proposition 3.43, let $G_i \in \mathbb{Z}_q^{k_i \times n_i}$ and $H_i \in \mathbb{Z}_q^{(n_i-k_i)\times n_i}$ be a generator and a parity check matrix, respectively, of the code $\mathscr{C}_i$ with $i = 1, 2$. Using Proposition 3.49 it is easy to check that

$$H = \left(\begin{array}{c|c} \hat{H}_1 & A \\ \hline \mathbf{0} & B \end{array} \right) \in \mathbb{Z}_q^{(n-k)\times n}$$

is a parity check matrix for $\mathscr{C}$, where:

- $\hat{H}_1$ is a submatrix of H_1 obtained by deleting its last three columns, that is $\hat{H}_1$ is a parity check matrix of the code $\mathscr{C}_{1_{\cdot\{n_1-2,n_1-1,n_1\}}}$.
- Similarly $\hat{H}_2$ is a submatrix of H_2 obtained by deleting its first three columns, i.e. $\hat{H}_2$ is a parity check matrix of the code $\mathscr{C}_{2_{\cdot\{1,2,3\}}}$.
- Now let B_1 denote the matrix formed by the rows of H_2 where the first three elements are zero, then B is obtained from B_1 by deleting the first three columns.

- Furthermore, let A_1 be an $(n_1 - k_1) \times n_2$ matrix whose rows are formed by linear combinations of the rows of H_2 in such a way that

$$\begin{pmatrix} H_1^{(n_1-2)} & H_1^{(n_1-1)} & H_1^{(n_1)} \end{pmatrix} = \begin{pmatrix} A_1^{(1)} & A_1^{(2)} & A_1^{(3)} \end{pmatrix}$$

  where $H_1^{(i)}$ denotes the i-th column of H_1 and $A_1^{(j)}$ indicates the j-th column of A_1. Then A is obtained from A_1 by deleting the first three columns.

Note that all rows of $\hat{H}_2$ are present in some way in the matrix $\left(\frac{A}{B}\right)$.

Let E_1 and E_2 be the index set of the columns of $\hat{H}_1$ and $H \setminus \hat{H}_1$ respectively. We consider any codeword $\mathbf{c}$ of the set $\mathscr{M}_{\mathscr{C}}$.

- If $\mathrm{supp}(\mathbf{c}) \cap E_2 = \emptyset$ then there exists a codeword $\mathbf{c}_1$ belonging to $\mathscr{M}_{\mathscr{C}_{1,\{n_1-2,n_1-1,n_1\}}}$ such that $\mathbf{c} = (\mathbf{c}_1 \mid \mathbf{0})$. Since $(1110\ldots0) \in \mathscr{M}_{\mathscr{C}_2}$, the other possibility we have is when

$$\mathbf{c}_1 = (\hat{\mathbf{c}}_1 \mid 000) + (0\ldots0111)$$

  with $(\hat{\mathbf{c}}_1 \mid 000) \in \mathscr{C}_1$, but notice that $\mathbf{c}_1 \,||_3\, (1110\ldots0) = (\hat{\mathbf{c}}_1 \mid \mathbf{0})$ where $\hat{\mathbf{c}}_1$ must belong to $\mathscr{M}_{\mathscr{C}_{1,\{n_1-2,n_1-1,n_1\}}}$, otherwise we obtain a contradiction to the minimality of $\mathbf{c}$.

- Equivalent, for the case $\mathrm{supp}(\mathbf{c}) \cap E_1 = \emptyset$, there must exist a codeword $\mathbf{c}_2$ belonging to $\mathscr{M}_{\mathscr{C}_{2,\{1,2,3\}}}$ such that $\mathbf{c} = (\mathbf{0} \mid \mathbf{c}_2)$.

- If $\mathrm{supp}(\mathbf{c})$ intersects both E_1 and E_2, say $\overline{E_1}$ and $\overline{E_2}$, then we know by matroid theory that the set of columns indexed by $\hat{E}_1 \cup \hat{E}_2$ is minimally dependent. That is to say, the set of columns of $\hat{H}_1$ indexed by $\hat{E}_1$ are linearly independent, so any codeword of $\mathscr{C}_1$ with support

$$\left\{ \overline{E_1} \cup \{n_1 - 2\},\quad \overline{E_1} \cup \{n_1 - 1\},\quad \overline{E_1} \cup \{n_1\} \right\}$$

  is a codeword of the set $\mathscr{M}_{\mathscr{C}_1}$. Likewise any codeword of $\mathscr{C}_2$ with support

$$\left\{ \{1\} \cup \overline{E_2},\quad \{2\} \cup \overline{E_2},\quad \{3\} \cup \overline{E_2} \right\}$$

  is a codeword of the set $\mathscr{M}_{\mathscr{C}_2}$.

  Since $\mathbf{c} \in \mathscr{C} = \mathscr{C}_1 \textcircled{g}_3 \mathscr{C}_2$, by Lemma 3.48 we know that there exist $\mathbf{c}_1 \in \mathscr{C}_1$ and $\mathbf{c}_2 \in \mathscr{C}_2$ such that $\mathbf{c} = \mathbf{c}_1 \,||_3\, \mathbf{c}_2$, but this decomposition is not unique in general. We can distinguish two cases:

  1. If there exists a codeword $\hat{\mathbf{c}}_1 \in \mathscr{C}_1$ of the form

$$\left(\mathbf{c}_{|E_1} \mid *00\right), \quad \left(\mathbf{c}_{|E_1} \mid 0*0\right) \quad \text{or} \quad \left(\mathbf{c}_{|E_1} \mid 00*\right),$$

     then we have that $\hat{\mathbf{c}}_1 \in \mathscr{M}_{\mathscr{C}_1}$. W.l.o.g. we can assume that $\hat{\mathbf{c}}_1 = \left(\mathbf{c}_{|E_1} \mid *00\right)$. If there is not a codeword $\hat{\mathbf{c}}_2 \in \mathscr{C}_2$ of the form $\left(-*00 \mid \mathbf{c}_{|E_2}\right)$ such that $\mathbf{c} = \hat{\mathbf{c}}_1 \,||_3\, \hat{\mathbf{c}}_2$ then we would have that a codeword of the form

$$\mathbf{c}_1 - \hat{\mathbf{c}}_1 - (*_1 - *)(\mathbf{0} \mid 111) = (\mathbf{0} \mid 0, *_2 - *_1 + *, *_3 - *_1 + *)$$

where $(*_1, *_2, *_3)$ are the last three elements of $\mathbf{c}_1$, must belongs to the code $\mathscr{C}_1$ which contradicts the minimality of the codeword $(0\ldots0111)$ in $\mathscr{C}_1$ except for the case $*_2 = *_3 = *_1 - *$. However, in this special case we would have that

$$\mathbf{c}_2 + *_2(111 \mid \mathbf{0}) = \left(-*_1 + *_2, 0, 0 \mid \mathbf{c}_{|E_2}\right)$$

is a codeword of the code $\mathscr{C}_2$ which contradicts the last hypothesis we made.

2. Otherwise we would have that the set of columns of $\hat{H}_1$ indexed by

$$\overline{E_1} \cup \{n_1 - 2\}, \quad \overline{E_1} \cup \{n_1 - 1\} \quad \text{and} \quad \overline{E_1} \cup \{n_1\}$$

are linearly independent. So any codeword of $\mathscr{C}_1$ with support

$$\overline{E_1} \cup \{n_1 - 2, n_1 - 1\}, \quad \overline{E_1} \cup \{n_1 - 2, n_1\} \quad \text{and} \quad \overline{E_1} \cup \{n_1 - 1, n_1\}$$

is a codeword of the set $\mathscr{M}_{\mathscr{C}_1}$. Moreover, we can always define a codeword with this characteristics since $\alpha_x + \beta_x + \gamma_x = 0$ then $\mathbf{c}_{|E_1}$ can be expressed uniquely as

$$\begin{aligned} \mathbf{c}_{|E_1} &= \hat{\mathbf{c}}_1 + \lambda_1\alpha_x + \lambda_2\beta_x + \lambda_3\gamma_x \\ &= \hat{\mathbf{c}}_1 + (\lambda_1 - \lambda_3)\alpha_x + (\lambda_2 - \lambda_3)\beta_x \end{aligned}$$

with $\hat{\mathbf{c}}_1 \in \mathscr{C}_{1_{\cdot\{n_1-2, n_1-1, n_1\}}}$ and similarly for the vector $\mathbf{c}_{|E_2}$.

In both cases we find $\mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_1}$ and $\mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_2}$ such that $\mathbf{c} = \mathbf{c}_1 \parallel_3 \mathbf{c}_2$.

□

Remark 3.51. The converse is not true in general. In the following example we will see that it is not always true that if $\mathbf{c}_1 \in \mathscr{M}_{\mathscr{C}_1}$ and $\mathbf{c}_2 \in \mathscr{M}_{\mathscr{C}_2}$ with

$$c_1^{(n_1-2)} + c_2^{(1)} = c_1^{(n_1-1)} + c_2^{(2)} = c_1^{(n_1)} + c_2^{(3)}$$

then $\mathbf{c} = \mathbf{c}_1 \parallel_3 \mathbf{c}_2$ belongs to $\mathscr{M}_{\mathscr{C}}$.

Example 3.52. Let $\mathscr{C}_1$ be the binary linear $[7,4]$-code with generator matrix

$$G_1 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \in \mathbb{Z}_2^{4\times 7}.$$

Associated to the code $\mathscr{C}_1$ we can define the following binomial ideal:

$$I(\mathscr{C}_1) = \left\langle \{x_2x_4 - 1, x_1x_3x_5 - 1, x_1x_7 - 1, x_3x_6 - 1\} \cup \left\{x_i^2 - 1\right\}_{i=1}^{7} \right\rangle.$$

The Graver basis of this ideal represents the set of codewords of minimal support for the code $\mathscr{C}_1$, which turns out to be:

$$\mathscr{M}_{\mathscr{C}_1} = \left\{ \begin{array}{c} (0101000), (1000001), (0010010), (1010100), \\ (1000110), (0010101), (0000111) \end{array} \right\}.$$

Now let $\mathscr{C}_2$ be the binary linear $[7,4]$-code with generator matrix

$$G_2 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \in \mathbb{Z}_2^{4\times 7}.$$

Associated to the code $\mathscr{C}_2$ we can define the following binomial ideal:

$$I(\mathscr{C}_2) = \left\langle \{x_1x_4x_6 - 1, x_2x_4 - 1, x_3x_6 - 1, x_5x_7 - 1\} \cup \left\{x_i^2 - 1\right\}_{i=1}^{7} \right\rangle.$$

The Graver basis of this ideal represents the set of codewords of minimal support for the code $\mathscr{C}_2$, which turns out to be:

$$\mathscr{M}_{\mathscr{C}_2} = \left\{ \begin{array}{c} (0000101), (1001010), (0101000), (0010010), \\ (1100010), (1011000), (1110000) \end{array} \right\}.$$

Since $\mathscr{C}_1$ and $\mathscr{C}_2$ verify the requirement of Definition 3.47, then their 3-gluing code can be constructed. The code $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_3 \mathscr{C}_2$ is the binary linear $[8,4]$ code with generator matrix:

$$G = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \in \mathbb{Z}_2^{4\times 8}.$$

Associated to the code $\mathscr{C}$ we can define the following binomial ideal:

$$I(\mathscr{C}) = \left\langle \{x_2x_4 - 1, x_3x_5x_7 - 1, x_1x_7 - 1, x_6x_8 - 1\} \cup \left\{x_i^2 - 1\right\}_{i=1}^{8} \right\rangle$$

The Graver basis of this ideal represents the set of codewords of minimal support for the code $\mathscr{C}$, which is:

$$\mathscr{M}_{\mathscr{C}} = \{(01010000), (00101000), (10000010), (00000101)\}.$$

Note that $\mathbf{c}_1 = (1010100)$ and $\mathbf{c}_2 = (1001010)$ belongs to $\mathscr{M}_{\mathscr{C}_1}$ and $\mathscr{M}_{\mathscr{C}_2}$, respectively, but $\mathbf{c} = \mathbf{c}_1 \,||_3\, \mathbf{c}_2 = (10101010)$ does not belongs $\mathscr{M}_{\mathscr{C}}$.

3.4.4 General case

In the general case, we have the following characterization for the set of codewords of the code $\mathscr{C}_1 \textcircled{g}_m \mathscr{C}_2$.

Lemma 3.53. $\mathbf{c} \in \mathscr{C}_1 \textcircled{g}_m \mathscr{C}_2$ *if and only if there exist two codewords* $\mathbf{c}_1 \in \mathscr{C}_1$ *and* $\mathbf{c}_2 \in \mathscr{C}_2$ *such that*

$$c_1^{(n_1)} + c_2^{(m)} = c_1^{(n_1-1)} + c_2^{(m-1)} = \ldots = c_1^{(n_1-m+1)} + c_2^{(1)} = 0$$

and $\mathbf{c} = \mathbf{c}_1 \,||_m\, \mathbf{c}_2$.

Proof. This proposition is a straight forward generalization of Lemma 3.39. □

In this section let $\mathbf{X}$ denotes the $n_1 - m$ variables $\{x_1, \ldots, x_{n_1-m}\}$ and $\mathbf{Y}$ denotes the $n_2 - m$ variables $\{y_{m+1}, \ldots, y_{n_2}\}$. We have the following result which states the connection between the m-gluing of two codes and the sum of its associated ideals.

Proposition 3.54. $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_m \mathscr{C}_2$ *if and only if*

$$I(\mathscr{C}) = I(\mathscr{C}_{1.\{n_1-m+1,\ldots,n_1\}}) + I(\mathscr{C}_{2.\{1,2,\ldots,m\}}) + \left\langle \mathbf{X}^{\blacktriangle \alpha_1} - \mathbf{Y}^{\blacktriangle \beta_1}, \ldots, \mathbf{X}^{\blacktriangle \alpha_m} - \mathbf{Y}^{\blacktriangle \beta_m} \right\rangle,$$

with:

- $(\alpha_1 \mid \mathbf{e}_1), \ldots, (\alpha_m \mid \mathbf{e}_m) \in \mathscr{C}_1$ *and* $\alpha_1 + \ldots + \alpha_m = 0$.
- $(\mathbf{e}_1 \mid \beta_1), \ldots, (\mathbf{e}_m \mid \beta_m) \in \mathscr{C}_2$ *and* $\beta_1 + \ldots + \beta_m = 0$.

Where $\{\mathbf{e}_1, \ldots, \mathbf{e}_m\}$ *denotes the canonical basis of* $\mathbb{Z}_q^m$.

Proof. Note that this Proposition is a natural extension to Proposition 3.40. □

Note that we have investigated the problem of reducing the complexity of computing the set of codewords of minimal support for an arbitrary code over $\mathbb{Z}_q$.

If the decomposition of a code $\mathscr{C}$ defined over $\mathbb{F}_q$, with q prime, as an m-gluing of smaller codes with $m \leq 3$ is known, then by definition the linear codes that appear in the decomposition have smaller length than $\mathscr{C}$ by Remark 3.29 and the computations may be carried out in parallel. Thus we have found an effective method to achieve our goal since the complexity of the problem is reduced from $\mathscr{O}(n^2 2^{n-k})$ to $\mathscr{O}(m^2 2^{m-k'})$ where $m = \max\{n_1, n_2\} < n$ and $k' \leq k$. The idea for future work is try to devise a test for the m-gluing condition whose complexity is competitive with the computation of the whole Gröbner basis.

However, there are still open problems related to our goals. Firstly, the decomposition theory has been presented in the context of modular codes. In particular, we obtained interesting results for linear codes defined on finite fields $\mathbb{F}_q$ with q prime. However, it seems that if we can define a Gröbner test set for codes over an arbitrary finite field (indeed we achieve to do it in Chapter 4) then it would be possible to develop a general decomposition theory for these codes analogous to that for prime finite fields exposed in this chapter.

Furthermore, we would like to know an efficient algorithm that produces, if possible, a decomposition of any linear code as the m-gluing of codes of smaller length. We suspect that such a decomposition can be obtained in polynomial time. The test to determine if an $[n,k]$-code $\mathscr{C}$ is expressible as an m-gluing of two codes $\mathscr{C}_1$ and $\mathscr{C}_2$, for $m \geq 3$, is equivalent to finding a generator matrix of the studied code which, after row operations and column permutation (denoted by π), is expressible as a matrix of the following form:

$$M = \left(\begin{array}{c|c} A_1 & \mathbf{0} \\ \hline B_1 & B_2 \\ \hline \mathbf{0} & A_2 \end{array}\right) \in \mathbb{Z}_q^{k\times n},$$

where the rank of B_i is $m-1$, for $i = 1,2$. Then $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_m \mathscr{C}_2$ where $\mathscr{C}_i$ are linear codes of parameters $[n_i, k_i]$ defined over the index set

$$I_i = \{j \in \{1,\dots,n\} \mid \pi(j) \in \{1,\dots,n_i\}\} \quad \text{for } i = 1,2,$$

with generator matrices

$$G_{\mathscr{C}_1} = \left(\begin{array}{c|c} A_1 & \mathbf{0} \\ \hline B_1 & *_1 I_m \end{array}\right) \in \mathbb{Z}_q^{k_1\times n_1} \quad \text{and} \quad G_{\mathscr{C}_2} = \left(\begin{array}{c|c} *_2 I_m & B_2 \\ \hline \mathbf{0} & A_2 \end{array}\right) \in \mathbb{Z}_q^{k_2\times n_2}$$

respectively, such that $*_1 + *_2 = 0$ and I_m denotes the identity matrix of size m. Similar tests for $m = 0$ and $m = 1$ are defined in Remark 3.36 and Remark 3.46, respectively.

Finally, another open problem is to obtain the set of codewords of minimal support for codes $\mathscr{C}$ from smaller codes $\mathscr{C}_1$ and $\mathscr{C}_2$ when $\mathscr{C} = \mathscr{C}_1 \textcircled{g}_m \mathscr{C}_2$ with $m > 3$.

4

Linear codes: Applications

Contents

Throughout this chapter $\mathscr{C}$ will be an $[n,k]$ linear code defined over a finite field $\mathbb{F}_q$. In this chapter we associate a binomial ideal to any arbitrary linear code given by the rows of a generator matrix and the relations given by the additive table of the defining field. The binomials involved in the reduced Gröbner basis of such ideal w.r.t. a degree ordering induce a uniquely defined test-set for the code which allows the description of an algebraic decoding algorithm. While the binomials involved in the Graver basis provides a universal test-set which turns out to be a set containing the

set of codewords of minimal support of the code. This chapter yields a generalization of [11, 14, 74] where these ideas were stated just for the binary case or for modular codes.

In the following, we present the structure of the chapter. First, in Section 4.1, we describe the structure of the binomial ideal $I_+(\mathscr{C})$ associated to an arbitrary linear code $\mathscr{C}$, which we prove that it can be described by a finite set of generators provided by a basis of the subspace $\mathscr{C}$ and the binomials attached to the additive table of the base field $\mathbb{F}_q$. Moreover, we show that the above ideal is also generated by the set of binomials associated with the vectors that belong to the $\mathbb{F}_q$-kernel of a matrix, which is the kernel of a ring homomorphism given by an elimination ideal. Note that this idea is an extension of that proposed by Ikegami and Kaji [60] to solve linear integer programming with modulo arithmetic conditions. Actually it is a generalization of the works [11, 74] to the non-modular case.

In Section 4.2 we present a structure associated to the class of linear codes called a Gröbner representation which will be an essential structure in the decoding process.

In Section 4.3, we show that the reduced Gröbner basis of $I_+(\mathscr{C})$ relative to a degree compatible ordering allows us a complete decoding algorithm that has some resemblances with the two gradient descent decoding algorithms known for binary codes. See Section 2.4 and the references given there. In this chapter, the test-set of a code is replaced by the Gröbner basis and addition is replaced by the reduction induced by the basis. The idea behind both algorithms can be stated in a Gröbner basis free theory *Step by Step decoding*, which is an old but quite recurrent technique in Coding Theory. A primer study on it can be found in [94].

Next, in Section 4.4, we discuss an alternative for the computation of the Gröbner basis of $I_+(\mathscr{C})$ w.r.t. a degree compatible ordering. This algorithm is based on the computation of the syzygy module of the ideal $I_+(\mathscr{C})$. It is an adaptation of the FGLM techniques presented in [11] for the binary case. A brief description of this technique as well as a complexity estimation can be found here. However we can not expect that the algorithm behaves in polynomial time since it will also do the complete decoding algorithm for general linear codes which is an NP-hard problem. But the proposed algorithm is better suited than the standard Buchberger algorithm for computing Gröbner basis.

In Section 4.5, we consider the Graver basis associated to $I_+(\mathscr{C})$ which turns out to be a set containing the set of codewords of minimal support of $\mathscr{C}$.

Finally, in Section 4.6 we apply the results obtained in this chapter to other classes of codes such as modular codes, codes defined over multiple alphabets or additive codes. Recall that modular codes were already discussed in Chapter 3 but this new approach allows the computation of a Gröbner test-set.

4.1 The ideal associated to any linear code

As usual by $\mathbb{K}$, $\mathbb{Z}$, $\mathbb{Z}_s$ and $\mathbb{F}_q$, where q is a prime power, we denote an arbitrary finite field, the ring of integers, the ring of integers modulo s and any representation of a finite field with q elements, respectively. For every finite field $\mathbb{F}_q$ the multiplicative

group $\mathbb{F}_q^*$ of nonzero elements of $\mathbb{F}_q$ is cyclic. A generator of the cyclic group $\mathbb{F}_q^*$ is called a primitive element of $\mathbb{F}_q$. Therefore, $\mathbb{F}_q$ consist of 0 and appropriate powers of that primitive element (see for instance [99]).

Let α be a primitive element of $\mathbb{F}_q$ and $\left\{\mathbf{e}_1,\ldots,\mathbf{e}_{q-1}\right\}$ be the canonical basis of $\mathbb{Z}^{q-1}$. We will use the following characteristic crossing functions:

$$\nabla:\ \{0,1\}^{q-1}\ \longrightarrow\ \mathbb{F}_q\ ,\qquad \Delta:\ \mathbb{F}_q\ \longrightarrow\ \{0,1\}^{q-1}$$

The map Δ replaces the class of the elements $\mathbf{a}=\alpha^j\in\mathbb{F}_q^*$ by the vector $\mathbf{e}_j$ and $0\in\mathbb{F}_q$ by the zero vector $\mathbf{0}\in\mathbb{Z}^{q-1}$. Whereas the map ∇ recovers the element $j_1\alpha+\ldots+j_{q-1}\alpha^{q-1}=j_1\alpha+\ldots+j_{q-2}\alpha^{q-2}+j_{q-1}$ of $\mathbb{F}_q$ from the $(q-1)$-tuple of binary elements $\left(j_1,\ldots,j_{q-1}\right)$. These maps will be used with matrices and vectors acting coordinate-wise.

Remark 4.1. Take into account that $\mathbb{F}_q$ contains $\phi(q-1)$ primitive elements, where ϕ is Euler's function (i.e. the number of integers less than and relative prime to $q-1$). Recall that, if the integer n has the prime factorization $n=p_1^{r_1}\ldots p_s^{r_s}$, then

$$\phi(n)\ =\ n\left(1-\tfrac{1}{p_1}\right)\ldots\left(1-\tfrac{1}{p_s}\right)\ =\ (p_1-1)\,p_1^{r_1-1}\ldots(p_s-1)\,p_s^{r_s-1}\ .$$

Every primitive element of $\mathbb{F}_q$ can serve as a defining element of the characteristic crossing functions ∇ and Δ. The difference lies in that the elements of $\mathbb{F}_q$ which represents the components x_{ij} appear in a different component of the vector variable X_i, depending on the chosen primitive element.

Let $\mathbf{X}$ denotes n vector variables $X_1,\ldots,X_n$ such that each variable X_i can be decomposed into $q-1$ components $x_{i1},\ldots,x_{iq-1}$ with $i=1,\ldots,n$. Let $\mathbf{a}=(a_1,\ldots,a_n)$ be an n-tuple of elements of the field $\mathbb{F}_q$. We will adopt the following notation:

$$\mathbf{X}^{\mathbf{a}}=X_1^{a_1}\cdots X_n^{a_n}=\left(x_{11}\cdots x_{1q-1}\right)^{\Delta a_1}\cdots\left(x_{n1}\cdots x_{nq-1}\right)^{\Delta a_n}.$$

This relationship allows us to work with monomials whose exponents are form by elements defined over the field $\mathbb{F}_q$ as monomials with integer exponents, more precisely $\{0,1\}$-exponents. Note that the degree of the monomial $\mathbf{X}^{\mathbf{a}}$ is defined as the support of the vector $\Delta\mathbf{a}$ view as an $n(q-1)$-tuple of zeros and ones.

Unless otherwise stated, we simply write $\mathscr{C}$ for an $[n,k]$ linear code defined over the finite field $\mathbb{F}_q$. We define the *ideal associated* to $\mathscr{C}$ as the binomial ideal:

$$I_+(\mathscr{C})=\left\langle\left\{\mathbf{X}^{\mathbf{a}}-\mathbf{X}^{\mathbf{b}}\mid\mathbf{a}-\mathbf{b}\in\mathscr{C}\right\}\right\rangle\subseteq\mathbb{K}[\mathbf{X}].\tag{4.1}$$

Given the rows of a generator matrix of $\mathscr{C}$, labelled by $\mathbf{w}_1,\ldots,\mathbf{w}_k$, we define the following ideal:

$$\Delta I\ =\ \left\langle\ \left\{\mathbf{X}^{(\alpha^j\mathbf{w}_i)}-1\right\}_{\substack{i=1,\ldots,k\\ j=1,\ldots,q-1}}\ \bigcup\ \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n}\ \right\rangle\subseteq\mathbb{K}[\mathbf{X}],\tag{4.2}$$

where $\mathscr{R}_{X_i}(T_+)$ consists of all the binomials on the variable X_i associated to the relations given by the additive table of the field $\mathbb{F}_q = \langle\alpha\rangle$, i.e.,

$$\mathscr{R}_{X_i}(T_+) = \left\{ \{x_{iu}x_{iv} - x_{iw} \mid \alpha^u + \alpha^v = \alpha^w\} \bigcup \{x_{iu}x_{iv} - 1 \mid \alpha^u + \alpha^v = 0\} \right\},$$

with $i = 1, \ldots, n$.

Remark 4.2. There are as many different binomials in $\mathscr{R}_{X_i}(T_+)$ as 2-combinations with repetitions from the set of variables $\left\{x_{i1}, \ldots, x_{iq-1}\right\}$, i.e. we have $\binom{q}{2}$ different binomials.

Theorem 4.3. $I_+(\mathscr{C}) = \Delta I$.

Proof. It is clear that $\Delta I \subseteq I_+(\mathscr{C})$ since all binomials in the generating set of ΔI belong to $I_+(\mathscr{C})$. Note that the binomials from $\mathscr{R}_{X_i}(T_+)$ poses no problem because they are multiples of the binomial $\mathbf{X^0} - 1$ which fit in $I_+(\mathscr{C})$.

To show the converse it suffices to show that each binomial $\mathbf{X^a} - \mathbf{X^b}$ of $I_+(\mathscr{C})$ belongs to ΔI. By the definition of $I_+(\mathscr{C})$ we have that $\mathbf{a} - \mathbf{b} \in \mathscr{C}$. Hence

$$\mathbf{a} - \mathbf{b} = \lambda_1\mathbf{w}_1 + \ldots + \lambda_k\mathbf{w}_k \text{ with } \lambda_1, \ldots, \lambda_k \in \mathbb{F}_q.$$

Note that, if the binomials $\mathbf{z}_1 - 1$ and $\mathbf{z}_2 - 1$ belongs to the ideal ΔI then $\mathbf{z}_1\mathbf{z}_2 - 1 = (\mathbf{z}_1 - 1)\mathbf{z}_2 + (\mathbf{z}_2 - 1)$ also belongs to ΔI. On account of the previous line, we have:

$$\begin{aligned}
\mathbf{X}^{(\mathbf{a}-\mathbf{b})} - 1 &= \left(\mathbf{X}^{(\lambda_1\mathbf{w}_1)} - 1\right)\prod_{i=2}^{k}\mathbf{X}^{(\lambda_i\mathbf{w}_i)} + \left(\prod_{i=2}^{k}\mathbf{X}^{(\lambda_i\mathbf{w}_i)} - 1\right) \\
&= \left(\mathbf{X}^{(\lambda_1\mathbf{w}_1)} - 1\right)\prod_{i=2}^{k}\mathbf{X}^{(\lambda_i\mathbf{w}_i)} + \left(\mathbf{X}^{(\lambda_2\mathbf{w}_2)} - 1\right)\prod_{i=3}^{k}\mathbf{X}^{(\lambda_i\mathbf{w}_i)} + \ldots + \\
&+ \left(\mathbf{X}^{(\lambda_{k-1}\mathbf{w}_{k-1})} - 1\right)\mathbf{X}^{(\lambda_k\mathbf{w}_k)} + \left(\mathbf{X}^{(\lambda_k\mathbf{w}_k)} - 1\right).
\end{aligned}$$

If at least one λ_i is nonzero with $i = 1, \ldots, k$, then the last equation forces that

$$\mathbf{X}^{(\mathbf{a}-\mathbf{b})} - 1 \in \left\langle \left\{\mathbf{X}^{(\alpha^j\mathbf{w}_i)} - 1\right\}_{\substack{i=1,\ldots,k \\ j=1,\ldots,q-1}} \right\rangle.$$

Otherwise $\mathbf{a} - \mathbf{b} = \mathbf{0}$, and thus,

$$\mathbf{X}^{(\mathbf{a}-\mathbf{b})} - 1 \in \left\langle \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n} \right\rangle.$$

We have actually proved that $\mathbf{X^a} - \mathbf{X^b} = \left(\mathbf{X}^{(\mathbf{a}-\mathbf{b})} - 1\right)\mathbf{X^b} \in \Delta I$, which completes the proof. □

Remark 4.4. Let $B \in \mathbb{F}_q^{m\times n}$ be a matrix, $B^\perp$ be the matrix whose rows generate the null-space of B and $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ be a set of generators of the row space of the matrix B. We can define the following binomial ideal:

$$I(B) = \left\langle \left\{\mathbf{X^u} - \mathbf{X^v} \mid B^\perp(\mathbf{u} - \mathbf{v})^T = 0 \text{ in } \mathbb{F}_q\right\} \right\rangle.$$

Therefore, the construction presented above for linear codes can be generalized for any matrix defined over an arbitrary finite field, i.e. we have actually proved that $I(B) = \Delta I$.

Let B be a $m \times n$ matrix defined over $\mathbb{F}_q$. Let $\mathbf{X}$ denotes n vector variables $X_1, \ldots, X_n$ such that $X_i = \left(x_{i1} \cdots x_{iq-1}\right)$ for $1 \leq i \leq n$ and $\mathbf{Y}$ denotes m vector variables $Y_1, \ldots, Y_m$ such that $Y_j = \left(y_{j1} \cdots y_{jq-1}\right)$ for $1 \leq j \leq m$. For a vector $\mathbf{u} \in \mathbb{F}_q^n$ we define $\theta(\mathbf{u}) = B\mathbf{u}^T \in \mathbb{F}_q^m$. The ring homomorphism

$$\Theta: \quad \mathbb{K}[\mathbf{X}] \quad \longrightarrow \quad \mathbb{K}[\mathbf{Y}]$$

is then defined by $\Theta(\mathbf{X}^{\mathbf{u}}) = \mathbf{Y}^{\theta(\mathbf{u})} = \mathbf{Y}^{B\mathbf{u}^T}$. More generally, for every polynomial $f = \sum c_{\mathbf{v}} X^{\mathbf{v}}$ of the ring of polynomials $\mathbb{K}[\mathbf{X}]$ we have that

$$\Theta(f) = f\left(\Theta(X_1), \ldots, \Theta(X_n)\right) = \sum c_{\mathbf{v}} \mathbf{Y}^{\theta(\mathbf{v})}.$$

Let $\mathscr{R}_{Y_j}\left(T_+\right) \subseteq \mathbb{K}[Y_j]$ be the binomial ideal consisting of all the binomials on the variables $\left\{y_{j1}, \ldots, y_{jq-1}\right\}$ associated to the relations given by the additive table of the field $\mathbb{F}_q = \langle \alpha \rangle$ with $1 \leq j \leq m$ and let

$$\mathscr{R}_{\mathbf{Y}}\left(T_+\right) = \left\langle \bigcup_{i=1}^{m} \mathscr{R}_{Y_i}\left(T_+\right) \right\rangle$$

be a binomial ideal in the polynomial ring $\mathbb{K}[\mathbf{Y}]$. We have the following result:

Lemma 4.5. *Let us consider the matrix $B \in \mathbb{F}_q^{m \times n}$ and the vectors $\mathbf{u} \in \mathbb{F}_q^n$ and $\mathbf{b} \in \mathbb{F}_q^m$. $B\mathbf{u}^T = \mathbf{b}$ in $\mathbb{F}_q$ if and only if $\Theta(\mathbf{X}^{\mathbf{u}}) = \mathbf{Y}^{\mathbf{b}}$ modulo the ideal $\mathscr{R}_{\mathbf{Y}}\left(T_+\right)$.*

Proof. This Lemma is a generalization of [60, Lemma 1]. Let $\mathbf{b} = B\mathbf{u}^T$ in $\mathbb{F}_q$ then $b_i = \sum_{j=1}^{n} b_{ij} u_j$ in $\mathbb{F}_q$ for $1 \leq i \leq m$, where b_{ij} denotes the element of matrix B in row i and column j . Let α be a primitive element of $\mathbb{F}_q$. We distinguish two cases.

- Case 1: $b_i = 0 \notin \mathbb{F}_q^* = \langle \alpha \rangle$. Then it is clear that $\sum_{j=1}^{n} b_{ij} u_j = 0$, and thus

$$Y_i^{\Delta b_i} = 1 = Y_i^{\Delta \sum_{j=1}^{n} b_{ij} u_j}.$$

- Case 2: $b_i \neq 0$. Then we have that $b_i = \alpha^{s_i}$ and $\sum_{j=1}^{n} b_{ij} u_j = \alpha^{r_i}$ with $\alpha^{s_i} - \alpha^{r_i} = \alpha^{s_i} + \alpha^{\hat{r}_i} = 0$ and $\alpha^{r_i} + \alpha^{\hat{r}_i} = 0$ for $1 \leq i \leq m$, or equivalently $y_{i,s_i} y_{i,\hat{r}_i} - 1$, $y_{i,r_i} y_{i,\hat{r}_i} - 1 \in \mathscr{R}_{Y_i}\left(T_+\right)$, i.e.

$$\left(y_{i,s_i} y_{i,\hat{r}_i} - 1\right) y_{i,r_i} = y_{i,s_i} - y_{i,r_i} \in \mathscr{R}_{Y_i}\left(T_+\right).$$

  Hence $Y_i^{\Delta \sum_{j=1}^{n} b_{ij} u_j} - Y_i^{\Delta b_i} \in \mathscr{R}_{Y_i}\left(T_+\right)$.

Thus, $\Theta(\mathbf{X}^{\mathbf{u}}) = \mathbf{Y}^{\mathbf{b}}$ modulo the ideal $\mathscr{R}_{\mathbf{Y}}\left(T_+\right)$. The converse inclusion is proved by reading the above backwards. □

Another ideal associated to the matrix $B \in \mathbb{F}_q^{m\times n}$ is defined by

$$I_B = \left\langle \ \{\Theta(X_i) - X_i\}_{i=1,\ldots,n} \quad \cup \quad \left\{\mathscr{R}_{Y_j}\left(T_+\right)\right\}_{j=1,\ldots,m} \ \right\rangle \subseteq \mathbb{K}[\mathbf{X},\mathbf{Y}]. \tag{4.3}$$

Lemma 4.6. *$f \in I_B \cap \mathbb{K}[\mathbf{X}]$ if and only if $f \in \mathbb{K}[\mathbf{X}]$ and $\Theta(f) \equiv 0 \mod \mathscr{R}_{\mathbf{Y}}\left(T_+\right)$.*

Proof. By Remark 4.2, for each $j = 1,\ldots,m$ we have $\binom{q}{2}$ different binomials in $\mathscr{R}_{Y_j}\left(T_+\right)$. We will denote by $r_{j,l}(Y_j)$ the polynomial at position l with respect to certain order in $\mathscr{R}_{Y_j}\left(T_+\right)$ with $j = 1,\ldots,m$.

Let $f \in I_B$, by representing f with the generators of I_B, we have that

$$f(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{n} \lambda_i \left(\Theta(X_i) - X_i\right) + \sum_{j=1}^{m}\sum_{l=1}^{\binom{q}{2}} \beta_{j,l} r_{j,l}(Y_j)$$

with $\quad \{\lambda_i\}_{i=1,\ldots,n} \quad$ and $\quad \left\{\beta_{j,l}\right\}_{\substack{j=1,\ldots,m \\ l=1,\ldots,\binom{q}{2}}} \quad \in \mathbb{K}[\mathbf{X},\mathbf{Y}]$.

Then

$$\begin{aligned}
\Theta(f) &= f(\Theta(X_1),\ldots,\Theta(X_n),Y_1,\ldots,Y_m) \\
&= \sum_{i=1}^{n} \Theta(\lambda_i)\left(\Theta(X_i) - \Theta(X_i)\right) + \sum_{j=1}^{m}\sum_{l=1}^{\binom{q}{2}} \Theta(\beta_j) r_{j,l}(Y_j) \\
&= \sum_{j=1}^{m}\sum_{l=1}^{\binom{q}{2}} \Theta(\beta_j) r_{j,l}(Y_j) \equiv 0 \mod \mathscr{R}_{\mathbf{Y}}\left(T_+\right).
\end{aligned}$$

To prove the converse, first note that given any vector $\mathbf{u} = (u_1,\ldots,u_n) \in \mathbb{F}_q^n$ the monomial $\mathbf{X}^{\mathbf{u}}$ can be written, for some $B_1,\ldots,B_n \in \mathbb{K}[\mathbf{X},\mathbf{Y}]$, as:

$$\begin{aligned}
X_1^{u_1}\cdots X_n^{u_n} &= (\Theta(X_1) + (X_1 - \Theta(X_1)))^{u_1}\cdots(\Theta(X_n) + (X_n - \Theta(X_n)))^{u_n} \\
&= \Theta(X_1)^{u_1}\cdots\Theta(X_n)^{u_n} + B_1(X_1 - \Theta(X_1)) + \ldots + B_n(X_n - \Theta(X_n)).
\end{aligned}$$

Hence, for all polynomial $f \in \mathbb{K}[\mathbf{X}]$ there exists polynomials $C_1,\ldots,C_n \in \mathbb{K}[\mathbf{X},\mathbf{Y}]$ such that $f(\mathbf{X}) = f(\Theta(X_1),\ldots,\Theta(X_n)) + \sum_{i=1}^{n} C_i(X_i - \Theta(X_i))$. Moreover, from the initial assumption we have that $\Theta(f) = f(\Theta(X_1),\ldots,\Theta(X_n)) \equiv 0 \mod \mathscr{R}_{\mathbf{Y}}\left(T_+\right)$, and thus,

$$f(X_1,\ldots,X_n) = \underbrace{f(\Theta(X_1),\ldots,\Theta(X_n))}_{\left\langle\left\{\mathscr{R}_{Y_j}\left(T_+\right)\right\}_{j=1,\ldots,m}\right\rangle \subset I_B} + \underbrace{\sum_{i=1}^{n} C_i(X_i - \Theta(X_i))}_{I_B} \in I_B$$

□

Remark 4.7. Lemmas 4.6 and 4.5 are technical results valid for any matrix B. However, on the following result we applied the above lemmas to a matrix $A^{\perp}$ where A is a generator matrix of the linear code $\mathscr{C}$.

Theorem 4.8. $\Delta I = I_{A^\perp} \cap \mathbb{K}[\mathbf{X}]$

Proof. It is easy to check that $\Delta I \subseteq I_{A^\perp} \cap \mathbb{K}[\mathbf{X}]$. By Lemma 4.6 it suffices to make the following observations:

- $\Theta\left(\mathbf{X}^{(\alpha^j \mathbf{w}_i)} - 1\right) = \mathbf{Y}^{A^\perp(\alpha^j \mathbf{w}_i)} - 1 = 0$ since $A \cdot A^\perp = 0$ for $1 \leq j \leq q-1$ and $1 \leq i \leq k$.
- Let $A_j \in \mathbb{F}_q^m$ denotes the jth column of A. We distinguish two different types of generators in $\mathscr{R}_{X_i}(T_+)$:
  - $x_{iu}x_{iv} - x_{iw}$ such that $\alpha^u + \alpha^v = \alpha^w$. Then we have that
$$\begin{aligned}\Theta\left(x_{iu}x_{iv} - x_{iw}\right) &= \Theta\left(X_i^{\alpha^u+\alpha^v}\right) - \Theta\left(X_i^{\alpha^w}\right) = \Theta\left(\mathbf{X}^{\mathbf{e}_i(\alpha^u+\alpha^v)}\right) - \Theta\left(\mathbf{X}^{\mathbf{e}_i\alpha^w}\right) \\ &= \mathbf{Y}^{A_i^T(\alpha^u+\alpha^v)} - \mathbf{Y}^{A_i^T\alpha^w} = \mathbf{Y}^{A_i^T}\left(y_{iu}y_{iv} - y_{iw}\right) \in \mathscr{R}_{\mathbf{Y}}(T_+)\end{aligned}$$
  - $x_{iu}x_{iv} - 1$ such that $\alpha^u + \alpha^v = 0$. Therefore,
$$\Theta\left(x_{iu}x_{iv} - 1\right) = \Theta\left(X_i^{\alpha^u+\alpha^v}\right) - 1 = \mathbf{Y}^{A_i^T}\left(y_{iu}y_{iv} - 1\right) \in \mathscr{R}_{\mathbf{Y}}(T_+)$$

Conversely, let $f = \mathbf{X}^{\mathbf{u}} - \mathbf{X}^{\mathbf{v}} \in I_{A^\perp} \cap \mathbb{K}[\mathbf{X}]$ with $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$. Lemma 4.6 implies that $\Theta(f) = \mathbf{Y}^{A^\perp\mathbf{u}^T} - \mathbf{Y}^{A^\perp\mathbf{v}^T} \equiv 0 \mod \mathscr{R}_{\mathbf{Y}}(T_+)$. Hence, by Lemma 4.5 we have that $A^\perp\mathbf{u}^T = A^\perp\mathbf{v}^T$ in $\mathbb{F}_q$, thus $f \in I(A^\perp) = \Delta I$ by Theorem 4.3. □

Example 4.9. Let us consider the $[7,2]$ linear code $\mathscr{C}$ over $\mathbb{F}_3$ with generator matrix
$$G = \begin{pmatrix} 1 & 0 & 1 & 2 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 1 & 0 & 2 \end{pmatrix} \in \mathbb{F}_3^{2\times 7},$$
where the primitive element $\alpha = 2$ generates the finite field $\mathbb{F}_3 = \left\{0, \alpha = 2, \alpha^2 = 1\right\}$ which gives us the following additive table:

T_+	1	2
1	1	0
2	0	2

Or equivalently, $\left\{\ \alpha + \alpha = \alpha^2, \quad \alpha^2 + \alpha = 0, \quad \alpha^2 + \alpha^2 = \alpha\ \right\}$. Therefore, we obtain the following binomials associated to the previous rules:
$$\mathscr{R}_{X_i}(T_+) = \left\{\ x_{i1}^2 - x_{i2}, \quad x_{i1}x_{i2} - 1, \quad x_{i2}^2 - x_{i1}\ \right\} \text{ with } i = 1, \ldots, 7.$$
Let us label the rows of G by $\mathbf{w}_1$ and $\mathbf{w}_2$. By Theorem 4.3, the ideal associated to the linear code $\mathscr{C}$ may be defined as the following binomial ideal:
$$\begin{aligned} I_+(\mathscr{C}) &= \left\langle \left\{\mathbf{X}^{\alpha^j\mathbf{w}_i} - 1\right\}_{j=1,2}^{i=1,2,3} \ \bigcup\ \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,7} \right\rangle \\ &= \left\langle \left\{ \begin{array}{r} x_{12}x_{32}x_{41}x_{52}x_{62}x_{72} - 1, \\ x_{11}x_{31}x_{42}x_{51}x_{61}x_{71} - 1, \\ x_{22}x_{31}x_{41}x_{52}x_{71} - 1, \\ x_{21}x_{32}x_{42}x_{51}x_{72} - 1 \end{array} \right\} \ \bigcup\ \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,7} \right\rangle \end{aligned}$$

Example 4.10. Consider $\mathscr{C}$ the $[3,1]$ linear code defined over $\mathbb{F}_9$ with generator matrix

$$G = \begin{pmatrix} 1 & \alpha & \alpha+1 \end{pmatrix} \in \mathbb{F}_9^{1\times 3}$$

Let α be a root of the irreducible polynomial $f(z) = z^2 + z + 1$. In particular, α is a primitive element of $\mathbb{F}_9$, i.e. $\mathbb{F}_9 = \mathbb{F}_3[\alpha]$. In other words,

$$\mathbb{F}_9 = \left\{0, \alpha, \alpha^2 = \alpha+1, \alpha^3 = 2\alpha+1, \alpha^4 = 2, \alpha^5 = 2\alpha, \alpha^6 = 2\alpha+2, \alpha^7 = \alpha+2, \alpha^8 = 1\right\}.$$

This representation yields to the following additive table:

T_+	1	2	3	4	5	6	7	8
1	5	3	8	7	0	4	6	2
2		6	4	1	8	0	5	7
3			7	5	2	1	0	6
4				8	6	3	2	0
5					1	7	4	3
6						2	8	5
7							3	1
8								4

Or equivalently,

$$\begin{cases} \alpha+\alpha = \alpha^5 & \alpha+\alpha^2 = \alpha^3 & \dots & \alpha+\alpha^8 = \alpha^2 \\ & \alpha^2+\alpha^2 = \alpha^6 & \dots & \alpha^2+\alpha^8 = \alpha^7 \\ & & & \vdots \\ & & & \alpha^8+\alpha^8 = \alpha^4 \end{cases}$$

Therefore, we obtain the following binomials associated to the previous rules:

$$\mathscr{R}_{X_i}(\mathscr{T}_+) = \left\{ \begin{array}{l} x_{i1}^2 - x_{i5}, x_{i1}x_{i2} - x_{i3}, x_{i1}x_{i3} - x_{i8}, x_{i1}x_{i4} - x_{i7}, x_{i1}x_{i5} - 1, x_{i1}x_{i6} - x_{i4}, \\ \quad x_{i1}x_{i7} - x_{i6}, x_{i1}x_{i8} - x_{i2}, \\ x_{i2}^2 - x_{i6}, x_{i2}x_{i3} - x_{i4}, x_{i2}x_{i4} - x_{i1}, x_{i2}x_{i5} - x_{i8}, x_{i2}x_{i6} - 1, x_{i2}x_{i7} - x_{i5}, \\ \quad x_{i2}x_{i8} - x_{i7}, \\ x_{i3}^2 - x_{i7}, x_{i3}x_{i4} - x_{i5}, x_{i3}x_{i5} - x_{i2}, x_{i3}x_{i6} - x_{i1}, x_{i3}x_{i7} - 1, x_{i3}x_{i8} - x_{i6}, \\ x_{i4}^2 - x_{i8}, x_{i4}x_{i5} - x_{i6}, x_{i4}x_{i6} - x_{i3}, x_{i4}x_{i7} - x_{i2}, x_{i4}x_{i8} - 1, \\ x_{i5}^2 - x_{i1}, x_{i5}x_{i6} - x_{i7}, x_{i5}x_{i7} - x_{i4}, x_{i5}x_{i8} - x_{i1}, \\ x_{i6}^2 - x_{i2}, x_{i6}x_{i7} - x_{i8}, x_{i6}x_{i8} - x_{i5}, \\ x_{i7}^2 - x_{i3}, x_{i7}x_{i8} - x_{i1}, \\ x_{i8}^2 - x_{34} \end{array} \right\}$$

for $i = 1, 2, 3$.

Let us label the row of G by $\mathbf{w}_1$. By Theorem 4.3, the ideal associated to the linear

code $\mathscr{C}$ may be defined as the following binomial ideal:

$$
\begin{aligned}
I_+(\mathscr{C}) &= \left\langle \left\{\mathbf{X}^{\alpha^j \mathbf{w}_1} - 1\right\}_{j=1,\ldots,8} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,7} \right\rangle \\
&= \left\langle \left\{ \begin{array}{l} x_{18}x_{21}x_{32} - 1 \\ x_{11}x_{22}x_{33} - 1 \\ x_{12}x_{23}x_{34} - 1 \\ x_{13}x_{24}x_{35} - 1 \\ x_{14}x_{25}x_{36} - 1 \\ x_{15}x_{26}x_{37} - 1 \\ x_{16}x_{27}x_{38} - 1 \\ x_{17}x_{28}x_{31} - 1 \end{array} \right\} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,2,3} \right\rangle
\end{aligned}
$$

4.2 Computing a Gröbner representation

For simplicity of notation, in what follows, we write $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ for the canonical basis of $\mathbb{Z}^n$ and $\{\mathbf{u}_1, \ldots, \mathbf{u}_{n(q-1)}\}$ for the canonical basis of $\mathbb{Z}^{n(q-1)}$.

Definition 4.11. An ordering $\prec$ on $\mathbb{F}_q^n$ is a *weight compatible* ordering if for any $\mathbf{a}$, $\mathbf{b} \in \mathbb{F}_q^n$ we say $\mathbf{a} \prec \mathbf{b}$ if $\mathrm{w}_H(\mathbf{a}) < \mathrm{w}_H(\mathbf{b})$ or $\mathrm{w}_H(\mathbf{a}) = \mathrm{w}_H(\mathbf{b})$ and $\Delta\mathbf{a} \prec_1 \Delta\mathbf{b}$ where $\prec_1$ is any admissible order on $\mathbb{N}^n$.

Definition 4.12. A Gröbner representation of an $[n,k]$ linear code $\mathscr{C}$ is a pair $(\mathscr{N}, \phi)$ where:

- $\mathscr{N}$ is a transversal of the cosets in $\mathbb{F}_q^n/\mathscr{C}$ (i.e. one element of each coset) verifying that $\mathbf{0} \in \mathscr{N}$ and for each $n \in \mathscr{N} \setminus \{\mathbf{0}\}$ there exists $i \in \{1, \ldots, n(q-1)\}$ such that $\mathbf{n} = \mathbf{n}' + \nabla\mathbf{u}_i$ with $\mathbf{n}' \in \mathscr{N}$.
- $\phi : \mathscr{N} \times \{\mathbf{u}_i\}_{i=1}^{n(q-1)} \longrightarrow \mathscr{N}$ is a function called Matphi function that maps each pair $(\mathbf{n}, \mathbf{u}_i)$ to the element of $\mathscr{N}$ representing the coset that contains $\mathbf{n} + \nabla\mathbf{u}_i$.

Remark the resemblance with Definition 2.1 which explains the relationship with Gröbner Bases. As usual, the function Matphi can be seen as the multiplication tables of the corresponding standard monomials times the variables x_{ij} with $1 \le i \le n$ and $1 \le j \le q-1$.

Note that by Theorem 4.3 we know a set of generators of the ideal $I_+(\mathscr{C})$ given by

$$
\left\langle \left\{\mathbf{X}^{(\alpha^j \mathbf{w}_i)} - 1\right\}_{\substack{i=1,\ldots,k \\ j=1,\ldots,q-1}} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n} \right\rangle,
$$

Such Gröbner representation can be computed with a slight modification of the FGLM algorithm using Algorithm 18. For the binary code, this extension can be found in [16]. Let $\prec$ be a weight compatible ordering, there are three functions needed to understand the Algorithm 18:

- `InsertNext[`$\mathbf{w}$`, List]` adds to `List` all the sums $\mathbf{w}+\nabla\mathbf{u}_i$ with $i \notin \mathrm{supp}(\Delta\mathbf{w})$, keeps the increasing order of the list `List` w.r.t. $\prec$ and removes duplicates.

- `NextTerm[List]` returns the first element of the list `List` and deletes it from `List`.

- `Member[`$\mathbf{v}, \{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$`]` returns j if $\mathbf{v} = \mathbf{v}_j$ and `false` otherwise.

Remark 4.13. A weight compatible ordering $\prec$ is in general not admissible on $\mathbb{F}_q^n$. Note that $\mathbf{a} \prec \mathbf{b}$ does not always imply that $\mathbf{ac} \prec \mathbf{bc}$ for all $\mathbf{c} \in \mathbb{F}_q^n$. However $\prec$ is a notherian order on $\mathbb{F}_q^n$ since $\mathbb{F}_q^n$ is finite. Moreover, for all $\mathbf{a} \in \mathbb{F}_q^n$ we have that $\deg(\mathbf{X}^{\mathbf{a}}) = \mathrm{w}_H(\Delta\mathbf{a})$, that is, a weight compatible ordering on $\mathbb{F}_q^n$ can be viewed as a total degree ordering on $\mathbb{K}[\mathbf{X}]$.

Algorithm 18: Computing a Gröbner representation of an $\mathbb{F}_q$-linear code

Data: A weight compatible ordering $\prec$ and the parity check matrix H of a linear code $\mathscr{C}$
Result: $(\mathscr{N}, \phi)$ a Gröbner representation for $\mathscr{C}$

```
1  List ⟵ [0]; S ⟵ ∅; 𝒩 ⟵ ∅; r ⟵ 0;
2  while List ≠ ∅ do
3  |  w ⟵ NextTerm[List];
4  |  s ⟵ wH^T;
5  |  j ⟵ Member[s, S];
6  |  if j ≠ false then
7  |  |  for k ∈ supp(Δw) : w = w' + ∇u_k with w' ∈ 𝒩
8  |  |  |  φ(w', u_k) ⟵ w_j;
9  |  |  endfor
10 |  else
11 |  |  r ⟵ r + 1;
12 |  |  w_r ⟵ w;
13 |  |  𝒩 ⟵ 𝒩 ∪ {w_r};
14 |  |  S ⟵ S ∪ {s};
15 |  |  List = InsertNexts[w, List];
16 |  |  for k ∈ supp(Δw) : w = w' + ∇u_k with w' ∈ 𝒩
17 |  |  |  φ(w', u_k) ⟵ w;
18 |  |  |  φ(w, u_k) ⟵ w';
19 |  |  endfor
20 |  endif
21 endw
```

Theorem 4.14 (Correctness of Algorithm 18)**.** *Algorithm 18 computes a Gröbner representation $(\mathscr{N}, \phi)$ for a given linear code $\mathscr{C}$ defined over a finite field $\mathbb{F}_q$.*

Proof. Let us first prove that $\mathcal{N}$ is a transversal of the cosets in $\mathbb{F}_q^n/\mathscr{C}$ that satisfies the properties of Definition 4.12. For this purpose we need to show the following items:

1. $\mathbf{0} \in \mathcal{N}$.
2. $|\mathcal{N}| = q^{n-k}$.
3. Two different words of $\mathcal{N}$ determine different cosets in $\mathbb{F}_q^n/\mathscr{C}$.
4. For all $\mathbf{n} \in \mathcal{N} \setminus \{\mathbf{0}\}$ there exists $i \in \{0, \ldots, n(q-1)\}$ such that $\mathbf{n} = \mathbf{n}' + \nabla \mathbf{u}_i$ with $\mathbf{n}' \in \mathcal{N}$.

The first property is guaranteed by Step 1 and Step 13. Note that Step 4 computes the syndrome of the new element and Step 5 checks whether the syndrome has already been considered in the algorithm. Hence, the third property is satisfied. The last property of $\mathcal{N}$ is a consequence of Step 13 and Step 15. To prove the second property we need to consider the following recursive function: $\mathrm{cf}: \mathbb{F}_q^n \longrightarrow \mathcal{N}$ where $\mathrm{cf}(\mathbf{0}) = \mathbf{0}$ and

$$\mathrm{cf}(\mathbf{v}) = \phi\left(\mathrm{cf}(\mathbf{w}), \mathbf{u}_{i_s}\right) \quad \text{if} \quad \Delta\mathbf{v} = \underbrace{\mathbf{u}_{i_1} + \ldots + \mathbf{u}_{i_{s-1}}}_{\Delta\mathbf{w}} + \mathbf{u}_{i_s}.$$

cf is well defined for all vectors $\mathbf{v} \in \mathbb{F}_q^n$ beginning with $\mathrm{cf}(\mathbf{0}) = \mathbf{0}$. Moreover observe that $\mathrm{cf}(\mathbf{v})$ and $\mathbf{v}$ have the same syndrome for all $\mathbf{v} = \nabla\left(\mathbf{u}_{i_1} + \ldots + \mathbf{u}_{i_s}\right) \in \mathbb{F}_q^n$.

$$\begin{aligned} H\left(\mathrm{cf}(\mathbf{v})\right)^T &= H\left(\phi\left(\mathrm{cf}\left(\nabla\left(\mathbf{u}_{i_1} + \ldots + \mathbf{u}_{i_{s-1}}\right)\right), \mathbf{u}_{i_s}\right)\right)^T \\ &= H\left(\mathrm{cf}\left(\nabla\left(\mathbf{u}_{i_1} + \ldots + \mathbf{u}_{i_{s-1}}\right)\right) + \nabla\mathbf{u}_{i_s}\right)^T \\ &= H\left(\mathrm{cf}\left(\nabla\left(\mathbf{u}_{i_1} + \ldots + \mathbf{u}_{i_{s-1}}\right)\right)\right)^T + H\left(\nabla\mathbf{u}_{i_s}\right)^T \\ &= \ldots = H\left(\mathrm{cf}(\mathbf{0})\right)^T + H\left(\nabla\left(\mathbf{u}_{i_1} + \ldots \mathbf{u}_{i_s}\right)\right)^T = H\mathbf{v}^T \end{aligned}$$

The second equality is the result of applying the definition of the Matphi function. By Step 15, all possible values of $\mathrm{cf}(\mathbf{v})$ with $\mathbf{v} \in \mathbb{F}_q^n$ are achieved. Thus we may conclude that there will be as many canonical forms as different syndromes in $\mathbb{F}_q^n/\mathscr{C}$. In other words, $|\mathcal{N}| = q^{n-k}$.

The task is now to show that the Matphi function computed by the algorithm fulfils Definition 4.12. Let $\mathbf{n} \in \mathcal{N}$ and $i \in \{1, \ldots, n(q-1)\}$. There are two possibilities for the pair $(\mathbf{n}, \mathbf{u}_i)$:

- If $H\left(\mathbf{n} + \nabla\mathbf{u}_i\right)^T$ has previously been considered in the algorithm, then $\phi(\mathbf{n}, \mathbf{u}_i)$ is defined in Step 8 satisfying that $H\left(\phi(\mathbf{n}, \mathbf{u}_i)\right)^T = H\left(\mathbf{n} + \Delta\mathbf{u}_i\right)^T$.
- Otherwise, $\phi(\mathbf{n}, \mathbf{u}_i)$ is defined in Step 18.

The fact that only the index k, such that $k \in \mathrm{supp}(\mathbf{n})$, is deemed to maintain the increase order of $\mathbf{n} > \mathbf{n}'$ with $\mathbf{n} = \mathbf{n}' + \nabla\mathbf{u}_k$ and thus does not define twice the element $\phi(\mathbf{n}, \mathbf{u}_i)$.

We have seen that $|\mathscr{N}| = q^{n-k}$. Moreover, each time that a new word is added to the set $\mathscr{N}$, $n(q-1)$ products are inserted in the list `List`. Therefore we have $|\texttt{List}| \leq n(q-1)q^{n-k}$ so we deduce that after a finite number of steps the algorithm ends. □

Example 4.15. Let us consider the $[5,2,3]$ linear code $\mathscr{C}$ over $\mathbb{F}_3$ with generator and parity check matrices

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 2 & 2 \end{pmatrix} \in \mathbb{F}_3^{2\times 5}, \quad H = \begin{pmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 & 2 \end{pmatrix} \in \mathbb{F}_3^{3\times 5}$$

respectively.

Let $\prec$ be a weight compatible ordering induced by the lexicographic ordering (`lex`) on $\mathbb{F}_q^n$, Algorithm 18 computes a Gröbner representation $(\mathscr{N}, \phi)$ of $\mathscr{C}$ which is represented on Tables 4.1 and 4.2. Note that on the representation of the function Matphi ϕ in Table 4.2, the first entry corresponds to the element $\mathbf{n}_i \in \mathscr{N}$, following the order established in the Table 4.1, and the second points to the values $\phi(\mathbf{n}_i, \mathbf{u}_j)$ for $j = 1, \ldots, 10$. That is, $[\mathbf{n}_j, [\phi(\mathbf{n}_j, \mathbf{u}_1), \ldots, \phi(\mathbf{n}_j, \mathbf{u}_{10})]]$.

Set of canonical forms $\mathscr{N}$	
Weight 0	$[\mathbf{0}]$
Weight 1	$[(2,0,0,0,0)], [(1,0,0,0,0)], [(0,2,0,0,0)], [(0,1,0,0,0)],$ $[(0,0,2,0,0)], [(0,0,1,0,0)], [(0,0,0,2,0)], [(0,0,0,1,0)],$ $[(0,0,0,0,2)], [(0,0,0,0,1)],$
Weight 2	[(2,0,0,1,0)], [(2,0,0,0,1)], [(1,0,0,2,0)], [(0,2,0,2,0)], [(0,2,0,0,1)], [(0,1,0,1,0)], [(0,1,0,0,2)], [0,1,0,0,1], [(0,0,2,2,0)], [(0,0,2,0,1)], [(0,0,1,1,0)], [(0,0,0,2,2)], [(0,0,0,2,1)], [(0,0,0,1,2)], [(0,0,0,1,1)]

Table 4.1: Set $\mathscr{N}$ of Example 4.15

An equivalent way of obtaining the pair $(\mathscr{N}, \phi)$ is to compute a reduced Gröbner basis of the following binomial ideal:

$$I(\mathscr{C}) = \left\langle \left\{ \begin{array}{cc} x_{12}x_{32}x_{52} - 1, & x_{11}x_{31}x_{51} - 1, \\ x_{22}x_{32}x_{41}x_{51} - 1, & x_{21}x_{31}x_{42}x_{52} - 1 \end{array} \right\} \bigcup \left\{ \mathscr{R}_{X_i}(T_+) \right\}_{i=1,\ldots,5} \right\rangle \subseteq \mathbb{K}[\mathbf{X}]$$

w.r.t. the degree reverse lexicographical (`degrevlex`) order with $x_{11} < x_{12} < x_{21} < x_{22} < x_{31} < x_{32} < x_{41} < x_{42} < x_{51} < x_{52}$.

Theorem 4.16. *Assume that $\mathscr{N}$ has been obtained using Algorithm 18. If $\mathbf{w} \in \mathscr{N}$ then $\mathbf{w} \in \mathrm{CL}(\mathscr{C})$.*

Function Maphi	
$[\mathbf{n}_1, [2,3,4,5,6,7,8,9,10,11]]$,	$[\mathbf{n}_2, [3,1,21,26,11,22,16,12,7,13]]$,
$[\mathbf{n}_3, [1,2,25,23,13,10,14,20,22,6]]$,	$[\mathbf{n}_4, [21,25,5,1,24,14,15,13,16,17]]$,
$[\mathbf{n}_5, [26,23,1,3,12,27,22,18,19,20]]$,	$[\mathbf{n}_6, [11,13,24,12,7,1,21,19,3,22]]$,
$[\mathbf{n}_7, [22,10,14,27,1,6,17,23,13,2]]$,	$[\mathbf{n}_8, [16,14,15,22,21,17,9,1,24,25]]$,
$[\mathbf{n}_9, [12,20,13,18,19,23,1,8,26,27]]$,	$[\mathbf{n}_{10}, [7,22,16,19,3,13,24,26,11,1]]$,
$[\mathbf{n}_{11}, [13,6,17,20,22,2,25,27,1,10]]$,	$[\mathbf{n}_{12}, [20,9,6,24,27,5,2,16,23,15]]$,
$[\mathbf{n}_{13}, [6,11,18,9,10,3,4,14,2,7]]$,	$[\mathbf{n}_{14}, [8,16,27,7,21,24,20,3,18,21]]$,
$[\mathbf{n}_{15}, [19,27,22,8,26,20,13,4,12,23]]$,	$[\mathbf{n}_{16}, [14,8,19,11,25,18,12,11,13,4]]$,
$[\mathbf{n}_{17}, [18,24,20,11,8,21,23,7,4,16]]$,	$[\mathbf{n}_{18}, [24,17,9,13,16,25,5,22,21,14]]$,
$[\mathbf{n}_{19}, [27,12,10,16,23,9,6,21,20,5]]$,	$[\mathbf{n}_{20}, [9,12,11,17,15,26,13,14,5,19]]$,
$[\mathbf{n}_{21}, [25,4,26,2,17,8,19,6,14,18]]$,	$[\mathbf{n}_{22}, [10,7,8,15,2,11,18,5,6,3]]$,
$[\mathbf{n}_{23}, [5,26,3,25,9,19,7,17,14,12]]$,	$[\mathbf{n}_{24}, [17,18,12,6,14,4,26,10,25,8]]$,
$[\mathbf{n}_{25}, [4,21,23,3,18,16,27,11,8,24]]$,	$[\mathbf{n}_{26}, [23,5,2,21,20,7,10,24,27,9]]$,
$[\mathbf{n}_{27}, [15,19,7,14,5,13,11,25,9,26]]$	

Table 4.2: Function Matphi of Example 4.15

Proof. This Theorem is a direct consequence of the fact that a weight compatible ordering on $\mathbb{F}_q^n$ is a total degree ordering on $\mathbb{K}[\mathbf{X}]$ (see Remark 4.13). Suppose, contrary to our claim, that there exists an element $\mathbf{w}$ in $\mathscr{N}$ such that $\mathbf{w} \notin \mathrm{CL}(\mathscr{C})$. Hence, there must exist a word $\hat{\mathbf{w}} \in \mathbb{F}_q^n$ with strictly smaller Hamming weight in the same coset as $\mathbf{w}$. Therefore, $\hat{\mathbf{w}}$ is considered before than $\mathbf{w}$ in Algorithm 18. And this clearly forces that $\hat{\mathbf{w}}$ is the representative of the coset $\mathbf{w}+\mathscr{C}$ instead of $\mathbf{w}$, which contradicts the fact that $\mathbf{w} \in \mathscr{N}$. □

In [16], another Gröbner representation associated to the class of linear codes defined over an arbitrary finite field was presented. The main difference lies in the chosen representation of the finite field. Let $\mathbb{F}_q$ be a finite field with $q = p^m$ elements and p prime, we could adopt the convention that $\mathbb{F}_q = \frac{\mathbb{F}_p[X]}{(f(X))}$ where $f(X)$ is chosen such that $f(X)$ is an irreducible polynomial over $\mathbb{F}_p$ of degree m. Let β be a root of $f(X)$, then an equivalent formulation of $\mathbb{F}_q$ is $\mathbb{F}_p[\beta]$, i.e. any element of $\mathbb{F}_q$ is represented as $a_0 + a_1\beta + \ldots + a_{m-1}\beta^{m-1}$ with $a_i \in \mathbb{F}_p$ for $i \in \{0,\ldots,m-1\}$.

Let $\mathbf{Y}$ denotes n variables $Y_1,\ldots,Y_n$ such that each variable Y_i is decomposed into m variables $y_{i1},\ldots,y_{im}$. Let $\mathbf{b} = (b_1,\ldots,b_n)$ be an n-tuple of elements of the field $\mathbb{F}_q$ where $b_i = a_{i0} + a_{i1}\beta + \ldots + a_{im-1}\beta^{m-1}$ with $a_{ij} \in \mathbb{F}_p$.

On that paper the authors selected the following notation:

$$\mathbf{Y}^{\mathbf{b}} = Y_1^{b_1}\cdots Y_n^{b_n} = \left(y_{11}^{a_{10}}\cdots y_{1m}^{a_{1m-1}}\right)\cdots\left(y_{n1}^{a_{n0}}\cdots y_{nm}^{a_{nm-1}}\right)$$

where they replace the elements $a_{ij} \in \mathbb{F}_p \equiv \mathbb{Z}_p$, which belongs to the class of $0,1,\ldots,p-1$ by the same symbol regarded as an integer (i.e. $a_{ij} = \blacktriangle a_{ij}$). Therefore, again, we identify monomials whose exponents belongs to $\mathbb{F}_q^n$ with monomials with integer exponents.

If we take the following ideal as the binomial ideal associated with the code $\mathscr{C}$:

$$I(\mathscr{C}) = \left\{ \mathbf{Y}^{\mathbf{a}} - \mathbf{Y}^{\mathbf{b}} \mid \mathbf{a} - \mathbf{b} \in \mathscr{C} \right\} \subseteq \mathbb{K}[\mathbf{Y}],$$

then a reduced Gröbner basis of $I(\mathscr{C})$ w.r.t. an error-vector ordering $\prec_e$ defines the set $\mathscr{N}$.

Note that the number of variables is smaller in that case (mn variables instead of $(q-1)n$). However, with this notation the representative elements of $\mathscr{N}$ is not necessarily a coset leader when the leader weight is bigger than the error correcting capacity t, as we will see in the following example.

However, since the multiplicative structure of Matphi is independent of the particular set of canonical forms, then Matphi is equivalent in both representations. That is, we could consider the function Matphi defined over the set of cosets instead of a particular set of canonical forms.

Example 4.17. Let us consider the finite field $\mathbb{F}_4$. We can use two different field representation:

1. Let α be a primitive element of $\mathbb{F}_4$, then $\mathbb{F}_4$ contains all powers of α and the zero element, i.e. $\mathbb{F}_4 = \left\{0, \alpha, \alpha^2, \alpha^3 = 1\right\}$.
2. Let β be a root of an irreducible binary polynomial of degree 2 namely $f(z) = z^2 + z + 1$. Then the elements of $\mathbb{F}_4 = \frac{\mathbb{K}[z]}{f(z)}$ are the 4 binary polynomials of degree at most one, i.e. $\mathbb{F}_4 = \left\{0, 1, \beta, \beta + 1 = \beta^2\right\}$.

Note that β has order 3 (i.e. is a primitive element of $\mathbb{F}_4$), but it is not always the case that a root of an irreducible polynomial is a primitive element.

Let us consider the $[5, 2, 1]$ linear code over $\mathbb{F}_4$ with generator matrix:

$$G = \begin{pmatrix} 1 & 0 & \alpha & 1 & \alpha \\ 0 & 1 & \alpha^2 & 0 & \alpha \end{pmatrix} = \begin{pmatrix} 1 & 0 & \beta & 1 & \beta \\ 0 & 1 & \beta+1 & 0 & \beta \end{pmatrix} \in \mathbb{F}_4^{2\times 5}.$$

In [16, Example 18] the authors compute a Gröbner representation for the code $\mathscr{C}$ on account of the fact that $\mathbb{F}_4 = \frac{\mathbb{F}_2[z]}{f(z)}$. They get 64 elements representing the set $\mathscr{N}$. Note that there is an element in $\mathscr{N}$, the element $n_{64} = (1+\beta, 1, \beta, 0, 0)$, which does not belong to the set of coset leaders of the code $\mathscr{C}$.

However, with our new approach all the elements of $\mathscr{N}$ are coset leaders. Therefore, if we compute a reduced Gröbner basis of the following binomial ideal

$$I(\mathscr{C}) = \left\langle \left\{ \begin{array}{ll} x_{13}x_{31}x_{43}x_{51} - 1, & x_{23}x_{32}x_{51} - 1, \\ x_{11}x_{32}x_{41}x_{52} - 1 & x_{21}x_{33}x_{52} - 1, \\ x_{12}x_{33}x_{42}x_{53} - 1, & x_{22}x_{31}x_{53} - 1 \end{array} \right\} \cup \left\{ \mathscr{R}_{X_i}\left(T_+\right) \right\}_{i=1,\ldots,5} \right\rangle$$

where

$$\mathscr{R}_{X_i}\left(T_+\right) = \left\{ \; x_{i1}^2 - 1, x_{i1}x_{i2} - x_{i3}, x_{i1}x_{i3} - x_{i2}, x_{i2}^2 - 1, x_{i2}x_{i3} - x_{i1}, x_{i3}^2 - 1 \; \right\}$$

w.r.t. the `degrevlex` order we obtain as representative on the set $\mathscr{N}$ of the coset $(1+\beta, 1, \beta, 0, 0) + \mathscr{C}$ the element $(0, 0, 0, \alpha^2, \alpha^2)$ which is a coset leader.

4.3 Decoding linear codes

4.3.1 Reduced Gröbner basis

Throughout this section, let $\mathscr{G} = \{g_1, \ldots, g_s\}$ be the reduced Gröbner basis of $I_+(\mathscr{C})$ with respect to $\succ$, where we take $\succ$ to be any degree compatible ordering with $X_1 < \ldots < X_n$. Moreover, for all $i \in \{1, \ldots, s\}$ we define

$$g_i = \mathbf{X}^{\mathbf{g}_i^+} - \mathbf{X}^{\mathbf{g}_i^-} \quad \text{with} \quad \mathbf{X}^{\mathbf{g}_i^+} > \mathbf{X}^{\mathbf{g}_i^-} \quad \text{and} \quad \mathbf{g}_i^+ - \mathbf{g}_i^- \in \mathscr{C}.$$

The following lemma expresses the equivalence of the linear code $\mathscr{C}$ and the Gröbner basis $\mathscr{G}$.

Lemma 4.18. *$\mathbf{X}^{\mathbf{c}_1} - \mathbf{X}^{\mathbf{c}_2} \in \langle \mathscr{G} \rangle$ if and only if $\mathbf{c}_1 - \mathbf{c}_2 \in \mathscr{C}$.*

Proof. It follows from the fact that if $\mathscr{G}$ is a Gröbner basis for $I_+(\mathscr{C})$, then $\mathscr{G}$ is a basis of $I_+(\mathscr{C})$. □

Theorem 4.19. *Let t be the error-correcting capacity of $\mathscr{C}$. If $\deg\left(\mathrm{Red}_\prec(\mathbf{X}^{\mathbf{a}}, \mathscr{G})\right) \leq t$, then the vector $\mathbf{e} \in \mathbb{F}_q^n$ verifying that $\mathbf{X}^{\mathbf{e}} = \mathrm{Red}_\prec(\mathbf{X}^{\mathbf{a}}, \mathscr{G})$ is the error vector corresponding to the received word $\mathbf{a} \in \mathbb{F}_q^n$. In other words, $\mathbf{c} = \mathbf{a} - \mathbf{e} \in \mathscr{C}$ is the closest codeword to $\mathbf{a} \in \mathbb{F}_q^n$. Otherwise $\mathbf{a}$ contains more than t errors.*

Proof. According to the definition of *reduction of a polynomial with respect to* $\mathscr{G}$; since $\mathbf{X}^{\mathbf{e}} = \mathrm{Red}_\prec(\mathbf{X}^{\mathbf{a}}, \mathscr{G})$ there exists polynomials $f_1, \ldots, f_s \in \mathbb{K}[\mathbf{X}]$ such that

$$\mathbf{X}^{\mathbf{a}} = f_1 g_1 + \ldots + f_s g_s + \mathbf{X}^{\mathbf{e}}, \text{ or equivalently } \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{e}} \in \langle \mathscr{G} \rangle .$$

Lemma 4.18 now leads to $\mathbf{a} - \mathbf{e} \in \mathscr{C}$.

Assume that there exists $\mathbf{e}_2 \in \mathbb{F}_q^n$ such that $\mathbf{a} - \mathbf{e}_2 \in \mathscr{C}$ and $\mathrm{w}_H(\mathbf{e}_2) < \mathrm{w}_H(\mathbf{e})$; i.e. the total degree of $\mathbf{X}^{\mathbf{e}_2}$ is strictly smaller than the total degree of $\mathbf{X}^{\mathbf{e}}$, $\deg(\mathbf{X}^{\mathbf{e}_2}) < \deg(\mathbf{X}^{\mathbf{e}})$. Then by Lemma 4.18 there exists $\hat{f}_1, \ldots, \hat{f}_s \in \mathbb{K}[\mathbf{X}]$ such that $\mathbf{X}^{\mathbf{a}} = \hat{f}_1 g_1 + \ldots + \hat{f}_s g_s + \mathbf{X}^{\mathbf{e}_2}$, which contradicts the uniqueness of the normal form.

We have actually proved that the exponent of the normal form of $\mathbf{X}^{\mathbf{a}}$ is the minimal element with respect to $\succ$ having the same syndrome as $\mathbf{a}$. Therefore, if $\deg(\mathbf{X}^{\mathbf{e}}) \leq t$ then it is clear that there cannot be another element $\hat{\mathbf{e}}$ such that $\mathrm{w}_H(\hat{\mathbf{e}}) \leq t$ (or equivalently, $\deg(\mathbf{X}^{\hat{\mathbf{e}}}) \leq t$) and $\mathbf{a} - \hat{\mathbf{e}} \in \mathscr{C}$. Indeed, this would mean that there exists two solutions for the linear system with weight at most t, which is not possible since t is the error-correcting capacity of the code. Otherwise $\mathbf{a}$ contains more than t errors. □

Remark 4.20. In any case, $\mathrm{Red}_\prec(\mathbf{X}^{\mathbf{a}}, \mathscr{G}) = \mathbf{X}^{\mathbf{e}}$ provides a coset leader even if $\mathrm{w}_H(\mathbf{e}) \geq t$ as we have proved in Theorem 4.16. Note that this was the main difference with what was presented in [16].

Proposition 4.21. *Let t be the error-correcting capacity of $\mathscr{C}$, then*

$$\begin{aligned} t &= \min\left\{\mathrm{w}_H(g_i^+) \mid g_i \in \mathscr{G} \setminus \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n}\right\} + 1 \\ &= \min\left\{\deg(g_i) \mid g_i \in \mathscr{G} \setminus \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n}\right\} + 1. \end{aligned}$$

Proof. This Proposition is analogous to [11, Theorem 3]. Let $\mathbf{c}$ be a minimum weight nonzero codeword of $\mathscr{C}$, i.e. $\mathrm{w}_H(\mathbf{c}) = d$, where d is the minimum distance of $\mathscr{C}$. Let $\mathbf{X}^{\mathbf{c}_1}$ and $\mathbf{X}^{\mathbf{c}_2}$ be two monomials in $\mathbb{K}[\mathbf{X}]$ such that $\mathbf{X}^{\mathbf{c}} = \mathbf{X}^{\mathbf{c}_1}\mathbf{X}^{\mathbf{c}_2}$ and $\mathrm{w}_H(\mathbf{c}_1) = t+1$, that is to say $\mathbf{X}^{\mathbf{c}_1} \succ \mathbf{X}^{\mathbf{c}_2}$.

Then $\mathbf{X}^{\mathbf{c}_1}\mathbf{X}^{\mathbf{c}_2} - 1 \in I_+(\mathscr{C})$, or equivalently $\mathbf{X}^{\mathbf{c}_1} - \mathbf{X}^{-\mathbf{c}_2} \in I_+(\mathscr{C})$. Note that $\mathrm{w}_H(\mathbf{c}_2) = \mathrm{w}_H(-\mathbf{c}_2)$, thus $\mathbf{X}^{\mathbf{c}_1} \succ \mathbf{X}^{-\mathbf{c}_2}$. Therefore, we get that $\mathbf{X}^{\mathbf{c}_1}$ belongs to the initial ideal in $\big(I_+(\mathscr{C})\big)$, so there must exists an index $i \in \{1,\ldots,s\}$ such that the leading term of $g_i \in \mathscr{G}$ divides $\mathbf{X}^{\mathbf{c}_1}$, and thus, $\mathrm{w}_H(g_i^+) \leq \mathrm{w}_H(\mathbf{c}_1) = t-1$.

Now suppose that there exists $g_i \in \mathscr{G}$ with $i \in \{1,\ldots,s\}$ such that $\mathrm{w}_H(g_i^+) < t-1$. By definition, $g_i^+ - g_i^- \in \mathscr{C}$, but $\mathrm{w}_H(g_i^+ - g_i^-) \leq \mathrm{w}_H(g_i^+) - \mathrm{w}_H(g_i^-) < 2(t-1) < d$, which contradicts the definition of minimum distance of $\mathscr{C}$. □

Proposition 4.22. $\mathrm{w}_H(g_i^+) - \mathrm{w}_H(g_i^-) \leq 1$ *for all* $i \in \{1,\ldots,s\}$.

Proof. Without loss of generality we assume that $i = 1$. We can distinguish two cases:

- The case when $\mathrm{supp}(g_1^+) \cap \mathrm{supp}(g_1^-) = \emptyset$.

  Let x_{ij} be any variable that belongs to the support of $\mathbf{X}^{g_1^+}$, i.e. $\mathbf{X}^{g_1^+} = x_{ij}\mathbf{X}^{\mathbf{w}}$ with $\mathrm{w}_H(\mathbf{w}) + 1 = \mathrm{w}_H(g_1^+)$. There must exist $l \in \{1,\ldots,q-1\}$ such that $x_{ij}x_{il} - 1 \in \mathscr{R}_{X_i}\left(T_+\right)$. Thus $x_{il}\left(\mathbf{X}^{g_1^+} - \mathbf{X}^{g_1^-}\right) = \mathbf{X}^{\mathbf{w}} - x_{il}\mathbf{X}^{g_1^-}$. If $\mathbf{X}^{\mathbf{w}} \succ x_{il}\mathbf{X}^{g_1^-}$, then $\mathbf{X}^{\mathbf{w}} \in \mathrm{LT}(\mathscr{G} \setminus \{g_1\})$, which contradicts the fact that $\mathscr{G}$ is reduced. Hence, $x_{il}\mathbf{X}^{g_1^-} \succ \mathbf{X}^{\mathbf{w}}$ or equivalently $\mathrm{w}_H(g_1^-)+1 > \mathrm{w}_H(\mathbf{w}) = \mathrm{w}_H(g_1^+)-1$, which completes the proof.

- A similar argument applies to the case $i \in \mathrm{supp}(g_1^+) \cap \mathrm{supp}(g_1^-)$.

  In other words, $g_1 = \mathbf{X}^{g_1^+} - \mathbf{X}^{g_1^-} = x_{ij}\mathbf{X}^{\mathbf{a}} - x_{il}\mathbf{X}^{\mathbf{b}}$. There must exists an integer $m \in \{1,\ldots,q-1\}$ such that $x_{ij}x_{im} - 1$ and $x_{il}x_{im} - x_{iv}$ belongs to $\mathscr{R}_{X_i}\left(T_+\right)$. Thus $x_{im}g_1 = \mathbf{X}^{\mathbf{a}} - x_{iv}\mathbf{X}^{\mathbf{b}} \in I_+(\mathscr{C})$. Suppose that $\mathbf{X}^{\mathbf{a}} \succ x_{iv}\mathbf{X}^{\mathbf{b}}$, then $\mathbf{X}^{\mathbf{a}} \in \mathrm{LT}(\mathscr{G} \setminus \{g_1\})$, is a contradiction. Therefore, $\mathrm{w}_H(\mathbf{b}) + 1 \geq \mathrm{w}_H(\mathbf{a})$ which establishes the desired formula.

  Note that it may happen that $l = j$. In this case we would have that $\mathrm{w}_H(\mathbf{b}) \geq \mathrm{w}_H(\mathbf{a})$, i.e. $\mathrm{w}_H(g_i^-) \geq \mathrm{w}_H(g_i^-)$ which is impossible except for the case of equality.

□

Definition 4.23. The monomial $\prod_{i=1}^{n}\prod_{j=1}^{q-1} x_{ij}^{\beta_{ij}}$ is said to be in *standard form* if $\sum_{j=1}^{q-1}\beta_{ij} = 1$ for all $i = 1,\ldots,n$. In other words, the exponents of each variable x_{ij} is 0 or 1, and two variables x_{ij} and x_{il} do not appear in the same monomial.

Definition 4.24. The reduction in one step $\to$ using $\mathscr{G}$ is defined as follows. For any $\mathbf{w} \in [\mathbf{X}]$:

1. Reduce $\mathbf{w}$ to its standard form $\mathbf{w}'$ using the relations $\mathscr{R}_{\mathbf{X}}\left(T_+\right)$.
2. Reduce $\mathbf{w}'$ w.r.t. $\mathscr{G}$ by the usual one step reduction.

This reduction process is well defined since it is confluent and noetherian as we will show in the following theorem:

Theorem 4.25. *Let* $\mathbf{w} \in [\mathbf{X}]$ *be an arbitrary term. Then:*

i) The reduction process $\to$ *is noetherian.*

ii) If $\mathbf{w} \to \mathbf{w}_1$, $\mathbf{w} \to \mathbf{w}_2$ *and* $\mathbf{w}_1$, $\mathbf{w}_2$ *are irreducible words modulo* $\to$, *then* $\mathbf{w}_1 = \mathbf{w}_2$.

Proof. Let $\mathbf{w} = \mathbf{w}_0 \to \mathbf{w}_1 \to \ldots \to \mathbf{w}_k \to \ldots$ be a descending chain of reductions. Note that from $\mathbf{w}_i \to \mathbf{w}_{i+1}$ it follows that $\mathbf{w}_i \succ \mathbf{w}_{i+1}$, where $\prec$ is a total degree ordering. Since the total degree ordering is admissible, there is no infinite strict descending chain of elements. Therefore, $\to$ is noetherian and i) follows.
Let us now prove that ii) holds. An easy verification shows that the one step reduction keeps the syndrome invariant. To this end, consider $\mathbf{w} = \mathbf{w}_1\mathbf{w}_2$ and $\mathbf{w}_1 - \mathbf{v}_1 \in \mathscr{G}$; then $\mathbf{w}$ is reduced to $\mathbf{w}' = \mathbf{v}_1\mathbf{w}_2$ where $S(\overline{\mathbf{v}_1}) = S(\overline{\mathbf{w}_1})$ and $\overline{\mathbf{u}} \in \mathbb{F}_q^n$ denotes the vector such that $\mathbf{u} = \mathbf{X}^{\overline{\mathbf{u}}}$. Hence

$$S(\overline{\mathbf{w}}) = S(\overline{\mathbf{w}_1}) + S(\overline{\mathbf{w}_2}) = S(\overline{\mathbf{v}_1}) + S(\overline{\mathbf{w}_2}) = S(\overline{\mathbf{w}'}).$$

Moreover, by definition, all the irreducible elements belongs to $\mathscr{N}$. Thus, under the conditions stated by ii), we have that $S(\overline{\mathbf{w}_1}) = S(\overline{\mathbf{w}_2}) = S(\mathbf{w})$. But by the properties of $\mathscr{N}$, there exists only one element of each coset in $\mathscr{N}$, so $\mathbf{w}_1 = \mathbf{w}_2$ and ii) is proven. □

Remark 4.26. The irreducible element corresponding to $\mathbf{w}$ will be called the *normal form* of $\mathbf{w}$ w.r.t. $\mathscr{G}$ and it will be denoted by $\mathrm{Red}(\mathbf{w}, \mathscr{G})$. The above theorem states that $\mathrm{Red}(\mathbf{w}, \mathscr{G})$ is unique and computable by a typical Buchberger's reduction process.

Example 4.27. Continuing with Example 4.9, note that the code has Hamming distance 5 so it corrects up to 2 errors. A reduced Gröbner basis $\mathscr{G}$ for the ideal $I_+(\mathscr{C})$ w.r.t. the `degrevlex` order with

$$\underbrace{x_{11} < x_{12}}_{X_1} < \underbrace{x_{21} < x_{22}}_{X_2} < \ldots < \underbrace{x_{71} < x_{72}}_{X_7}$$

has 193 elements. It is easy to check that the binomial $G_1 = x_{31}x_{62}x_{71} - x_{11}x_{22}$ and all the generators of the ideal $\mathscr{R}_{\mathbf{X}}\left(T_+\right)$ are elements of the reduced Gröbner basis.

Let us take the codeword $\mathbf{c} = (1,2,2,0,0,1,2)$ and add the error vector $\mathbf{e} = (2,2,0,0,0,0,0)$. Then the received word is $\mathbf{y} = (0,1,2,0,0,1,2) = \mathbf{c} + \mathbf{e}$ which corresponds to the monomial $w = x_{22}x_{31}x_{62}x_{71}$. Let us reduce w using $\mathscr{G}$:

$$w = x_{22}x_{31}x_{62}x_{71} \xrightarrow{G_1 = x_{31}x_{62}x_{71} - x_{11}x_{22}} x_{11}x_{22}x_{22} \xrightarrow{x_{22}^2 - x_{21} \in \mathscr{R}_{X_2}\left(T_+\right)} x_{11}x_{21}$$

The normal form of w modulo $\mathscr{G}$ is $x_{11}x_{21}$ which has weight 2, then $(2,2,0,0,0,0,0)$ is the error vector corresponding to w and the closest codeword is $x_{12}x_{21}x_{32}x_{62}x_{71}$ or $\mathbf{c} = \mathbf{y} + \mathbf{e}$.

Remark 4.28. In Section 3.2 we have defined another ideal $I_m(\mathscr{C})$ associated with modular codes, i.e. codes defined over $\mathbb{Z}_q$, in particular for codes over $\mathbb{F}_q$ with q prime, but not for the case p^r since $\mathbb{F}_{p^r} \neq \mathbb{Z}_{p^r}$. However, for $q \neq 2$ such ideal does not allow complete decoding since the reduction does not provide the minimum Hamming weight representative in the coset. In the following lines we give an example of what is discussed in this note.

Example 4.29. Continuing with the Example 4.9, now suppose that we consider our code as a linear code over the alphabet $\mathbb{Z}_3 = \mathbb{F}_3$. Then we can define the lattice ideal associated with $\mathscr{C}$ as the ideal generated by the following set of binomials (see Theorem 3.9 for the definition of this ideal and the references given there)

$$I_m(\mathscr{C}) = \left\langle \left\{ \begin{array}{c} y_1y_3y_4^2y_5y_6y_7 - 1, \\ y_2y_3^2y_4^2y_5y_7^2 - 1 \end{array} \right\} \bigcup \left\{ y_i^3 - 1 \right\}_{i=1,\ldots,7} \right\rangle \subseteq \mathbb{K}[y_1, \ldots, y_7]$$

If we compute a reduced Gröbner basis $\mathscr{G}$ of $I_m(\mathscr{C})$ w.r.t. a `degrevlex` ordering with $y_1 < y_2 < \ldots < y_7$ we obtain 62 binomials. The elements

$$G_1 = y_3^2y_6y_7^2 - y_1^2y_2 \quad \text{and} \quad G_2 = y_1^2y_2^2 - y_4y_5^2y_6$$

are elements of our reduced Gröbner basis.

Similarly to Example 4.27, let us take the codeword $\mathbf{c} = (1,2,2,0,0,1,2)$ and add the error $\mathbf{e} = (2,2,0,0,0,0,0)$. Then the received word is $\mathbf{y} = (0,1,2,0,0,1,2) = \mathbf{c} + \mathbf{e}$ which corresponds to the monomial $w = y_2y_3^2y_6y_7^2$. Let us reduce w using $\mathscr{G}$:

$$w = y_2y_3^2y_6y_7^2 \xrightarrow{G_1 = y_3^2y_6y_7^2 - y_1^2y_2} y_1^2y_2^2 \xrightarrow{G_2 = y_1^2y_2^2 - y_4y_5^2y_6} y_4y_5^2y_6$$

The normal form of w modulo $\mathscr{G}$ is $y_4y_5^2y_6$ which does not correspond to the error vector.

Proposition 4.30. *The set* $\mathscr{T} = \left\{ \mathbf{g}_i^+ - \mathbf{g}_i^- \mid i = 1, \ldots, s \right\}$ *is a test-set for* $\mathscr{C}$.

Proof. Let $\mathbf{a} \in \mathbb{F}_q^n$ and suppose that $\mathbf{a} \notin D(\mathbf{0})$. Then there exists a non-zero codeword $\mathbf{c} \in \mathscr{C}$ such that $\mathrm{w}_H(\mathbf{a} - \mathbf{c}) < \mathrm{w}_H(\mathbf{a})$. Hence there exists a vector $\mathbf{e} = \mathbf{a} - \mathbf{c} \in \mathbb{F}_q^n$ such that $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{e}} \in I_+(\mathscr{C})$ with $\mathbf{X}^{\mathbf{a}} \succ \mathbf{X}^{\mathbf{e}}$, i.e. $\deg(\mathbf{X}^{\mathbf{a}}) = \mathrm{w}_H(\mathbf{a}) > \mathrm{w}_H(\mathbf{e}) = \deg(\mathbf{X}^{\mathbf{e}})$. Thus, we get that $\mathbf{X}^{\mathbf{a}}$ is a multiple of $\mathrm{LT}_\prec(g_i)$ for some $i = 1, \ldots, s$. Or equivalently, $\mathrm{supp}\left(\Delta\mathbf{g}_i^+\right) \subseteq \mathrm{supp}(\Delta\mathbf{a})$, i.e. $\mathrm{w}_H(\mathbf{a} - \mathbf{g}_i^+) = \mathrm{w}_H(\mathbf{a}) - \mathrm{w}_H(\mathbf{g}_i^+)$. And consequently,

$$\mathrm{w}_H(\mathbf{a} - \mathbf{g}_i^+ + \mathbf{g}_i^-) \leq \mathrm{w}_H(\mathbf{a}) - \mathrm{w}_H(\mathbf{g}_i^+) + \mathrm{w}_H(\mathbf{g}_i^-) < \mathrm{w}_H(\mathbf{a}).$$

Note that the second inequality is due to the fact that $\mathbf{X}^{\mathbf{g}_i^+} \succ \mathbf{X}^{\mathbf{g}_i^-}$. Moreover, on the first inequality, the equality holds if and only if $\mathrm{supp}(\mathbf{a} - \mathbf{g}_i^+) \cap \mathrm{supp}(\mathbf{g}_i^-) = \emptyset$. □

4.3.2 Reduced and Border basis

Definition 4.31. Let $\mathscr{C}$ be an $[n,k]$ linear code defined over the finite field $\mathbb{F}_q$. Associated to the Gröbner representation $(\mathscr{N}, \phi)$ for $\mathscr{C}$ obtained by Algorithm 18 we

can define the *Border basis* $\mathscr{B}(\mathscr{C})$ for $\mathscr{C}$ as the set:

$$\mathscr{B}(\mathscr{C}) = \left\{ \left(\mathbf{n}_1 + \nabla\mathbf{u}_i, \mathbf{n}_2\right) \;\middle|\; \begin{array}{c} \mathbf{n}_1 + \nabla\mathbf{u}_i \neq \mathbf{n}_2,\ \mathbf{n}_1, \mathbf{n}_2 \in \mathscr{N} \\ S(\mathbf{n}_1 + \Delta\mathbf{u}_i) = S(\mathbf{n}_2) \\ i \in \{1, \ldots, n(q-1)\} \end{array} \right\}.$$

Note that both components of an element $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2) \in \mathscr{B}(\mathscr{C})$ belong to the same coset, that is, $\mathbf{b}_1 - \mathbf{b}_2$ is a codeword of the code. Moreover, the Border basis is related with the function Matphi:

$$\mathscr{B}(\mathscr{C}) = \left\{ \left(\mathbf{n} + \nabla\mathbf{u}_i, \phi(\mathbf{n}, \mathbf{u}_i)\right) \;\middle|\; \begin{array}{c} \mathbf{n} \in \mathscr{N}, \\ i \in \{1, \ldots, n(q-1)\} \end{array} \right\} \setminus \{\mathbf{0}\}.$$

Remark 4.32. The Border of a code $\mathscr{C}$ is associated to the $\mathscr{O}$-Border basis of the ideal $I_+(\mathscr{C})$ where $\mathscr{O} = [\mathbf{X}]/\mathrm{LT}_\prec\left(I_+(\mathscr{C})\right)$ (see Definition 1.44).

Let $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2) \in \mathscr{B}(\mathscr{C})$, we define the `head` and the `tail` of $\mathbf{b}$ as:

$$\mathtt{head}(\mathbf{b}) = \mathbf{b}_1 \in \mathbb{F}_q^n \quad \text{and} \quad \mathtt{tail}(\mathbf{b}) = \mathbf{b}_2 \in \mathbb{F}_q^n \ .$$

As pointed above, $\mathtt{head}(\mathbf{b}) - \mathtt{tail}(\mathbf{b})$ is a codeword of $\mathscr{C}$ for all $\mathbf{b} \in \mathscr{B}(\mathscr{C})$.

The information in the Border is somehow redundant. We will now introduce a smaller structure analogous to the *reduced* Gröbner basis of $I(\mathscr{C})$ w.r.t. a degree compatible ordering.

Definition 4.33. The subset $\mathscr{R}(\mathscr{C})$ of the Border basis $\mathscr{B}(\mathscr{C})$ is called the *reduced Border of the code* $\mathscr{C}$ if it fulfills the following conditions:

1. For each element in the Border $\mathbf{b} \in \mathscr{B}(\mathscr{C})$ there exists an element $\mathbf{h} \in \mathscr{R}(\mathscr{C})$ such that $\mathrm{supp}\left(\mathtt{head}(\mathbf{h})\right) \subseteq \mathrm{supp}\left(\mathtt{head}(\mathbf{b})\right)$.

2. For every two different elements $\mathbf{h}_1$ and $\mathbf{h}_2$ in $\mathscr{R}(\mathscr{C})$, neither $\mathrm{supp}\left(\mathtt{head}(\mathbf{h}_1)\right) \subseteq \mathrm{supp}\left(\mathtt{head}(\mathbf{h}_2)\right)$ nor $\mathrm{supp}\left(\mathtt{head}(\mathbf{h}_2)\right) \subseteq \mathrm{supp}\left(\mathtt{head}(\mathbf{h}_1)\right)$ is verified.

Remark 4.34. The definition of reduced basis of the code $\mathscr{C}$ and reduced Gröbner basis of $I(\mathscr{C})$ are very similar. Indeed, let $\mathscr{G}$ be a reduced Gröbner basis of $I(\mathscr{C})$ w.r.t. a degree compatible ordering $\prec$, then $\mathbf{b} \in \mathscr{R}(\mathscr{C})$ if and only if $\mathbf{X}^{\mathrm{head}(\mathbf{b})} - \mathbf{X}^{\mathrm{tail}(\mathbf{b})} \in \mathscr{G}$.

Theorem 4.35. *Let us consider the set of codewords in* $\mathscr{C}$ *given by*

$$M_{\mathrm{Red}_\prec}(\mathscr{C}) = \{\mathrm{head}(\mathbf{b}) - \mathrm{tail}(\mathbf{b}) \mid \mathbf{b} \in \mathscr{R}(\mathscr{C})\}.$$

Then $M_{\mathrm{Red}_\prec}(\mathscr{C})$ *is a test-set for* $\mathscr{C}$.

Proof. This result is a direct consequence of Proposition 4.30 and Remark 4.34. □

Take the following recursive function $\mathrm{cf}: \mathbb{F}_q^n \longrightarrow \mathscr{N}$ where $\mathrm{cf}(\mathbf{0}) = \mathbf{0}$ and

$$\mathrm{cf}(\mathbf{v}) = \phi(\mathrm{cf}(\mathbf{w}), \mathbf{u}_{i_s}) \quad \text{if} \quad \Delta\mathbf{v} = \underbrace{\mathbf{u}_{i_1} + \ldots + \mathbf{u}_{i_{s-1}}}_{\Delta\mathbf{w}} + \mathbf{u}_{i_s},$$

defined in the proof of Theorem 4.14. Note that $\mathrm{Red}(\mathbf{w}, \mathscr{G})$ gives the same result as $\mathrm{cf}(\mathbf{w})$. However, the first one uses reduced border for decreasing while the second one uses either the function Matphi or the Border basis. In [42, 73, 88] it was shown that reduction by means of Matphi or Border basis gives an efficient algorithm. These two ways of understanding the reduction process will lead in the following subsection to two different gradient descent decoding procedures. However, the two algorithms can be seen as two different types of reductions associated to the Gröbner representation of the code.

Remark 4.36. In the binary case, we show in Proposition 2.51 that the set $\mathrm{M}_{\mathrm{Red}_\prec}(\mathscr{C})$ is a subset of the set of codewords of minimal support of $\mathscr{C}$, denoted by $\mathscr{M}_\mathscr{C}$. However for $q \neq 2$, the statement is not true in general. A counterexample will be given in the following lines.

Example 4.37. Let $\mathscr{C}$ be an $[6, 2, 4]$ linear code defined over the finite field $\mathbb{F}_3$ with generator matrix given by:

$$G = \begin{pmatrix} 1 & 0 & 2 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 & 0 & 2 \end{pmatrix} \in \mathbb{F}_3^{2\times 6}.$$

If we compute a reduced Gröbner basis for $I(\mathscr{C})$ w.r.t. a degree reverse lexicographic order, we get 74 elements representing the following nonzero words in $\mathscr{C}$:

$$\begin{array}{cccc} (2,1,2,2,2,1) & (1,2,1,1,1,2) & (2,2,0,1,2,0) & (1,1,0,2,1,0) \\ (2,0,1,0,2,2) & (1,0,2,0,1,1) & (0,1,1,2,0,2) & (0,2,2,1,0,1) \end{array}$$

Note that the first two codewords do not belong to $\mathscr{M}_\mathscr{C}$.

4.3.3 Gradient Descent Decoding

Given an $[n, k]$ linear code $\mathscr{C}$ defined over $\mathbb{F}_q$ and its corresponding Gröbner representation $(\mathscr{N}, \phi)$, we can accomplish two types of Gradient Descent Decoding Algorithms which can be seen as two ways of understanding the reduction associated to the Gröbner representation of the code.

Leader Gradient Descent Decoding

By Theorem 4.16, if a weight compatible ordering $\prec$ is chosen, Algorithm 18 returns a Gröbner representation $(\mathscr{N}, \phi)$ such that the representatives given by $\mathscr{N}$ corresponds to coset leaders of $\mathscr{C}$.

Definition 4.38. We shall define the reduction of an element $\mathbf{n} \in \mathscr{N}$ relative to the unit vector $\mathbf{u}_i$ with $i \in \{1, \ldots, n(q-1)\}$ as the element $\mathbf{n}' = \phi(\mathbf{n}, \mathbf{u}_i) \in \mathscr{N}$, denoted by $\mathbf{n} \longrightarrow_i \mathbf{n}'$.

Therefore, if we write each element $\mathbf{y} \in \mathbb{F}_q^n$ as $\mathbf{y} = \Delta\left(\sum_{j=1}^{s} \mathbf{u}_{i_j}\right)$, we can conclude that iterating a finite number of reduction yields to a representative of the coset $\mathbf{y} + \mathscr{C}$, that is, an element of the subset $\mathrm{CL}(\mathbf{y})$.

These ideas gives us a gradient descent decoding algorithm, described below as Algorithm 19.

Algorithm 19: Leader Gradient Descent Decoding on $\mathbb{F}_q$-linear codes

Data: $(\mathscr{N}, \phi)$ a Gröbner representation for $\mathscr{C}$ obtained by Algorithm 18 and the received word $\mathbf{y} \in \mathbb{F}_q^n$
Result: A codeword $\mathbf{c} \in \mathscr{C}$ that minimized the Hamming distance $d_H(\mathbf{c}, \mathbf{y})$

```
1 y = ∇(u_{i_1} + ... + u_{i_s});
2 n ⟵ 0;
3 for j ⟵ 1 to s
4 |   n ⟶_j n' // i.e. n' = φ(n, u_{i_j});
5 |   n ⟵ n';
6 endfor
7 Return: y − n ∈ C
```

Note that at the end of Algorithm 19 we terminate with a coset leader $\mathbf{n}$ of the coset $\mathbf{y} + \mathscr{C}$. This algorithm is related to the reduction by the function Matphi or the Border basis.

Test-set Gradient Descent Decoding

Given a received word $\mathbf{y} \in \mathbb{F}_q^n$ and suppose that a test-set $\mathscr{T}$ of codewords of $\mathscr{C}$ has been precomputed and stored in the memory. This algorithm recursively search an element $\mathbf{t} \in \mathscr{T}$ such that $\mathrm{w}_H(\mathbf{y} - \mathbf{t}) < \mathrm{w}_H(\mathbf{y})$ and replaces $\mathbf{y}$ by $\mathbf{y}' = \mathbf{y} - \mathbf{t}$.

Algorithm 20: Test-set Gradient Descent Decoding on $\mathbb{F}_q$-linear codes

Data: The received word $\mathbf{y} \in \mathbb{F}_q^n$ and a test-set $\mathscr{T}$ for $\mathscr{C}$
Result: A codeword $\mathbf{c} \in \mathscr{C}$ that minimized the Hamming distance $d_H(\mathbf{c}, \mathbf{y})$

```
1 c ⟵ 0;
2 while there exists t ∈ T such that w_H(y − t) < w_H(y) do
3 |   c ⟵ c + t;
4 |   y ⟵ y + t;
5 endw
6 Return: c
```

Of course, $\mathbf{y} - \mathbf{t}$ belongs to the same coset of $\mathbf{y}$, since $\mathbf{t} \in \mathscr{T} \subseteq \mathscr{C}$. The algorithm terminates when we arrive to a coset leader corresponding to the coset $\mathbf{y} + \mathscr{C}$, that is, when $\mathbf{y}' \in \mathrm{CL}(\mathbf{y})$.

Note that this algorithm requires a nontrivial preprocessing for the construction of a test-set of codewords. By Theorem 4.35 we know that the reduced basis of $\mathscr{C}$, which is related to its Gröbner representation, is a test-set for the code.

The main difference from Algorithm 20 is that it stays entirely in one coset of the code and systematically seeks out a coset leader, while Algorithm 19 changes between different coset of $\mathbb{F}_q^n/\mathscr{C}$ until it arrives to the $\mathbf{0}$-coset, i.e. the code itself.

4.4 FGLM technique to compute a Gröbner basis

In this section we present an algorithm to compute a reduced Gröbner basis of the ideal $I_+(\mathscr{C})$ which is associated to a linear code $\mathscr{C}$ defined over the finite field $\mathbb{F}_q$. This algorithm goes back to the work of Faugère et al. [42] and generalizes that of [11, 45, 46].

We will follow the notation of the theory of Gröbner Bases for submodules introduced in Section 3.3.

The main idea of the algorithm is to fix a generator matrix $G \in \mathbb{F}_q^{k\times n}$ of the $[n,k]$ linear code $\mathscr{C}$ defined over $\mathbb{F}_q$, whose rows are labelled by $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\}$. Then consider the set of binomials

$$F = \left\{ f_{ij} = \mathbf{X}^{\alpha^j \mathbf{w}_i} - 1 \right\}_{\substack{i=1,\ldots,k \\ j=1,\ldots,q-1}} \subseteq \mathbb{K}[\mathbf{X}].$$

By Theorem 4.3, the set $F \cup \left\{ \mathscr{R}_{X_i}\left(T_+\right) \right\}_{i=1,\ldots,n}$ generates the ideal $I_+(\mathscr{C})$. We claim that the set

$$\mathscr{G}_1 = \left\{ g_{ij} = \mathbf{v}_1 f_{ij} + \mathbf{v}_{((i-1)(q-1)+j+1)} \right\}_{\substack{i=1,\ldots,k \\ j=1,\ldots,q-1}},$$

where $\mathbf{v}_i$ denotes the unit vector of length $k(q-1)+1$ with a one in the i-th position, is a basis for the syzygy module M in $\mathbb{K}[\mathbf{X}]^{k(q-1)+1}$ with generating set

$$\hat{F} = \left\{ -1, f_{11},\ldots,f_{1q-1},\ldots,f_{k1},\ldots,f_{kq-1} \right\}.$$

Where the binomials $\left\{ \mathscr{R}_{X_i}\left(T_+\right) \right\}_{i=1,\ldots,n}$ are considered implicit on the operations.

Moreover, $\mathscr{G}_1$ is a Gröbner basis of M relative to a POT ordering $\prec_{\mathbf{w}}$ induced by an ordering $\prec$ in $\mathbb{K}[\mathbf{X}]$ and the weight vector

$$\mathbf{w} = (1, \mathrm{LT}_{\prec}(f_{11}),\ldots,\mathrm{LT}_{\prec}(f_{kq-1})).$$

Note that the leading term of g_{ij} with respect to $\prec_{\mathbf{w}}$ is $\mathbf{v}_{((i-1)(q-1)+j+1)}$.

Therefore, using the generalized FGLM algorithm [42] and running through the terms of $\mathbb{K}[\mathbf{X}]^{k(q-1)+1}$ using a TOP ordering we obtain a Gröbner basis $\mathscr{G}_2$ relative to a new order. Note that each syzygy corresponds to a solution of the following equation:

$$-\beta_0 + \sum_{i=1}^{k}\sum_{j=1}^{q-1} \beta_{(i-1)(q-1)+j} f_{ij} = 0 \ \text{ with } \beta_i \in \mathbb{K}[\mathbf{X}] \text{ for } i = 1,\ldots,k(q-1).$$

Hence, the first component of any syzygy of the module M indicates an element of the ideal generated by F. Moreover, it is immediate that the normal form with respect

to $\mathscr{G}_1$ of any element is zero except in the first component, that is to say, the linear combinations that Fitzpatrick's algorithm [45] look for, take place in this component.

Let $r = k(q-1)$. Three structures are used in the algorithm:

- The list `List` whose elements are of the specific type $\mathbf{v} = (\mathbf{v}[1], \mathbf{v}[2])$ where $\mathbf{v}[2]$ represents an element in $\mathbb{K}[\mathbf{X}]$ which can be expressed as

$$\mathbf{v}[2] = \lambda + \sum_{i=1}^{r} \lambda_i f_i \text{ with } \lambda, \lambda_1, \ldots, \lambda_r \in \mathbb{K}[\mathbf{X}].$$

  Thus, the coefficient vector $(\lambda, \lambda_1, \ldots, \lambda_r) \in \mathbb{K}[\mathbf{X}]^{r+1}$ is the associated vector of $\mathbf{v}[2]$ on the module M. Then $\mathbf{v}[1]$ represents the first component of such vector.

- The list G_T which ends up being a reduced Gröbner basis of $I_+(\mathscr{C})$ w.r.t. a degree compatible ordering $\prec_T$.

- The list $\mathscr{N}$ of terms that are reduced with respect to G_T, i.e. the set of standard monomials.

We also require the following subroutines:

- `InsertNexts(w, List)` inserts the product wx for $x \in [\mathbf{X}]$ in `List` and removes the duplicates, where the binomials of $\left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n}$ are considered as implicit in the computation. Then the elements of `List` are sorted by increasing order w.r.t. $\prec_T$ in the first component of the pairs and in case of equality by comparing the second component. Recall that

$$\mathbf{X} = \{X_1, \ldots, X_n\} = \{x_{11}, \ldots, x_{1q-1}, \ldots, x_{n1}, \ldots, x_{nq-1}\}.$$

- `NextTerm(List)` removes the first element from the list `List` and returns it.

- `Member(v,[`$\mathbf{v}_1, \ldots, \mathbf{v}_r$`])` returns j if $\mathrm{v} = \mathbf{v}_j$ or `false` otherwise.

Remark 4.39. Note that the computation of $\mathbf{X}^{\mathbf{a}} x_{ij}$ modulo the ideal $\mathscr{R}_{\mathbf{X}}(T_+)$, with $\mathbf{a} \in \mathbb{F}_q^n$, acts like the operation $\mathbf{a} + \alpha^j \mathbf{e}_i$ in the finite field $\mathbb{F}_q^n$ where $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ denotes a standard basis of $\mathbb{F}_q^n$.

Theorem 4.40. *Algorithm 21 computes a reduced Gröbner basis of the ideal associated to a given linear code $\mathscr{C}$ of parameters $[n,k]$ defined over $\mathbb{F}_q$.*

Proof. The proof of the algorithm is an extension of that in [45, Algorithm2.1] and therefore, is also a generalization of the FGLM algorithm [42]. Let $G \in \mathbb{F}_q^{k \times n}$ be a generator matrix of $\mathscr{C}$. We label the rows of G by $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\} \subseteq \mathbb{F}_q^n$.

By Theorem 4.3 the ideal associated to the linear code $\mathscr{C}$ may be defined as the following binomial ideal:

$$\begin{aligned} I_+(\mathscr{C}) &= \left\langle \left\{\mathbf{X}^{\alpha^j \mathbf{w}_i}\right\}_{\substack{i=1,\ldots,k \\ j=1,\ldots,q-1}} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n} \right\rangle \\ &= \left\langle \{f_1, \ldots, f_s\} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n} \right\rangle. \end{aligned}$$

Algorithm 21: Adapted FGLM algorithm for $I_+(\mathscr{C})$

Data: The rows $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\} \subseteq \mathbb{F}_q^n$ of a generator matrix of an $[n,k]$ linear code $\mathscr{C}$ defined over $\mathbb{F}_q$ and a degree compatible ordering $<_T$ on $\mathbb{K}[\mathbf{X}]$.
Result: A reduced Gröbner basis G_T of the ideal $I_+(\mathscr{C})$ w.r.t. $<_T$.

1 $\texttt{List} \longleftarrow \left[(1,1), \left\{(1,\alpha^j\mathbf{w}_i)\right\}_{\substack{i=1,\ldots,k \\ j=1,\ldots,q-1}}\right]$; $G_T \longleftarrow \emptyset$; $\mathscr{N} \longleftarrow \emptyset$; $r \longleftarrow 0$;
2 **while** $\texttt{List} \neq \emptyset$ **do**
3 $\mathbf{w} \longleftarrow \texttt{NextTerm(List)}$;
4 **if** $\mathbf{w}[1] \notin \mathrm{LT}_{<_T}(G_T)$ **then**
5 $j = \texttt{Member}(\mathbf{w}[2], [\mathbf{v}_1[2],\ldots,\mathbf{v}_r[2]])$;
6 **if** $j \neq \texttt{false}$ **then**
7 $G_T \longleftarrow G_T \cup \{\mathbf{w}[1] - \mathbf{v}_j[1]\}$;
8 **for** $i = 1$ *to* r
9 **if** $\mathbf{v}_i[1]$ *is a multiple of* $\mathbf{w}[1]$ **then**
10 Removes $\mathbf{v}_i[1]$ from $\mathscr{N}$
11 **endif**
12 **endfor**
13 **else**
14 $r \longleftarrow r+1$;
15 $\mathbf{v}_r \longleftarrow \mathbf{w}$;
16 $\mathscr{N} \longleftarrow \mathscr{N} \cup \{\mathbf{v}_r[1]\}$;
17 $\texttt{List} = \texttt{InsertNexts}(\mathbf{w}, \texttt{List})$;
18 **endif**
19 **endif**
20 **endw**

We first show that G_T is a subset of binomials of the ideal $I_+(\mathscr{C})$. The proof is based on the following observation: $\mathbf{X}^\mathbf{a} - \mathbf{X}^\mathbf{b} \in G_T$ if and only if it corresponds to the first component of a syzygy in the module M. In other words,

$$\mathbf{X}^\mathbf{a} - \mathbf{X}^\mathbf{b} = \sum_{i=1}^{s} \lambda_i f_i \text{ with } \lambda_i \in \mathbb{K}[\mathbf{X}]\ ,\ i = 1,\ldots,r,$$

or equivalently, $\mathbf{X}^\mathbf{a} - \mathbf{X}^\mathbf{b} \in I_+(\mathscr{C})$.

Moreover, we claim that the initial ideal of $I_+(\mathscr{C})$ is generated by the leading terms of polynomials in G_T. Indeed, by Theorem 4.3, any binomial $f(\mathbf{X})$ of the ideal $I_+(\mathscr{C})$ can be written uniquely as a linear combination of vectors in the generator set $F = \{f_1,\ldots,f_s\}$ modulo the ideal $\mathscr{R}_\mathbf{X}(T_+)$, i.e.

$$f(\mathbf{X}) = \sum_{i=1}^{s} \lambda_i f_i(\mathbf{X}) \mod \mathscr{R}_\mathbf{X}(T_+) \text{ with } \lambda_i \in \mathbb{K}[\mathbf{X}].$$

Therefore, $\mathrm{LT}_{<_T}(f(\mathbf{X}))$ is a multiple of the leading term of an element $f_r(\mathbf{X}) =$

$\mathbf{X}^{\alpha^j \mathbf{w}_i} - 1 \in F$ that appears on its decomposition. But the leading term of any element $f_r = \mathbf{X}^{\alpha^j \mathbf{w}_i} - 1 \in F$ cannot be in $\mathscr{N}$. To see this, note that the first element introduced in the set $\mathscr{N}$ is always $(1,1)$. Moreover,

$$1 = \mathbf{X}^{\alpha^j \mathbf{w}_i} + f_r \text{ i.e. } \mathrm{Red}_{\prec}(\mathbf{X}^{\alpha^j \mathbf{w}_i}, F) = 1 \quad \text{and} \quad 1 = 1 + \textstyle\sum_{i=1}^{s} 0 \cdot f_i$$

i.e. $\mathrm{Red}_{\prec}(1, F) = 1$ which implies that $\mathbf{X}^{\alpha^j \mathbf{w}_i} - 1 \in G_T$.

By definition, G_T is reduced since we only consider on the algorithm terms which are not divisible by any leading term of the Gröbner basis.

Finally, since $I_+(\mathscr{C})$ has finite dimension, then the number of terms in $\mathscr{N}$ is bounded. Note that at each iteration of the main loop either the size of `List` decreases or the size of $\mathscr{N}$ increases, thus there are only a finite number of iterations. This completes the proof of the algorithm. □

Remark 4.41. Recall that the dimension of the quotient vector space $\mathbb{F}_q^n / \mathscr{C}$ is q^{n-k}. Moreover, if $\mathscr{C}$ can correct up to t errors, then every word $\mathbf{e}$ of weight $\mathrm{w}_H(\mathbf{e}) \leq t$ is the unique coset leader (vectors of minimal weight in their cosets) of its coset modulo $\mathscr{C}$. In other words, all monomials of degree less than t modulo the ideal $\mathscr{R}_{\mathbf{X}}(T_+)$ should be a standard monomials for G_T.

Note that the writing rules given by the ideal $\mathscr{R}_{\mathbf{X}}(T_+)$ implies that "the exponent of each variable x_{ij} is 0 or 1" and "two different variables x_{ij} and x_{il} can not appear in a monomial". Thus, the number of standard monomials of a t-error correcting code is at least

$$M = \sum_{l=1}^{t} (q-1)^l \binom{n}{l}. \tag{4.4}$$

Accordingly, if $q^{n-k} = M$, then all cosets has a unique coset leader of weight smaller or equal to t. Codes that achieve this equality are the so-called *perfect codes*. Otherwise, there must appear some cosets leaders of weight at most $\rho(\mathscr{C})$, where $\rho(\mathscr{C})$ denotes the covering radius of $\mathscr{C}$, but never as the unique leader, or equivalently there exists standard monomials of degree up to $\rho(\mathscr{C})$. Recall that $\rho(\mathscr{C})$ coincide with the largest weight among all the cosets of $\mathscr{C}$.

By Proposition 4.22 in the worst case, a minimal generator of the initial ideal $\mathrm{in}_{\prec}(I_+(\mathscr{C}))$ has degree $\rho(\mathscr{C}) + 1$ where $\prec$ is a degree compatible ordering.

Theorem 4.42. *Let $\mathscr{C}$ be an $[n,k]$ linear code that corrects up to t errors and M be defined by Equation 4.4. If the basis field operations need an unit time, then Algorithm 21 needs a total time of $\mathscr{O}\left(Dn^2(q-1)\right)$, where*

$$D = \begin{cases} \sum_{i=1}^{t+1} (q-1)^i \binom{n}{i}, & \text{if } \mathscr{C} \text{ is a perfect code.} \\ \sum_{i=1}^{\rho(\mathscr{C})+1} (q-1)^i \binom{n}{i}, & \text{otherwise.} \end{cases}$$

Proof. The main time of the algorithm is devoted to the management of `InsertNexts`. In each main loop iteration this function first introduces $n(q-1)$ new elements to the list `List`, then compares all the elements and finally eliminates redundancy.

Note that comparing two monomials in $\mathbb{K}[\mathbf{X}]$ is equivalent to comparing vectors in $\mathbb{F}_q^n$, thus we need a time of $\mathcal{O}(n)$.

At iteration i, after inserting the new elements in the list `List` we would have at most D_i elements where

$$D_i = \underbrace{(q-1)k}_{\substack{\text{Elements that}\\ \text{initialized List}}} + \underbrace{i\left(n(q-1)\right)-i}_{\substack{\text{At each iteration}\\ \text{the first element is removed}\\ \text{and we add n(q-1)new elements}}}$$

By Remark 4.41 we have an upper bound D for the number of times that `InsertNexts` should be called. This gives a total time of

$$\mathcal{O}\left(n\left((q-1)k + D\left(n(q-1)\right) - D\right)\right) \sim \mathcal{O}\left(Dn^2(q-1)\right).$$

□

The following example shows how Algorithm 21 works.

Example 4.43. Consider $\mathscr{C}$ the $[4,2,3]$ ternary code with generator matrix

$$G_{\mathscr{C}} = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 2 \end{pmatrix} \in \mathbb{F}_3^{2\times 4}$$

We find that

$$I_+(\mathscr{C}) = \left\langle \begin{Bmatrix} x_{12}x_{32}x_{41}-1, & x_{11}x_{31}x_{42}-1 \\ x_{22}x_{31}x_{41}-1, & x_{21}x_{32}x_{42}-1 \end{Bmatrix} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,2,3,4} \right\rangle$$

with $\mathscr{R}_{X_i}(T_+) = \left\{x_{i1}^2 - x_{i2}, x_{i1}x_{i2}-1, x_{i2}^2 - x_{i1}\right\}$ for $i = 1,2,3,4$.

We compute the reduced Gröbner basis of $I_+(\mathscr{C})$ w.r.t. the `degrevlex` order with $\underbrace{x_{11} < x_{12}}_{X_1} < \underbrace{x_{21} < x_{22}}_{X_2} < \underbrace{x_{31} < x_{32}}_{X_3} < \underbrace{x_{41} < x_{42}}_{X_4}$.

We get 36 binomials representing all the codewords of the code:

$$\begin{Bmatrix} x_{11}x_{22}-x_{31}, & x_{11}x_{32}-x_{21}, & x_{22}x_{32}-x_{12}, \\ x_{12}x_{21}-x_{32}, & x_{12}x_{31}-x_{22}, & x_{21}x_{31}-x_{11} \end{Bmatrix} \rightarrow \begin{Bmatrix} (2,1,1,0) \\ (1,2,2,0) \end{Bmatrix}$$

$$\begin{Bmatrix} x_{11}x_{21}-x_{42}, & x_{11}x_{41}-x_{22}, & x_{21}x_{41}-x_{12}, \\ x_{12}x_{22}-x_{41}, & x_{12}x_{42}-x_{21}, & x_{22}x_{42}-x_{11} \end{Bmatrix} \rightarrow \begin{Bmatrix} (2,2,0,2) \\ (1,1,0,1) \end{Bmatrix}$$

$$\begin{Bmatrix} x_{11}x_{31}-x_{41}, & x_{11}x_{42}-x_{32}, & x_{31}x_{42}-x_{12}, \\ x_{12}x_{32}-x_{42}, & x_{12}x_{41}-x_{31}, & x_{32}x_{41}-x_{11} \end{Bmatrix} \rightarrow \begin{Bmatrix} (2,0,2,1) \\ (1,0,1,2) \end{Bmatrix}$$

$$\begin{Bmatrix} x_{22}x_{31}-x_{42}, & x_{22}x_{41}-x_{32}, & x_{31}x_{41}-x_{21}, \\ x_{21}x_{32}-x_{41}, & x_{21}x_{42}-x_{31}, & x_{32}x_{42}-x_{22} \end{Bmatrix} \rightarrow \begin{Bmatrix} (0,1,2,2) \\ (0,2,1,1) \end{Bmatrix}$$

$$\mathscr{R}_{X_i}\left(T_+\right) \text{ for } i=1,2,3,4 \quad \longrightarrow \quad (0,0,0,0)$$

Now we apply Algorithm 21 and we compare the obtained result.
The algorithm is initialized with the following data:

	$\mathbb{F}_3^n$-Representation of the exponent of $\mathbf{v}[2]$					
$\mathbf{v}[1]$	1	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
1	1					1
1		1	0	1	2	$x_{12}x_{32}x_{41}$
1		0	1	2	2	$x_{22}x_{31}x_{41}$
1		2	0	2	1	$x_{11}x_{31}x_{42}$
1		0	2	1	1	$x_{21}x_{32}x_{42}$

Note that at each step, we add new elements to the list `List` which we represent in a table. Moreover, at each Step the first element form the the list `List` is removed by the subroutine `NextTerm` and the elements are sorted by increasing order w.r.t. the `deglex` order in the first component of the pairs, and in case of equality by comparing the second component.

1. $\texttt{NextTerm}(\texttt{List}) = (1,1)$

   $\mathbf{v}_1 = (1,1)$

   $\mathscr{N} = \{1\}$

   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_1, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
x_{11}	2				x_{11}
x_{12}	1				x_{12}
x_{21}		2			x_{21}
x_{22}		1			x_{22}
x_{31}			2		x_{31}
x_{32}			1		x_{32}
x_{41}				2	x_{41}
x_{42}				1	x_{42}

2. $\texttt{NextTerm}(\texttt{List}) = (1, x_{11}x_{31}x_{42})$

   $\mathbf{v}_2 = (1, x_{11}x_{31}x_{42})$

   $\mathscr{N} = \{1\}$

   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_2, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
x_{11}	1	0	2	1	$x_{12}x_{31}x_{42}$
x_{12}	0	0	2	1	$x_{31}x_{42}$
x_{21}	2	2	2	1	$x_{11}x_{21}x_{31}x_{42}$
x_{22}	2	1	2	1	$x_{11}x_{22}x_{31}x_{42}$
x_{31}	2	0	1	1	$x_{11}x_{32}x_{42}$
x_{32}	2	0	0	1	$x_{11}x_{42}$
x_{41}	2	0	2	0	$x_{11}x_{31}$
x_{42}	2	0	2	2	$x_{11}x_{31}x_{41}$

3. $\texttt{NextTerm}(\texttt{List}) = (1, x_{12}x_{32}x_{41})$

   $\mathbf{v}_3 = (1, x_{12}x_{32}x_{41})$

   $\mathscr{N} = \{1\}$

   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_3, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
x_{11}	0	0	1	2	$x_{32}x_{41}$
x_{12}	2	0	1	2	$x_{11}x_{32}x_{41}$
x_{21}	1	2	1	2	$x_{12}x_{21}x_{32}x_{41}$
x_{22}	1	1	1	2	$x_{12}x_{22}x_{32}x_{41}$
x_{31}	1	0	0	2	$x_{12}x_{41}$
x_{32}	1	0	2	2	$x_{12}x_{31}x_{41}$
x_{41}	1	0	1	1	$x_{12}x_{32}x_{42}$
x_{42}	1	0	1	0	$x_{12}x_{32}$

4. $\texttt{NextTerm}(\texttt{List}) = (1, x_{21}x_{32}x_{42})$
   $\mathbf{v}_4 = (1, x_{21}x_{32}x_{42})$
   $\mathscr{N} = \{1\}$
   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_4, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
x_{11}	2	2	1	1	$x_{11}x_{21}x_{32}x_{42}$
x_{12}	1	2	1	1	$x_{12}x_{21}x_{32}x_{42}$
x_{21}	0	1	1	1	$x_{22}x_{32}x_{42}$
x_{22}	0	0	1	1	$x_{32}x_{42}$
x_{31}	0	2	0	1	$x_{21}x_{42}$
x_{32}	0	2	2	1	$x_{21}x_{31}x_{42}$
x_{41}	0	2	1	0	$x_{21}x_{32}$
x_{42}	0	2	1	2	$x_{21}x_{32}x_{41}$

5. $\texttt{NextTerm}(\texttt{List}) = (1, x_{22}x_{31}x_{41})$
   $\mathbf{v}_5 = (1, x_{22}x_{31}x_{41})$
   $\mathscr{N} = \{1\}$
   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_5, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
x_{11}	2	1	2	2	$x_{11}x_{22}x_{31}x_{41}$
x_{12}	1	1	2	2	$x_{12}x_{22}x_{31}x_{41}$
x_{21}	0	0	2	2	$x_{31}x_{41}$
x_{22}	0	2	2	2	$x_{21}x_{31}x_{41}$
x_{31}	0	1	1	2	$x_{22}x_{32}x_{41}$
x_{32}	0	1	0	2	$x_{22}x_{41}$
x_{41}	0	1	2	1	$x_{22}x_{31}x_{42}$
x_{42}	0	1	2	0	$x_{22}x_{31}$

6. $\texttt{NextTerm}(\texttt{List}) = (x_{11}, x_{11})$
   $\mathbf{v}_6 = (x_{11}, x_{11})$
   $\mathscr{N} = \{1, x_{11}\}$
   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_6, \texttt{List})$.

	$\mathbb{F}_3^n$-Representation of the exponent of v[2]					
v[1]	1	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{11}$		1	0	0	0	x_{12}
$x_{11}x_{12}$	1	0	0	0	0	1
$x_{11}x_{21}$		2	2	0	0	$x_{11}x_{21}$
$x_{11}x_{22}$		2	1	0	0	$x_{11}x_{22}$
$x_{11}x_{31}$		2	0	2	0	$x_{11}x_{31}$
$x_{11}x_{32}$		2	0	1	0	$x_{11}x_{32}$
$x_{11}x_{41}$		2	0	0	2	$x_{11}x_{41}$
$x_{11}x_{42}$		2	0	0	1	$x_{11}x_{42}$

7. $\texttt{NextTerm}(\texttt{List}) = (x_{11}, x_{32}x_{41})$
   $\mathbf{v}_7 = (x_{11}, x_{32}x_{41})$
   $\mathscr{N} = \{1, x_{11}\}$
   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_7, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{11}$	2	0	1	2	$x_{11}x_{32}x_{41}$
$x_{11}x_{12}$	1	0	1	2	$x_{12}x_{32}x_{41}$
$x_{11}x_{21}$	0	2	1	2	$x_{21}x_{32}x_{41}$
$x_{11}x_{22}$	0	1	1	2	$x_{11}x_{32}x_{41}$
$x_{11}x_{31}$	0	0	0	2	x_{41}
$x_{11}x_{32}$	0	0	2	2	$x_{31}x_{41}$
$x_{11}x_{41}$	0	0	1	1	$x_{32}x_{42}$
$x_{11}x_{42}$	0	0	1	0	x_{32}

8. $\texttt{NextTerm}(\texttt{List}) = (x_{11}, x_{12}x_{31}x_{42})$
   $\mathbf{v}_8 = (x_{11}, x_{12}x_{31}x_{42})$
   $\mathscr{N} = \{1, x_{11}\}$
   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_8, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{11}$	0	0	2	1	$x_{31}x_{42}$
$x_{11}x_{12}$	1	0	2	1	$x_{12}x_{31}x_{42}$
$x_{11}x_{21}$	1	2	2	1	$x_{12}x_{21}x_{31}x_{42}$
$x_{11}x_{22}$	1	1	2	1	$x_{12}x_{22}x_{31}x_{42}$
$x_{11}x_{31}$	1	0	1	1	$x_{12}x_{32}x_{42}$
$x_{11}x_{32}$	1	0	0	1	$x_{12}x_{42}$
$x_{11}x_{41}$	1	0	2	0	$x_{12}x_{31}$
$x_{11}x_{42}$	1	0	2	2	$x_{12}x_{31}x_{41}$

9. $\texttt{NextTerm}(\texttt{List}) = (x_{11}, x_{11}x_{21}x_{32}x_{42})$
   $\mathbf{v}_9 = (x_{11}, x_{11}x_{21}x_{32}x_{42})$
   $\mathscr{N} = \{1, x_{11}\}$
   $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_9, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{11}$	1	2	1	1	$x_{12}x_{21}x_{32}x_{42}$
$x_{11}x_{12}$	0	2	1	1	$x_{21}x_{32}x_{42}$
$x_{11}x_{21}$	2	1	1	1	$x_{11}x_{22}x_{32}x_{42}$
$x_{11}x_{22}$	2	0	1	1	$x_{11}x_{32}x_{42}$
$x_{11}x_{31}$	2	2	0	1	$x_{11}x_{21}x_{42}$
$x_{11}x_{32}$	2	2	2	1	$x_{11}x_{21}x_{31}x_{42}$
$x_{11}x_{41}$	2	2	1	0	$x_{11}x_{21}x_{32}$
$x_{11}x_{42}$	2	2	1	2	$x_{11}x_{21}x_{32}x_{41}$

10. $\texttt{NextTerm}(\texttt{List}) = (x_{11}, x_{11}x_{22}x_{31}x_{41})$
    $\mathbf{v}_{10} = (x_{11}, x_{11}x_{22}x_{31}x_{41})$
    $\mathscr{N} = \{1, x_{11}\}$
    $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{10}, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{11}$	1	1	2	2	$x_{12}x_{22}x_{31}x_{41}$
$x_{11}x_{12}$	0	1	2	2	$x_{22}x_{31}x_{41}$
$x_{11}x_{21}$	2	0	2	2	$x_{11}x_{31}x_{41}$
$x_{11}x_{22}$	2	2	2	2	$x_{11}x_{21}x_{31}x_{41}$
$x_{11}x_{31}$	2	1	1	2	$x_{11}x_{22}x_{32}x_{41}$
$x_{11}x_{32}$	2	1	0	2	$x_{11}x_{22}x_{41}$
$x_{11}x_{41}$	2	1	2	1	$x_{11}x_{22}x_{31}x_{42}$
$x_{11}x_{42}$	2	1	2	0	$x_{11}x_{22}x_{31}$

11. $\texttt{NextTerm}(\texttt{List}) = (x_{12}, x_{12})$
    $\mathbf{v}_{11} = (x_{12}, x_{12})$
    $\mathscr{N} = \{1, x_{11}, x_{12}\}$
    $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{11}, \texttt{List})$.

	$\mathbb{F}_3^n$-Representation of the exponent of v[2]					
v[1]	1	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{12}$	1	0	0	0	0	1
$x_{12}x_{12}$		2	0	0	0	x_{11}
$x_{12}x_{21}$		1	2	0	0	$x_{12}x_{21}$
$x_{12}x_{22}$		1	1	0	0	$x_{12}x_{22}$
$x_{12}x_{31}$		1	0	2	0	$x_{12}x_{31}$
$x_{12}x_{32}$		1	0	1	0	$x_{12}x_{32}$
$x_{12}x_{41}$		1	0	0	2	$x_{12}x_{41}$
$x_{12}x_{42}$		1	0	0	1	$x_{12}x_{42}$

12. $\texttt{NextTerm}(\texttt{List}) = (x_{12}, x_{31}x_{42})$
    $\mathbf{v}_{12} = (x_{12}, x_{31}x_{42})$
    $\mathscr{N} = \{1, x_{11}, x_{12}\}$
    $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{12}, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{12}$	2	0	2	1	$x_{11}x_{31}x_{42}$
$x_{12}x_{12}$	1	0	2	1	$x_{12}x_{31}x_{42}$
$x_{12}x_{21}$	0	2	2	1	$x_{21}x_{31}x_{42}$
$x_{12}x_{22}$	0	1	2	1	$x_{22}x_{31}x_{42}$
$x_{12}x_{31}$	0	0	1	1	$x_{32}x_{42}$
$x_{12}x_{32}$	0	0	0	1	x_{42}
$x_{12}x_{41}$	0	0	2	0	x_{31}
$x_{12}x_{42}$	0	0	2	2	$x_{31}x_{41}$

13. $\texttt{NextTerm}(\texttt{List}) = (x_{12}, x_{11}x_{32}x_{41})$
    $\mathbf{v}_{13} = (x_{12}, x_{11}x_{32}x_{41})$
    $\mathscr{N} = \{1, x_{11}, x_{12}\}$
    $\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{13}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{12}$	1	0	1	2	$x_{12}x_{32}x_{41}$
$x_{12}x_{12}$	0	0	1	2	$x_{32}x_{41}$
$x_{12}x_{21}$	2	2	1	2	$x_{11}x_{21}x_{32}x_{41}$
$x_{12}x_{22}$	2	1	1	2	$x_{11}x_{22}x_{32}x_{41}$
$x_{12}x_{31}$	2	0	0	2	$x_{11}x_{41}$
$x_{12}x_{32}$	2	0	2	2	$x_{11}x_{31}x_{41}$
$x_{12}x_{41}$	2	0	1	1	$x_{11}x_{32}x_{42}$
$x_{12}x_{42}$	2	0	1	0	$x_{11}x_{32}$

14. $\texttt{NextTerm}(\texttt{List}) = (x_{12}, x_{12}x_{21}x_{32}x_{42})$
$\mathbf{v}_{14} = (x_{12}, x_{12}x_{21}x_{32}x_{42})$
$\mathcal{N} = \{1, x_{11}, x_{12}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{14}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{12}$	0	2	1	1	$x_{21}x_{32}x_{42}$
$x_{12}x_{12}$	2	2	1	1	$x_{11}x_{21}x_{32}x_{42}$
$x_{12}x_{21}$	1	1	1	1	$x_{12}x_{22}x_{32}x_{42}$
$x_{12}x_{22}$	1	0	1	1	$x_{12}x_{32}x_{42}$
$x_{12}x_{31}$	1	2	0	1	$x_{12}x_{21}x_{42}$
$x_{12}x_{32}$	1	2	2	1	$x_{12}x_{21}x_{31}x_{42}$
$x_{12}x_{41}$	1	2	1	0	$x_{12}x_{21}x_{32}$
$x_{12}x_{42}$	1	2	1	2	$x_{12}x_{21}x_{32}x_{41}$

15. $\texttt{NextTerm}(\texttt{List}) = (x_{12}, x_{12}x_{22}x_{31}x_{41})$
$\mathbf{v}_{15} = (x_{12}, x_{12}x_{22}x_{31}x_{41})$
$\mathcal{N} = \{1, x_{11}, x_{12}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{15}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{12}$	0	1	2	2	$x_{22}x_{31}x_{41}$
$x_{12}x_{12}$	2	1	2	2	$x_{11}x_{22}x_{31}x_{41}$
$x_{12}x_{21}$	1	0	2	2	$x_{12}x_{31}x_{41}$
$x_{12}x_{22}$	1	2	2	2	$x_{12}x_{21}x_{31}x_{41}$
$x_{12}x_{31}$	1	1	1	2	$x_{12}x_{22}x_{32}x_{41}$
$x_{12}x_{32}$	1	1	0	2	$x_{12}x_{22}x_{41}$
$x_{12}x_{41}$	1	1	2	1	$x_{12}x_{22}x_{31}x_{42}$
$x_{12}x_{42}$	1	1	2	0	$x_{12}x_{22}x_{31}$

16. $\texttt{NextTerm}(\texttt{List}) = (x_{21}, x_{21})$
$\mathbf{v}_{16} = (x_{21}, x_{21})$
$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{16}, \texttt{List})$.

$\mathbf{v}[1]$	$\mathbb{F}_3^n$-Representation of the exponent of $\mathbf{v}[2]$					$\mathbf{v}[2]$
	1	X_1	X_2	X_3	X_4	
$x_{11}x_{21}$		2	2	0	0	$x_{11}x_{21}$
$x_{12}x_{21}$		1	2	0	0	$x_{12}x_{21}$
$x_{21}x_{21}$		0	1	0	0	x_{22}
$x_{21}x_{22}$	1	0	0	0	0	1
$x_{21}x_{31}$		0	2	2	0	$x_{21}x_{31}$
$x_{21}x_{32}$		0	2	1	0	$x_{21}x_{32}$
$x_{21}x_{41}$		0	2	0	2	$x_{21}x_{41}$
$x_{21}x_{42}$		0	2	0	1	$x_{21}x_{42}$

17. $\texttt{NextTerm}(\texttt{List}) = (x_{21}, x_{31}x_{41})$
$\mathbf{v}_{17} = (x_{21}, x_{31}x_{41})$
$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{17}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{21}$	2	0	2	2	$x_{11}x_{31}x_{41}$
$x_{12}x_{21}$	1	0	2	2	$x_{12}x_{31}x_{41}$
$x_{21}x_{21}$	0	2	2	2	$x_{21}x_{31}x_{41}$
$x_{21}x_{22}$	0	1	2	2	$x_{22}x_{31}x_{41}$
$x_{21}x_{31}$	0	0	1	2	$x_{32}x_{41}$
$x_{21}x_{32}$	0	0	0	2	x_{41}
$x_{21}x_{41}$	0	0	2	1	$x_{31}x_{42}$
$x_{21}x_{42}$	0	0	2	0	x_{31}

18. $\texttt{NextTerm}(\texttt{List}) = (x_{21}, x_{22}x_{32}x_{42})$
$\mathbf{v}_{18} = (x_{21}, x_{22}x_{32}x_{42})$
$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{18}, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{21}$	2	1	1	1	$x_{11}x_{22}x_{32}x_{42}$
$x_{12}x_{21}$	1	1	1	1	$x_{12}x_{22}x_{32}x_{42}$
$x_{21}x_{21}$	0	0	1	1	$x_{32}x_{42}$
$x_{21}x_{22}$	0	2	1	1	$x_{21}x_{32}x_{42}$
$x_{21}x_{31}$	0	1	0	1	$x_{22}x_{42}$
$x_{21}x_{32}$	0	1	2	1	$x_{22}x_{31}x_{42}$
$x_{21}x_{41}$	0	1	1	0	$x_{22}x_{32}$
$x_{21}x_{42}$	0	1	1	2	$x_{22}x_{32}x_{41}$

19. $\texttt{NextTerm}(\texttt{List}) = (x_{21}, x_{11}x_{21}x_{31}x_{42})$
$\mathbf{v}_{19} = (x_{21}, x_{11}x_{21}x_{31}x_{42})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{19}, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{21}$	1	2	2	1	$x_{12}x_{21}x_{31}x_{42}$
$x_{12}x_{21}$	0	2	2	1	$x_{21}x_{31}x_{42}$
$x_{21}x_{21}$	2	1	2	1	$x_{11}x_{22}x_{31}x_{42}$
$x_{21}x_{22}$	2	0	2	1	$x_{11}x_{31}x_{42}$
$x_{21}x_{31}$	2	2	1	1	$x_{11}x_{21}x_{32}x_{42}$
$x_{21}x_{32}$	2	2	0	1	$x_{11}x_{21}x_{42}$
$x_{21}x_{41}$	2	2	2	0	$x_{11}x_{21}x_{31}$
$x_{21}x_{42}$	2	2	2	2	$x_{11}x_{21}x_{31}x_{41}$

20. $\texttt{NextTerm}(\texttt{List}) = (x_{21}, x_{12}x_{21}x_{32}x_{41})$
$\mathbf{v}_{20} = (x_{21}, x_{12}x_{21}x_{32}x_{41})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{20}, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{21}$	0	2	1	2	$x_{21}x_{32}x_{41}$
$x_{12}x_{21}$	2	2	1	2	$x_{11}x_{21}x_{32}x_{41}$
$x_{21}x_{21}$	1	1	1	2	$x_{12}x_{22}x_{32}x_{41}$
$x_{21}x_{22}$	1	0	1	2	$x_{12}x_{32}x_{41}$
$x_{21}x_{31}$	1	2	0	2	$x_{12}x_{21}x_{41}$
$x_{21}x_{32}$	1	2	2	2	$x_{12}x_{21}x_{31}x_{41}$
$x_{21}x_{41}$	1	2	1	1	$x_{12}x_{21}x_{32}x_{42}$
$x_{21}x_{42}$	1	2	1	0	$x_{12}x_{21}x_{32}$

21. $\texttt{NextTerm}(\texttt{List}) = (x_{22}, x_{22})$
$\mathbf{v}_{21} = (x_{22}, x_{22})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{21}, \texttt{List})$.

	$\mathbb{F}_3^n$-Representation of the exponent of v[2]					
v[1]	1	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{22}$		2	2	0	0	$x_{11}x_{22}$
$x_{12}x_{22}$		1	2	0	0	$x_{12}x_{22}$
$x_{21}x_{22}$	1	0	0	0	0	1
$x_{22}x_{22}$		0	1	0	0	x_{21}
$x_{22}x_{31}$		0	2	2	0	$x_{22}x_{31}$
$x_{22}x_{32}$		0	2	1	0	$x_{22}x_{32}$
$x_{22}x_{41}$		0	2	0	2	$x_{22}x_{41}$
$x_{22}x_{42}$		0	2	0	1	$x_{22}x_{42}$

22. $\texttt{NextTerm}(\texttt{List}) = (x_{22}, x_{32}x_{42})$
$\mathbf{v}_{22} = (x_{22}, x_{32}x_{42})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{22}, \texttt{List})$.

	Exponent of v[2]				
v[1]	X_1	X_2	X_3	X_4	v[2]
$x_{11}x_{22}$	2	0	1	1	$x_{11}x_{32}x_{42}$
$x_{12}x_{22}$	1	0	1	1	$x_{12}x_{32}x_{42}$
$x_{21}x_{22}$	0	2	1	1	$x_{21}x_{32}x_{42}$
$x_{22}x_{22}$	0	1	1	1	$x_{22}x_{32}x_{42}$
$x_{22}x_{31}$	0	0	0	1	x_{42}
$x_{22}x_{32}$	0	0	2	1	$x_{31}x_{42}$
$x_{22}x_{41}$	0	0	1	0	x_{32}
$x_{22}x_{42}$	0	0	1	2	$x_{32}x_{41}$

23. $\texttt{NextTerm}(\texttt{List}) = (x_{22}, x_{21}x_{31}x_{41})$
$\mathbf{v}_{23} = (x_{22}, x_{21}x_{31}x_{41})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{23}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{22}$	2	2	2	2	$x_{11}x_{21}x_{31}x_{41}$
$x_{12}x_{22}$	1	2	2	2	$x_{12}x_{21}x_{31}x_{41}$
$x_{21}x_{22}$	0	1	2	2	$x_{22}x_{31}x_{41}$
$x_{22}x_{22}$	0	0	2	2	$x_{31}x_{41}$
$x_{22}x_{31}$	0	2	1	2	$x_{21}x_{32}x_{41}$
$x_{22}x_{32}$	0	2	0	2	$x_{21}x_{41}$
$x_{22}x_{41}$	0	2	2	1	$x_{21}x_{31}x_{42}$
$x_{22}x_{42}$	0	2	2	0	$x_{21}x_{31}$

24. $\texttt{NextTerm}(\texttt{List}) = (x_{22}, x_{11}x_{32}x_{42})$
$\mathbf{v}_{24} = (x_{22}, x_{11}x_{22}x_{31}x_{42})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{24}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{22}$	1	0	1	1	$x_{12}x_{32}x_{42}$
$x_{12}x_{22}$	0	0	1	1	$x_{32}x_{42}$
$x_{21}x_{22}$	2	2	1	1	$x_{11}x_{21}x_{32}x_{42}$
$x_{22}x_{22}$	2	1	1	1	$x_{11}x_{22}x_{32}x_{42}$
$x_{22}x_{31}$	2	0	0	1	$x_{11}x_{42}$
$x_{22}x_{32}$	2	0	2	1	$x_{11}x_{31}x_{42}$
$x_{22}x_{41}$	2	0	1	0	$x_{11}x_{32}$
$x_{22}x_{42}$	2	0	1	2	$x_{11}x_{32}x_{41}$

25. $\texttt{NextTerm}(\texttt{List}) = (x_{22}, x_{12}x_{22}x_{32}x_{41})$
$\mathbf{v}_{25} = (x_{22}, x_{12}x_{22}x_{32}x_{41})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{25}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{22}$	0	1	1	2	$x_{22}x_{32}x_{41}$
$x_{12}x_{22}$	2	1	1	2	$x_{11}x_{22}x_{32}x_{41}$
$x_{21}x_{22}$	1	0	1	2	$x_{12}x_{32}x_{41}$
$x_{22}x_{22}$	1	2	1	2	$x_{12}x_{21}x_{32}x_{41}$
$x_{22}x_{31}$	1	1	0	2	$x_{12}x_{22}x_{41}$
$x_{22}x_{32}$	1	1	2	2	$x_{12}x_{22}x_{31}x_{41}$
$x_{22}x_{41}$	1	1	1	1	$x_{12}x_{22}x_{32}x_{42}$
$x_{22}x_{42}$	1	1	1	0	$x_{12}x_{22}x_{32}$

26. $\texttt{NextTerm}(\texttt{List}) = (x_{31}, x_{31})$
$\mathbf{v}_{26} = (x_{31}, x_{31})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{26}, \texttt{List})$.

$\mathbf{v}[1]$	$\mathbb{F}_3^n$-Representation of the exponent of $\mathbf{v}[2]$					$\mathbf{v}[2]$
	1	X_1	X_2	X_3	X_4	
$x_{11}x_{31}$		2	0	2	0	$x_{11}x_{31}$
$x_{12}x_{31}$		1	0	2	0	$x_{12}x_{31}$
$x_{21}x_{31}$		0	2	2	0	$x_{21}x_{31}$
$x_{22}x_{31}$		0	1	2	0	$x_{22}x_{31}$
$x_{31}x_{31}$		0	0	1	0	x_{32}
$x_{31}x_{32}$	1	0	0	0	0	1
$x_{31}x_{41}$		0	0	2	2	$x_{31}x_{41}$
$x_{31}x_{42}$		0	0	2	1	$x_{31}x_{42}$

27. $\texttt{NextTerm}(\texttt{List}) = (x_{31}, x_{12}x_{41})$
$\mathbf{v}_{27} = (x_{31}, x_{12}x_{41})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{27}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{31}$	0	0	0	2	x_{41}
$x_{12}x_{31}$	2	0	0	2	$x_{11}x_{41}$
$x_{21}x_{31}$	1	2	0	2	$x_{12}x_{21}x_{41}$
$x_{22}x_{31}$	1	1	0	2	$x_{12}x_{22}x_{41}$
$x_{31}x_{31}$	1	0	2	2	$x_{12}x_{31}x_{41}$
$x_{31}x_{32}$	1	0	1	2	$x_{12}x_{32}x_{41}$
$x_{31}x_{41}$	1	0	0	1	$x_{12}x_{42}$
$x_{31}x_{42}$	1	0	0	0	x_{12}

28. $\texttt{NextTerm}(\texttt{List}) = (x_{31}, x_{21}x_{42})$
$\mathbf{v}_{28} = (x_{31}, x_{21}x_{42})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{28}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{31}$	2	2	0	1	$x_{11}x_{21}x_{42}$
$x_{12}x_{31}$	1	2	0	1	$x_{12}x_{21}x_{42}$
$x_{21}x_{31}$	0	1	0	1	$x_{22}x_{42}$
$x_{22}x_{31}$	0	0	0	1	x_{42}
$x_{31}x_{31}$	0	2	2	1	$x_{21}x_{31}x_{42}$
$x_{31}x_{32}$	0	2	1	1	$x_{21}x_{32}x_{42}$
$x_{31}x_{41}$	0	2	0	0	x_{21}
$x_{31}x_{42}$	0	2	0	2	$x_{21}x_{41}$

29. $\texttt{NextTerm}(\texttt{List}) = (x_{31}, x_{11}x_{32}x_{42})$

$\mathbf{v}_{29} = (x_{31}, x_{11}x_{32}x_{42})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{29}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{31}$	1	0	1	1	$x_{12}x_{32}x_{42}$
$x_{12}x_{31}$	0	0	1	1	$x_{32}x_{42}$
$x_{21}x_{31}$	2	2	1	1	$x_{11}x_{21}x_{32}x_{42}$
$x_{22}x_{31}$	2	1	1	1	$x_{11}x_{22}x_{32}x_{42}$
$x_{31}x_{31}$	2	0	0	1	$x_{11}x_{42}$
$x_{31}x_{32}$	2	0	2	1	$x_{11}x_{31}x_{42}$
$x_{31}x_{41}$	2	0	1	0	$x_{11}x_{32}$
$x_{31}x_{42}$	2	0	1	2	$x_{11}x_{32}x_{41}$

30. $\texttt{NextTerm}(\texttt{List}) = (x_{31}, x_{22}x_{32}x_{41})$

$\mathbf{v}_{30} = (x_{31}, x_{22}x_{32}x_{41})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{30}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{31}$	2	1	1	2	$x_{11}x_{22}x_{32}x_{41}$
$x_{12}x_{31}$	1	1	1	2	$x_{12}x_{22}x_{32}x_{41}$
$x_{21}x_{31}$	0	0	1	2	$x_{32}x_{41}$
$x_{22}x_{31}$	0	2	1	2	$x_{21}x_{32}x_{41}$
$x_{31}x_{31}$	0	1	0	2	$x_{22}x_{41}$
$x_{31}x_{32}$	0	1	2	2	$x_{22}x_{31}x_{41}$
$x_{31}x_{41}$	0	1	1	1	$x_{22}x_{32}x_{42}$
$x_{31}x_{42}$	0	1	1	0	$x_{22}x_{32}$

31. $\texttt{NextTerm}(\texttt{List}) = (x_{32}, x_{32})$

$\mathbf{v}_{31} = (x_{32}, x_{32})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{31}, \texttt{List})$.

	$\mathbb{F}_3^n$-Representation of the exponent of $\mathbf{v}[2]$					
$\mathbf{v}[1]$	1	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{32}$		2	0	1	0	$x_{11}x_{32}$
$x_{12}x_{32}$		1	0	1	0	$x_{12}x_{32}$
$x_{21}x_{32}$		0	2	1	0	$x_{21}x_{32}$
$x_{22}x_{32}$		0	1	1	0	$x_{22}x_{32}$
$x_{31}x_{32}$	1	0	0	0	0	1
$x_{32}x_{32}$		0	0	2	0	x_{31}
$x_{32}x_{41}$		0	0	1	2	$x_{32}x_{41}$
$x_{32}x_{42}$		0	0	1	1	$x_{32}x_{42}$

32. $\texttt{NextTerm}(\texttt{List}) = (x_{32}, x_{11}x_{42})$

$\mathbf{v}_{32} = (x_{32}, x_{11}x_{42})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{32}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{32}$	1	0	0	1	$x_{12}x_{42}$
$x_{12}x_{32}$	0	0	0	1	x_{42}
$x_{21}x_{32}$	2	2	0	1	$x_{11}x_{21}x_{42}$
$x_{22}x_{32}$	2	1	0	1	$x_{11}x_{22}x_{42}$
$x_{31}x_{32}$	2	0	2	1	$x_{11}x_{31}x_{42}$
$x_{32}x_{32}$	2	0	1	1	$x_{11}x_{32}x_{42}$
$x_{32}x_{41}$	2	0	0	0	x_{11}
$x_{32}x_{42}$	2	0	0	2	$x_{11}x_{41}$

33. $\texttt{NextTerm}(\texttt{List}) = (x_{32}, x_{22}x_{41})$

$\mathbf{v}_{33} = (x_{32}, x_{22}x_{41})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{33}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{32}$	2	1	0	2	$x_{11}x_{22}x_{41}$
$x_{12}x_{32}$	1	1	0	2	$x_{12}x_{22}x_{41}$
$x_{21}x_{32}$	0	0	0	2	x_{41}
$x_{22}x_{32}$	0	2	0	2	$x_{21}x_{41}$
$x_{31}x_{32}$	0	1	2	2	$x_{22}x_{31}x_{41}$
$x_{32}x_{32}$	0	1	1	2	$x_{22}x_{32}x_{41}$
$x_{32}x_{41}$	0	1	0	1	$x_{22}x_{42}$
$x_{32}x_{42}$	0	1	0	0	x_{22}

34. $\texttt{NextTerm}(\texttt{List}) = (x_{32}, x_{12}x_{31}x_{41})$

$\mathbf{v}_{34} = (x_{32}, x_{12}x_{31}x_{41})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{34}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{32}$	0	0	2	2	$x_{31}x_{41}$
$x_{12}x_{32}$	2	0	2	2	$x_{11}x_{31}x_{41}$
$x_{21}x_{32}$	1	2	2	2	$x_{12}x_{21}x_{31}x_{41}$
$x_{22}x_{32}$	1	1	2	2	$x_{12}x_{22}x_{31}x_{41}$
$x_{31}x_{32}$	1	0	1	2	$x_{12}x_{32}x_{41}$
$x_{32}x_{32}$	1	0	0	2	$x_{12}x_{41}$
$x_{32}x_{41}$	1	0	2	1	$x_{12}x_{31}x_{42}$
$x_{32}x_{42}$	1	0	2	0	$x_{12}x_{31}$

35. $\texttt{NextTerm}(\texttt{List}) = (x_{32}, x_{21}x_{31}x_{42})$

$\mathbf{v}_{35} = (x_{32}, x_{21}x_{31}x_{42})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{35}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{32}$	2	2	2	1	$x_{11}x_{21}x_{31}x_{42}$
$x_{12}x_{32}$	1	2	2	1	$x_{12}x_{21}x_{31}x_{42}$
$x_{21}x_{32}$	0	1	2	1	$x_{22}x_{31}x_{42}$
$x_{22}x_{32}$	0	0	2	1	$x_{31}x_{42}$
$x_{31}x_{32}$	0	2	1	1	$x_{21}x_{32}x_{42}$
$x_{32}x_{32}$	0	2	0	1	$x_{21}x_{42}$
$x_{32}x_{41}$	0	2	2	0	$x_{21}x_{31}$
$x_{32}x_{42}$	0	2	2	2	$x_{21}x_{31}x_{41}$

36. $\texttt{NextTerm}(\texttt{List}) = (x_{41}, x_{41})$

$\mathbf{v}_{36} = (x_{41}, x_{41})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{36}, \texttt{List})$.

$\mathbf{v}[1]$	$\mathbb{F}_3^n$-Representation of the exponent of $\mathbf{v}[2]$					$\mathbf{v}[2]$
	1	X_1	X_2	X_3	X_4	
$x_{11}x_{41}$		2	0	0	2	$x_{11}x_{41}$
$x_{12}x_{41}$		1	0	0	2	$x_{12}x_{41}$
$x_{21}x_{41}$		0	2	0	2	$x_{21}x_{41}$
$x_{22}x_{41}$		0	1	0	2	$x_{22}x_{41}$
$x_{31}x_{41}$		0	0	2	2	$x_{31}x_{41}$
$x_{32}x_{41}$		0	0	1	2	$x_{32}x_{41}$
$x_{41}x_{41}$		0	0	0	1	x_{42}
$x_{41}x_{42}$	1	0	0	0	0	1

37. $\texttt{NextTerm}(\texttt{List}) = (x_{41}, x_{11}x_{31})$

$\mathbf{v}_{37} = (x_{41}, x_{11}x_{31})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{37}, \texttt{List})$.

$\mathbf{v}[1]$	Exponent of $\mathbf{v}[2]$				$\mathbf{v}[2]$
	X_1	X_2	X_3	X_4	
$x_{11}x_{41}$	1	0	2	0	$x_{12}x_{31}$
$x_{12}x_{41}$	0	0	2	0	x_{31}
$x_{21}x_{41}$	2	2	2	0	$x_{11}x_{21}x_{31}$
$x_{22}x_{41}$	2	1	2	0	$x_{11}x_{22}x_{31}$
$x_{31}x_{41}$	2	0	1	0	$x_{11}x_{32}$
$x_{32}x_{41}$	2	0	0	0	x_{11}
$x_{41}x_{41}$	2	0	2	2	$x_{11}x_{31}x_{41}$
$x_{41}x_{42}$	2	0	2	1	$x_{11}x_{31}x_{42}$

38. $\texttt{NextTerm}(\texttt{List}) = (x_{41}, x_{21}x_{32})$

$\mathbf{v}_{38} = (x_{41}, x_{21}x_{32})$

$\mathcal{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{38}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{41}$	2	2	1	0	$x_{11}x_{21}x_{32}$
$x_{12}x_{41}$	1	2	1	0	$x_{12}x_{21}x_{32}$
$x_{21}x_{41}$	0	1	1	0	$x_{22}x_{32}$
$x_{22}x_{41}$	0	0	1	0	x_{32}
$x_{31}x_{41}$	0	2	0	0	x_{21}
$x_{32}x_{41}$	0	2	2	0	$x_{21}x_{31}$
$x_{41}x_{41}$	0	2	1	2	$x_{21}x_{32}x_{41}$
$x_{41}x_{42}$	0	2	1	1	$x_{21}x_{32}x_{42}$

39. $\texttt{NextTerm}(\texttt{List}) = (x_{41}, x_{12}x_{32}x_{42})$

$\mathbf{v}_{39} = (x_{41}, x_{12}x_{32}x_{42})$

$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{39}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{41}$	0	0	1	1	$x_{32}x_{42}$
$x_{12}x_{41}$	2	0	1	1	$x_{11}x_{32}x_{42}$
$x_{21}x_{41}$	1	2	1	1	$x_{12}x_{21}x_{32}x_{42}$
$x_{22}x_{41}$	1	1	1	1	$x_{12}x_{22}x_{32}x_{42}$
$x_{31}x_{41}$	1	0	0	1	$x_{12}x_{42}$
$x_{32}x_{41}$	1	0	2	1	$x_{12}x_{31}x_{42}$
$x_{41}x_{41}$	1	0	1	0	$x_{12}x_{32}$
$x_{41}x_{42}$	1	0	1	2	$x_{12}x_{32}x_{41}$

40. $\texttt{NextTerm}(\texttt{List}) = (x_{41}, x_{22}x_{31}x_{42})$

$\mathbf{v}_{40} = (x_{41}, x_{22}x_{31}x_{42})$

$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{40}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{41}$	2	1	2	1	$x_{11}x_{22}x_{31}x_{42}$
$x_{12}x_{41}$	1	1	2	1	$x_{12}x_{22}x_{31}x_{42}$
$x_{21}x_{41}$	0	0	2	1	$x_{31}x_{42}$
$x_{22}x_{41}$	0	2	2	1	$x_{21}x_{31}x_{42}$
$x_{31}x_{41}$	0	1	1	1	$x_{22}x_{32}x_{42}$
$x_{32}x_{41}$	0	1	0	1	$x_{22}x_{42}$
$x_{41}x_{41}$	0	1	2	0	$x_{22}x_{31}$
$x_{41}x_{42}$	0	1	2	2	$x_{22}x_{31}x_{41}$

41. $\texttt{NextTerm}(\texttt{List}) = (x_{42}, x_{42})$

$\mathbf{v}_{41} = (x_{42}, x_{42})$

$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}, x_{42}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{41}, \texttt{List})$.

	$\mathbb{F}_3^n$-Representation of the exponent of $\mathbf{v}[2]$					
$\mathbf{v}[1]$	1	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{42}$		2	0	0	1	$x_{11}x_{42}$
$x_{12}x_{42}$		1	0	0	1	$x_{12}x_{42}$
$x_{21}x_{42}$		0	2	0	1	$x_{21}x_{42}$
$x_{22}x_{42}$		0	1	0	1	$x_{22}x_{42}$
$x_{31}x_{42}$		0	0	2	1	$x_{31}x_{42}$
$x_{32}x_{42}$		0	0	1	1	$x_{32}x_{42}$
$x_{41}x_{42}$	1	0	0	0	0	1
$x_{42}x_{42}$		0	0	0	2	x_{41}

42. $\texttt{NextTerm}(\texttt{List}) = (x_{42}, x_{12}x_{32})$

$\mathbf{v}_{42} = (x_{42}, x_{12}x_{32})$

$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}, x_{42}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{42}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{42}$	0	0	1	0	x_{32}
$x_{12}x_{42}$	2	0	1	0	$x_{11}x_{32}$
$x_{21}x_{42}$	1	2	1	0	$x_{12}x_{21}x_{32}$
$x_{22}x_{42}$	1	1	1	0	$x_{12}x_{22}x_{32}$
$x_{31}x_{42}$	1	0	0	0	x_{12}
$x_{32}x_{42}$	1	0	2	0	$x_{12}x_{31}$
$x_{41}x_{42}$	1	0	1	2	$x_{12}x_{32}x_{41}$
$x_{42}x_{42}$	1	0	1	1	$x_{12}x_{32}x_{42}$

43. $\texttt{NextTerm}(\texttt{List}) = (x_{42}, x_{22}x_{31})$

$\mathbf{v}_{43} = (x_{42}, x_{22}x_{31})$

$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}, x_{42}\}$

$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{43}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{42}$	2	1	2	0	$x_{11}x_{22}x_{31}$
$x_{12}x_{42}$	1	1	2	0	$x_{12}x_{22}x_{31}$
$x_{21}x_{42}$	0	0	2	0	x_{31}
$x_{22}x_{42}$	0	2	2	0	$x_{21}x_{31}$
$x_{31}x_{42}$	0	1	1	0	$x_{22}x_{32}$
$x_{32}x_{42}$	0	1	0	0	x_{22}
$x_{41}x_{42}$	0	1	2	2	$x_{22}x_{31}x_{41}$
$x_{42}x_{42}$	0	1	2	1	$x_{22}x_{31}x_{42}$

44. $\texttt{NextTerm}(\texttt{List}) = (x_{42}, x_{11}x_{31}x_{41})$
$\mathbf{v}_{44} = (x_{42}, x_{11}x_{31}x_{41})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}, x_{42}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{44}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{42}$	1	0	2	2	$x_{12}x_{31}x_{41}$
$x_{12}x_{42}$	0	0	2	2	$x_{31}x_{41}$
$x_{21}x_{42}$	2	2	2	2	$x_{11}x_{21}x_{31}x_{41}$
$x_{22}x_{42}$	2	1	2	2	$x_{11}x_{22}x_{31}x_{41}$
$x_{31}x_{42}$	2	0	1	2	$x_{11}x_{32}x_{41}$
$x_{32}x_{42}$	2	0	0	2	$x_{11}x_{41}$
$x_{41}x_{42}$	2	0	2	1	$x_{11}x_{31}x_{42}$
$x_{42}x_{42}$	2	0	2	0	$x_{11}x_{31}$

45. $\texttt{NextTerm}(\texttt{List}) = (x_{42}, x_{21}x_{32}x_{41})$
$\mathbf{v}_{45} = (x_{42}, x_{21}x_{32}x_{41})$
$\mathscr{N} = \{1, x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, x_{41}, x_{42}\}$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{45}, \texttt{List})$.

	Exponent of $\mathbf{v}[2]$				
$\mathbf{v}[1]$	X_1	X_2	X_3	X_4	$\mathbf{v}[2]$
$x_{11}x_{42}$	2	2	1	2	$x_{11}x_{21}x_{32}x_{41}$
$x_{12}x_{42}$	1	2	1	2	$x_{12}x_{21}x_{32}x_{41}$
$x_{21}x_{42}$	0	1	1	2	$x_{22}x_{32}x_{41}$
$x_{22}x_{42}$	0	0	1	2	$x_{32}x_{41}$
$x_{31}x_{42}$	0	2	0	2	$x_{21}x_{41}$
$x_{32}x_{42}$	0	2	2	2	$x_{21}x_{31}x_{41}$
$x_{41}x_{42}$	0	2	1	1	$x_{21}x_{32}x_{42}$
$x_{42}x_{42}$	0	2	1	0	$x_{21}x_{32}$

46. $\texttt{NextTerm}(\texttt{List}) = (x_{11}^2, x_{12})$
$j = 11$
$G_T = \{x_{11}^2 - x_{12}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of x_{11}^2.

47. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{12}, 1)$
$j = 1$
$G_T = G_T \cup \{x_{11}x_{12} - 1\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{11}x_{12}$.

48. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{21}, x_{11}x_{21})$
$\mathbf{v}_{46} = (x_{11}x_{21}, x_{11}x_{21})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{46}, \texttt{List})$

49. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{21}, x_{11}x_{31}x_{41})$
$j = 44$
$G_T = G_T \cup \{x_{11}x_{21} - x_{42}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{11}x_{21}$.

50. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{22}, x_{11}x_{22})$
$\mathbf{v}_{47} = (x_{11}x_{22}, x_{11}x_{22})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{47}, \texttt{List})$

51. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{22}, x_{11}x_{32}x_{42})$
$j = 29$
$G_T = G_T \cup \{x_{11}x_{22} - x_{31}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{11}x_{22}$.

52. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{31}, x_{41})$
$j = 36$
$G_T = G_T \cup \{x_{11}x_{31} - x_{41}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{11}x_{31}$.

53. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{32}, x_{11}x_{32})$
$\mathbf{v}_{48} = (x_{11}x_{32}, x_{11}x_{32})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{48}, \texttt{List})$

54. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{32}, x_{12}x_{42})$
$\mathbf{v}_{49} = (x_{11}x_{32}, x_{12}x_{42})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{49}, \texttt{List})$

55. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{32}, x_{31}x_{41})$
$j = 17$
$G_T = G_T \cup \{x_{11}x_{32} - x_{21}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{11}x_{32}$.
56. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{41}, x_{11}x_{41})$
$\mathbf{v}_{50} = (x_{11}x_{41}, x_{11}x_{41})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{50}, \texttt{List})$
57. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{42}, x_{32}x_{42})$
$j = 22$
$G_T = G_T \cup \{x_{11}x_{41} - x_{22}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{11}x_{41}$.
58. $\texttt{NextTerm}(\texttt{List}) = (x_{11}x_{42}, x_{32})$
$j = 31$
$G_T = G_T \cup \{x_{11}x_{42} - x_{32}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{11}x_{42}$.
59. $\texttt{NextTerm}(\texttt{List}) = (x_{12}^2, x_{11})$
$j = 6$
$G_T = G_T \cup \{x_{12}^2 - x_{11}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of x_{12}^2.
60. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{21}, x_{12}x_{21})$
$\mathbf{v}_{51} = (x_{12}x_{21}, x_{12}x_{21})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{51}, \texttt{List})$
61. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{21}, x_{12}x_{31}x_{41})$
$j = 34$
$G_T = G_T \cup \{x_{12}x_{21} - x_{32}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{12}x_{21}$.
62. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{22}, x_{12}x_{22})$
$\mathbf{v}_{52} = (x_{12}x_{22}, x_{12}x_{22})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{52}, \texttt{List})$
63. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{22}, x_{12}x_{32}x_{42})$
$j = 39$
$G_T = G_T \cup \{x_{12}x_{22} - x_{41}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{12}x_{22}$.
64. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{31}, x_{12}x_{31})$
$\mathbf{v}_{53} = (x_{12}x_{31}, x_{12}x_{31})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{53}, \texttt{List})$
65. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{31}, x_{32}x_{42})$
$j = 22$
$G_T = G_T \cup \{x_{12}x_{31} - x_{22}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{12}x_{31}$.
66. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{32}, x_{42})$
$j = 41$
$G_T = G_T \cup \{x_{12}x_{32} - x_{42}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{12}x_{32}$.
67. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{41}, x_{31})$
$j = 26$
$G_T = G_T \cup \{x_{12}x_{41} - x_{31}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{12}x_{41}$.
68. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{42}, x_{11}x_{32})$
$\mathbf{v}_{54} = (x_{12}x_{42}, x_{11}x_{32})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{54}, \texttt{List})$
69. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{42}, x_{12}x_{42})$
$\mathbf{v}_{55} = (x_{12}x_{42}, x_{12}x_{42})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{55}, \texttt{List})$
70. $\texttt{NextTerm}(\texttt{List}) = (x_{12}x_{42}, x_{31}x_{41})$
$j = 17$
$G_T = G_T \cup \{x_{12}x_{42} - x_{21}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{12}x_{42}$.
71. $\texttt{NextTerm}(\texttt{List}) = (x_{21}^2, x_{22})$
$j = 21$
$G_T = G_T \cup \{x_{21}^2 - x_{22}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of x_{21}^2.

72. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{22}, 1)$
$j = 1$
$G_T = G_T \cup \{x_{21}x_{22} - 1\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{21}x_{22}$.
73. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{31}, x_{21}x_{31})$
$\mathbf{v}_{56} = (x_{21}x_{31}, x_{21}x_{31})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{56}, \texttt{List})$
74. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{31}, x_{22}x_{42})$
$\mathbf{v}_{57} = (x_{21}x_{31}, x_{22}x_{42})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{57}, \texttt{List})$
75. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{31}, x_{32}x_{41})$
$j = 7$
$G_T = G_T \cup \{x_{21}x_{31} - x_{11}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{21}x_{31}$.
76. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{32}, x_{41})$
$j = 36$
$G_T = G_T \cup \{x_{21}x_{32} - x_{41}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{21}x_{32}$.
77. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{41}, x_{21}x_{41})$
$\mathbf{v}_{58} = (x_{21}x_{41}, x_{21}x_{41})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{58}, \texttt{List})$
78. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{41}, x_{22}x_{42})$
$\mathbf{v}_{59} = (x_{21}x_{41}, x_{22}x_{42})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{59}, \texttt{List})$
79. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{41}, x_{31}x_{42})$
$j = 12$
$G_T = G_T \cup \{x_{21}x_{41} - x_{12}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{21}x_{41}$.
80. $\texttt{NextTerm}(\texttt{List}) = (x_{21}x_{42}, x_{32})$
$j = 31$
$G_T = G_T \cup \{x_{21}x_{42} - x_{32}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{21}x_{42}$.
81. $\texttt{NextTerm}(\texttt{List}) = (x_{22}^2, x_{21})$
$j = 16$
$G_T = G_T \cup \{x_{22}^2 - x_{21}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of x_{22}^2.
82. $\texttt{NextTerm}(\texttt{List}) = (x_{22}x_{31}, x_{42})$
$j = 41$
$G_T = G_T \cup \{x_{22}x_{31} - x_{42}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{22}x_{31}$.
83. $\texttt{NextTerm}(\texttt{List}) = (x_{22}x_{32}, x_{21}x_{41})$
$\mathbf{v}_{60} = (x_{22}x_{32}, x_{21}x_{41})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{60}, \texttt{List})$
84. $\texttt{NextTerm}(\texttt{List}) = (x_{22}x_{32}, x_{22}x_{32})$
$\mathbf{v}_{61} = (x_{22}x_{32}, x_{22}x_{32})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{61}, \texttt{List})$
85. $\texttt{NextTerm}(\texttt{List}) = (x_{22}x_{32}, x_{31}x_{42})$
$j = 12$
$G_T = G_T \cup \{x_{22}x_{32} - x_{12}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{22}x_{32}$.
86. $\texttt{NextTerm}(\texttt{List}) = (x_{22}x_{41}, x_{32})$
$j = 31$
$G_T = G_T \cup \{x_{22}x_{41} - x_{32}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{22}x_{41}$.
87. $\texttt{NextTerm}(\texttt{List}) = (x_{22}x_{42}, x_{22}x_{42})$
$\mathbf{v}_{62} = (x_{22}x_{42}, x_{22}x_{42})$
$\texttt{List} = \texttt{InsertNexts}(\mathbf{v}_{62}, \texttt{List})$
88. $\texttt{NextTerm}(\texttt{List}) = (x_{22}x_{42}, x_{32}x_{41})$
$j = 7$
$G_T = G_T \cup \{x_{22}x_{42} - x_{11}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{22}x_{42}$.
89. $\texttt{NextTerm}(\texttt{List}) = (x_{31}^2, x_{32})$
$j = 31$
$G_T = G_T \cup \{x_{31}^2 - x_{32}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of x_{31}^2.

90. $\texttt{NextTerm}(\texttt{List}) = (x_{31}x_{32}, 1)$
$j = 1$
$G_T = G_T \cup \{x_{31}x_{32} - 1\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{31}x_{32}$.

91. $\texttt{NextTerm}(\texttt{List}) = (x_{31}x_{41}, x_{21})$
$j = 16$
$G_T = G_T \cup \{x_{31}x_{41} - x_{21}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{31}x_{41}$.

92. $\texttt{NextTerm}(\texttt{List}) = (x_{31}x_{42}, x_{12})$
$j = 11$
$G_T = G_T \cup \{x_{31}x_{42} - x_{12}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{31}x_{42}$.

93. $\texttt{NextTerm}(\texttt{List}) = (x_{41}^2, x_{42})$
$j = 41$
$G_T = G_T \cup \{x_{41}^2 - x_{42}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of x_{41}^2.

94. $\texttt{NextTerm}(\texttt{List}) = (x_{41}x_{42}, 1)$
$j = 1$
$G_T = G_T \cup \{x_{41}x_{42} - 1\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of $x_{41}x_{42}$.

95. $\texttt{NextTerm}(\texttt{List}) = (x_{42}^2, x_{41})$
$j = 36$
$G_T = G_T \cup \{x_{42}^2 - x_{41}\}$
We eliminate from the list `List` all the elements $\mathbf{w} \in \texttt{List}$ such that $\mathbf{w}[1]$ is a multiple of x_{42}^2.

At the end of this last step the list `List` is empty and the algorithm ends.

By Proposition 4.35, the set of codewords related with the exponents of a reduced Gröbner basis of the ideal associated with a linear code $\mathscr{C}$ with respect to a degree compatible ordering induced a test-set $\mathscr{T}$ for $\mathscr{C}$. However, not all the codewords of this test-set are codewords of minimal support, i.e. this set is somehow redundant. We can reduce the number of codewords to the set $\mathscr{T} \cap \mathscr{M}_{\mathscr{C}}$, which is still a test-set for the code $\mathscr{C}$, using Algorithm 22. The obtained test-set will be called a *minimal Gröbner test-set*.

Example 4.44. The reduced Gröbner basis $\mathscr{G}$ of the ideal $I(\mathscr{C})$ associated to the ternary code $\mathscr{C}$ of Example 4.15 contains 74 elements representing the following nonzero codewords of $\mathscr{C}$.

$$\begin{array}{cccc} (2,1,2,2,2,1) & (1,2,1,1,1,2) & (2,2,0,1,2,0) & (1,1,0,2,1,0) \\ (2,0,1,0,2,2) & (1,0,2,0,1,1) & (0,1,1,2,0,2) & (0,2,2,1,0,1) \end{array}$$

Note that the first two codewords do not belong to $\mathscr{M}_{\mathscr{C}}$. The first codeword is related to the binomial $x_{11}x_{22}x_{31} - x_{42}x_{52}x_{61} \in \mathscr{G}$. Note that $x_{11}x_{22}x_{31} \notin \mathrm{LT}_{\prec}(\mathscr{G})$. Notwithstanding

$$\frac{x_{11}x_{22}x_{31}}{x_{42}x_{52}x_{61}} = x_{11}x_{22}x_{31}x_{41}x_{51}x_{62} \in \mathrm{LT}_{\prec}(\mathscr{G})$$

since $x_{11}x_{51} - x_{31}x_{62} \in \mathscr{G}$. Moreover, the second codeword is related to the binomial $x_{12}x_{21}x_{32} - x_{41}x_{51}x_{62}$. Again, $x_{12}x_{21}x_{32} \notin \mathrm{LT}_{\prec}(\mathscr{C})$, but

$$\frac{x_{12}x_{21}x_{32}}{x_{41}x_{51}x_{62}} = x_{12}x_{21}x_{32}x_{42}x_{52}x_{61} \in \mathrm{LT}_{\prec}(\mathscr{G})$$

Algorithm 22: Algorithm for computing a minimal Gröbner test-set for $\mathscr{C}$

Data: The rows $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\} \subseteq \mathbb{F}_q^n$ of a generator matrix of an $[n,k]$ linear code $\mathscr{C}$ defined over $\mathbb{F}_q$ and a degree compatible ordering $<_T$ on $\mathbb{K}[\mathbf{X}]$.
Result: A minimal Gröbner test-set for $\mathscr{C}$ w.r.t. $<_T$.

1 $\texttt{List} \longleftarrow \left[(1,1), \left\{(1, \alpha^j \mathbf{w}_i)\right\}_{\substack{i=1,\ldots,k \\ j=1,\ldots,q-1}}\right]$; $G_T \longleftarrow \emptyset$; $\mathscr{N} \longleftarrow \emptyset$;
2 **while** $\texttt{List} \neq \emptyset$ **do**
3 $\mathbf{w} \longleftarrow \texttt{NextTerm(List)}$;
4 **if** $\mathbf{w}[1] \notin \mathrm{LT}_{<_T}(G_T)$ **then**
5 $j = \texttt{Member}(\mathbf{w}[2], [\mathbf{v}_1[2], \ldots, \mathbf{v}_r[2]])$;
6 **if** $j \neq \texttt{false}$ **then**
7 **if** $\left(\mathbf{w}[1]/\mathbf{v}_j[1]\right) \notin \mathrm{LT}_{<_T}(G_T)$ **then**
8 $G_T \longleftarrow G_T \cup \left\{\mathbf{w}[1] - \mathbf{v}_j[1]\right\}$;
9 **for** $i = 1$ *to* r
10 **if** $\mathbf{v}_i[1]$ *is a multiple of* $\mathbf{w}[1]$ **then**
11 Removes $\mathbf{v}_i[1]$ from $\mathscr{N}$
12 **endif**
13 **endfor**
14 **endif**
15 **else**
16 $r \longleftarrow r+1$;
17 $\mathbf{v}_r \longleftarrow \mathbf{w}$;
18 $\mathscr{N} \longleftarrow \mathscr{N} \cup \{\mathbf{v}_r[1]\}$;
19 $\texttt{List} = \texttt{InsertNexts}(\mathbf{w}, \texttt{List})$;
20 **endif**
21 **endif**
22 **endw**

since $x_{21}x_{42} - x_{32}x_{61} \in \mathrm{LT}_{\prec}(\mathscr{C})$. Therefore, Algorithm 22 ensures a minimal Gröbner test-set.

4.5 Set of codewords of minimal support

We define the *Universal Gröbner basis* of $I_+(\mathscr{C})$, denoted by $\mathscr{U}_{\mathscr{C}}$, to be the union of all reduced Gröbner Bases $\mathscr{G}_{\prec}$ of $I_+(\mathscr{C})$ as $\prec$ runs over all terms orders over $\mathbb{K}[\mathbf{X}]$. A binomial $\mathbf{X}^{\mathbf{u}_1} - \mathbf{X}^{\mathbf{u}_2}$ in $I_+(\mathscr{C})$ is called *primitive* if there exists no other binomial $\mathbf{X}^{\mathbf{v}_1} - \mathbf{X}^{\mathbf{v}_2} \in I_+(\mathscr{C})$ such that $\mathbf{X}^{\mathbf{v}_1}$ divides $\mathbf{X}^{\mathbf{u}_1}$ and $\mathbf{X}^{\mathbf{v}_2}$ divides $\mathbf{X}^{\mathbf{u}_2}$, or equivalently, $\mathrm{supp}(\Delta\mathbf{v}_1) \not\subset \mathrm{supp}(\Delta\mathbf{u}_1)$ and $\mathrm{supp}(\Delta\mathbf{v}_2) \not\subset \mathrm{supp}(\Delta\mathbf{u}_2)$.

Remark 4.45. Recall that the map Δ transforms elements from the finite field $\mathbb{F}_q$ into a $(q-1)$-tuple of integers so the division algorithm in $\mathbb{K}[\mathbf{X}]$ is the division algorithm for polynomials in the usual sense.

Lemma 4.46. *Every binomial in $\mathcal{U}_{\mathscr{C}}$ is primitive.*

Proof. It is a straightforward generalization of [107, Lemma 4.6]. Let us fix an arbitrary term ordering $\prec$ in $\mathbb{K}[\mathbf{X}]$, and let $\mathcal{G}_{\prec}$ be the reduced Gröbner basis of $I_+(\mathscr{C})$ w.r.t. $\prec$. By definition, for any binomial $\mathbf{X}^{\mathbf{u}_1} - \mathbf{X}^{\mathbf{u}_2}$ in $\mathcal{G}_{\prec}$ with $\mathbf{X}^{\mathbf{u}_1} \succ \mathbf{X}^{\mathbf{u}_2}$, $\mathbf{X}^{\mathbf{u}_1}$ is a minimal generator of the initial ideal $\mathrm{in}_{\prec}\left(I_+(\mathscr{C})\right)$ and $\mathbf{X}^{\mathbf{u}_2}$ is a standard monomial. Now suppose that $\mathbf{X}^{\mathbf{u}_1} - \mathbf{X}^{\mathbf{u}_2}$ is not primitive, or equivalently there exists another binomial $\mathbf{X}^{\mathbf{v}_1} - \mathbf{X}^{\mathbf{v}_2}$ in $I_+(\mathscr{C})$ such that $\mathbf{X}^{\mathbf{v}_1}$ divides $\mathbf{X}^{\mathbf{u}_1}$ and $\mathbf{X}^{\mathbf{v}_2}$ divides $\mathbf{X}^{\mathbf{u}_2}$. We distinguish two cases:

- If $\mathbf{X}^{\mathbf{v}_1} \succ \mathbf{X}^{\mathbf{v}_2}$, then $\mathbf{X}^{\mathbf{u}_1}$ is not a minimal generator of the initial ideal $\mathrm{in}_{\prec}\left(I_+(\mathscr{C})\right)$.
- If $\mathbf{X}^{\mathbf{v}_1} \prec \mathbf{X}^{\mathbf{v}_2}$, then $\mathbf{X}^{\mathbf{u}_2}$ is not in standard form.

Both cases contradicts our assumption. □

We call the set of all primitive binomials of $I_+(\mathscr{C})$ the *Graver basis* of $I_+(\mathscr{C})$ and denote it by $\mathrm{Gr}_{\mathscr{C}}$.

Corollary 4.47. $\mathcal{U}_{\mathscr{C}} \subseteq \mathrm{Gr}_{\mathscr{C}}$

Proof. The result is a direct consequence of Lemma 4.46. □

The following theorem suggests an algorithm for computing the Graver basis of the ideal $I_+(\mathscr{C})$. For this purpose we define the *Lawrence lifting* of the ideal $I_+(\mathscr{C})$ as the ideal

$$I_{\Lambda(\mathscr{C})} = \left\{\mathbf{X}^{\mathbf{w}_1}\mathbf{Z}^{\mathbf{w}_2} - \mathbf{X}^{\mathbf{w}_2}\mathbf{Z}^{\mathbf{w}_1} \mid \mathbf{w}_1 - \mathbf{w}_2 \in \mathscr{C}\right\}$$

in the polynomial ring $\mathbb{K}[\mathbf{X}, \mathbf{Z}]$. Recall that

$$\mathbb{K}[\mathbf{X}, \mathbf{Y}] = \mathbb{K}[\underbrace{x_{11}, \ldots, x_{1q-1}, \ldots, x_{n1}, \ldots, x_{nq-1}}_{\mathbf{X}}, \underbrace{z_{11}, \ldots, z_{1q-1}, \ldots, z_{n1}, \ldots, z_{nq-1}}_{\mathbf{Z}}].$$

Theorem 4.48. *The set of binomials of the Graver basis of $I_{\Lambda(\mathscr{C})}$ coincides with any reduced Gröbner basis of $I_{\Lambda(\mathscr{C})}$*

Proof. The proof starts with the observation that a binomial $\mathbf{X}^{\mathbf{u}_1} - \mathbf{X}^{\mathbf{u}_2}$ is primitive in the ideal $I_+(\mathscr{C})$ if and only if the corresponding binomial $\mathbf{X}^{\mathbf{u}_1}\mathbf{Z}^{\mathbf{u}_2} - \mathbf{X}^{\mathbf{u}_2}\mathbf{Z}^{\mathbf{u}_1}$ in the lifting ideal $I_{\Lambda(\mathscr{C})}$ is primitive. Therefore, between the Graver basis of the ideals $I_+(\mathscr{C})$ and $I_{\Lambda(\mathscr{C})}$ there exists the following relation:

$$\mathrm{Gr}_{\Lambda(\mathscr{C})} = \left\{\mathbf{X}^{\mathbf{u}_1}\mathbf{Z}^{\mathbf{u}_2} - \mathbf{X}^{\mathbf{u}_2}\mathbf{Z}^{\mathbf{u}_1} \mid \mathbf{X}^{\mathbf{u}_1} - \mathbf{X}^{\mathbf{u}_2} \in \mathrm{Gr}_{\mathscr{C}}\right\}.$$

Now, take any element $g = \mathbf{X}^{\mathbf{u}_1}\mathbf{Z}^{\mathbf{u}_2} - \mathbf{X}^{\mathbf{u}_2}\mathbf{Z}^{\mathbf{u}_1}$ in $\mathrm{Gr}_{\Lambda(\mathscr{C})}$. Let B be the set of all binomials in $I_{\Lambda(\mathscr{C})}$ except g and assume that B generates the ideal $I_{\Lambda(\mathscr{C})}$. Therefore g can be written as a linear combination of the elements of B. In other words, there exists a binomial $\mathbf{X}^{\mathbf{v}_1}\mathbf{Z}^{\mathbf{v}_2} - \mathbf{X}^{\mathbf{v}_2}\mathbf{Z}^{\mathbf{v}_1}$ in B such that one of its terms divides the leading term of g. Replacing $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)$ by $-\mathbf{v} = (-\mathbf{v}_1, -\mathbf{v}_2)$ in $\mathbb{F}_q$ if necessary, we may assume that $\mathbf{X}^{\mathbf{v}_1}\mathbf{Z}^{\mathbf{v}_2}$ divides $\mathbf{X}^{\mathbf{u}_1}\mathbf{Z}^{\mathbf{u}_2}$, contrary to the fact that $\mathbf{X}^{\mathbf{u}_1} - \mathbf{X}^{\mathbf{u}_2}$ is primitive in $I_+(\mathscr{C})$. So some non-zero scalar multiple of g must appear in any reduced Gröbner basis of $I_{\Lambda(\mathscr{C})}$ which is also a minimal generating set of $I_{\Lambda(\mathscr{C})}$. □

Algorithm 23: Algorithm for computing the Graver basis of $I_+(\mathscr{C})$

Data: An $[n,k]$ linear code $\mathscr{C}$ defined over $\mathbb{F}_q$.
Result: The Graver basis of the ideal $I_+(\mathscr{C})$, $\mathrm{Gr}_{\mathscr{C}}$.
1 Choose any term order $\prec$ on $\mathbb{K}[\mathbf{X},\mathbf{Z}]$;
2 Define the Lawrence ideal $I_{\Lambda(\mathscr{C})}$;
3 Compute a reduced Gröbner basis of $I_{\Lambda(\mathscr{C})}$ w.r.t. $\prec$;
4 Substitute the variable $\mathbf{Z}$ by $\mathbf{1}$;

This theorem gives us an algorithm to compute a Graver basis of the ideal $I_+(\mathscr{C})$. Note that Step 3 of the algorithm can be executed by applying Algorithm 21. Here is another way of defining the ideal $I_{\Lambda(\mathscr{C})}$.

Theorem 4.49. *Let $\mathscr{C}$ be an $[n,k]$ linear code defined over $\mathbb{F}_q$ and $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\}$ be the rows of a generator matrix of $\mathscr{C}$. We define the ideal:*

$$I_3 = \left\langle \left\{\mathbf{X}^{\alpha^j\mathbf{w}_i} - \mathbf{Z}^{\alpha^j\mathbf{w}_i}\right\}_{\substack{i=1,\ldots,k\\ j=1,\ldots,q-1}} \bigcup \left\{\mathscr{R}_{X_i}\left(T_+\right)\right\}_{i=1,\ldots,n} \bigcup \left\{\mathscr{R}_{Z_i}\left(T_+\right)\right\}_{i=1,\ldots,n} \right\rangle.$$

Then $I_{\Lambda(\mathscr{C})} = I_3$.

Proof. The following result may be proved in the same way as Theorem 4.3. We claim that all the binomials of the generating set of I_3 belongs to $I_{\Lambda(\mathscr{C})}$. Indeed, the exponents of all the binomials of the sets $\mathscr{R}_{X_i}\left(T_+\right)$ and $\mathscr{R}_{Z_i}\left(T_+\right)$ correspond to the codeword $\mathbf{0} \in \mathscr{C}$.

Conversely, we need to show that each binomial $\mathbf{X^aZ^b} - \mathbf{X^bZ^a}$ in $I_{\Lambda(\mathscr{C})}$ belongs to I_3. Applying the definition of the ideal $I_{\Lambda(\mathscr{C})}$ we can rewrite $\mathbf{a}-\mathbf{b} \in \mathscr{C}$ as

$$\mathbf{a}-\mathbf{b} = \lambda_1\mathbf{w}_1 + \ldots + \lambda_k\mathbf{w}_k \text{ with } \lambda_1,\ldots,\lambda_k \in \mathbb{F}_q.$$

We have that

$$\begin{aligned}
\mathbf{X}^{\mathbf{a}-\mathbf{b}}\mathbf{Z}^{\mathbf{b}-\mathbf{a}} - 1 &= \left(\mathbf{X}^{\lambda_1\mathbf{w}_1}\mathbf{Z}^{-\lambda_1\mathbf{w}_1} - 1\right)\prod_{i=2}^{k}\mathbf{X}^{\lambda_i\mathbf{w}_i}\mathbf{Z}^{-\lambda_i\mathbf{w}_i} + \left(\prod_{i=2}^{k}\mathbf{X}^{\lambda_i\mathbf{w}_i}\mathbf{Z}^{-\lambda_i\mathbf{w}_i} - 1\right)\\
&= \left(\mathbf{X}^{\lambda_1\mathbf{w}_1}\mathbf{Z}^{-\lambda_1\mathbf{w}_1} - 1\right)\prod_{i=2}^{k}\mathbf{X}^{\lambda_i\mathbf{w}_i}\mathbf{Z}^{-\lambda_i\mathbf{w}_i} +\\
&+ \left(\mathbf{X}^{\lambda_2\mathbf{w}_2}\mathbf{Z}^{-\lambda_2\mathbf{w}_2} - 1\right)\prod_{i=3}^{k}\mathbf{X}^{\lambda_i\mathbf{w}_i}\mathbf{Z}^{-\lambda_i\mathbf{w}_i} + \ldots +\\
&+ \left(\mathbf{X}^{\lambda_{k-1}\mathbf{w}_{k-1}}\mathbf{Z}^{-\lambda_{k-1}\mathbf{w}_{k-1}} - 1\right)\mathbf{X}^{\lambda_k\mathbf{w}_k}\mathbf{Z}^{-\lambda_k\mathbf{w}_k} + \left(\mathbf{X}^{\lambda_k\mathbf{w}_k}\mathbf{Z}^{-\lambda_k\mathbf{w}_k} - 1\right).
\end{aligned}$$

If at least one λ_i is nonzero with $i = 1,\ldots,k$, then the last equation forces that

$$\mathbf{X}^{\mathbf{a}-\mathbf{b}}\mathbf{Z}^{\mathbf{b}-\mathbf{a}} - 1 \in \left\langle \left\{\mathbf{X}^{\alpha^j\mathbf{w}_i}\mathbf{Z}^{-\alpha^j\mathbf{w}_i} - 1\right\}_{\substack{i=1,\ldots,k\\ j=1,\ldots,q-1}} \right\rangle.$$

Otherwise $\mathbf{a}-\mathbf{b}=\mathbf{0}$ and we deduce that:

$$\mathbf{X}^{\mathbf{a}-\mathbf{b}}\mathbf{Z}^{\mathbf{b}-\mathbf{a}}-1\in\left\langle\ \left\{\mathscr{R}_{X_i}\left(T_+\right)\right\}_{i=1,\ldots,n}\ \bigcup\ \left\{\mathscr{R}_{Z_j}\left(T_+\right)\right\}_{j=1,\ldots,n}\ \right\rangle.$$

Note that we have actually proved that $\mathbf{X}^{\mathbf{a}}\mathbf{Z}^{\mathbf{b}}-\mathbf{X}^{\mathbf{b}}\mathbf{Z}^{\mathbf{a}}=\left(\mathbf{X}^{\mathbf{a}-\mathbf{b}}\mathbf{Z}^{\mathbf{b}-\mathbf{a}}-1\right)\mathbf{X}^{\mathbf{b}}\mathbf{Z}^{\mathbf{a}}\in I_3$ which completes the proof. □

The following lemma is straightforward. However, for convenience of the reader we write its proof, thus making our exposition self-contained.

Lemma 4.50. *Two minimal support codewords in $\mathscr{C}$ with the same support should be one scalar multiple of the other.*

Proof. Suppose the lemma were false. Then we could find two codewords $\mathbf{m}_1,\ \mathbf{m}_2$ of minimal support of $\mathscr{C}$ such that $\mathrm{supp}(\mathbf{m}_1)=\mathrm{supp}(\mathbf{m}_2)$ but $\mathbf{m}_1\neq\lambda\mathbf{m}_2$ for any $\lambda\in\mathbb{F}_q^*$. Let us choose $\lambda\in\mathbb{F}_q^*$ such that $m_1^{(i)}=\lambda m_2^{(i)}$ for at least one index $i\in\mathrm{supp}(\mathbf{m}_1)$, then

$$\mathbf{m}_1-\lambda\mathbf{m}_2\in\mathscr{C}\setminus\{\mathbf{0}\}\quad\text{ and }\quad\mathrm{supp}(\mathbf{m}_1-\lambda\mathbf{m}_2)\subset\mathrm{supp}(\mathbf{m}_1)$$

which contradicts the minimality of $\mathbf{m}_2$. □

Theorem 4.51. *The set of codewords of minimal support of the code $\mathscr{C}$ is a subset of the vectors related to the Graver basis of the ideal associated to $\mathscr{C}$.*

Proof. Let $\mathbf{m}\in\mathscr{M}_{\mathscr{C}}$. Suppose the theorem were false, then no binomial of type $\mathbf{X}^{\mathbf{m}_1}-\mathbf{X}^{\mathbf{m}_2}\in I_+(\mathscr{C})$ with $\mathbf{m}_1-\mathbf{m}_2=\mathbf{m}$ would be primitive. Choose any binomial $\mathbf{X}^{\mathbf{m}_1}-\mathbf{X}^{\mathbf{m}_2}$ with $\mathbf{m}_1-\mathbf{m}_2=\mathbf{m}$ such that the below condition does not hold for $i=1,\ldots,n$:

$$\text{If}\quad x_{ir}\in\mathrm{supp}\left(\mathbf{X}^{\mathbf{m}_1}\right)\quad\text{ and }\quad x_{is}\in\mathrm{supp}\left(\mathbf{X}^{\mathbf{m}_2}\right)\quad\text{ then }\quad x_{ir}x_{is}-1\in\mathscr{R}_{X_i}\left(T_+\right).$$

Therefore, there exists a different binomial $\mathbf{X}^{\mathbf{v}_1}-\mathbf{X}^{\mathbf{v}_2}\in I_+(\mathscr{C})$ such that $\mathbf{X}^{\mathbf{v}_1}$ divides $\mathbf{X}^{\mathbf{m}_1}$ and $\mathbf{X}^{\mathbf{v}_2}$ divides $\mathbf{X}^{\mathbf{m}_2}$, or equivalently,

$$\mathrm{supp}(\Delta\mathbf{v}_1)\subseteq\mathrm{supp}(\Delta\mathbf{m}_1)\quad\text{ and }\quad\mathrm{supp}(\Delta\mathbf{v}_2)\subseteq\mathrm{supp}(\Delta\mathbf{m}_2).$$

Hence, $\mathrm{supp}\left(\Delta(\mathbf{v}_1-\mathbf{v}_2)\right)\subseteq\mathrm{supp}\left(\Delta(\mathbf{m}_1-\mathbf{m}_2)\right)$ which contradicts the minimality of $\mathbf{m}$. □

Remark 4.52. We could get rid of the leftover codewords from the set obtained by the above theorem using Algorithm 22.

Corollary 4.53. *The set of codewords of minimal support of any linear code $\mathscr{C}$ can be computed from the ideal*

$$I_3=\left\langle\ \left\{\mathbf{X}^{\alpha^j\mathbf{w}_i}-\mathbf{Z}^{\alpha^j\mathbf{w}_i}\right\}_{\substack{i=1,\ldots,k\\ j=1,\ldots,q-1}}\ \bigcup\ \left\{\mathscr{R}_{X_i}\left(T_+\right)\right\}_{i=1,\ldots,n}\ \bigcup\ \left\{\mathscr{R}_{Z_i}\left(T_+\right)\right\}_{i=1,\ldots,n}\ \right\rangle.$$

Proof. This result follows directly from Theorems 4.49 and 4.51. □

In the following example we will see how to use the Graver basis to obtain the set of codewords of minimal support of a linear code.

Example 4.54. Consider $\mathscr{C}$ the $[6,3]$ ternary code with generator matrix

$$G_{\mathscr{C}} = \begin{pmatrix} 1 & 0 & 0 & 2 & 2 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 2 & 1 \end{pmatrix} \in \mathbb{F}_3^{3\times 6}.$$

This code has $3^3 = 27$ codewords.

- The zero codeword.
- 16 codewords of minimal support. Note that by Lemma 4.50, if a codeword $\mathbf{c}$ is a codeword minimal support codeword, then all its multiples are also codewords of minimal support. So these 16 codewords represent 8 different supports.

1.	$(1,0,0,2,2,0)$	$(2,0,0,1,1,0)$
2.	$(0,1,0,1,1,0)$	$(0,2,0,2,2,0)$
3.	$(1,1,0,0,0,0)$	$(2,2,0,0,0,0)$
4.	$(0,0,1,1,2,1)$	$(0,0,2,2,1,2)$
5.	$(1,0,1,0,1,1)$	$(2,0,2,0,2,2)$
6.	$(2,0,1,2,0,1)$	$(1,0,2,1,0,2)$
7.	$(0,1,1,2,0,1)$	$(0,2,2,1,0,2)$
8.	$(0,2,1,0,1,1)$	$(0,1,2,0,2,2)$

- Another 10 codewords which do not have minimal support.

$(2,1,0,2,2,0)$	$(1,2,0,1,1,0)$	$(2,1,1,0,1,1)$	$(1,2,2,0,2,2)$
$(1,2,1,2,0,1)$	$(2,1,2,1,0,2)$		
$(2,2,1,1,2,1)$	$(1,1,2,2,1,2)$	$(1,1,1,1,2,1)$	$(2,2,2,2,1,2)$

Let $\alpha = 2$ be a primitive element of $\mathbb{F}_3$ and let us label the rows of G by $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$. By Theorem 4.3, the ideal associated to $\mathscr{C}$ may be defined as the following ideal:

$$\left\langle \left\{\mathbf{x}^{\Delta(\alpha^j \mathbf{w}_i)} - 1\right\}_{\substack{i=1,\dots,3 \\ j=1,\dots,2}} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\dots,n} \right\rangle$$

where $\mathscr{R}_{X_i}(T_+)$ consists of the following binomials

$$\mathscr{R}_{X_i}(T_+) = \left\{ x_{i1}^2 - x_{i2}, \quad x_{i1}x_{i2} - 1, \quad x_{i2}^2 - x_{i1} \right\} \text{ with } i = 1,\dots,6.$$

If we compute a Gröbner basis of $I_+(\mathscr{C})$ w.r.t. a `degrev` ordering we get 41 binomials representing the following set of codewords:

$(0,0,0,0,0,0)$	
$(0,2,2,1,0,2)$	$(0,1,1,2,0,1)$
$(0,1,2,0,2,2)$	$(0,2,1,0,1,1)$
$(0,2,0,2,2,0)$	$(0,1,0,1,1,0)$
$(0,0,2,2,1,2)$	$(0,0,1,1,2,1)$
$(2,2,0,0,0,0)$	$(1,1,0,0,0,0)$

Note that all nonzero codewords are codewords of minimal support but not all codewords of minimal support are represented in the above set.

If we compute a Graver basis of $I_+(\mathscr{C})$, we obtain 4212 binomials representing the following codewords:

$$\begin{array}{ll}
(2,1,2,1,0,2) & (1,2,1,2,0,1)\\
(1,2,2,0,2,2) & (2,1,1,0,1,1)\\
(1,0,2,1,0,2) & (2,0,1,2,0,1)\\
(2,0,2,0,2,2) & (1,0,1,0,1,1)\\
(0,2,2,1,0,2) & (0,1,1,2,0,1)\\
(0,1,2,0,2,2) & (0,2,1,0,1,1)\\
(0,0,2,2,1,2) & (0,0,1,1,2,1)\\
(2,1,0,2,2,0) & (1,2,0,1,1,0)\\
(2,0,0,1,1,0) & (1,0,0,2,2,0)\\
(0,1,0,1,1,0) & (0,2,0,2,2,0)\\
(1,1,0,0,0,0) & (2,2,0,0,0,0)\\
(0,0,0,0,0,0) &
\end{array}$$

Observe that the set $\mathscr{M}_{\mathscr{C}}$ is contained in the previous set.

4.6 Applications to other classes of codes

4.6.1 Modular codes

Chapter 3 is devoted to the study of modular codes $\mathscr{C}$ defined over the ring $\mathbb{Z}_s$, in other words, submodules of $\left(\mathbb{Z}_s^n,+\right)$. The important point in that chapter was the fact that a Graver basis of the lattice ideal associated with a modular code provides the set of codewords of minimal support of our modular code. In this subsection we will provide a *complete decoding procedure* for them. Recall that the reduced Gröbner basis of the lattice ideal associated to the modular code, following the method of Chapter 3, does not allow decoding, see Example 4.29.

Throughout this subsection $\mathscr{C}$ will be a modular code of parameters $[n,k]$ defined over $\mathbb{Z}_s$ and let $\{\mathbf{e}_1,\ldots,\mathbf{e}_{s-1}\}$ be the canonical basis of $\mathbb{Z}^{s-1}$. We will consider on this section the following characteristic crossing functions:

$$\Delta:\ \mathbb{Z}_s \longrightarrow \{0,1\}^{s-1} \qquad \text{and} \qquad \nabla:\ \{0,1\}^{s-1} \longrightarrow \mathbb{Z}_s$$

where the map Δ replace the element $j\in\mathbb{Z}_s\setminus\{0\}$ by the vector $\mathbf{e}_j\in\mathbb{Z}^{s-1}$ and $\mathbf{0}$ by the zero vector $\mathbf{0}\in\mathbb{Z}^{s-1}$. Whereas the map ∇ recovers the element $j_1+2j_2+\ldots+(s-1)j_{s-1}$ from the binary vector $(j_1,\ldots,j_{s-1})$.

Now let $\mathbf{X}$ denote n vector variables $X_1,\ldots,X_n$ such that each variable X_i can be decomposed into $s-1$ components $x_{i1},\ldots,x_{is-1}$ with $i=1,\ldots,n$, representing the nonzero elements of $\mathbb{Z}_s$. Let $\mathbf{a}=(a_1,\ldots,a_n)$ be an n-tuple of elements in the ring $\mathbb{Z}_s$. We will adopt the following notation:

$$\mathbf{X}^{\mathbf{a}}=X_1^{a_1}\cdots X_n^{a_n}=(x_{11}\cdots x_{1s-1})^{\Delta a_1}\cdots(x_{n1}\cdots x_{ns-1})^{\Delta a_n}.$$

This relationship allows us to work with monomials with non-integer exponents as monomials with binary exponents. Note that the degree of the monomial $\mathbf{X}^{\mathbf{a}}$ is defined as the support of the vector $\mathbf{a}$.

As we have already described in this chapter, we define the ideal associated to $\mathscr{C}$ as the binomial ideal:

$$I_+(\mathscr{C}) = \left\langle \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \mathbf{a} - \mathbf{b} \in \mathscr{C} \right\rangle \subseteq \mathbb{K}[\mathbf{X}].$$

Given the rows of a generator matrix of $\mathscr{C}$, labelled by $\mathbf{w}_1, \ldots, \mathbf{w}_k$, we define the following binomial ideal:

$$\Delta I = \left\langle \ \{\mathbf{X}^{\mathbf{w}_i} - 1\}_{i=1,\ldots,k} \quad \bigcup \quad \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n} \ \right\rangle$$

where $\mathscr{R}_{X_i}(T_+)$ consists of all the binomials on the variable X_i associated to the relations given by the additive table of the ring $\mathbb{Z}_s$, i.e.

$$\mathscr{R}_{X_i}(T_+) = \left\{ \begin{array}{c} \{x_{iu}x_{iv} - x_{iw} \mid u+v \equiv w \mod s\} \\ \{x_{iu}x_{iv} - 1 \mid u+v \equiv 0 \mod s\} \end{array} \right\} \text{ with } i = 1, \ldots, n.$$

Proposition 4.55. *The following conditions are equivalent:*

1. $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \in I_+(\mathscr{C})$.
2. *There exists* $\lambda_1, \ldots, \lambda_k \in \mathbb{Z}$ *such that* $\mathbf{X}^{\mathbf{a}+(s-1)\mathbf{b}} = \prod_{i=1}^{k} \mathbf{X}^{\lambda_i \mathbf{w}_i}$.

Proof. Given any binomial $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}}$ in the ideal $I_+(\mathscr{C})$. By definition we have that $\mathbf{a} - \mathbf{b} = \mathbf{a} + (s-1)\mathbf{b} \in \mathscr{C}$. Hence, $\mathbf{a} + (s-1)\mathbf{b} \equiv \lambda_{1,s}\mathbf{w}_1 + \ldots + \lambda_{k,s}\mathbf{w}_k \mod s$ with $\lambda_{i,s} \in \mathbb{Z}_s$ for $i = 1, \ldots, k$. Therefore, $\mathbf{a} + (s-1)\mathbf{b} \equiv \lambda_1\mathbf{w}_1 + \ldots + \lambda_k\mathbf{w}_k \mod s$ where $\lambda_i = \blacktriangle\lambda_{i,s} \in \mathbb{Z}$ for $i = 1, \ldots, k$. The map $\blacktriangle$ was defined on Chapter 2 and replaces the class of $0, 1, \ldots, s-1$ in $\mathbb{Z}_s$ by the same symbols regarded as integers.

The converse inclusion is the above proof read backwards. □

Theorem 4.56. $I_+(\mathscr{C}) = \Delta I$

Proof. This theorem is a fairly straightforward generalization of Theorem 4.3 and Theorem 3.9. It is clear that $\Delta I \subseteq I_+(\mathscr{C})$ since all the binomials in the generating set of ΔI belongs to $I_+(\mathscr{C})$.

To show the converse it suffices to show that each binomial $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}}$ of $I_+(\mathscr{C})$ belongs to ΔI. We will use Proposition 4.55 together with the observation that

$$\mathbf{z}_1 - 1, \ \ \mathbf{z}_2 - 1 \in \Delta I \Longleftrightarrow \mathbf{z}_1\mathbf{z}_2 - 1 \in \Delta I. \tag{4.5}$$

Consider any binomial $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \in I_+(\mathscr{C})$. By Proposition 4.55, there exists $\lambda_1, \ldots, \lambda_k \in \mathbb{Z}$ such that $\mathbf{X}^{\mathbf{a}+(s-1)\mathbf{b}} = \prod_{i=1}^{k} \mathbf{X}^{\lambda_i \mathbf{w}_i}$. Repeated application of Equation 4.5 enables us to write:

$$\prod_{i=1}^{k} \mathbf{X}^{\lambda_i \mathbf{w}_i} - 1 = \left(\prod_{j=1}^{k} \mathbf{X}^{\mathbf{w}_j} - 1 \right) \left(\prod_{j|\lambda_j > 1} \mathbf{X}^{(\lambda_j - 1)\mathbf{w}_j} \right) + \ldots + \left(\prod_{j|\lambda_j > r-1} \mathbf{X}^{\mathbf{w}_j} - 1 \right) \in \Delta I$$

where $r = \max\left\{\lambda_j \mid j = 1, \ldots, k\right\}$. We have proof more, namely that $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \in \Delta I$ since

$$\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} = \mathbf{X}^{\mathbf{b}} \underbrace{\left(\mathbf{X}^{\mathbf{a}+(s-1)\mathbf{b}} - 1\right)}_{\in \Delta I} - \mathbf{X}^{\mathbf{a}} \underbrace{\left(\mathbf{X}^{s\mathbf{b}} - 1\right)}_{\in \Delta I}.$$

□

Remark 4.57. Note that the main difference of the set of generators describing the ideal associated with a modular code, respect to the set of generators of the ideal related with a $\mathbb{F}_q$-linear code, is the fact that for the second set we need to add all the multiples in $\mathbb{F}_q$ of each row $\mathbf{w}_i$, while for the first ideal this is not necessary as consequence of Proposition 4.55. Moreover, the previous result can be extended for codes over $\mathbb{F}_p$ with p prime since $\mathbb{F}_p \equiv \mathbb{Z}_p$.

Example 4.58. Continuing with Example 3.25 where we considered a $[5, 3, 1]$ modular code defined over $\mathbb{Z}_4$ with generator and parity check matrices:

$$G = \begin{pmatrix} 2 & 1 & 0 & 1 & 1 \\ 1 & 2 & 3 & 1 & 0 \\ 2 & 3 & 0 & 0 & 3 \end{pmatrix} \in \mathbb{Z}_4^{3\times 5} \quad \text{and} \quad H = \begin{pmatrix} 1 & 0 & 1 & 0 & 2 \\ 0 & 1 & 2 & 0 & 3 \end{pmatrix} \in \mathbb{Z}_4^{2\times 5},$$

respectively. The following table is related to the additive structure of $\mathbb{Z}_4$:

T_+	1	2	3
1	2	3	0
2		0	1
3			2

This table yields to the following binomials:

$$\mathscr{R}_{X_i}(T_+) = \left\{ \begin{array}{l} x_{i1}^2 - x_{i2}, x_{i1}x_{i2} - x_{i3}, x_{i1}x_{i3} - 1, \\ x_{i2}^2 - 1, x_{i2}x_{i3} - x_{i1}, \\ x_{i3}^2 - x_{i2}, \end{array} \right\} \text{ with } i = 1, \ldots, 5.$$

Thus, we may define the ideal $I_+(\mathscr{C})$ as the following ideal:

$$I_+(\mathscr{C}) = \left\langle \left\{ \begin{array}{l} x_{12}x_{21}x_{41}x_{51} - 1 \\ x_{11}x_{22}x_{33}x_{41} - 1 \\ x_{12}x_{23}x_{53} - 1 \end{array} \right\} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,5} \right\rangle \subseteq \mathbb{K}[X_1, \ldots, X_5]$$

where we decompose each variable X_i with $i = 1, \ldots, 5$ into 3 variables x_{i1}, x_{i2}, x_{i3} representing each nonzero element of $\mathbb{Z}_4$.

Taking into account the new definition of the ideal associated to a modular code we can apply all the results of this chapter to these types of codes. That is to say, we can use FGLM techniques to obtain a Gröbner basis of our ideal (using a direct

adaptation of algorithm 21). Moreover, Section 4.2 gives us a method to compute a Gröbner representation of our modular code (just apply Algorithm 18 with the new set of generators). All outcomes of Section 4.3 are also valid for modular codes. We present below only the main results.

Let $\mathscr{G} = \{g_1, \ldots, g_r\}$ be the reduced Gröbner basis of $I_+(\mathscr{C})$ w.r.t. any degree compatible ordering $\prec$. Moreover for all $i \in \{1, \ldots, r\}$ we define

$$g_i = \mathbf{X}^{g_i^+} - \mathbf{X}^{g_i^-} \quad \text{with} \quad \mathbf{X}^{\mathbf{g}_i^+} \succ \mathbf{X}^{\mathbf{g}_i^-} \quad \text{and} \quad \mathbf{g}_i^+ - \mathbf{g}_i^- \in \mathscr{C}.$$

Theorem 4.59. *Let t be the error-correcting capacity of $\mathscr{C}$. If $\deg\left(\mathrm{Red}_\prec(\mathbf{X}^{\mathbf{a}}, \mathscr{G})\right) \leq t$, then the vector $\mathbf{e} \in \mathbb{Z}_s^n$, verifying that $\mathbf{X}^{\mathbf{e}} = \mathrm{Red}_\prec(\mathbf{X}^{\mathbf{a}}, \mathscr{G})$, is the error vector corresponding to the received word $\mathbf{a} \in \mathbb{Z}_s^n$. Otherwise $\mathbf{a}$ contains more than t errors.*

However, in any case, $\mathrm{Red}_\prec(\mathbf{X}^{\mathbf{a}}, \mathscr{G})$ provides a coset leader even if $\mathrm{w}_H(\mathbf{e}) \geq t$.

Proof. The proof is straightforward from Theorem 4.19. □

Proposition 4.60. *The set $\mathscr{T} = \left\{\mathbf{g}_i^+ - \mathbf{g}_i^- \mid i = 1, \ldots, r\right\}$ is a test-set for $\mathscr{C}$.*

Proof. The proof is analogous to the proof of Proposition 4.30. □

Example 4.61. Continuing with Example 4.58. We can enumerate the different 16 cosets of $\mathscr{C}$ and compute their syndromes to arrive at the syndrome lookup table displayed in Table 4.3.

Syndrome	Coset leader(s)
$(0,0)$	zero vector
$(0,1)$	$(0,1,0,0,0)$
$(0,2)$	$(0,0,0,0,2)$, $(0,2,0,0,0)$
$(0,3)$	$(0,3,0,0,0)$
$(1,0)$	$(1,0,0,0,0)$
$(1,1)$	$(0,0,3,0,1)$, $(0,3,1,0,0)$, $(1,1,0,0,0)$, $(3,0,0,0,3)$
$(1,2)$	$(0,0,1,0,0)$
$(1,3)$	$(0,0,3,0,3)$, $(0,1,1,0,0)$, $(1,3,0,0,0)$, $(3,0,0,0,1)$
$(2,0)$	$(0,0,2,0,0)$, $(2,0,0,0,0)$
$(2,1)$	$(0,0,0,0,3)$
$(2,2)$	$(0,0,2,0,2)$, $(0,1,0,0,3)$, $(0,2,2,0,0)$, $(0,3,0,0,1)$, $(1,0,1,0,0)$, $(2,0,0,0,2)$, $(2,2,0,0,0)$, $(3,0,3,0,0)$
$(2,3)$	$(0,0,0,0,1)$
$(3,0)$	$(3,0,0,0,0)$
$(3,1)$	$(0,0,1,0,1)$, $(0,3,3,0,0)$, $(1,0,0,0,3)$, $(3,1,0,0,0)$
$(3,2)$	$(0,0,3,0,0)$
$(3,3)$	$(0,0,1,0,3)$, $(0,1,3,0,0)$

Table 4.3: Syndrome lookup table of Example 4.61

If we compute a reduced Gröbner basis $\mathscr{G}$ of $I_+(\mathscr{C})$ w.r.t. a degree reverse lexicographic order, induced by the following ordering on the variables

$$\underbrace{x_{11} > x_{12} > x_{13}}_{X_1} > \underbrace{x_{21} > x_{22} > x_{23}}_{X_2} > \ldots > \underbrace{x_{51} > x_{52} > x_{53}}_{X_5}.$$

And we take the vectors $\mathbf{w} \in \mathbb{Z}_4^5$ such that $\mathbf{X}^{\mathbf{w}}$ is a standard monomial in $\mathscr{G}$, we get the following elements:

$$\begin{array}{ccc} & (0,1,0,0,0) & (0,0,0,0,2) & (0,3,0,0,0) \\ (1,0,0,0,0) & (3,0,0,0,3) & (0,0,1,0,0) & (0,0,3,0,3) \\ (0,0,2,0,0) & (0,0,0,0,3) & (0,1,0,0,3) & (0,0,0,0,1) \\ (3,0,0,0,0) & (1,0,0,0,3) & (0,0,3,0,0) & (0,0,1,0,3) \end{array}$$

Note that all standard monomials in $\mathscr{G}$ represent a coset leader of $\mathscr{C}$.

If we adapt the ideas of Subsection 4.3.2, we manage to define the Border basis and its reduced structure, the reduced basis, of a modular code. Therefore, again, we have two ways of reducing a monomial using either the reduced basis or the Matphi function of its Gröbner representation. These two ways of understanding the reduction process lead to two different descent decoding procedures: the one where the search is done changing the coset representative (similar to Liebler's algorithm which was defined just for binary codes) and the one given by descending within the same coset (which is an extension of the algorithm of Ashikhmin and Barg for the non-binary case).

4.6.2 Multiple Alphabets

Along this section $\mathscr{C}$ will a submodule of dimension k defined over the multiple alphabets $\mathbb{Z}_{s_1} \times \ldots \times \mathbb{Z}_{s_n}$. For simplicity of notation we write $\left\{\mathbf{e}_1^s, \ldots, \mathbf{e}_{s-1}^s\right\}$ for the canonical basis of $\mathbb{Z}^{s-1}$. We will consider the following characteristic crossing functions (which are the same as the one considered in the previous subsection):

$$\Delta_s : \quad \mathbb{Z}_s \quad \longrightarrow \quad \{0,1\}^{s-1} \qquad \text{and} \qquad \nabla_s : \quad \{0,1\}^{s-1} \quad \longrightarrow \quad \mathbb{Z}_s$$

where the map Δ_s replaces the element $j \in \mathbb{Z}_s \setminus \{0\}$ by the vector $\mathbf{e}_j^s \in \mathbb{Z}^{s-1}$ and $\mathbf{0}$ by the zero vector $\mathbf{0} \in \mathbb{Z}^{s-1}$. Whereas the map ∇_s recovers the element $j_1 + 2j_2 + \ldots + (s-1)j_{s-1}$ from the binary vector $(j_1, \ldots, j_{s-1})$.

Let $\mathbf{X}$ stands for n vector variables $X_1, \ldots, X_n$ such that each variable X_i can be decomposed into $s_i - 1$ components $x_{i1}, \ldots, x_{is_i-1}$ with $i = 1, \ldots, n$ representing the non zero element of $\mathbb{Z}_{s_i}$. Let $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_{s_1} \times \ldots \times \mathbb{Z}_{s_n}$. We will adopt the following notation:

$$\mathbf{X}^{\mathbf{a}} = X_1^{a_1} \cdots X_n^{a_n} = \left(x_{11} \cdots x_{1s_1-1}\right)^{\Delta_{s_1} a_1} \cdots \left(x_{n1} \cdots x_{ns_n-1}\right)^{\Delta_{s_n} a_n}.$$

Similar to the modular case, given the rows of a generator matrix of $\mathscr{C}$, labelled by $\mathbf{w}_1, \ldots, \mathbf{w}_k$, we may define the ideal associated to $\mathscr{C}$ as the following binomial ideal:

$$I_+(\mathscr{C}) = \left\langle \ \{\mathbf{X}^{\mathbf{w}_i} - 1\}_{i=1,\ldots,k} \ \bigcup \ \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n} \ \right\rangle.$$

Remark 4.62. The main difference with the modular case is that the relations $\mathscr{R}_{X_i}(T_+)$ could be different for each $i \in \{1,\ldots,n\}$.

Example 4.63. For this example let us take a linear subgroup $\mathscr{C}$ defined in $\mathbb{Z}_2 \times \mathbb{Z}_6 \times \mathbb{Z}_6 \times \mathbb{Z}_2$ with generator matrix

$$G = \begin{pmatrix} 1 & 1 & 4 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

This code has 12 codewords consisting of the following set:

$$\begin{array}{cccc} (1,1,4,0) & (0,2,2,0) & (1,3,0,0) & (0,4,4,0) \\ (1,5,2,0) & (1,1,1,1) & (1,3,3,1) & (1,5,5,1) \\ (0,0,3,1) & (0,2,5,1) & (0,4,1,1) & (0,0,0,0) \end{array}$$

In $\mathbb{Z}_2$ there is only one nonzero element whose additive structure yields to the following binomial:

$$\mathscr{R}_{X_i}(T_+) = \left\{x_i^2 - 1\right\} \text{ with } i = 1,4.$$

We represent each nonzero element of $\mathbb{Z}_6$ by a distinct variable : $x_{i1},\ldots,x_{i5}$ with $i = 2,3$. The following table is related to the additive structure of $\mathbb{Z}_6$:

T_+	1	2	3	4	5
1	2	3	4	5	0
2		4	5	0	1
3			0	1	2
4				2	3
5					4

This table yields to the following binomials:

$$\mathscr{R}_{X_i}(T_+) = \left\{ \begin{array}{l} x_{i1}^2 - x_{i2}, x_{i1}x_{i2} - x_{i3}, x_{i1}x_{i3} - x_{i4}, x_{i1}x_{i4} - x_{i5}, x_{i1}x_{i5} - 1, \\ x_{i2}^2 - x_{i4}, x_{i2}x_{i3} - x_{i5}, x_{i2}x_{i4} - 1, x_{i2}x_{i5} - x_{i1}, \\ x_{i3}^2 - 1, x_{i3}x_{i4} - x_{i1}, x_{i3}x_{i5} - x_{i2}, \\ x_{i4}^2 - x_{i2}, x_{i4}x_{i5} - x_{i3}, \\ x_{i5}^2 - x_{i4} \end{array} \right\}$$

for $i = 2,3$. Let us label the rows of G by $\mathbf{w}_1$ and $\mathbf{w}_2$. The ideal associated to $\mathscr{C}$ may be defined as the following binomial ideal:

$$\begin{aligned} I_+(\mathscr{C}) &= \left\langle \{\mathbf{X}^{\mathbf{w}_i} - 1\}\, i = 1,2 \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,2,3,4} \right\rangle \\ &= \left\langle \left\{ \begin{array}{ll} x_1x_{21}x_{34} - 1, & x_1x_{21}x_{31}x_4 - 1, \\ x_{22}x_{32} - 1, & \\ x_1x_{23} - 1, & x_1x_{23}x_{33}x_4 - 1, \\ x_{24}x_{34} - 1, & x_{24}x_{34} - 1, \\ x_1x_{25}x_{32} - 1, & x_1x_{25}x_{35}x_4 - 1 \end{array} \right\} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,2,3,4} \right\rangle \end{aligned}$$

which is an ideal on $\mathbb{K}[x_1, x_{21},\ldots,x_{25}, x_{31},\ldots,x_{35}, x_4]$. The exponents of a Gröbner basis of the above ideal gives the complete set of codewords of $\mathscr{C}$.

It is clear that all the results of $\mathbb{F}_q$-linear codes could be applied to these types of codes.

4.6.3 Additive codes

Let $\mathbb{F}_{q_1}$ be an algebraic extension of $\mathbb{F}_{q_2}$, i.e. $q_1 = p^{r_1}$ and $q_2 = p^{r_2}$ where p is a prime number and r_2 divides r_1. An $\mathbb{F}_{q_2}$-additive code $\mathscr{C}$ of parameters $[n,k]$ over $\mathbb{F}_{q_1}$ is a submodule of $\mathbb{F}_{q_1}^n$, which is closed under the addition in $\mathscr{C}$ and closed under scalar multiplication over the field $\mathbb{F}_{q_2}$. In other words, given the rows of a generator matrix of $\mathscr{C}$ labelled by $\mathbf{w}_1, \ldots, \mathbf{w}_k \in \mathbb{F}_{q_1}^n$, the set of codewords of $\mathscr{C}$ may be defined as:

$$\left\{\alpha_1\mathbf{w}_1 + \ldots + \alpha_k\mathbf{w}_k \mid \alpha_i \in \mathbb{F}_{q_2} \text{ for } i = 1, \ldots, k\right\}.$$

Let α be a primitive element of $\mathbb{F}_{q_2}$. We check at once that the binomial ideal associated to $\mathscr{C}$ is defined by the following binomial ideal:

$$I_+(\mathscr{C}) = \left\langle \left\{\mathbf{X}^{\alpha^j\mathbf{w}_i} - 1\right\}_{\substack{i=1,\ldots,k \\ j=1,\ldots,q_2-1}} \bigcup \left\{\mathscr{R}_{X_i}\left(T_+\right)\right\}_{i=1,\ldots,n} \right\rangle$$

where $\mathscr{R}_{X_i}\left(T_+\right)$ consist of all the binomials on the variable X_i associated to the relations given by the additive table of the field $\mathbb{F}_{q_1}$.

Of course, the results obtained for $\mathbb{F}_q$-linear codes could be adapted to additive codes.

Remark 4.64. In the case that q_2 is a prime number, the number of generators of $I_+(\mathscr{C})$ would be reduced taking account of Remark 4.57.

Example 4.65. Let α be a primitive element of $\mathbb{F}_9$, i.e.

$$\mathbb{F}_9 = \left\{0, \alpha, \alpha^2 = \alpha + 1, \alpha^3 = 2\alpha + 1, \alpha^4 = 2, \alpha^5 = 2\alpha, \alpha^6 = 2\alpha + 2, \alpha^7 = \alpha + 2, \alpha^8 = 1\right\}.$$

This representation yields to the following additive table:

T_+	1	2	3	4	5	6	7	8
1	5	3	8	7	0	4	6	2
2		6	4	1	8	0	5	7
3			7	5	2	1	0	6
4				8	6	3	2	0
5					1	7	4	3
6						2	8	5
7							3	1
8								4

Hence, we obtain the following binomials associated to the previous rules:

$$\mathscr{R}_{X_i}(\mathscr{T}_+) = \left\{ \begin{array}{l} x_{i1}^2 - x_{i5}, x_{i1}x_{i2} - x_{i3}, x_{i1}x_{i3} - x_{i8}, x_{i1}x_{i4} - x_{i7}, x_{i1}x_{i5} - 1, x_{i1}x_{i6} - x_{i4}, \\ \quad x_{i1}x_{i7} - x_{i6}, x_{i1}x_{i8} - x_{i2}, \\ x_{i2}^2 - x_{i6}, x_{i2}x_{i3} - x_{i4}, x_{i2}x_{i4} - x_{i1}, x_{i2}x_{i5} - x_{i8}, x_{i2}x_{i6} - 1, x_{i2}x_{i7} - x_{i5}, \\ \quad x_{i2}x_{i8} - x_{i7}, \\ x_{i3}^2 - x_{i7}, x_{i3}x_{i4} - x_{i5}, x_{i3}x_{i5} - x_{i2}, x_{i3}x_{i6} - x_{i1}, x_{i3}x_{i7} - 1, x_{i3}x_{i8} - x_{i6}, \\ x_{i4}^2 - x_{i8}, x_{i4}x_{i5} - x_{i6}, x_{i4}x_{i6} - x_{i3}, x_{i4}x_{i7} - x_{i2}, x_{i4}x_{i8} - 1, \\ x_{i5}^2 - x_{i1}, x_{i5}x_{i6} - x_{i7}, x_{i5}x_{i7} - x_{i4}, x_{i5}x_{i8} - x_{i1}, \\ x_{i6}^2 - x_{i2}, x_{i6}x_{i7} - x_{i8}, x_{i6}x_{i8} - x_{i5}, \\ x_{i7}^2 - x_{i3}, x_{i7}x_{i8} - x_{i1}, \\ x_{i8}^2 - x_{34} \end{array} \right\}$$

for $i = 1, 2, 3$. Consider the $\mathbb{F}_3$-additive code $\mathscr{C}$ of parameters $[3, 2]$ with generator matrix:

$$G = \begin{pmatrix} 1 & 0 & \alpha + 1 \\ 0 & 1 & 0 \end{pmatrix} \in \mathbb{F}_9^{2\times 3}.$$

Let us label the rows of G by $\mathbf{w}_1$ and $\mathbf{w}_2$. Note that this code has 9 codewords which consist of the set $\{\beta_1\mathbf{w}_1 + \beta_2\mathbf{w}_2 \mid \beta_1, \beta_2 \in \mathbb{F}_3\}$. Therefore, we may define the ideal $I_+(\mathscr{C})$ as the following ideal:

$$I_+(\mathscr{C}) \quad = \quad \left\langle \quad \left\{\mathbf{X}^{\alpha^j \mathbf{w}_i} - 1\right\}_{\substack{i=1,\ldots,k \\ j=4,8}} \quad \bigcup \quad \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n} \quad \right\rangle.$$

5
A semigroup approach

Contents

The purpose of this chapter is to show how some error-correcting codes can be understood by means of appropriate commutative semigroups with given generators. This approach involves the description of several kinds of objects such as abelian groups, lattices, algebras and binomial ideals. Our approach provides the relationship among the above objects for better understanding the techniques used in this thesis and brings a new perspective for future developments in the area.

Given an error-correcting code $\mathscr{C}$ we define the ideal associated to this object as the binomial ideal

$$I(\mathscr{C}) = \left\langle \left\{ \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \mathbf{a} - \mathbf{b} \in \mathscr{C} \right\} \right\rangle \subseteq \mathbb{K}[\mathbf{X}].$$

Take notice that $\mathbf{X}^{\gamma}$ denotes an usual term on $\mathbb{K}[\mathbf{X}]$ where γ belongs to the algebraic structure of the code $\mathscr{C}$. Therefore, it seems natural to ask if the above ideal match

a semigroup ideal $I(S)$ given by some specific semigroup S. This construction establishes a strong relation between codes and semigroups and constitutes a means to apply numerous results in the field of semigroups to problems in information theory.

Thus some problems of Coding Theory could be addressed using techniques inspired by toric mathematics from semigroups. References [22, 26, 55, 107, 32] can be consulted for a detailed exposition of semigroups and its applications.

The chapter is organized as follows. Section 5.1 is intended to be a brief review of some basics definitions and known results from the theory of semigroups. We will restrict our attention to commutative and finitely generated semigroups classes.

Section 5.2 is devoted to the study of several representations for the semigroup associated with a modular code while Section 5.3 deals with the case of linear codes. We emphasize that, although there exists different semigroups ideals associated to the same semigroup depending on the chosen set of generators, the choice of *digital* representations seems to be the best adapted to perform complete decoding on the selected codes.

For the convenience of the reader we repeat the relevant material from the previous chapters thus making the exposition of this chapter self-contained.

5.1 Overview of semigroups

Let S be a commutative semigroup with an identity element denoted as $\mathbf{0} \in S$. In other words, S is a set endowed with an internal commutative operation denoted by $+$ such that $\mathbf{0}+\mathbf{a}=\mathbf{a}+\mathbf{0}=\mathbf{a}$, $\forall a \in S$.

All semigroups in this chapter are assumed to be finitely generated, thus, there exists a fixed finitely system of generators $\mathbf{n}_1,\ldots,\mathbf{n}_r$ in S such that every element $\mathbf{m} \in S$ can be written in the form

$$\mathbf{m}=\sum_{i=1}^{r}\alpha_i\mathbf{n}_i \text{ with } \alpha_i \in \mathbb{N}.$$

The notion of cancellative semigroup is related with the cancellation property. That is to say, if $\mathbf{m}+\mathbf{n}=\mathbf{m}+\mathbf{n}'$ with $\mathbf{m},\mathbf{n},\mathbf{n}' \in S$ then $\mathbf{n}=\mathbf{n}'$.

Moreover, S is combinatorially finite if there is only finitely many ways to write every $\mathbf{a} \in S \setminus \{\mathbf{0}\}$ as a sum $\mathbf{a}=\mathbf{a}_1+\ldots+\mathbf{a}_s$ with $\mathbf{a}_i \in S \setminus \{\mathbf{0}\}$.

Let $G(S)$ be the associated commutative group of S, i.e. every homomorphism from S to a group passes through a unique semigroup homomorphism

$$i: \quad S \quad \longrightarrow \quad G(S) \ .$$

The commutative group $G(S)$ exists and is unique up to isomorphism. Furthermore, $G(S)$ is finitely generated when S is. Note that $G(S)$ is also defined as the group of classes of pairs $(\mathbf{m},\mathbf{n}) \in S \times S$ under the relation $\sim$ where $(\mathbf{m},\mathbf{n}) \sim (\mathbf{m}',\mathbf{n}')$ if and only if $\mathbf{m}+\mathbf{n}'=\mathbf{m}'+\mathbf{n}$. Note that every abelian group is a subgroup of itself and a semigroup. Moreover, S is cancellative if and only if it can be embedded into a group,

that is if the canonical homomorphism i is injective. In this case, S is combinatorially finite if and only if $S \cap (-S) = \{\mathbf{0}\}$.

Any subset of a semigroup with an identity element which is closed under the semigroup operation is known as subsemigroup. Any subsemigroups H of an abelian group G is cancellative and its associated commutative group $G(H)$ is identified to the smallest subgroup of G containing H. When S is cancellative it can be naturally identify with a subsemigroup of $G(S)$. Therefore, we may view cancellative semigroups (up to isomorphism) as subsemigroups of abelian groups. Thus, $G(S)$ is, up to isomorphism, the smallest abelian group of which S is a subsemigroup, in other words, $G(S)$ is the smallest group containing S.

If S is finitelly generated then we may assume that

$$S \subset \mathbb{Z}^n \oplus \mathbb{Z}/a_1\mathbb{Z} \oplus \ldots \oplus \mathbb{Z}/a_s\mathbb{Z},$$

where $a_1, \ldots, a_s$ are non-zero integers and $1 < a_1/a_2/\ldots/a_s$ are uniquely determined. In particular, $G(S) \simeq \mathbb{Z}^n \oplus T$ where T is the torsion subgroup of $G(S)$ i.e.

$$T = \{\mathbf{g} \in G(S) \mid \exists m \in \mathbb{N} \setminus \{0\} \ : \ m\mathbf{g} = 0\}.$$

That is, an element $\mathbf{a}$ of a group G is called a torsion element of the group if it has finite order, i.e. if there exists a positive integer $m \in \mathbb{N}$ such that $m\mathbf{a} = \mathbf{0}$. A group is called a torsion group if all its elements are torsion elements, i.e. all its elements has finite order. Therefore all finite groups are torsion groups.

The choice of a system of generators $\{\mathbf{n}_1, \ldots, \mathbf{n}_r\}$ of S induces a natural semigroup morphism $\pi : \ \mathbb{N}^r \longrightarrow S$ given by $\pi(\mathbf{e}_i) = \mathbf{n}_i$, where $\{\mathbf{e}_i \mid i = 1, \ldots, r\}$ denotes the canonical basis of $\mathbb{N}^r$. And, more generally, $\pi(\mathbf{a}) = \sum_{i=1}^{r} a_i \mathbf{n}_i$ for every $\mathbf{a} \in \mathbb{N}^r$.

Let $\mathbb{K}$ denotes an arbitrary field and $\mathbb{K}[\mathbf{X}] = \mathbb{K}[X_1, \ldots, X_r]$ denotes the polynomial ring in r variables over $\mathbb{K}$. We write $\mathbb{K}[S]$ for the $\mathbb{K}$-vector space:

$$\mathbb{K}[S] = \left\{ \sum_{\mathbf{n} \in S} a_{\mathbf{n}} \mathbf{t}^{\mathbf{n}} \mid a_{\mathbf{n}} \in \mathbb{K} \right\}$$

endowed with a multiplication which is $\mathbb{K}$-linear and satisfies that $\mathbf{t}^{\mathbf{a}} \cdot \mathbf{t}^{\mathbf{b}} = \mathbf{t}^{\mathbf{a}+\mathbf{b}}$ with $\mathbf{a}, \mathbf{b} \in S$. Thus $\mathbb{K}[S]$ has a natural $\mathbb{K}$-algebra structure and we will refer to it as the semigroup algebra of S.

The semigroup morphism π described above defines a $\mathbb{K}$-algebra morphism:

$$\begin{array}{rccc} \varphi : & \mathbb{K}[\mathbf{X}] & \longrightarrow & \mathbb{K}[S] \\ & X_i & \longmapsto & \mathbf{t}^{\mathbf{n}_i} \end{array}$$

The ideal $I(S) = \ker(\varphi)$ is called the semigroup ideal associated to S. It is well known (see [55]) that $I(S)$ is a binomial ideal finitely generated by:

$$I(S) = \left\langle \left\{ \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \sum_{i=1}^{r} a_i \mathbf{n}_i = \sum_{i=1}^{r} b_i \mathbf{n}_i \text{ with } \mathbf{a}, \mathbf{b} \in \mathbb{N}^r \right\} \right\rangle.$$

Notice that, $I(S)$ is dependent on the chosen system of generators and we can expect a wide range of generating set choice. When no confusion arise, we will simply refer to these ideals as $I(S)$. Otherwise we should specify the representation chosen to define S, that is, we will denote by $I_F(S)$ the semigroup ideal associated to S when S is defined by the set of generators F.

The semigroup ideal $I(S)$ is called a toric ideal when $G(S)$ is torsion free, see [107, Chapter 4]. By Nakayama's lemma (see [22]) if S is combinatorially finite, then every minimal binomial generating sets of $I(S)$ have the same cardinality.

In what follows S stands for a commutative cancellative and finitely generated semigroup with zero element. The following lemma allows a characterization of combinatorially finite semigroups.

Lemma 5.1. *Given a semigroup S and let $I(S)$ be its semigroup ideal over $\mathbb{K}[\mathbf{X}]$. S is combinatorially finite if and only if there are no binomial in $I(S)$ of the form $\mathbf{X}^{\mathbf{a}} - 1$.*

Proof. See [115, Lemma I-B.3]. Fix a set of generators $\{\mathbf{n}_1, \ldots, \mathbf{n}_r\}$ of S. By definition $I(S) = \ker(\varphi)$. Thus, if $\mathbf{X}^{\mathbf{a}} - 1 \in I(S)$ then, $\sum_{i=1}^{r} a_i \mathbf{n}_i = \mathbf{0}$ with $\mathbf{a} = (a_1, \ldots, a_r) \in \mathbb{N}^r$. That is to say $S \cap (-S) \neq \{\mathbf{0}\}$. The converse is proved by reading the above backwards. □

Given a lattice $\mathscr{L} \subset \mathbb{Z}^r$, the binomial ideal

$$I_{\mathscr{L}} = \left\langle \left\{ \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \mathbf{a} - \mathbf{b} \in \mathscr{L} \right\} \right\rangle$$

is called the lattice ideal associated to $\mathscr{L}$.

Let us describe the lattice $\mathscr{L}$ as the set of integer solutions of the linear system $A\mathbf{X} = \mathbf{0}$ where $A = \{\mathbf{n}_1, \ldots, \mathbf{n}_r\}$ is a fix system of generators of S, i.e.

$$\mathscr{L} = \left\{ \mathbf{u} \in \mathbb{Z}^r \mid \sum_{i=1}^{r} u_i \mathbf{n}_i = \mathbf{0} \right\} \subseteq \mathbb{Z}^r.$$

Induced by the semigroup homomorphism π, any set of generators $\{\mathbf{n}_1, \ldots, \mathbf{n}_r\}$ provides a group homomorphism

$$\Pi : \quad \mathbb{Z}^r \quad \longrightarrow \quad G(S)$$

given by $\Pi(\mathbf{e}_i) = \mathbf{n}_i$ where $\{\mathbf{e}_i \mid i = 1, \ldots, r\}$ denotes the canonical basis of $\mathbb{Z}^r$. Thus $\Pi(\mathbf{a}) = \sum_{i=1}^{r} a_i \mathbf{n}_i$ for every $\mathbf{a} \in \mathbb{Z}^r$. Notices that, for the semigroup $\mathbb{N}^r$ one has $G(\mathbb{N}^r) = \mathbb{Z}^r$.

Therefore, if $\ker \Pi$ is equal to $\mathscr{L}$ then the lattice ideal $I_{\mathscr{L}}$ match the semigroup ideal $I(S)$, i.e.

$$I_{\mathscr{L}} = I(S) = \left\langle \left\{ \mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} \mid \mathbf{a} - \mathbf{b} \in \mathscr{L} \right\} \right\rangle.$$

This case induces an exact sequence of abelian groups given by:

$$0 \quad \longrightarrow \quad \mathscr{L} \quad \longrightarrow \quad G(\mathbb{N}^r) = \mathbb{Z}^r \quad \longrightarrow \quad G(S) \quad \longrightarrow \quad 0.$$

5.2 The semigroup associated with a modular code

Let us consider the integer $m \geq 2$. We define the following characteristic crossing functions:

$$\blacktriangledown : \ \mathbb{Z}^s \longrightarrow \mathbb{Z}_m^s \quad \text{and} \quad \blacktriangle : \ \mathbb{Z}_m^s \longrightarrow \mathbb{Z}^s$$

where s is determined by context. The map $\blacktriangledown$ is reduction modulo m while the map $\blacktriangle$ replace the class of $0, 1, \ldots, m-1$ by the same symbols regarded as integers. Both maps act coordinate-wise.

Let $\mathbf{X}$ denotes n variables $X_1, \ldots, X_n$. Note that, for every $\mathbf{a} \in \mathbb{Z}_m^r$ we have that $\mathbf{X}^{\blacktriangle \mathbf{a}} = X_1^{\blacktriangle a_1} \cdots X_n^{\blacktriangle a_n} \in \mathbb{K}[\mathbf{X}]$. Therefore, these functions enable us to go back to the usual definition of terms in $\mathbb{K}[\mathbf{X}]$.

Throughout this section $\mathscr{C}$ will be a modular code defined over $\mathbb{Z}_m$, where $\mathbb{Z}_m$ denotes the ring of integer modulo m. Recall that a modular code $\mathscr{C}$ over $\mathbb{Z}_m$ of length n and rank k is an additive subgroup of $\left(\mathbb{Z}_m^n, +\right)$ which has a basis form by k codewords. By a basis for $\mathscr{C}$ we understand a set of codewords that are independent, modular independent and generate $\mathscr{C}$. For a deeper discussion of basis for modular codes we refer to Section 1.1.2.

Given a generator matrix $G \in \mathbb{Z}_m^{k \times n}$ of $\mathscr{C}$ and let label its rows by $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\} \subseteq \mathbb{Z}_m^n$. By Theorem 3.9 we know that the lattice ideal $I_m(\mathscr{C})$ associated with $\mathscr{C}$ is the ideal generated by

$$I_m(\mathscr{C}) = \left\langle \ \{\mathbf{X}^{\mathbf{w}_i} - 1\}_{i=1,\ldots,k} \ \bigcup \ \left\{X_j^m - 1\right\}_{j=1,\ldots,n} \ \right\rangle \subseteq \mathbb{K}[\mathbf{X}].$$

Let $H \in \mathbb{Z}_m^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. The above ideal is also the semigroup ideal associated to the commutative semigroup S finitely generated by $\mathbf{h}_1, \ldots, \mathbf{h}_n$ where $\mathbf{h}_j \in \mathbb{Z}_m^{n-k}$ denotes the j-th column of H.

Proposition 5.2. *Let $\mathscr{C}$ be a modular code of length n and* rank *k defined over $\mathbb{Z}_m$ and let $H \in \mathbb{Z}_m^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. Consider the commutative semigroup S finitely generated by $\mathbf{h}_1, \ldots, \mathbf{h}_n$ where $\mathbf{h}_j$ denotes the j-th column of H. Then $I(S) = I_m(\mathscr{C})$.*

Proof. It is clear that $I_m(\mathscr{C}) \subseteq I(S)$ since all binomials in the generating set of $I_m(\mathscr{C})$ belong to $I(S)$. Indeed we distinguish two cases:

- The equality $GH^T \equiv 0 \mod m$, which holds for any generator matrix $G \in \mathbb{Z}_m^{k\times n}$ of $\mathscr{C}$, implies that $\mathbf{X}^{\mathbf{w}_i} - 1 \in I(S)$ since $\sum_{j=1}^n (\blacktriangle w_{ij})\mathbf{h}_j \equiv 0 \mod m$, where the element $\mathbf{w}_i = \left(w_{i1}, \ldots, w_{in}\right) \in \mathbb{Z}_m^n$ denotes the i-th row of G for all $i = 1, \ldots, k$.

- Moreover $m\mathbf{h}_j \equiv 0 \mod m$ for all $j = 1, \ldots, n$, since $\mathbf{h}_j \in \mathbb{Z}_m^{n-k}$. We thus get $X_j^m - 1 \in I(S)$.

To show the converse, it suffice to make the following observations:

$$\begin{aligned}
\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}} &\iff \sum_{i=1}^{r} a_i \mathbf{h}_i \equiv \sum_{i=1}^{r} b_i \mathbf{h}_i \mod m \text{ with } \mathbf{a}, \mathbf{b} \in \mathbb{N}^r \\
&\iff \sum_{i=1}^{r} (a_i - b_i)\, \mathbf{h}_i \equiv \mathbf{0} \mod m \text{ with } \mathbf{a}, \mathbf{b} \in \mathbb{N}^r \\
&\iff (\mathbf{a} - \mathbf{b}) \in \blacktriangle\mathscr{C} + (m\mathbb{Z}^n)
\end{aligned}$$

In other words we can see the semigroup ideal $I(S)$ as an elimination ideal of the $\mathbb{Z}$-kernel of the matrix $A \in \mathbb{Z}^{(2n-k)\times(n-k)}$ where

$$A = \begin{pmatrix} \blacktriangle H & m\mathrm{Id}_{n-k} \end{pmatrix} \in \mathbb{Z}^{(2n-k)\times(n-k)}$$

and Id_{n-k} denotes the identity matrix of size $n-k$. By Remark 3.4 this ideal is also the ideal related the $\mathbb{Z}_m$-kernel of the matrix $H \in \mathbb{Z}_m^{(n-k)\times n}$. Therefore, by applying Theorem 3.9 we conclude the proof. □

Following the above construction for the semigroup S, note that $S \subseteq \mathbb{Z}_m^{n-k}$. Furthermore, by Lemma 5.1, we may deduce that S is not combinatorially finite. Moreover, we can drawn further conclusions for the associated commutative group of S, denoted by $G(S)$. First note that $G(S)$ is a torsion group since $m\mathbf{a} \equiv \mathbf{0} \mod m$ for all elements $\mathbf{a} \in S$. Moreover $S = G(S)$, or equivalently $S = -S$. In fact, this property is true for any cancellative semigroup for which $G(S)$ is a torsion group.

Proposition 5.3. *Let S be the semigroup associated to an $[n,k]$-modular code $\mathscr{C}$ over $\mathbb{Z}_m$ defined as above. Then*

$$S = G(S) \subseteq \mathbb{Z}_m^{n-k},$$

where $G(S)$ denotes the associated commutative group of S.

Proof. By definition, it is clear that $S \subseteq G(S)$. To show the converse consider any element $\mathbf{m} \in G(S)$, then $\mathbf{m}$ can be written as

$$\mathbf{m} = \sum_{i=1}^{n} a_i \mathbf{h}_i \text{ with } \mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n.$$

We distinguish two cases:

- If $a_i \geq 0$ for all $i = 1, \ldots, n$ then $\mathbf{m} \in S$.
- Otherwise, there must exists an index $i \in \{1, \ldots, n\}$ such that $a_i < 0$. Replacing a_i by $\hat{a}_i := a_i + K(m-1) \geq 0$ with $K \in \mathbb{N}$ we can rewrite $\mathbf{m}$ as

$$\mathbf{m} \equiv \sum_{\substack{j=1 \\ j \neq i}} a_j \mathbf{h}_j + \hat{a}_i \mathbf{h}_i \mod m.$$

  By repeating the previous substitution on all possible indexes $j \in \{1, \ldots, n\}$ such that $a_j < 0$ we conclude that $\mathbf{m} \in S$.

□

Remark 5.4. Let $\mathscr{C}$ be a modular code of parameters $[n,k]$ over $\mathbb{Z}_m$ and let $H \in \mathbb{Z}_m^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. The proof of Proposition 5.2 allows us to deduce that the lattice $\mathscr{L}_1$ described as

$$\mathscr{L}_1 = \left\{ \mathbf{u} \in \mathbb{Z}^n \mid \sum_{i=1}^{n} u_i \mathbf{h}_i \equiv 0 \mod m \right\}$$

where $\mathbf{h}_i \in \mathbb{Z}_m^{n-k}$ denotes the i-th column of H is equal to the set $\blacktriangle\mathscr{C} + (m\mathbb{Z}^n)$. Thus we have the following exact sequence of abelian groups:

$$0 \longrightarrow \mathscr{L}_1 \longrightarrow G(\mathbb{N}^n) = \mathbb{Z}^n \longrightarrow G(S) = S \longrightarrow 0$$

and one has $I_m(\mathscr{C}) = I_{\mathscr{L}_1}$.

5.2.1 Another representation for modular codes

In the previous section we have described the semigroup associated to the lattice ideal of a modular code $\mathscr{C}$. Although this lattice ideal provides the set of codewords of minimal support of $\mathscr{C}$, see for instance Theorem 3.20, it does not allow complete decoding, we refer the reader to the Example 4.17. This is why we define on Chapter 4 another ideal $I_+(\mathscr{C})$ associated to $\mathscr{C}$.

For this purpose we need to define new characteristic crossing functions:

$$\overline{\nabla}: \{0,1\}^{m-1} \longrightarrow \mathbb{Z}_m \quad \text{and} \quad \underline{\Delta}: \mathbb{Z}_m \longrightarrow \{0,1\}^{m-1} .$$

Let $\{\mathbf{e}_1, \ldots, \mathbf{e}_{m-1}\}$ be the canonical basis of $\mathbb{Z}^{m-1}$. Here the map $\underline{\Delta}$ replace the element $j \in \mathbb{Z}_m \setminus \{0\}$ by the unit vector $\mathbf{e}_j \in \mathbb{Z}^{m-1}$ and 0 by the zero vector $\mathbf{0} \in \mathbb{Z}^{m-1}$. Whereas the map $\overline{\nabla}$ recovers the element $j_1 + 2j_2 + \ldots + (m-1)j_{m-1}$ from the binary vector $(j_1, \ldots, j_{m-1})$.

Now let $\mathbf{X}$ denotes n vector variables $X_1, \ldots, X_n$ such that each variable X_i can be decomposed into $m-1$ components $x_{i1}, \ldots, x_{im-1}$ with $i = 1, \ldots, n$ representing the nonzero elements of $\mathbb{Z}_m$. Let $\mathbf{a} \in \mathbb{Z}_m^n$, we adopt the following notation:

$$\mathbf{X}^{\mathbf{a}} = X_1^{a_1} \cdots X_n^{a_n} = \left(x_{11} \cdots x_{1m-1}\right)^{\underline{\Delta} a_1} \cdots \left(x_{n1} \cdots x_{nm-1}\right)^{\underline{\Delta} a_n} .$$

This relationship allows us to work with monomial of type $\mathbf{X}^{\mathbf{a}}$ with $\mathbf{a} \in \mathbb{Z}_m^n$ as monomial with binary exponents. In this case, note that the degree of the monomial $\mathbf{X}^{\mathbf{a}}$ is defined as the Hamming weight of the vector $\mathbf{a} \in \mathbb{Z}_m^n$.

By Theorem 3.9, given the rows of a generator matrix $G \in \mathbb{Z}_m^{k\times n}$ of $\mathscr{C}$, labelled by $\mathbf{w}_1, \ldots, \mathbf{w}_k$, we define the following binomial ideal associated to $\mathscr{C}$:

$$I_+(\mathscr{C}) = \left\langle \{\mathbf{X}^{\mathbf{w}_i} - 1\}_{i=1,\ldots,k} \bigcup \left\{\mathscr{R}_{X_i}(T_+)\right\}_{i=1,\ldots,n} \right\rangle$$

where $\mathscr{R}_{X_i}(T_+)$ consists of all binomials on the vector variable X_i associated to the relations given by the additive table of the ring $\mathbb{Z}_m$ i.e.

$$\mathscr{R}_{X_i}(T_+) = \left\{ \begin{array}{c} \{x_{iu}x_{iv} - x_{iw} \mid u+v \equiv w \mod m\} \\ \{x_{iu}x_{iv} - 1 \mid u+v \equiv 0 \mod m\} \end{array} \right\} \text{ with } i = 1,\ldots,n.$$

Let $H \in \mathbb{Z}_m^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. The above ideal can be viewed also as the semigroup ideal associated to the commutative semigroup S finitely generated by $n \times (m-1)$ elements:

$$\{\mathbf{n}_{11},\ldots,\mathbf{n}_{1m-1},\ldots,\mathbf{n}_{n1},\ldots,\mathbf{n}_{nm-1}\}$$

with $\mathbf{n}_{ij} = j\mathbf{h}_i$ where $\mathbf{h}_i \in \mathbb{Z}_m^{n-k}$ denotes the i-th column H.

Remark 5.5. Let $H \in \mathbb{Z}_m^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$ whose columns are labelled by $\{\mathbf{h}_1,\ldots,\mathbf{h}_n\}$. We define two different set of elements of $\mathbb{Z}_m^{n-k}$:

- F_1 is a set of n elements given by $\{\mathbf{h}_1,\ldots,\mathbf{h}_n\}$. This set was considered as generating set of S in Section 5.2 since its ideal semigroup match the lattice ideal $I_{\mathscr{L}_1} = I_m(\mathscr{C})$.
- F_2 has cardinality $n \times (m-1)$ and is defined by $\{j\mathbf{h}_i\}_{\substack{i=1,\ldots,n \\ j=1,\ldots,m-1}}$. Next result shows that $I_{F_2}(S) = I_+(\mathscr{C})$, where $I_+(\mathscr{C})$ was first described in Section 4.6.1.

It follows immediate that both F_1 and F_2 generates the same semigroup S but they provide two different semigroup ideals.

Proposition 5.6. *Let $\mathscr{C}$ be a modular code of length n and* rank *k defined over $\mathbb{Z}_m$ and let $H \in \mathbb{Z}_m^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. Let S be the commutative semigroup finitely generated by $F_2 = \{j\mathbf{h}_i\}_{\substack{i=1,\ldots,n \\ j=1,\ldots,m-1}}$ where $\mathbf{h}_i$ denotes the i-th column of H. Then $I_{F_2}(S) = I_+(\mathscr{C})$.*

Proof. We begin by proving that all the binomials of the generating set of $I_+(\mathscr{C})$ belongs to $I_{F_2}(S)$. Indeed, we distinguish two types of binomials:

- $\mathbf{X}^{\mathbf{w}} - 1$ where $\mathbf{w} = (w_1,\ldots,w_n) \in \mathbb{Z}_m^n$ denotes a row of a generator matrix $G \in \mathbb{Z}_m^{k\times n}$ of $\mathscr{C}$. Note that

$$\underline{\triangle}\mathbf{w} \cdot \left(\begin{array}{ccc} \mathbf{n}_{11},\ldots,\mathbf{n}_{1m-1} & , \cdots , & \mathbf{n}_{n1},\ldots,\mathbf{n}_{nm-1} \end{array} \right)^T = \sum_{i=1}^{n}\sum_{j=1}^{m-1} w_{ij}\mathbf{n}_{ij}$$

  where if $w_i = s_i \in \mathbb{Z}_m$, then $w_{ij} = \left\{ \begin{array}{ll} 0 & \text{, if } j \neq s_i \\ 1 & \text{, otherwise.} \end{array} \right.$

  Thus,

$$\sum_{i=1}^{n}\sum_{j=1}^{m-1} w_{ij}\mathbf{n}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{m-1} w_{ij}j\mathbf{h}_i = \sum_{i=1}^{n} s_i\mathbf{h}_i = \mathbf{w}H^T = 0,$$

  since $GH^T = 0$ in $\mathbb{F}_q$, and so $\mathbf{X}^{\mathbf{w}} - 1 \in I_{F_2}(S)$.

- $x_{iu}x_{iv} - x_{iw} \in \mathscr{R}_{X_i}\left(T_+\right)$ which comes from the additive rule $u+v \equiv w \mod m$. Therefore $u\mathbf{h}_i + v\mathbf{h}_i \equiv w\mathbf{h}_i \mod m$. Or equivalently $\mathbf{n}_{iu} + \mathbf{n}_{iv} - \mathbf{n}_{iw} \equiv 0 \mod m$ and so $x_{iu}x_{iv} - x_{iw} \in I_{F_2}(S)$.
- $x_{iu}x_{iv} - 1 \in \mathscr{R}_{X_i}\left(T_+\right)$ which implies that $u+v \equiv 0 \mod m$. Thus, $u\mathbf{h}_i + v\mathbf{h}_i \equiv 0 \mod m$ and so $x_{iu}x_{iv} - 1 \in I_{F_2}(S)$.

To show the converse it suffice to make the following observation:

$$\begin{aligned}\mathbf{X}^{\underline{\Delta}\mathbf{a}} - \mathbf{X}^{\underline{\Delta}\mathbf{b}} \in I_{F_2}(S) \iff & \sum_{i=1}\sum_{j=1} a_{ij}\mathbf{n}_{ij} \equiv \sum_{i=1}\sum_{j=1} b_{ij}\mathbf{n}_{ij} \mod m \text{ with } \mathbf{a},\mathbf{b} \in \mathbb{N}^n \\ \iff & \sum_{i=1}\sum_{j=1} \left(a_{ij} - b_{ij}\right) j\mathbf{h}_i \equiv 0 \mod m \text{ with } \mathbf{a},\mathbf{b} \in \mathbb{N}^n\end{aligned}$$

$$\iff \left(\sum_{i=1}\sum_{j=1}\left(a_{1j} - b_{1j}\right) j \quad ,\ldots, \quad \sum_{i=1}\sum_{j=1}\left(a_{nj} - b_{nj}\right) j \right) \in \mathscr{C}$$

In other words $(\mathbf{a} - \mathbf{b} \mod m) \in \mathscr{C}$. Thus, $\mathbf{X}^{\underline{\Delta}\mathbf{a}} - \mathbf{X}^{\underline{\Delta}\mathbf{b}} \in I(\mathscr{C})$. Thus Theorem 3.9 gives that $\mathbf{X}^{\underline{\Delta}\mathbf{a}} - \mathbf{X}^{\underline{\Delta}\mathbf{b}} \in I_+(\mathscr{C})$ which completes the proof. □

Remark 5.7. Let $\mathscr{C}$ be a modular code of parameters $[n,k]$ over $\mathbb{Z}_m$ and let $H \in \mathbb{Z}_m^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. The proof of Proposition 5.6 allows us to deduce that the lattice $\mathscr{L}_2$ described as

$$\mathscr{L}_2 = \left\{ \mathbf{u} \in \mathbb{Z}^{n(m-1)} \mid \sum_{i=1}^{n}\sum_{j=1}^{m-1} u_{ij} j\mathbf{h}_i \equiv 0 \mod m \right\}$$

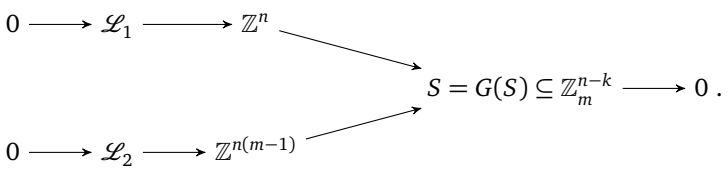
where $\mathbf{h}_i \in \mathbb{Z}_m^{n-k}$ denotes the i-th column of H is equal to the set $\underline{\Delta}\mathscr{C} + (m\mathbb{Z}^{n(m-1)})$. Thus we have the following exact sequence of abelian groups:

$$0 \longrightarrow \mathscr{L}_2 \longrightarrow G\left(\mathbb{N}^{n(m-1)}\right) = \mathbb{Z}^{n(m-1)} \longrightarrow G(S) = S \longrightarrow 0.$$

As above, let $\mathscr{L}_i$ denotes the lattice related to the semigroup ideal $I_{F_i}(S)$ with $i = 1, 2$ where

$$F_1 = \left\{\mathbf{h}_i\right\}_{i=1,\ldots,n} \quad \text{and} \quad F_2 = \left\{j\mathbf{h}_i\right\}_{\substack{i=1,\ldots,n \\ j=1,\ldots,m-1}}.$$

These lattices were described in detail in Remarks 5.4 and 5.7. We have the following exact sequences:



5.2.2 Identify equivalent representations

An elementary row operation on a matrix of type $A \in \mathbb{Z}_m^{r\times s}$ is a sequence of three types of row operations: interchanging two rows, multiplying a row with a unit in $\mathbb{Z}_m$ or adding one row to another row.

Proposition 5.8. *Let H_1 and H_2 be two $(n-k)\times n$ parity check matrices of the modular codes $\mathscr{C}_1$ and $\mathscr{C}_2$ over $\mathbb{Z}_m$. Then the following statements are equivalent:*

$$\mathscr{C}_1 = \mathscr{C}_2 \iff \text{There is an invertible matrix } M \in \mathbb{Z}_m^{(n-k)\times(n-k)} \,:\, H_2 = MH_1.$$

Proof. Assume that $\mathscr{C}_1 = \mathscr{C}_2$, then the row space of H_1 and H_2 are the same, that is H_1 and H_2 are row equivalent. Let $M \in \mathbb{Z}_m^{(n-k)\times(n-k)}$ be the matrix which represents the set of elementary row operations made over H_1 to obtain H_2. Thus, $H_2 = MH_1$.

Conversely, let $H_2 = MH_1$ where $M \in \mathbb{Z}_m^{(n-k)\times(n-k)}$ is an invertible matrix over $\mathbb{Z}_m$. Then, we check at once that each codeword $\mathbf{c}_2 \in \mathscr{C}_2$ verifies that

$$H_2\mathbf{c}_2^T \equiv \mathbf{0} \mod m \Rightarrow MH_1\mathbf{c}_2^T \equiv \mathbf{0} \mod m \Rightarrow H_1\mathbf{c}_2^T \equiv 0 \mod m.$$

Therefore $\mathbf{c}_2 \in \mathscr{C}_1$. Similar arguments apply to the case $\mathscr{C}_1 \subseteq \mathscr{C}_2$, and the proof is complete. □

Let S be the semigroup associated to an $[n,k]$-modular code $\mathscr{C}$ over the ring $\mathbb{Z}_m$ and F be the chosen generating set of S. Consider the matrix $H \in \mathbb{Z}_m^{(n-k)\times n}$ which is the parity check matrix of $\mathscr{C}$ related with F, that is,

$$F = \{\mathbf{h}_i\}_{i=1,\ldots,n} \quad \text{or} \quad F = \{j\mathbf{h}_i\}_{\substack{i=1,\ldots,n\\ j=1,\ldots,n}}$$

where $\mathbf{h}_i$ denotes the i-th column of H.

Proposition 5.8 implies that performing elementary row operations on H yields to a new set of generators $\hat{F}$ of S. Or equivalently, if we consider a new set of generators $\hat{F}$ defined by a matrix $\hat{H}$ such that $\hat{H} = MH$ where M is an invertible matrix $M \in \mathbb{Z}_m^{(n-k)\times(n-k)}$. Then $\hat{F}$ is also a generating set of S.

However, any permutation of the order of the set F or multiplying any element of F by a unit in $\mathbb{Z}_m$ gives the same semigroup S_2 associated with another modular code $\mathscr{C}_2$ which is equivalent to $\mathscr{C}$ thus, both codes have the same parameters.

5.3 The semigroup associated with a linear code

Let α be a primitive element of $\mathbb{F}_q$ and $\{\mathbf{e}_1,\ldots,\mathbf{e}_{q-1}\}$ be the canonical basis of $\mathbb{Z}^{q-1}$. We will use throughout this section the following characteristic crossing functions:

$$\nabla: \quad \{0,1\}^{q-1} \longrightarrow \mathbb{F}_q \qquad \text{and} \qquad \Delta: \quad \mathbb{F}_q \longrightarrow \{0,1\}^{q-1}$$

The map Δ replaces the class of the elements $\mathbf{a} = \alpha^j \in \mathbb{F}_q^*$ by the vector $\mathbf{e}_j$ and $0 \in \mathbb{F}_q$ by the zero vector $\mathbf{0} \in \mathbb{Z}^{q-1}$. Whereas the map ∇ recovers the element

$$j_1\alpha + \ldots j_{q-2}\alpha^{q-2} + j_{q-1} \in \mathbb{F}_q$$

from the $(q-1)$-tuple of binary elements $(j_1,\ldots,j_{q-1})$.

Let $\mathbf{X}$ denotes n vector variables $X_1,\ldots,X_n$ such that each variable X_i can be decomposed into $q-1$ components $x_{i1},\ldots,x_{iq-1}$ with $i=1,\ldots,n$. Let $\mathbf{a}\in\mathbb{F}_q^n$, we will adopt the following notation:

$$\mathbf{X}^{\mathbf{a}}=X_1^{a_1}\cdots X_n^{a_n}=\left(x_{11}\cdots x_{1q-1}\right)^{\Delta a_1}\cdots\left(x_{n1}\cdots x_{nq-1}\right)^{\Delta a_n}.$$

This relationship allows us to work with monomial whose exponents are form by elements defined over the field $\mathbb{F}_q$ as monomials with integer exponents.

On this section $\mathscr{C}$ will be an $[n,k]$ linear code defined over a finite field $\mathbb{F}_q$. Given the rows of a generator matrix $G\in\mathbb{F}_q^{k\times n}$ of $\mathscr{C}$, labelled by $\mathbf{w}_1,\ldots,\mathbf{w}_k$, we define the ideal associated with $\mathscr{C}$ as the binomial ideal generated by

$$I_+(\mathscr{C})=\left\langle\ \left\{\mathbf{X}^{\alpha^j\mathbf{w}_i}-1\right\}_{\substack{i=1,\ldots,k\\ j=1,\ldots,q-1}}\ \bigcup\ \left\{\mathscr{R}_{X_i}\left(T_+\right)\right\}_{i=1,\ldots,n}\ \right\rangle\subseteq\mathbb{K}[\mathbf{X}]$$

where $\mathscr{R}_{X_i}\left(T_+\right)$ stands for all the binomials on the variable X_i associated to the relations given by the additive table of the field $\mathbb{F}_q$ (see Section 4.1 for more details), i.e.

$$\mathscr{R}_{X_i}\left(T_+\right)=\left\{\ \left\{x_{iu}x_{iv}-x_{iw}\mid\alpha^u+\alpha^v=\alpha^w\right\}\ \bigcup\ \left\{x_{iu}x_{iv}-1\mid\alpha^u+\alpha^v=0\right\}\ \right\}$$

Let $H\in\mathbb{F}_q^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$ whose columns are labelled by $\mathbf{h}_1,\ldots,\mathbf{h}_n$. The ideal $I_+(\mathscr{C})$ is also a semigroup ideal associated to the commutative semigroup S finitely generated by $n\times(q-1)$ elements defined as

$$\left\{\mathbf{n}_{11},\ldots,\mathbf{n}_{1q-1},\ldots,\mathbf{n}_{n1},\ldots,\mathbf{n}_{nq-1}\right\}\ \text{with}\ \mathbf{n}_{ij}=\alpha^j\mathbf{h}_i\in\mathbb{F}_q^{n-k}.$$

Proposition 5.9. *Let $\mathscr{C}$ be an $[n,k]$-linear code over $\mathbb{F}_q$ and $H\in\mathbb{F}_q^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. Let S be the commutative semigroup finitely generated by $\left\{\alpha^j\mathbf{h}_i\right\}_{\substack{i=1,\ldots,n\\ j=1,\ldots,q-1}}$ where $\mathbf{h}_i$ denotes the i-th column of a H and α is any primitive element of $\mathbb{F}_q$. Then $I(S)=I_+(\mathscr{C})$.*

Proof. It is easy to check that $I_+(\mathscr{C})\subseteq I(S)$ since all binomials in the generating set of $I_+(\mathscr{C})$ belong to $I(S)$. To prove this, take any element of the generating of $I_+(\mathscr{C})$ of the form:

- $\mathbf{X}^{\alpha^s\mathbf{w}}-1$ with $s=1,\ldots,q-1$ where $\mathbf{w}=(w_1,\ldots,w_n)\in\mathbb{F}_q^n$ denotes a row of a generator matrix $G\in\mathbb{F}_q^{k\times n}$ of $\mathscr{C}$. Note that

$$\alpha^s\Delta\mathbf{w}\left(\ \mathbf{n}_{11},\ldots,\mathbf{n}_{1q-1}\quad,\ldots,\quad\mathbf{n}_{n1},\ldots,\mathbf{n}_{nq-1}\ \right)^T=\alpha^s\sum_{i=1}^{n}\sum_{j=1}^{q-1}w_{ij}\mathbf{n}_{ij}$$

  where if $w_i=\alpha^{r_i}\in\mathbb{F}_q$ then $w_{ij}=\begin{cases}0 & ,\text{ if } j\neq r_i\\ 1 & ,\text{ otherwise.}\end{cases}$

Thus

$$\alpha^s \sum_{i=1}^{n}\sum_{j=1}^{q-1} w_{ij}\mathbf{n}_{ij} = \alpha^s \sum_{i=1}^{n}\sum_{j=1}^{q-1} w_{ij}\alpha^j \mathbf{h}_i = \alpha^s \sum_{i=1}^{n} \alpha^{r_i}\mathbf{h}_i = \mathbf{w}H^T = 0,$$

since $GH^T = 0$ in $\mathbb{F}_q$, and so $\mathbf{X}^{\alpha^s\mathbf{w}} - 1 \in I(S)$.

- $x_{iu}x_{iv} - x_{iw} \in \mathscr{R}_{X_i}\left(T_+\right)$ which comes from the additive rule $\alpha^u + \alpha^v = \alpha^w$ in $\mathbb{F}_q$. Therefore $\alpha^u\mathbf{h}_i + \alpha^v\mathbf{h}_i - \alpha^w\mathbf{h}_i = 0$, or equivalently, $\mathbf{n}_{iu} + \mathbf{n}_{iv} - \mathbf{n}_{iw} = 0$, and consequently $x_{iu}x_{iv} - x_{iw} \in I(S)$.
- $x_{iu}x_{iv} - 1 \in \mathscr{R}_{X_i}\left(T_+\right)$ which implies that $\alpha^u + \alpha^v = 0$ in $\mathbb{F}_q$. This gives $\mathbf{n}_{iu} + \mathbf{n}_{iv} = \mathbf{0}$ and hence $x_{iu}x_{iv} - 1 \in I(S)$.

To show the converse, it suffice to make the following observation:

$$\begin{aligned}
\mathbf{X}^{\Delta\mathbf{a}} - \mathbf{X}^{\Delta\mathbf{b}} \in I(\mathscr{C}) \quad &\Longleftrightarrow \quad \sum_{i=1}^{n}\sum_{j=1}^{q-1} a_{ij}\mathbf{n}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{q-1} b_{ij}\mathbf{n}_{ij} \text{ in } \mathbb{F}_q \text{ with } \mathbf{a},\mathbf{b} \in \mathbb{N}^n \\
&\Longleftrightarrow \quad \sum_{i=1}^{n}\sum_{j=1}^{q-1} \left(a_{ij} - b_{ij}\right)\alpha^j \mathbf{h}_i = 0 \text{ in } \mathbb{F}_q \text{ with } \mathbf{a},\mathbf{b} \in \mathbb{N}^n \\
&\Longleftrightarrow \quad \left(\sum_{i=1}^{q-1}\left(a_{1j} - b_{1j}\right)\alpha^j \quad ,\ldots, \quad \sum_{i=1}^{q-1}\left(a_{nj} - b_{nj}\right)\alpha^j \ \right) \in \mathscr{C}
\end{aligned}$$

In other words $\left(\mathbf{a} - \mathbf{b} \text{ in } \mathbb{F}_q\right) \in \mathscr{C}$. Thus, $\mathbf{X}^{\Delta\mathbf{a}} - \mathbf{X}^{\Delta\mathbf{b}} \in I(\mathscr{C}) = I_+(\mathscr{C})$. Recall that the last equality is due to Theorem 4.3. □

Let $q = p^s$ with p prime. Following the above construction for the semigroup S, note that $S \subseteq \mathbb{F}_q^{n-k}$. Again, by Lemma 5.1, we may deduce that S is not combinatorially finite. Moreover S is cancellative and $G(S)$, which denotes the associated commutative group of S, is a torsion group since $p\mathbf{a} = \mathbf{0}$ in $\mathbb{F}_q$ for all elements $\mathbf{a} \in S$. This implies $S = G(S)$ but we will also show that $G(S) = \mathbb{F}_q^{n-k}$.

Proposition 5.10. *Let S be the semigroup associated to an $[n,k]$-linear code $\mathscr{C}$ over $\mathbb{F}_q$ defined as above. Then*

$$S = G(S) = \mathbb{F}_q^{n-k},$$

where $G(S)$ denotes the associated commutative group of S.

Proof. By construction, it is clear that $S = G(S)$. Moreover, since every parity check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ of $\mathscr{C}$ has rank $n-k$, then $n-k$ columns of its reduced echelon form $R = \operatorname{rref}(H)$ form the identity matrix Id_{n-k}. Let us labelled such columns by $\{\mathbf{r}_1, \ldots, \mathbf{r}_{n-k}\}$.

Note that if H_1, H_2 are two parity check matrices of $\mathscr{C}$ then its columns generates the same semigroup. Moreover every element $\mathbf{v} \in \mathbb{F}_q^{n-k}$ can be rewritten as

$$\mathbf{v} = v_1\mathbf{r}_1 + \ldots + v_{n-k}\mathbf{r}_{n-k} = \alpha^{i_1}\mathbf{r}_1 + \ldots + \alpha^{i_{n-k}}\mathbf{r}_{n-k} \in G(S),$$

which gives our claim. □

Remark 5.11. Let $\mathscr{C}$ be an $[n,k]$-linear code over $\mathbb{F}_q$ and let $H \in \mathbb{F}_q^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. The proof of Proposition 5.9 allows us to deduce that the lattice $\mathscr{L}_2$ described as

$$\mathscr{L}_2 = \left\{ \mathbf{u} \in \mathbb{Z}^{n(q-1)} \mid \sum_{i=1}^{n} \sum_{j=1}^{q-1} u_{ij} \alpha^j \mathbf{h}_i = 0 \text{ in } \mathbb{F}_q \right\}$$

where $\mathbf{h}_i \in \mathbb{F}_q^{n-k}$ denotes the i-th column of H and α is a primitive element of $\mathbb{F}_q$, is equal to the set $\Delta\mathscr{C} + \left(p\mathbb{Z}^{n(q-1)}\right)$. Thus we have the following exact sequence of abelian groups:

$$0 \longrightarrow \mathscr{L}_2 \longrightarrow G\left(\mathbb{N}^{n(q-1)}\right) = \mathbb{Z}^{n(q-1)} \longrightarrow G(S) = S = \mathbb{F}_q^{n-k} \longrightarrow 0 \ .$$

5.3.1 Another representation for linear codes

Let $\mathbb{F}_q$ be a finite field then q is a prime power, i.e. $q = p^s$ where p is a prime. Let $f(X)$ be any irreducible polynomial of degree s over $\mathbb{F}_p$ and β be a root of $f(X)$. Then, every element of $\mathbb{F}_q$ can be uniquely represented in the form

$$a_0 + a_1\beta + \ldots + a_{s-1}\beta^{s-1} \text{ with } a_0, \ldots, a_{s-1} \in \mathbb{F}_q.$$

That is, we can express $\mathbb{F}_q$ in the form $\mathbb{F}_p[\beta]$.

Note that every primitive element α of $\mathbb{F}_q$ can serve as a defining element of $\mathbb{F}_p$ over $\mathbb{F}_q$, i.e. $\mathbb{F}_p[\alpha] = \mathbb{F}_q$. However, it is not in fact necessary for β to be a multiplicative generator of $\mathbb{F}_q^*$.

In this way we could define new characteristic crossing functions:

$$\overline{\blacktriangledown}: \ \mathbb{Z}^s \longrightarrow \mathbb{F}_q \qquad \text{and} \qquad \underline{\blacktriangle}: \ \mathbb{F}_q \longrightarrow \mathbb{Z}^s$$

Here the map $\underline{\blacktriangle}$ replaces the class of the elements $\mathbf{a} = a_0 + a_1\beta + \ldots + a_{s-1}\beta^{s-1} \in \mathbb{F}_q$ with $(a_0, a_1, \ldots, a_{s-1}) \in \mathbb{F}_p^s$ by the vector $\blacktriangle(a_0, \ldots, a_{s-1}) \in \mathbb{Z}^s$. While the map $\overline{\blacktriangledown}$ recovers the element $\blacktriangledown a_0 + \blacktriangledown a_1\beta + \ldots + \blacktriangledown a_{s-1}\beta^{s-1}$ from the integer vector $(a_0, a_1, \ldots, a_{s-1})$.

Note that the map $\blacktriangledown$ means reduction modulo p while the map $\blacktriangle$ replace the class of $0, 1, \ldots, p-1$ by the same symbols regarded as integers. These maps where defined on Section 5.2.

Let $\mathbf{Y}$ denotes n vector variables $Y_1, \ldots, Y_n$ such that each variable Y_i is decomposed into s components $y_{i1}, \ldots, y_{is}$. Let $\mathbf{b} \in \mathbb{F}_q^n$, we adopt the following notation:

$$\mathbf{Y}^{\mathbf{b}} = Y_1^{b_1} \cdots Y_n^{b_n} = (y_{11} \cdots y_{1s})^{\underline{\blacktriangle} b_1} \cdots (y_{n1} \cdots y_{ns})^{\underline{\blacktriangle} b_n}.$$

Therefore, again we identify monomials whose exponents belongs to $\mathbb{F}_q^n$ with the usual terms of $\mathbb{K}[\mathbf{Y}]$.

Let $G \in \mathbb{F}_q^{k\times n}$ be a generator matrix of $\mathscr{C}$, where its rows are tagged as $\{\mathbf{w}_1,\ldots,\mathbf{w}_k\} \subseteq \mathbb{F}_q^n$. The following binomial ideal is another ideal associated to $\mathscr{C}$.

$$I_m(\mathscr{C}) = \left\langle \ \{\mathbf{Y}^{\mathbf{w}_i} - 1\}_{i=1,\ldots,k} \quad \bigcup \quad \left\{y_{ij}^p - 1\right\}_{\substack{i=1,\ldots,n \\ j=1,\ldots,s}} \ \right\rangle \subseteq \mathbb{K}[\mathbf{Y}].$$

Note that this ideal is in certain sense equivalent to the binomial ideal for modular code defined on Section 5.2 denoted also by $I_m(\mathscr{C})$.

Let $H \in \mathbb{F}_q^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$ whose columns are labelled by $\{\mathbf{h}_1,\ldots,\mathbf{h}_n\} \subseteq \mathbb{F}_q^{n-k}$. The above ideal can be viewed also as a semigroup ideal associated to the commutative semigroup S finitely generated by $n \times m$ elements

$$\{\mathbf{n}_{11},\ldots,\mathbf{n}_{1s},\ldots,\mathbf{n}_{n1},\ldots,\mathbf{n}_{ns}\} \text{ with } \mathbf{n}_{ij} = \beta^{j-1}\mathbf{h}_i.$$

Remark 5.12. Let $H \in \mathbb{F}_q^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$ whose columns are labelled by $\{\mathbf{h}_1,\ldots,\mathbf{h}_n\}$. We have define two generating sets of S.

- Let α be a primitive element of $\mathbb{F}_q$. We define the set F_2 as a set of $n \times (q-1)$ elements given by
$$\left\{\mathbf{n}_{ij}\right\}_{\substack{i=1,\ldots,n \\ j=1,\ldots,q-1}} \text{ with } \mathbf{n}_{ij} = \alpha^j\mathbf{h}_i.$$
This set was considered as generating set of S in Section 5.3 since its ideal semigroup coincide with the ideal $I_+(\mathscr{C})$.

- Let β be a root of any irreducible polynomial of degree s over $\mathbb{F}_p$. We define the set F_1 with cardinality $n \times s$ defined by the elements
$$\left\{\mathbf{n}_{ij}\right\}_{\substack{i=1,\ldots,n \\ j=1,\ldots,s}} \text{ with } \mathbf{n}_{ij} = \beta^{j-1}\mathbf{h}_i.$$
We will see that $I_{F_1}(S) = I_m(\mathscr{C})$.

They are easily seen to generate the same semigroup S. However they provide two different semigroups ideals.

Proposition 5.13. *Let $\mathscr{C}$ be an $[n,k]$-linear code over $\mathbb{F}_q$ with $q = p^s$ and let $H \in \mathbb{F}_q^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. Let S be the commutative semigroup finitely generated by $F_1 = \left\{\beta^{j-1}\mathbf{h}_i\right\}_{\substack{i=1,\ldots,n \\ j=1,\ldots,s}}$ where $\mathbf{h}_i$ denotes the i-th column H and β is any root of an irreducible polynomial of degree s over $\mathbb{F}_p$. Then $I_{F_1}(S) = I_m(\mathscr{C})$.*

Proof. The following result may be proved in the same way as Proposition 4.55. We claim that all the binomials of the generating set of $I_m(\mathscr{C})$ belongs to $I_{F_1}(S)$. Indeed, we distinguish two types of binomials:

- $\mathbf{Y}^{\mathbf{w}} - 1$ where $\mathbf{w} = (w_1,\ldots,w_n) \in \mathbb{F}_q$ denotes a row of any generator matrix $G \in \mathbb{F}_q^{k\times n}$ of $\mathscr{C}$. Note that each component $w_i \in \mathbb{F}_q$ can be rewritten as:
$$w_i = a_{i0} + a_{i1}\beta + \ldots + a_{is-1}\beta^{s-1}.$$

Hence,

$$\underline{\blacktriangle}\mathbf{w}\left(\begin{array}{ccc}\mathbf{n}_{11},\ldots,\mathbf{n}_{1s} & \ldots, & \mathbf{n}_{n1},\ldots,\mathbf{n}_{ns}\end{array}\right)^T = \sum_{i=1}^{n}\left(a_{i0}\mathbf{n}_{i1}+\ldots+a_{is-1}\mathbf{n}_{is}\right)$$
$$= \sum_{i=1}^{n}\left(a_{i0}+a_{i1}\beta+\ldots+a_{is-1}\beta^{s-1}\right)\mathbf{h}_i = \mathbf{w}H^T = \mathbf{0}$$

Since $GH^T = \mathbf{0}$ in $\mathbb{F}_q$, and so $\mathbf{Y}^{\mathbf{w}} - 1 \in I_{F_1}(S)$.

- $y_{ij}^p - 1$ with $i = 1,\ldots,n$ and $j = 1,\ldots,s$. Note that $p\beta^{j-1}\mathbf{h}_i = \mathbf{0}$ in $\mathbb{F}_q$, since $\mathrm{char}(\mathbb{F}_q) = p$. Thus $y_{ij}^p - 1 \in I_{F_1}(S)$.

To show the converse it suffice to make the following observation:

$$\begin{aligned}\mathbf{Y}^{\underline{\blacktriangle}\mathbf{a}} - \mathbf{Y}^{\underline{\blacktriangle}\mathbf{b}} \in I_{F_1}(S) &\iff \sum_{i=1}^{n}\sum_{j=1}^{s} a_{ij}\mathbf{n}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{s} b_{ij}\mathbf{n}_{ij} \text{ with } \mathbf{a},\mathbf{b}\in\mathbb{N}^n \\ &\iff \sum_{i=1}^{n}\sum_{j=1}^{s}\left(a_{ij}-b_{ij}\right)\mathbf{n}_{ij} = \mathbf{0} \text{ with } \mathbf{a},\mathbf{b}\in\mathbb{N}^n \\ &\iff \left(\sum_{i=1}^{s}(a_{1i}-b_{1i})\beta^{i-1},\ldots,\sum_{i=1}^{s}(a_{ni}-b_{ni})\beta^{i-1},\right)\in\mathscr{C}\end{aligned}$$

Thus $\mathbf{a}-\mathbf{b}\in\mathscr{C}$ over $\mathbb{F}_q$, or equivalently, $\mathbf{Y}^{\underline{\blacktriangle}\mathbf{a}} - \mathbf{Y}^{\underline{\blacktriangle}\mathbf{b}} \in I_m(\mathscr{C})$. □

Remark 5.14. Let $\mathscr{C}$ be an $[n,k]$-linear code over $\mathbb{F}_q$ and let $H \in \mathbb{F}_q^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. The proof of Proposition 5.13 allows us to deduce that the lattice $\mathscr{L}_1$ described as

$$\mathscr{L}_1 = \left\{\mathbf{u}\in\mathbb{Z}^{ns} \mid \sum_{i=1}^{n}\sum_{j=1}^{s} u_{ij}\beta^j\mathbf{h}_i = 0 \text{ in } \mathbb{F}_q\right\}$$

where $\mathbf{h}_i \in \mathbb{F}_q^{n-k}$ denotes the i-th column of H and β is a root of an irreducible polynomial of degree s over $\mathbb{F}_q$, is equal to the set $\underline{\blacktriangle}\mathscr{C} + \left(p\mathbb{Z}^{ns}\right)$. Thus we have the following exact sequence of abelian groups:

$$0 \longrightarrow \mathscr{L}_1 \longrightarrow G(\mathbb{N}^{ns}) = \mathbb{Z}^{ns} \longrightarrow G(S) = S = \mathbb{F}_q^{n-k} \longrightarrow 0$$

Let $\mathscr{L}_i$ denotes the lattice related to the semigroup ideal $I_{F_i}(S)$ with $i = 1,2$ where

$$F_2 = \left\{\alpha^j\mathbf{h}_i\right\}_{\substack{i=1,\ldots,n\\ j=1,\ldots,q-1}} \quad \text{and} \quad F_1 = \left\{\beta^{j-1}\mathbf{h}_i\right\}_{\substack{i=1,\ldots,n\\ j=1,\ldots,s}}.$$

These lattices were described in detail in Remarks 5.11 and 5.14. We have the following exact sequences:

$$\begin{array}{ccccccccc} 0 & \longrightarrow & \mathscr{L}_2 & \longrightarrow & \mathbb{Z}^{n(q-1)} & & & & \\ & & & & & \searrow & & & \\ & & & & & & S = G(S) = \mathbb{F}_q^{n-k} & \longrightarrow & 0 \\ & & & & & \nearrow & & & \\ 0 & \longrightarrow & \mathscr{L}_1 & \longrightarrow & \mathbb{Z}^{ns} & & & & \end{array}$$

5.3.2 Identify equivalent representations

The following result will provide us a test to determine what happen with the linear code $\mathscr{C}$ associated to a commutative semigroup S when we make some particular modifications over the generating set of S.

Definition 5.15. A matrix is in *reduced echelon form* if it satisfies the following conditions:

1. All nonzero rows are above any row form by the zero vector.
2. Every leading coefficient (i.e. the first nonzero element of each nonzero-row) is 1 and is the only nonzero entry in its column.
3. The leading coefficient is always strictly to the right of the leading coefficient of the row above it.

Let $H \in \mathbb{F}_q^{(n-k)\times n}$ be a parity check matrix of $\mathscr{C}$. By Linear Algebra, we can transform H in a row reduced echelon, denoted by $\operatorname{rred}(H)$, form by a sequence of three elementary row operations: interchanging two rows, multiplying a row with a nonzero constant or adding one row to another row.

Note that, although a parity check matrix H of a code $\mathscr{C}$ is not unique, the matrix $\operatorname{rref}(H)$, which is also a parity check matrix of $\mathscr{C}$ is unique.

Proposition 5.16. *Let H_1 and H_1 be two $(n-k)\times n$ parity check matrices of the codes $\mathscr{C}_1$ and $\mathscr{C}_2$ over $\mathbb{F}_q$. Then the following statements are equivalent:*

1. $\mathscr{C}_1 = \mathscr{C}_2$.
2. $\operatorname{rref}(H_1) = \operatorname{rref}(H_2)$.
3. *There is an invertible matrix $M \in \mathbb{F}_q^{(n-k)\times(n-k)}$ such that $H_2 = MH_1$.*

Proof. Let us assume that $\mathscr{C}_1 = \mathscr{C}_2$, then the row space of H_1 and H_2 are the same. So H_1 and H_2 are row equivalent, i.e. $\operatorname{rref}(H_1) = \operatorname{rref}(H_2)$.

Furthermore, $\operatorname{rref}(H_1) = E_1 \cdots E_l H_1$ where $E_1, \ldots, E_l$ are the elementary matrices that corresponds to the elementary row operations made to transform H_1 on $\operatorname{rref}(H_1)$. Let $M_1 = E_1 \cdots E_l$, then $M_1 \in \mathbb{F}_q^{(n-k)\times(n-k)}$ is an invertible matrix since E_i are invertible and $\operatorname{rref}(H_1) = M_1 H_1$. Likewise $\operatorname{rref}(H_2) = M_2 H_2$ where $M_2 \in \mathbb{F}_q^{(n-k)\times(n-k)}$ is an invertible matrix.

Now assume that $\operatorname{rref}(H_1) = \operatorname{rref}(H_2)$ and let $M = M_2 M_1^{-1}$, then

$$MH_1 = M_2 M_1^{-1} M_1 \operatorname{rref}(H_1) = M_2 \operatorname{rref}(H_2) = H_2.$$

Finally suppose that $H_2 = MH_1$ for some invertible matrix $M \in \mathbb{F}_q^{(n-k)\times(n-k)}$. Then for every codeword $\mathbf{c}_2 \in \mathscr{C}_2$ we have that

$$H_2 \mathbf{c}_2^T = \mathbf{0} \Rightarrow MH_1 \mathbf{c}_2^T = \mathbf{0} \Rightarrow H_1 \mathbf{c}_2^T = \mathbf{0}.$$

So $\mathscr{C}_2$ is a subcode of $\mathscr{C}_1$. Similarly $\mathscr{C}_1 \subseteq \mathscr{C}_2$ since $H_1 = M^{-1} H_2$ which proves the desired result. □

Let S be the semigroup associated to an $[n,k]$ linear code $\mathscr{C}$ over $\mathbb{F}_q$ and F be the generating set of S. Consider the matrix $H \in \mathbb{F}_q^{(n-k)\times(n-k)}$ which is the parity check matrix of $\mathscr{C}$ defining the set F, i.e. we distinguish two cases:

$$F = \left\{\alpha^j \mathbf{h}_i\right\}_{\substack{i=1,\dots,n \\ j=1,\dots,q-1}} \quad \text{or} \quad F = \left\{\beta^{j-1}\mathbf{h}_i\right\}_{\substack{i=1,\dots,n \\ j=1,\dots,s}}$$

where $\mathbf{h}_i$ denotes the i-th column of H, α denotes a primitive element of $\mathbb{F}_q$ and β represent a root of any irreducible polynomial of degree s over $\mathbb{F}_p$ with $q = p^s$.

Proposition 5.16 implies that performing elementary row operations on H yields to a new generating set of S. That is to say, if we consider a new generating set $\hat{F}$ defined by the matrix $\hat{H}$ such that $\operatorname{rref}(H) = \operatorname{rref}(\hat{H})$. Then $\hat{F}$ is also a generating set of S.

However a permutation or a multiplication by a nonzero constant on the generating set F define the same semigroup S associated with another linear code $\mathscr{C}_2$ which is equivalent to $\mathscr{C}$, thus, both codes have the same parameters.

5.4 Some conclusions

We have found a semigroup S related to linear and modular codes but we can consider several generating sets of it which yields to different binomials semigroups ideals. Summarizing, we have two different situations:

1. If $\mathscr{C}$ is an $[n,k]$-modular code over $\mathbb{Z}_m$ and H is a parity check matrix of $\mathscr{C}$ we have analyzed two generating sets of S:

   - The set F_1 described by the elements $\{\mathbf{h}_i\}_{i=1,\dots,n}$, where $\mathbf{h}_i$ denotes the i-th column of H. Proposition 5.2 states the equivalence $I_{F_1}(S) = I_m(\mathscr{C})$ where $I_m(\mathscr{C})$ describes the binomial ideal of $\mathscr{C}$ by taking care of the arithmetic of modular integers.

     This ideal was first described in Section 3.2 where we proved that it can be also view as the binomial ideal of a modular integer program. Moreover a Graver basis of this ideal provides the set of codewords of minimal support of $\mathscr{C}$, see Theorem 3.20. However, such ideal does not allow complete decoding as Example 4.29 shows.

   - The set F_2 given by the elements $\left\{j\mathbf{h}_i\right\}_{\substack{i=1,\dots,n \\ j=1,\dots m-1}}$, where $\mathbf{h}_i$ denotes the i-th column of H. Proposition 5.6 shows that $I_{F_2}(S) = I_+(\mathscr{C})$, where $I_+(\mathscr{C})$ is the binomial ideal of $\mathscr{C}$ given by the additive rules of the ring $\mathbb{Z}_m$.

     This ideal was introduced in Section 4.6.1 where it was shown that not only it describes the set of codewords of minimal support but it also provides a complete decoding procedure for $\mathscr{C}$, see Proposition 4.60. Although the cardinality of this set is $n \times (m-1)$ which is much larger than the cardinality of F_1, in Section 4.4 we discuss an alternative for the computation of a Gröbner basis of this ideal which is better suited than the standard Buchberger's algorithm.

2. The same conclusion can be drawn for linear codes. Let $\mathscr{C}$ be an $[n,k]$-linear code over $\mathbb{F}_q$ with $q = p^s$ and H be a parity check matrix of $\mathscr{C}$. We have examined two generating sets of S:

   - Let α be a primitive element of the finite field $\mathbb{F}_q$. The set F_2 is given by $n \times (q-1)$ elements of the form: $\left\{\alpha^j \mathbf{h}_i\right\}_{\substack{i=1,\dots,n \\ j=1,\dots,q-1}}$, where $\mathbf{h}_i$ denotes the i-th column of H. Proposition 5.9 shows that $I_{F_2}(S) = I_+(\mathscr{C})$ where $I_+(\mathscr{C})$ is the binomial ideal of $\mathscr{C}$ attached to the additive table of the field $\mathbb{F}_q$.

     This ideal was depth studied in Chapter 4 where we show that the reduced Gröbner basis of $I_+(\mathscr{C})$ relative to a degree compatible ordering provides us with two complete decoding algorithms, see Section 4.3. Moreover a Graver basis of $I_+(\mathscr{C})$ allows us to obtain the set of codewords of minimal support of $\mathscr{C}$.

   - Let β be a root of any irreducible polynomial of degree s over $\mathbb{F}_p$. The set F_1 is given by $n \times s$ elements of the form: $\left\{\beta^{j-1}\mathbf{h}_i\right\}_{\substack{i=1,\dots,n, \\ j=1,\dots,s}}$ where $\mathbf{h}_i$ denotes the i-th column of H. Proposition 5.13 claims that $I_{F_1}(S) = I_m(\mathscr{C})$ where $I_m(\mathscr{C})$ describes the binomial ideal of $\mathscr{C}$ by taking care of the arithmetic over fields of characteristic p.

     This ideal was already studied in [16] where the authors were unable to describe a complete decoding algorithm for $\mathscr{C}$. See for instance Example 4.17.

Therefore only the semigroups ideals related to the binomial ideals $I_+(\mathscr{C})$ allow us to solve the complete decoding problem, which is the main aim of this thesis. The advantage of using the ideals $I_+(\mathscr{C})$ lies in the fact that on these cases the degree of the monomials $\mathbf{X}^{\mathbf{a}}$ match the Hamming weight of the word $\mathbf{a}$, which could be defined over the ring $\mathbb{Z}_m^n$ or over the finite field $\mathbb{F}_q^n$.

Remark 5.17. Note that the number of generators of the ideal $I_+(\mathscr{C})$ varies depending whether the code $\mathscr{C}$ belongs to one of the following cases:

1. The code $\mathscr{C}$ is modular or $\mathscr{C}$ is linear a defined over a prime field.

2. The code $\mathscr{C}$ is linear and is defined over a finite field $\mathbb{F}_q$ where q is not prime.

See Remark 4.57 for a deeper discussion.

Definition 5.18. The generating set $F = \{\mathbf{n}_1, \dots, \mathbf{n}_r\}$ of a semigroup S is called a *digital representation* of S if every element $\mathbf{m} \in S$ can be written as

$$\sum_{i=1}^{r} a_i \mathbf{n}_i \text{ with } a_1, \dots, a_r \in \{0,1\} \subseteq \mathbb{N}.$$

It is worth pointing out that the choice of digital representations of S, where S defines the semigroup associated with a code $\mathscr{C}$, provides not only complete decoding

algorithms but also the set of codewords of minimal support of $\mathscr{C}$, where $\mathscr{C}$ could be either a modular or a linear code.

Moreover the result obtained in this chapter could be adapted to other classes of codes such as codes defined over multiple alphabets or additive codes. Notice that semigroups in this context are always cancellatives and finites, so digital representations for them always exists and a rather a natural choice.

Bibliography

[1] W. W. Adams and P. Loustaunau. *An introduction to Gröbner bases*, volume 3 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1994. (Cited on page 24.)

[2] A. Ashikhmin and A. Barg. Minimal vectors in linear codes. *IEEE Trans. Inform. Theory*, 44(5):2010–2017, 1998. (Cited on pages 5, 13, 50, 69, 70, 80, and 88.)

[3] A. Barg. Complexity issues in coding theory. In *Handbook of coding theory, Vol. I, II*, pages 649–754. North-Holland, Amsterdam, 1998. (Cited on pages 32, 33, 34, 50, 66, 71, and 90.)

[4] A. Becker, A. Joux, A. May, and A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In *EUROCRYPT*, pages 520–536, 2012. (Cited on page 35.)

[5] E. R. Berlekamp. *Algebraic Coding Theory*. No. M-6. Aegean Park Press, 1984. (Cited on page 24.)

[6] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Information Theory*, IT-24(3):384–386, 1978. (Cited on pages 3, 11, 32, and 90.)

[7] D. J. Bernstein, T. Lange, and C. Peters. Attacking and Defending the McEliece Cryptosystem. In Johannes Buchmann and Jintai Ding, editors, *PQCrypto 2008*, volume 5299 of *Lecture Notes in Computer Science*, pages 31–46. Springer-Verlag Berlin Heidelberg, 2008. (Cited on page 35.)

[8] D. J. Bernstein, T. Lange, and C. Peters. Smaller Decoding Exponents: Ball-Collision Decoding. In *Crypto*, pages 743–760, 2011. (Cited on page 35.)

[9] R. E. Blahut. *Theory and practice of error control codes*. Addison-Wesley Pub. Co., 1983. (Cited on page 24.)

[10] T. Bogart, A. N. Jensen, and R. R. Thomas. The circuit ideal of a vector configuration. *Journal of Algebra*, 309(2):518 – 542, 2007. (Cited on page 91.)

[11] M. Borges-Quintana, M. A. Borges-Trenard, P. Fitzpatrick, and E. Martínez-Moro. Gröbner bases and combinatorics for binary codes. *Applicable Algebra in Engineering Communication and Computing (AAECC)*, 19(5):393–411, 2008.

(Cited on pages 5, 6, 8, 12, 14, 15, 51, 53, 57, 81, 85, 87, 90, 94, 97, 101, 126, 140, and 146.)

[12] M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella, and E. Martínez-Moro. Computing coset leaders and leader codewords of binary codes. Submitted, 2012. (Cited on pages 10, 17, and 49.)

[13] M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro. GBLA-LC: Gröbner Bases by Linear Algebra and Linear Codes. In *ICM 2006. Mathematical Software*, pages 604–605. EMS, 2006. (Cited on pages 5, 12, 51, and 57.)

[14] M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro. A general framework for applying FGLM techniques to linear codes. In *Applied algebra, algebraic algorithms and error-correcting codes*, volume 3857 of *Lecture Notes in Computer Science (LNCS)*, pages 76–86. Springer, Berlin, 2006. (Cited on pages 5, 12, 51, 57, and 126.)

[15] M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro. A Gröbner representation for linear codes. In *Advances in coding theory and cryptography*, volume 3 of *Series on Coding Theory Cryptology*, pages 17–32. World Sci. Publ., Hackensack, NJ, 2007. (Cited on pages 5, 12, 51, and 97.)

[16] M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro. On a Gröbner bases structure associated to linear codes. *J. Discrete Math. Sci. Cryptogr.*, 10(2):151–191, 2007. (Cited on pages 5, 12, 51, 54, 56, 57, 101, 133, 137, 138, 139, and 194.)

[17] M Borges-Quintana, M. A. Borges-Trenard, and E Martínez-Moro. Gröbner representations of Binary Matroids. In *Applicable Algebra in Engineering Communication and Computing (AAECC)*, pages 227–230, 2009. (Cited on page 48.)

[18] M. Borges-Quintana, M.A. Borges-Trenard, I. Márquez-Corbella, and E. Martínez-Moro. An algebraic view to gradient descent decoding. In *Information Theory Workshop (ITW), 2010 IEEE*, pages 1 –4, 30 2010-sept. 3 2010. (Cited on pages 9, 16, and 49.)

[19] Y. Borissov and N. Manev. Minimal codewords in linear codes. *Serdica Math. J.*, 30:303–324, 2004. (Cited on pages 3 and 11.)

[20] I. Borosh and L. B. Treybig. Bounds on positive integral solutions of linear diophantine equations. *Proceedings of the American Mathematical Society*, 55(2):299–304, 1976. (Cited on page 82.)

[21] M. Bras-Amorós and T. Høholdt, editors. *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 18th International Symposium, AAECC-18 2009, Tarragona, Catalonia, Spain, June 8-12, 2009. Proceedings*, volume 5527 of *Lecture Notes in Computer Science*. Springer, 2009. (Cited on page 48.)

[22] E. Briales, A. Campillo, C. Marijuán, and P. Pisón. Minimal systems of generators for ideals of semigroups. *Journal of Pure and Applied Algebra*, 124(1-3):7–30, 1998. (Cited on pages 178 and 180.)

[23] J. Bruck and M. Naor. The hardness of decoding linear codes with preprocessing. *IEEE Transactions on Information Theory*, 36(2):381–385, 1990. (Cited on pages 3, 11, and 90.)

[24] B. Buchberger. Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *Journal of Symbolic Computation*, 41(3-4):475–511, 2006. (Cited on pages 24 and 40.)

[25] A. R. Calderbank. The art of signaling: fifty years of coding theory. *IEEE Trans. Inform. Theory*, 44(6):2561–2595, 1998. Information theory: 1948–1998. (Cited on page 69.)

[26] A. Campillo and P. Pisón. Toric mathematics from semigroup viewpoint. In *Ring theory and algebraic geometry (León, 1999)*, volume 221 of *Lecture Notes in Pure and Appl. Math.*, pages 95–112. Dekker, New York, 2001. (Cited on page 178.)

[27] A. Canteaut and H. Chabanne. A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem. In Pascale Charpin, editor, *EUROCODE 94*, pages 169–173, 1994. (Cited on page 35.)

[28] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998. (Cited on page 35.)

[29] A. Canteaut and N. Sendrier. Cryptanalysis of the original McEliece cryptosystem. In Kazuo Ohta and Dingyi Pei, editors, *ASIACRYPT'98*, volume 1514 of *Lecture Notes in Computer Science*, pages 187–199. Springer-Verlag Berlin Heidelberg, 1998. (Cited on page 35.)

[30] D. Choi and J. D. H. Smith. Greedy loop transversal codes for correcting error bursts. *Discrete Math.*, 264(1-3):37–43, March 2003. (Cited on page 76.)

[31] T. Chunming, G. Shuhong, and Z. Chengli. The Optimal Linear Secret Sharing Scheme for Any Given Access Structure. *Cryptology ePrint Archive*, 2011. (Cited on pages 4 and 11.)

[32] A. H. Clifford and G. B. Preston. *The algebraic theory of semigroups. Vol. II*. Mathematical Surveys, No. 7. American Mathematical Society, Providence, R.I., 1967. (Cited on page 178.)

[33] P. Conti and C. Traverso. Buchberger algorithm and integer programming. In *Applied algebra, algebraic algorithms and error-correcting codes (New Orleans, LA, 1991)*, volume 539 of *Lecture Notes in Comput. Sci.*, pages 130–139. Springer, Berlin, 1991. (Cited on pages 6, 13, 24, 80, and 82.)

[34] D.A. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Graduate Texts in Mathematics. Springer, 2005. (Cited on page 24.)

[35] D.A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Number v. 10 in Undergraduate Texts in Mathematics. Springer, 2007. (Cited on pages 24, 37, 39, and 41.)

[36] R. Crandall. Some notes on steganography, 1998. (Cited on pages 4 and 12.)

[37] B. Debraize and I. Márquez-Corbella. Fault analysis of the stream cipher snow 3g. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2009 Workshop on*, pages 103–110, 2009. (Cited on pages 10 and 17.)

[38] F. Di Biase and R. Urbanke. An Algorithm to Calculate the Kernel of Certain Polynomial Ring Homomorphisms. *Experimental Mathematics*, 4(3):227–234, 1995. (Cited on pages 6, 14, 81, 85, and 87.)

[39] S. T. Dougherty and H. Liu. Independence of vectors in codes over rings. *Designs, Codes and Cryptography*, 51:55–68, 2009. (Cited on pages 30, 31, and 32.)

[40] D. Eisenbud and B. Sturmfels. Binomial ideals. *Duke Mathematical Journal*, 84(1):1–45, 1996. (Cited on page 24.)

[41] P. Elias. List decoding for noisy channels. *Technical report (Massachusetts Institute of Technology. Research Laboratory of Electronics); 335*, 1957. (Cited on page 35.)

[42] 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):329–344, 1993. (Cited on pages 41, 51, 96, 144, 146, and 147.)

[43] M. Finiasz, P. Gaborit, and N. Sendrier. Improved fast syndrome based cryptographic hash functions. In *Proceedings of ECRYPT Hash Workshop 2007*, 2007. `http://www-roc.inria.fr/secret/Matthieu.Finiasz/research/2007/finiasz-gaborit-sendrier-ecrypt-hash-workshop07.pdf`. (Cited on page 35.)

[44] M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In *ASIACRYPT*, pages 88–105, 2009. (Cited on page 35.)

[45] P. Fitzpatrick. Solving a multivariable congruence by change of term order. *J. Symbolic Comput.*, 24(5):575–589, 1997. (Cited on pages 96, 146, and 147.)

[46] P. Fitzpatrick and J. Flynn. A Gröbner basis technique for Padé approximation. *J. Symbolic Comput.*, 13(2):133–138, 1992. (Cited on page 146.)

[47] E. Gabidulin and T. Klove. On a bound involving the covering radius and the newton radius. In *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on*, pages 433–, 1998. (Cited on page 59.)

[48] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.12*, 2009. (Cited on page 57.)

[49] J. I. García-García, M. A. Moreno-Frías, and A. Vigneron-Tenorio. On glued semigroups. *arXiv:1104.2835v2*, 2011. (Cited on pages 81 and 102.)

[50] M. J. E. Golay. Notes on digital coding. *Proc. IEEE.*, 37:657, 1949. (Cited on page 23.)

[51] J. E. Graver. On the foundations of linear and integer linear programming. I. *Math. Programming*, 9(2):207–226, 1975. (Cited on page 83.)

[52] G. M. Greuel, G. Pfister, O. Bachmann, C. Lossen, and H. Schönemann. *A Singular Introduction to Commutative Algebra*. Springer, 2007. (Cited on page 24.)

[53] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 26:147–160, 1950. (Cited on page 23.)

[54] T. Helleseth and T. Klove. The newton radius of codes. *IEEE Trans. Inf. Theor.*, 43(6):1820–1831, November 1997. (Cited on page 59.)

[55] J. Herzog. Generators and relations of abelian semigroups and semigroup rings. *Manuscripta mathematica*, 3:175–194, 1970. (Cited on pages 178 and 179.)

[56] D.G. Hoffman. *Coding Theory: The Essentials*. Pure and applied mathematics. M. Dekker, 1991. (Cited on page 24.)

[57] F. L. Hsu, F. A. Hummer, and J. D. H. Smith. Logarithms, syndrome functions, and the information rates of greedy loop transversal codes. *J. Comb. Math. Comb. Comp.*, 22:33 – 49, 1996. (Cited on page 76.)

[58] W. C. Huffman and V. Pless. *Fundamentals of error-correcting codes*. Cambridge University Press, Cambridge, 2003. (Cited on pages 24, 29, and 61.)

[59] T. Y. Hwang. Decoding linear block codes for minimizing word error rate. *IEEE Trans. Inform. Theory*, 25(6):733–737, 1979. (Cited on page 90.)

[60] D. Ikegami and Y. Kaji. Maximum Likelihood Decoding for Linear Block Codes using Grobner Bases. *IEICE Trans. Fund. Electron. Commun. Comput. Sci.*, E86-A(3):643–651, 2003. (Cited on pages 6, 13, 80, 83, 84, 85, 87, 90, 126, and 129.)

[61] R. Jurrius and R. Pellikaan. *Algebraic Geometry Modeling in Information Theory*, volume 8 of *Coding Theory and Cryptology*, chapter Codes, Arrangements and Matroids. World Scientific Publishing Company Incorporated, 2013. (Cited on page 24.)

[62] J. Justesen and T. Høholdt. *A Course In Error-Correcting Codes*. EMS Textbooks in Mathematics. European Mathematical Society, 2004. (Cited on pages 24 and 26.)

[63] N. Kashyap. A decomposition theory for binary linear codes. *IEEE Trans. Inform. Theory*, 54(7):3035–3058, 2008. (Cited on pages 48, 81, 102, 104, and 108.)

[64] N. Kashyap. Matroid pathwidth and code trellis complexity. *SIAM J. Discret. Math.*, 22(1):256–272, February 2008. (Cited on page 80.)

[65] N. Kashyap. On minimal tree realizations of linear codes. *IEEE Trans. Inform. Theory*, 55(8):3501–3519, 2009. (Cited on pages 81, 102, and 104.)

[66] A. Kehrein and M. Kreuzer. Characterizations of border bases. *Journal of Pure and Applied Algebra*, pages 251–270, 2004. (Cited on page 44.)

[67] M. Kreuzer and L. Robbiano. *Computational Commutative Algebra 1*. Computational Commutative Algebra. Springer, 2008. (Cited on page 24.)

[68] J. P. S. Kung. *A source book in matroid theory*. Birkhāuser, 1986. (Cited on page 24.)

[69] P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In Christoph G. Günther, editor, *EUROCRYPT '88*, volume 330 of *Lecture Notes in Computer Science*, pages 275–280. Springer-Verlag Berlin Heidelberg, 1988. (Cited on page 35.)

[70] J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988. (Cited on page 35.)

[71] R. A. Liebler. Implementing gradient descent decoding. *Michigan Math. J.*, 58(1):285–291, 2009. (Cited on pages 5, 13, 50, 69, 74, 80, and 88.)

[72] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier/North Holland, Amsterdam, 1977. (Cited on page 24.)

[73] M. G. Marinari, H. M. Moller, and T. Mora. Gröbner bases of ideals denned by functionals with an application to ideals of projective points. *Applicable Algebra in Engineering Communication and Computing (AAECC)*, 4:103–145, 1993. (Cited on page 144.)

[74] I. Márquez-Corbella and E. Martínez-Moro. Algebraic structure of the minimal support codewords set of some linear codes. *Adv. Math. Commun.*, 5(2):233–244, 2011. (Cited on pages 9, 16, 50, 80, 81, 105, and 126.)

[75] I. Márquez-Corbella and E. Martínez-Moro. Decomposition of Modular Codes for Computing Test Sets and Graver Basis. *Mathematics in Computer Science*, 6:147–165, 2012. (Cited on pages 9, 16, and 81.)

[76] I. Márquez-Corbella and E. Martínez-Moro. *Algebraic Geometry Modeling in Information Theory*, volume 8 of *Coding Theory and Cryptology*, chapter An Introduction to LDPC codes, pages 129–166. World Scientific Publishing Company Incorporated, 2013. (Cited on pages 9 and 16.)

[77] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. The non-gap sequence of a subcode of a generalized Reed–Solomon code. *Des. Codes Cryptogr.*, 66(1-3):317–333, 2013. (Cited on pages 9 and 16.)

[78] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. On the unique representation of very strong algebraic geometry codes. *Des. Codes Cryptogr.*, pages 1–16, 2013. (Cited on pages 9 and 16.)

[79] I. Márquez-Corbella, E. Martínez-Moro, R. Pellikaan, and D. Ruano. Computational aspects of retrieving a representation of an algebraic geometry code. Submitted, 2013. (Cited on pages 10 and 17.)

[80] I. Márquez-Corbella, E. Martínez-Moro, and E. Suárez-Canedo. Error-correcting pairs for public-key cryptosystem. Submitted, 2013. (Cited on pages 10 and 17.)

[81] I. Márquez-Corbella, E. Martínez-Moro, and E. Suárez-Canedo. On the composition of secret sharing schemes related to codes. Submitted, 2013. (Cited on pages 4, 10, and 17.)

[82] I. Márquez-Corbella, E. Martínez-Moro, and E. Suárez-Canedo. On the ideal associated to a linear code. Submitted, 2013. (Cited on pages 10 and 17.)

[83] E. Martínez-Moro, J. Mozo-Fernández, and C. Munuera. Compounding secret sharing schemes. *Australian Journal of Combinatorics*, 30, 2004. (Cited on page 4.)

[84] J. L. Massey. Minimal codewords and secret sharing, 1993. (Cited on pages 4, 11, 50, and 90.)

[85] A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\mathcal{O}(2^{0.054n})$. In *ASIACRYPT*, pages 107–124, 2011. (Cited on page 35.)

[86] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory, 1978. Jet Propulsion Laboratory DSN Progress Report 42–44. `http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF`. (Cited on page 35.)

[87] E. Miller and B. Sturmfels. *Combinatorial Commutative Algebra*. Graduate Texts in Mathematics. Springer, 2004. (Cited on page 24.)

[88] T. Mora. *Solving polynomial equation systems II = Macaulay's paradigm and Gröbner technology*. Cambridge University Press, Encyclopedia of Mathematics and its Applications 99, 2005. (Cited on page 144.)

[89] C. Munuera. *Algebraic Geometry Modeling in Information Theory*, volume 8 of *Coding Theory and Cryptology*, chapter Steganography from a coding theory point of view. World Scientific Publishing Company Incorporated, 2013. (Cited on pages 4 and 12.)

[90] J. G. Oxley. *Matroid Theory*. Oxford Graduate Texts in Mathematics Series. Oxford University Press, 2006. (Cited on pages 24 and 80.)

[91] C. H. Papadimitriou. On the complexity of integer programming. *J. ACM*, 28(4):765–768, October 1981. (Cited on page 82.)

[92] Y. Park. Modular independence and generator matrices for codes over $\mathbb{Z}_m$. *Designs, Codes and Cryptography*, 50:147–162, 2009. (Cited on pages 31 and 32.)

[93] C. Peters. Information-set decoding for linear codes over $\mathbf{F}_q$. In *PQCrypto 2010*, pages 81–94, 2010. (Cited on page 35.)

[94] E. Prange. Step-by-step decoding in groups with weight function. part 1. AIR FORCE CAMBRIDGE RESEARCH LABS HANSCOM AFB MA, 1961. (Cited on pages 34 and 126.)

[95] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, September 1962. (Cited on page 35.)

[96] A. Renvall, C. Ding, J. Pieprzyk, and J. Seberry. *Information Security and Privacy*, volume 1172, pages 67–78. Springer Berlin / Heidelberg, 1996. (Cited on pages 4 and 11.)

[97] J. C. Rosales. On presentations of subsemigroups of $\mathbf{N}^n$. *Semigroup Forum*, 55(2):152–159, 1997. (Cited on pages 81 and 102.)

[98] M. Sala, T. Mora, and L. Perret. *Gröbner Bases, Coding, and Cryptography*. Springer, 2009. (Cited on pages 24, 46, and 47.)

[99] P. Samuel and A. J. Silberger. *Algebraic Theory Of Numbers*. Dover Books on Mathematics Series. Dover Publications, 2008. (Cited on page 127.)

[100] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986. (Cited on page 82.)

[101] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423,623–656, 1948. `http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`. (Cited on pages 23 and 24.)

[102] J. D. H. Smith. Loop transversals to linear codes. *J. Combin. Inform. System Sci.*, 17:1–8, 1992. (Cited on page 76.)

[103] R. P. Stanley. *Combinatorics and Commutative Algebra:*. Progress in Mathematics. Birkhäuser Boston, 2004. (Cited on page 24.)

[104] W. A. Stein et al. *Sage Mathematics Software (Version 4.7.2)*. The Sage Development Team, 2012. `http://www.sagemath.org`. (Cited on pages 107 and 116.)

[105] J. Stern. A method for finding codewords of small weight. In Gérard D. Cohen and Jacques Wolfmann, editors, *Coding theory and applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer-Verlag Berlin Heidelberg New York, 1989. (Cited on page 35.)

[106] B. Sturmfels. Grobner bases of toric varieties. *Tohoku mathematical journal. Second series*, 43(2):249–261, june 1991. (Cited on page 24.)

[107] B. Sturmfels. *Gröbner bases and convex polytopes*, volume 8 of *University Lecture Series*. American Mathematical Society, Providence, RI, 1996. (Cited on pages 24, 81, 82, 88, 89, 165, 178, and 180.)

[108] B. Sturmfels and R. R. Thomas. Variation of cost functions in integer programming. *Mathematical Programming*, 77:357–387, 1994. (Cited on page 24.)

[109] A. Thomas. Construction of set theoretic complete intersections via semigroup gluing. *Beiträge Algebra Geom.*, 41(1):195–198, 2000. (Cited on pages 81 and 102.)

[110] R. R. Thomas. A geometric buchberger algorithm for integer programming. *Mathematics of Operations Research*, 20:864–884, 1995. (Cited on page 24.)

[111] W. T. Tutte. *Lectures on matroids*, volume 69B of *Journal of Research of the National Bureau of Standards-B. Mathematics and Mathematical Physics*. National Bureau of Standards, 1965. (Cited on page 80.)

[112] J. H. Van Lint. *Introduction to Coding Theory*. Graduate Texts in Mathematics. Springer, 1999. (Cited on page 24.)

[113] J. van Tilburg. On the McEliece public-key cryptosystem. In Shafi Goldwasser, editor, *CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 119–131. Springer-Verlag Berlin Heidelberg, 1990. (Cited on page 35.)

[114] J. van Tilburg. *Security-analysis of a class of cryptosystems based on linear error-correcting codes*. PhD thesis, Eindhoven University of Technology, Netherlands, 1994. (Cited on page 35.)

[115] A. Vigneron-Tenorio. *Álgebra de Semigrupos y Aplicaciones*. PhD thesis, Universidad de Sevilla, 2000. (Cited on page 180.)

[116] T. Wadayama, K. Nakamura, M. Yagita, Y. Funahashi, S. Usami, and I. Takumi. Gradient descent bit flipping algorithms for decoding ldpc codes. *Communications, IEEE Transactions on*, 58(6):1610 –1614, june 2010. (Cited on page 34.)

[117] B. L. van der. Waerden. *Modern Algebra*, volume 1. Springer Verlang, second edition, 1937. (Cited on page 24.)

[118] D. J. A. Welsh. *Matroid Theory*. Dover Books on Mathematics. Dover Publications, 2010. (Cited on page 24.)

[119] N. White. *Theory of Matroids*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2008. (Cited on page 24.)

[120] H. Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57:509–533, 1935. (Cited on page 24.)

[121] J. M. Wozencraft. List decoding. *Quarterly Progress Report*, pages 90–95, Jan. 1958. (Cited on page 35.)

Index

Curriculum Vitae

Contact Information

University of Valladolid
Department of Algebra, Geometry and Topology
Facultad de Ciencias
Paseo de Belén, 7
Valladolid, 47011 (Spain)

office: +34 98342 3046
e-mail: imarquez@agt.uva.es

Education

BSc in Mathematics — Sept. 2003 - Jun. 2008

University of La Laguna (Spain)

MSc in Mathematics and Computer Science — Sept. 2008 - Sept. 2009

Diderot - Paris VII University (France)

Speciality in Cryptologie, Networks and Protocols.

MRes in Mathematics — Sept. 2008 - Sept. 2010

University of Valladolid (Spain)

Doctoral program in Mathematics with Quality Mention.

Dissertation on "*Presentaciones binomiales de programas lineales modulares: Aplicaciones a los códigos correctores de errores*".

Journal articles

1. I. Márquez-Corbella and E. Martínez-Moro. *Decomposition of Modular Codes for Computing Test Sets and Graver Basis*. Mathematics in Computer Science, 6(2), pp. 147-165, 2012.
2. I. Márquez-Corbella, E. Martínez-Moro and G. R. Pellikaan. *On the unique representation of very strong algebraic geometry codes*. Designs, Codes and Cryptography, pp. 1-16, 2012.
3. I. Márquez-Corbella, E. Martínez-Moro and G. R. Pellikaan. *The non-gap sequence of a subcode of a generalized Reed-Solomon code*. Designs, Codes and Cryptography, pp. 1-17, 2012.
4. I. Márquez-Corbella and E. Martínez-Moro. *Algebraic structure of the minimal support codewords set of some linear codes*. Adv. Math. Commun., 5(2), pp. 233-244, 2011.

Main conference papers

1. I. Márquez-Corbella, E. Martínez-Moro, G. R. Pellikaan and D. Ruano. *Computational Aspects of Retrieving a Representation of an Algebraic Geometry Code*. 3^{rd} Int. conf. on Symbolic Computation and Cryptography, CIEM- Castro Urdiales (Spain), July 11-13 2012, pp. 147-151.
2. I. Márquez-Corbella and E. Martínez-Moro. *Efficient computation of the set of codewords of minimal support* XIII Encuentro de Álgebra Computacional y Aplicaciones, Alcalá de Henares (Spain), July 13-15, 2012, pp. 127-130.
3. I. Márquez-Corbella, E. Martínez-Moro and E. Suárez-Canedo. *Construcciones minimales asociadas a códigos concatenados generalizados* XIII Encuentro de Álgebra Computacional y Aplicaciones, Alcalá de Henares (Spain), July 13-15, 2012, pp.131-134.
4. I. Márquez-Corbella, E. Martínez-Moro and E. Suárez-Canedo. *On the Composition of Secret Sharing Schemes Related to Codes*. Workshop on Computation Security. Centre de Recerca Matematica (CRM), Barcelona (Spain), November 28 - December 2, 2011.
5. I. Márquez-Corbella, E. Martínez-Moro and G. R. Pellikaan. *Evaluation of public-key cryptosystems based on algebraic geometry codes*. 3^{rd} Int. Castle Meeting on Coding Theory and Applications, Castell de Cardona, Barcelona (Spain), September 11-15, 2011, pp. 199-204.
6. I. Márquez-Corbella, E. Martínez-Moro and G. R. Pellikaan. *The non-gap sequence of a subcode of a GRS code*. The 7^{th} International Workshop on Coding and Cryptography 2011, Paris (France), April 11-15, 2011, pp. 183-193.
7. M. Borges-Quintana, M. A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro, *An algebraic view to gradient descent decoding*. Information Theory Workshop (ITW), 2010 IEEE, Dublin (Ireland), August 30 - September 3, 2010, pp.1-4.
8. M. Borges-Quintana, M.A. Borges-Trenard, I. Márquez-Corbella and E. Martínez-Moro. *Descodificación por gradiente como reducción*. XII Encuentro de Álgebra Computacional y Aplicaciones, Santiago de Compostela (Spain), July 19-21, 2010.
9. I. Márquez-Corbella and E. Martínez-Moro. *Programación lineal modular y bases de Graver: Cálculo de soportes minimales de códigos lineales*. VII Jornadas de Matemática Discreta y Algorítmica, Castro Urdiales (Spain), July 7-9, 2010.
10. B. Debraize and I. Márquez-Corbella. *Fault Analysis of the Stream Cipher Snow 3G*. Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), Lausanne (Switzerland), September 6, 2009, pp. 103-110.

Book chapters

1. I. Márquez-Corbella and E. Martínez-Moro. *An Introduction to LDPC codes*. In Algebraic Geometry Modeling in Information Theory, Series on Coding Theory and Cryptology, pp. 129-166. ISBN: 978-981-4335-75-1.

Papers submitted for publication

1. I. Márquez-Corbella, E. Martínez-Moro and E. Suárez-Canedo. *On the Composition of Secret Sharing Schemes Related to Codes*.

2. M. B. Quintana, M. A. B. Trenard, I. Márquez-Corbella and E. Martínez-Moro. *Computing coset leaders and leader codewords of binary codes*.
3. I. Márquez-Corbella, E. Martínez-Moro and E. Suárez-Canedo. *On the ideal associated to a linear code.*
4. I. Márquez-Corbella and R. Pellikaan. *Error-correcting pairs for public-key cryptosystem*.

Courses taught

- An invited course at the Research School "Algebra for Secure and Reliable Communication Modeling (ASReCoM)"
  - *An overview of connections between coding and cryptography through case studies*.
  - This course has the duration of 10 teaching hours.
  - The School was held at Morelia (Mexico) under the sponsorship of CIMPA-UNESCO-MESR-MICINN, during 1-13 of October 2012.
- *Linear Algebra for Mechanical Engineering (2Y650)* at the Eindhoven University of Technology (The Netherlands) with a total of 3 ECTS, first semester year course 2011-2012.
- *Matemática Discreta* third year of mathematics degree at University of Valladolid (Spain) with a total of 1,5 ECTS, second semester year course 2011-2012 and 2012-2013.
- *Códigos Correctores* fourth year of mathematics degree at University of Valladolid (Spain) with a total of 1,5 ECTS, second semester year course 2011-2012 and 2012-2013.
- *A case study of the combinatorics of linear codes with SAGE* for 5 hours during the ACAGM Summer School. Leuven (Belgium),10-22 July, 2011.

Organizer of the following events

- *Soria Summer Seminar on Computational Mathematics*, Soria (Spain), 17 July, 2012.
- *Congreso de jóvenes investigadores de la RSME*, Soria (Spain), 5-9 September, 2011.
- Mini-symposium with title *Applications of Matroid Theory in Coding Theory I* at the Canadian Discrete and Algorithmic Mathematics Conference (CanaDAM), Victoria (Canada), 31 May - 3 June, 2011.
- *Soria Summer School on Computational Mathematics: Algebraic Geometric Modeling in Information Theory*, Soria (Spain), 12-16 July, 2010.
- *Soria Summer School on Computational Mathematics: Applied computational Algebraic Geometric Modeling*, Soria (Spain), 13-24 July, 2009.

Editorial work

- Guest editor with I. S. Kotsireas and E. Martínez-Moro of the Special Issue of Mathematics in Computer Science on Matroids in Coding Theory and Related Topics. Vol. 6, Issue 2, June 2012. ISSN: 1661-8270 (Print) 1661-8289 (Online).

Main talks and posters

- **Invited talks**
  - Talk title: *Nuevo Algoritmo de Descodificación Completa para Códigos lineales*. Seminario de Doctorandos de Matemáticas ULL, La Laguna (Spain), February 14, 2013.
  - Talk title: *Enfoque Algebraico a la descodificación de códigos lineales*. Congreso de la RSME 2013, Santiago de Compostela (Spain), January 21-25, 2013.
  - Talk title: *On the ideal associated to a code*. Soria Summer Seminar on Computational Mathematics, Soria (Spain), July 17, 2012.
  - Talk title: *Structural attacks against code-based cryptosystems*. First International IMAC Workshop on Algebraic Applications to Information Theory, Castellón (Spain), April 17, 2012.
  - Talk title: *Are AG codes secure for code based PKC?*. Cryptography Working Group EIDMA/DIAMANT, Utrecht (The Netherlands), December 2, 2011.
  - Talk title: *Modular integer programming and applications to coding theory*. EIDMA Seminar Combinatorial Theory, Eindhoven (The Netherlands), December 1, 2010.
  - Talk title: *Descripción algebraica del conjunto de palabras de soporte mínimo de un código*. Seminario de especialización de álgebra, geometría algebraica y singularidades de la Universidad de La Laguna, La Laguna (Spain), June 28 - July 2, 2010.
  - Talk title: *Algebraic Description of Modular Integer Programming: Applications to Coding Theory*. Claude Shannon Institute Workshop on Coding and Cryptography, Cork (Ireland), May 17-18, 2010.
  - Talk title: *Computing Graver bases and applications to coding theory*. SINGACOM Computing Team (SCT), Valladolid (Spain), April 5, 2010.
  - Talk title: *Security of stream ciphers against fault attacks*. Thematic Seminar on Singularities, Algebraic Geometry, Computing and Information, Segovia (Spain), October 16, 2009.
  - Talk title: *Seguridad de los cifrados en flujo contra los ataques de inyección de fallos*. Seminario de Álgebra, Geometría Algebraica y Singularidades, La Laguna (Spain), October 13, 2009.
- **Other talks**
  - Talk title: *Estudio de la seguridad de varias familias de códigos para criptosistemas de clave pública*. Encuentro de Jóvenes Investigadores en Matemáticas, La Laguna (Spain), September 27-29, 2012.
  - Talk title: *Efficient computation of the set of codewords of minimal support*. XIII Encuentro de Álgebra Computacional y Aplicaciones, Alcalá de Henares (Spain), July 13-15, 2012.
  - Talk title: *A Characterization of MDS codes that have an Error-correcting Pair*. Code-based Cryptography Workshop (CBC), Lyngby (Denmark), May 9-11, 2012.
  - Talk title: *Vulnerabilities of code-based cryptography*. Spanish Cryptography Days, Murcia (Spain), November 18-19, 2012.

  - Talk title: *Criptografía basada en códigos correctores*. Primer Encuentro de Jóvenes Investigadores en Matemáticas (PEJIM). La Laguna (Spain), September 28-30, 2011.
  - Talk title: *Evaluation of Public-key Cryptosystems Based on Algebraic Geometry*. Third International Castle Meeting on Coding Theory and Applications. Cardona Castle, Barcelona, (Spain), September 11-15, 2011.
  - Talk title: *Matroid decomposition and minimal codewords*. Canadian Discrete and Algorithmic Mathematics Conference (CanaDAM), Victoria (Canada), May 31 - June 3, 2011.
  - Talk title: *The non-gap sequence of a subcode of a generalized Reed-Solomon code*. The Seventh International Workshop on Coding and Cryptography (WCC), Paris (France), April 11-15, 2011.
  - Talk title: *Computing minimal codewords of certain linear codes*. DIAMANT / EIDMA Symposium, Het Bosgoed (The Netherlands), November 25-26, 2010.
  - Talk title: *Descodificación por gradiente como reducción*. XII Encuentro de Álgebra Computacional y Aplicaciones (EACA). Santiago de Compostela (Spain), July 19-21, 2010.
  - Talk title: *The relationship between GDD algotihms and the Gröbner representation of a linear code*. S3CM Soria Summer School on Computational Mathematics, Soria (Spain), July 12-16, 2010.
  - Talk title: *Combinatorics of minimal codewords of binary codes*. ALCOMA 10 Designs and Codes, Thurnau (Germany), April 11-18, 2010.
  - Talk title: *Fault Analysis of the Stream Cipher Snow 3G*. Fault Diagnosis and Tolerance in Cryptography (FDTC). Lausanne (Switzerland), September 5, 2009.
  - Talk title: *Searching for Low Weight Codewords*. S3CM on Applied computational Algebraic Geometric Modelling, Soria(Spain), July 13-24, 2009.

- **Posters**
  - Poster title: *Computation of Graver Test Sets. Applications to Coding Theory*. Congreso de la RSME 2011, Ávila (Spain), February 1-5, 2011.
  - Poster title: *Combinatorics of the Graver Basis associated to certain linear codes*. EACAs First International School on Computer Algebra and its Applications, Costa Adeje (Spain), January 24-28, 2011.
  - Poster title: *Attack against Snow3G*. Sixth Meeting for Young Mathematicias in Segovia (YMIS), Segovia (Spain), February 7 - 12, 2010.

Visits to foreign institutions

- Department of Mathematics and Computer Science, Technische Universiteit Eindhoven (The Netherlands).
  - First stay: 15/09/2010-15/12/2010.
  - Second stay: 10/10/2011-15/02/2012.
  - Third stay: 20/08/2012-20/09/2012.

- Internship at the Security Lab in Gemalto, Meudon (France) as part of the Master degree in Mathematics and Computer Science speciality in Cryptologie, Networks and Protocols in Diderot - Paris VII University, April-October 2009.

Professional memberships

- Member of the European Cooperation in Science and Technology (COST) Action IC1104, Random Network Coding and Designs over GF(q), since 2012.
- Member of the research institute of mathematics at Valladolid (IMUVa), since 2011.
- Member of Red EACA (Red Temática de Cálculo Simbólico, Álgebra Computacional y Aplicaciones), since 2010.
- Member of the Spanish Royal Society of Mathematics (RSME), since 2008.
- Member of the research group SINGACOM (SINGularities, Algebraic Geometry, COmmutative Algebra, COding, COMbinatorics, COMputation and Optimization), since 2008.

Attendance to other courses

- *First European Training School in Network Coding*. Barcelona (Spain), February 4-8, 2013.
- *Workshop in Code-Based Cryptography*. Eindhoven (The Netherlands), May 11-12, 2011.
- *Risc/Intercity Number Theory Seminar on Crypto, Coding and Geometry*. Amsterdam (The Netherlands), November 18-19, 2011.
- *EIDMA/DIAMANT Cryptography Working Group*. Utrecht (The Netherlands), October 1-December 3, 2010.
- *Sage Days Workshop* at the Field Institute. Toronto (Canada), May 3-7, 2010.
- *Second Iberian Meeting on Numerical Semigroups IMNS*. Granada (Spain), February 3-5, 2010.
- *International School on Combinatorics Pilar-Pisón-Casares*. Sevilla (Spain), January 25-30, 2010.
- *CHES 2009 - Workshop on Cryptographic Hardware and Embedded Systems*. Lausanne (Switzerland), September 6-9, 2009.
- *Seminar on Singularities: Algebraic Methods*. Garachico (Spain), September 2-6, 2008.
- *S3CM 2008: Soria Summer School on Computation and Mathematics: Algebraic Coding Theory*. Soria (Spain), July 2-11, 2008.
- *Seminar on Algebraic Geometry* held at the IDU of Algebra at University of La Laguna (Spain), October 2006 - June 2007.

Scholarships and awards

- Short Stays granted by the Spanish Government to visit the Department of Mathematics and Computer Science at Technische Universiteit Eindhoven (The Netherlands).

  - First stay: 15/09/2010 - 15/12/2010.
  - Second stay: 10/10/2011 - 15/02/2012.
- *FPU* grant from the Spanish Government for PhD studies, Sept. 2009 - ongoing.
- *"La Caixa" foundation and Government of France: Master's Program in France*, Paris, Sept. 2008 - Sept. 2009.
- *University of Valladolid: Introduction to research*, Spain, June 2008.
- *Bancaja-University of La Laguna : Scholarship with honors Erasmus*, July 2008.

Peer reviewer for

- ZBMATH Online Database since 2012.
- Aplicable Algebra in Engineering, Communication and Computing. Special Issue on Computer Algebra in Coding Theory and Cryptography.
- The third International Castle Meeting on Coding Theory and Applications (3ICMCTA).

Language skills

- Spanish: Mother tongue.
- English: Good knowledge of written and oral language.
  - Cambridge First Certificate in English.
- French: Good knowledge of written and oral language.
  - Master degree at the University of Paris VII - Diderot.

Irene Márquez-Corbella was born on September 20, 1985 in Tenerife, Canary Islands. She received her diploma in mathematics in 2008 from the University of La Laguna. In 2009, she was awarded by "La Caixa" foundation and the Government of France with an scholarship for Msc studies which allow her to accomplish a Master degree at the University of Diderot-Paris VII and an intership at the Security Lab in Gemalto, Meudon (France). In 2010 she got her MsC in Mathematics at the Univesity of Valladolid and started her PhD studies under the support of the FPU scholarship from the Spanish Governement and the research group SINGACOM (from the Institute of Mathematics, UVa). The present dissertation contains the result of part of her work from 2009 to 2013.

'Combinatorial Commutative Algebra Approach to Complete Decoding'

This thesis aims to explore the bridge between the algebraic structure of a code and the complete decoding process. Although complete decoding is an NP-problem, even if preprocessing is allowed, we propose alternative algorithms and new techniques to tackle such problem. This problem has many applications not only in Coding Theory but also in other areas of Information Security such as Cryptography and Steganography.
This memory is intended to be as much self contained as possible, providing the whole theoretical setting behing the structure used as well as the computational tools needed, namely Gröbner and Border basis.

Institute of Mathematics University of Valladolid • http://www.imuva.uva.es/

SINGACOM Group • http://www.singacom.uva.es