



**UNIVERSIDAD
DE GRANADA**

TRABAJO FIN DE GRADO
INGENIERÍA EN INFORMÁTICA

Estudio de la estructura

arbórea de los semigrupos numéricos

Autor

Mario Casas Pérez

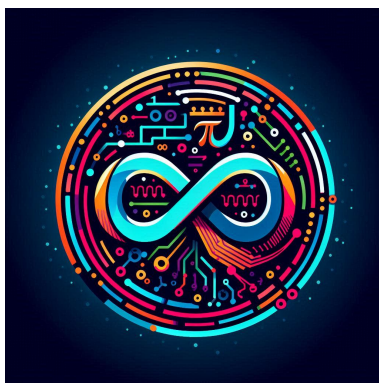
Directores

José Antonio Jiménez Madrid



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, 16 de junio de 2025



Estudio de la estructura

arbórea de los semigrupos numéricos

Autor

Mario Casas Pérez

Directores

José Antonio Jiménez Madrid

Estudio de la estructura arbórea de los semigrupos numéricos

Mario Casas Pérez

Palabras clave: número de Frobenius, género, generadores minimales, autómatas, conductor, serie de Hillbert, conjuntos de Apéry, sitio web, criptografía, matemáticas, informática, sistema minimal de generadores infinito, multiplicidad, huecos especiales, algoritmos

Resumen

La motivación de la realización de este trabajo de fin de grado es obtener una comprensión detallada sobre los semigrupos numéricos y su estudio en el ámbito de la informática y de las matemáticas. Un semigrupo numérico es un subconjunto de los números naturales \mathbb{N} que contiene el cero, es cerrado bajo la suma y tiene un complemento finito en \mathbb{N} . Estos semigrupos pueden describirse mediante generadores finitos y poseen propiedades clave como el número de Frobenius, el elemento conductor y la cantidad de huecos.

Los semigrupos numéricos desempeñan un papel fundamental en matemáticas, especialmente en álgebra conmutativa, teoría de números y geometría algebraica. Son esenciales en el estudio de los dominios de valoración y en la clasificación de singularidades en curvas algebraicas. Además, encuentran aplicaciones en informática, como en la optimización discreta, la teoría de autómatas, la criptografía y la teoría de la información. Un ejemplo clásico es el problema de la moneda, que consiste en determinar el mayor valor que no se puede representar con un conjunto dado de denominaciones monetarias.

Se ha realizado adicionalmente una página web en donde aquellos usuarios interesados en el estudio y análisis de los semigrupos numéricos puedan aprender y estudiar de forma dinámica todas sus propiedades matemáticas así como su inclusión en el área de la informática. Esta página facilitará el conocimiento mediante ejemplos prácticos donde el usuario podrá interactuar a través de diversos programas para ver el comportamiento de estos.

Este trabajo aborda un análisis teórico y práctico de los semigrupos numéricos, destacando su importancia en problemas combinatorios y en el diseño de algoritmos eficientes para la resolución de problemas en ciencias de la computación.

Study of the tree structure of numerical semigroups

Mario, Casas Pérez

Keywords: Frobenius number, genus, minimal generators, automata, conductor, Hilbert series, Apéry sets, cryptography, mathematics, computer science, infinite minimal generating system, multiplicity, special gaps, algorithms

Abstract

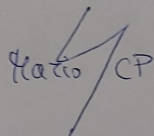
The motivation for conducting this final project is to gain a detailed understanding of numerical semigroups and their study in the context of computer science and mathematics. A numerical semigroup is a subset of the natural numbers \mathbb{N} that includes zero, is closed under addition, and has a finite complement in \mathbb{N} . These semigroups can be described using finite generators and possess key properties such as the Frobenius number, the conductor element, and the number of gaps.

Numerical semigroups play a fundamental role in mathematics, especially in commutative algebra, number theory, and algebraic geometry. They are essential in the study of valuation domains and the classification of singularities in algebraic curves. Furthermore, they find applications in computer science, such as in discrete optimization, automata theory, cryptography, and information theory. A classic example is the coin problem, which involves determining the largest value that cannot be represented with a given set of coin denominations.

In addition, a website has been developed where users interested in the study and analysis of numerical semigroups can learn and explore their mathematical properties dynamically, as well as their application in the field of computer science. This platform will enhance understanding through practical examples, allowing users to interact with various programs to observe their behavior.

This work presents a theoretical and practical analysis of numerical semigroups, highlighting their importance in combinatorial problems and the design of efficient algorithms for solving problems in computer science.

Yo, **Mario Casas Pérez**, alumno de la titulación INGENIERÍA INFORMÁTICA de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 75941571X, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

A rectangular box containing a handwritten signature in blue ink. The signature appears to be 'Mario CP' with a stylized flourish above the 'P'.

Fdo: Mario Casas Pérez

Granada a 16 de junio de 2025 .

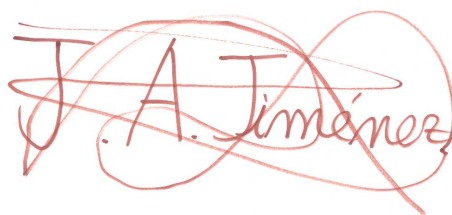
D. **José Antonio Jiménez Madrid**, Profesor del Área de informática del Departamento de Álgebra de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Estudio de la estructura arbórea de los semigrupos numéricos***, ha sido realizado bajo su supervisión por **Mario Casas Pérez**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 16 de junio de 2025 .

El director:

A handwritten signature in red ink, appearing to read "J. A. Jiménez", with a large, stylized flourish or loop extending from the end of the name.

José Antonio Jimenez Madrid

Agradecimientos

Me gustaría agradecer a mi familia por ayudarme e inspirarme siempre en todo lo que hago, a mis amigos por ser personas que suman en mi vida, y al profesorado que me ha ayudado en todo lo que he necesitado y me ha dado grandes aprendizajes necesarios para completar esta carrera.

Índice general

1. Introducción	19
1.1. Definición y estructura	19
1.1.1. Naturaleza y cierre aditivo	19
1.1.2. Complemento finito y conductor	20
1.1.3. Generación finita y representación	20
1.1.4. Número de Frobenius y género	20
1.1.5. Conjuntos de Apéry	21
1.1.6. Interpretación como solución de ecuaciones diofánticas	21
1.1.7. Aplicaciones y representaciones alternativas	21
1.2. Principios clave	22
1.2.1. Número de Frobenius $F(S)$	22
1.2.2. Género	22
1.2.3. Tipo	23
1.2.4. Elemento conductor $c(S)$	23
1.2.5. Relaciones entre los principios clave	24
1.3. Importancia de la estructura arbórea de los semigrupos	24
1.3.1. Conexiones con la teoría de números	24
1.3.2. Impacto en el álgebra conmutativa y la geometría algebraica	25
1.3.3. Aplicaciones en combinatoria y optimización	25
1.3.4. Ventajas en la computación y el desarrollo de algoritmos	25
1.3.5. Análisis y perspectivas futuras	26
1.4. Aplicaciones enfocadas a la informática	26
1.4.1. Codificación y teoría de la información	26
1.4.2. Optimización y algoritmos	27
1.4.3. Criptografía	27
1.4.4. Lenguajes formales y teoría de autómatas	27
1.4.5. Análisis de complejidad	28
1.4.6. Visión informática	28
1.5. Ejemplo del problema de la mochila	28
1.5.1. Contextualización del problema de la mochila	28
1.5.2. Relación con semigrupos numéricos	29
1.5.3. Análisis del ejemplo: mochila y semigrupo	29

1.5.4.	Implicaciones en la resolución del problema	30
1.5.5.	Conclusión	30
2.	Semigrupos numéricos	31
2.1.	Número de Frobenius	31
2.1.1.	Propiedades	32
2.1.2.	Ejemplos	34
2.1.3.	Bases de Gröbner	35
2.1.4.	Algoritmos y Métodos	36
2.1.5.	Aplicaciones	37
2.1.6.	Programa de cálculo del número de Frobenius	38
2.2.	Género	40
2.2.1.	Propiedades	41
2.2.2.	Semigrupos Equilibrados	43
2.2.3.	Ejemplos	43
2.2.4.	Algoritmos y Métodos	44
2.2.5.	Aplicaciones	45
2.2.6.	Programa de cálculo de género de un semigrupo numérico	46
2.3.	Conductor	48
2.3.1.	Propiedades	49
2.3.2.	Ejemplos	50
2.3.3.	Algoritmos y Métodos	50
2.3.4.	Aplicaciones	51
2.3.5.	Programa de cálculo del género de un semigrupo numérico	52
2.4.	Serie de Hilbert	53
2.4.1.	Propiedades	54
2.4.2.	Ejemplos	55
2.4.3.	Algoritmos y Métodos	57
2.4.4.	Aplicaciones	59
2.4.5.	Programa de cálculo de la serie de Hilbert	61
2.5.	Sistema minimal de generadores	63
2.5.1.	Propiedades	64
2.5.2.	Ejemplos	66
2.5.3.	Algoritmos y Métodos	67
2.5.4.	Aplicaciones	68
2.5.5.	Programa de cálculo de sistemas minimales de generadores	70
2.6.	Sistema de generadores minimal infinito	71
2.6.1.	Propiedades	72
2.6.2.	Ejemplos	73
2.6.3.	Algoritmos y Métodos	74
2.6.4.	Aplicaciones	77
2.6.5.	Programa de cálculo de sistemas minimales infinitos	79

2.7. Conjuntos de Apéry	80
2.7.1. Propiedades	81
2.7.2. Ejemplos	82
2.7.3. Algoritmos y Métodos para los Conjuntos de Apéry en Semigrupos Numéricos	83
2.7.4. Aplicaciones	84
2.7.5. Programa de cálculo de conjuntos de Apéry	86
3. Semigrupos numéricos basados en informática	89
3.1. Autómatas	89
3.1.1. Propiedades	91
3.1.2. Ejemplos	93
3.1.3. Utilidades	94
3.1.4. Aplicaciones	95
3.1.5. Programa de cálculo de un autómata	96
3.2. Seguridad	98
3.2.1. Propiedades	99
3.2.2. Ejemplos	100
3.2.3. Utilidades	101
3.2.4. Aplicaciones	103
3.2.5. Programa de cálculo sobre contraseñas ultra seguras .	104
3.3. Problema de la mochila	107
3.3.1. Propiedades	107
3.3.2. Ejemplos	109
3.3.3. Utilidades	110
3.3.4. Aplicaciones	112
3.3.5. Programa de cálculo del problema de la mochila con semigrupos numéricos	114
3.4. Semigrupos numéricos de género fijo	115
3.4.1. Propiedades	116
3.4.2. Ejemplos	117
3.4.3. Utilidades	118
3.4.4. Aplicaciones	120
3.4.5. Programa de cálculo de semigrupos numéricos de gé- nero fijo	123
3.5. Semigrupos numéricos con número de Frobenius fijo	129
3.5.1. Propiedades	130
3.5.2. Ejemplos	131
3.5.3. Utilidades	132
3.5.4. Aplicaciones	134
3.5.5. Programa para calcular los semigrupos numéricos dado un número de Frobenius fijo	135
4. Sitio web de estudio y análisis	141

Capítulo 1

Introducción

1.1. Definición y estructura

A pesar de la aparente simplicidad en su definición, los semigrupos numéricos esconden una estructura rica y multifacética que se entrelaza profundamente con la teoría de números, el álgebra conmutativa y la combinatoria. En esta sección se presenta un análisis extenso que abarca desde sus propiedades fundamentales hasta sus aplicaciones prácticas.

1.1.1. Naturaleza y cierre aditivo

La propiedad de estar cerrados bajo la suma es el pilar fundamental de cualquier semigrupo numérico. Formalmente, si $a, b \in S$ entonces $a + b \in S$. Al exigir además que $0 \in S$ y que $\mathbb{N} \setminus S$ sea finito ($\text{mcd}(S) = 1$), distinguimos a los semigrupos numéricos de los submonoides aditivos arbitrarios de \mathbb{N} .

Esta característica permite interpretar S como la imagen de un morfismo de monoides

$$\varphi : \mathbb{N}^n \longrightarrow \mathbb{N}, \quad \varphi(e_i) = g_i,$$

donde los generadores g_i son los “ladrillos básicos” y los coeficientes naturales k_i indican cuántas veces se repite cada uno:

$$S = \left\{ \sum_{i=1}^n k_i g_i \mid k_i \in \mathbb{N} \right\}.$$

Comprobar la pertenencia $a \in S$ equivale entonces a resolver la ecuación diofántica lineal

$$g \cdot k = a, \quad \text{donde } g \cdot k = \sum_{i=1}^n k_i g_i \text{ y } k \in \mathbb{N}^n.$$

1.1.2. Complemento finito y conductor

Una propiedad distintiva de los semigrupos numéricos es que el conjunto de huecos, es decir, los elementos en $\mathbb{N} \setminus S$, es finito. Esto implica la existencia de un **conductor**, que es el mínimo número c a partir del cual se tiene:

$$n \geq c, \quad \forall n \in S.$$

Esta propiedad tiene varias implicaciones:

- **Estabilidad asintótica:** La existencia de un conductor garantiza que la estructura de S se estabiliza y se vuelve regular para números suficientemente grandes.
- **Condición del MCD:** La finitud del complemento es equivalente a que el máximo común divisor de los generadores de S sea 1. Esto asegura que el semigrupo se extienda, salvo por unos pocos huecos, a casi todo \mathbb{N} .

1.1.3. Generación finita y representación

Cada semigrupo numérico es **finitamente generado**. Es decir, existe un conjunto finito $A \subset \mathbb{N}$ tal que:

$$S = \{a_1x_1 + a_2x_2 + \cdots + a_nx_n \mid x_i \in \mathbb{N}\}.$$

Esta representación no solo simplifica la descripción del semigrupo, sino que también es fundamental para el análisis algorítmico y computacional de sus propiedades. La noción de **generadores mínimos** surge en este contexto, permitiendo una descripción óptima y reducida de la estructura de S .

1.1.4. Número de Frobenius y género

Entre los invariantes más estudiados se encuentra el **número de Frobenius**, definido como el mayor entero que no se puede expresar como combinación lineal no negativa de los generadores. Este concepto es especialmente relevante en problemas de optimización y en el análisis de sistemas discretos.

Asimismo, se define el **género del semigrupo** como el número total de huecos en $\mathbb{N} \setminus S$. Estos dos invariantes ofrecen una medida de la “deficiencia” y complejidad del semigrupo:

- Un género pequeño suele asociarse a una estructura más regular.
- Un género elevado puede reflejar relaciones complejas entre los generadores.

1.1.5. Conjuntos de Apéry

Los **conjuntos de Apéry** son herramientas analíticas de gran utilidad para estudiar la estructura interna de S . Para un elemento fijo $m \in S$, el conjunto de Apéry se define como:

$$\text{Ap}(S, m) = \{s \in S \mid s - m \notin S\}.$$

Este conjunto contiene representantes únicos de las clases residuales módulo m y permite:

- Calcular invariantes como el número de Frobenius.
- Estudiar la distribución de los huecos dentro del semigrupo.

La utilización de los conjuntos de Apéry aporta un enfoque combinatorio y aritmético para comprender en profundidad la estructura de S .

1.1.6. Interpretación como solución de ecuaciones diofánticas

Una interpretación alternativa y enriquecedora de los semigrupos numéricos es su conexión con las **ecuaciones diofánticas lineales** de la forma:

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b,$$

donde los coeficientes a_i son enteros positivos y primos relativos. Bajo estas condiciones, el conjunto de soluciones $(x_1, x_2, \dots, x_n) \in \mathbb{N}^n$ se identifica precisamente con un semigrupo numérico. Esta relación establece un puente entre la teoría de semigrupos y problemas clásicos de la teoría de números, como el problema del cambio o el corte de barra.

1.1.7. Aplicaciones y representaciones alternativas

La versatilidad de los semigrupos numéricos se refleja en sus múltiples representaciones y aplicaciones:

- **Álgebra conmutativa:** En el estudio de anillos semigraduados y módulos, donde la condición de Noetherianidad se relaciona con la finitud de los huecos y la existencia de generadores mínimos.
- **Combinatoria:** En problemas de partición y conteo, donde la estructura del semigrupo codifica información esencial para la enumeración.
- **Optimización:** En programación entera y problemas logísticos, donde los invariantes del semigrupo (como el número de Frobenius) permiten identificar límites y restricciones en sistemas discretos.

Cada una de estas representaciones ofrece distintas “ventanas” para analizar y comprender la compleja interacción entre los generadores, el conductor y los huecos, lo que enriquece tanto la teoría abstracta como las aplicaciones prácticas en diversas ramas de las matemáticas.

1.2. Principios clave

Los semigrupos numéricos se distinguen por propiedades fundamentales que permiten caracterizar y clasificar su estructura. En esta sección se presenta un análisis detallado de los principios clave: el número de Frobenius, el género, el tipo y el elemento conductor.

1.2.1. Número de Frobenius $F(S)$

El **número de Frobenius** se define como el mayor número natural que no pertenece al semigrupo S . Este invariante es crucial por las siguientes razones:

- **Límite de inalcanzabilidad:** Marca el umbral máximo de los números que no pueden expresarse como combinación lineal no negativa de los generadores de S . Es decir, todos los enteros mayores a $F(S)$ son representables.
- **Aplicaciones prácticas:** Se utiliza en problemas clásicos como el de la “moneda” o “cambio”, donde se busca determinar el mayor monto que no puede obtenerse mediante ciertas denominaciones.
- **Indicador de densidad:** Un número de Frobenius pequeño sugiere que el semigrupo se “llena” más rápidamente, mientras que uno grande indica una mayor dispersión de los huecos en S .

1.2.2. Género

El **género** de un semigrupo numérico es el número total de huecos o enteros positivos que no están presentes en S . Se puede expresar de forma alternativa como:

$$\text{Género} = F(S) + 1 - |S \cap \{0, 1, 2, \dots, F(S)\}|,$$

lo que representa la cantidad de números que no pueden formarse a partir de los generadores. Sus implicaciones son:

- **Medida de complejidad:** Un género elevado implica que existen muchos huecos, reflejando una estructura más “dispersa” en el semigrupo.

- **Relación con otros invariantes:** La cantidad de huecos se relaciona directamente con $F(S)$ y con la eficiencia de la representación de enteros en S .
- **Aplicaciones teóricas:** Se emplea en la caracterización de propiedades geométricas y algebraicas, por ejemplo, en la teoría de curvas algebraicas donde los huecos pueden asociarse a ciertas irregularidades.

1.2.3. Tipo

El **tipo** de un semigrupo numérico se refiere al número de elementos mínimos necesarios para expresar cualquier entero mayor que $F(S)$ como combinación de sus generadores. En otras palabras, mide la *redundancia* o *minimalidad* en la representación de los elementos. Este concepto es relevante porque:

- **Eficiencia en la representación:** Un tipo bajo indica que pocos generadores son suficientes para representar todos los números a partir de cierto umbral, lo que refleja una estructura ordenada y eficiente.
- **Complejidad interna:** Un tipo alto sugiere múltiples formas mínimas de expresar enteros, implicando una mayor complejidad combinatoria dentro del semigrupo.
- **Implicaciones algorítmicas:** El estudio del tipo facilita el desarrollo de algoritmos que determinan representaciones mínimas, lo cual es esencial en aplicaciones computacionales y de optimización.

1.2.4. Elemento conductor $c(S)$

El **elemento conductor** es el mínimo número natural $c(S)$ tal que para todo $x \geq c(S)$ se tiene $x \in S$:

$$c(S) = \min\{n \in \mathbb{N} \mid \forall m \geq n, m \in S\}.$$

Este concepto tiene las siguientes implicaciones:

- **Cobertura eventual:** Garantiza que, aunque S pueda tener huecos, estos se limitan a los enteros menores que $c(S)$; a partir de este punto, S abarca completamente \mathbb{N} .
- **Determinación de otros invariantes:** La existencia del conductor permite calcular invariantes como el número de Frobenius y el género, proporcionando una base sólida para el análisis de la estructura.
- **Conexión con la representabilidad:** Indica la cota inferior a partir de la cual todos los enteros son representables, lo que es fundamental para resolver problemas en optimización y teoría de números.

1.2.5. Relaciones entre los principios clave

Estos principios no son independientes, sino que se interrelacionan de manera significativa:

- La existencia de un **conductor** $c(S)$ implica la finitud del número de huecos y, por ende, la existencia de un **número de Frobenius** finito.
- La representación de los enteros mayores que $F(S)$ en términos de los generadores está intrínsecamente ligada al **tipo**, ya que mide la eficiencia y minimalidad de dicha representación.
- El **género** ofrece una medida cuantitativa de los huecos en S y se relaciona directamente tanto con $F(S)$ como con la estructura generadora, permitiendo una caracterización global del semigrupo.

En resumen, los principios clave —el número de Frobenius, el género, el tipo y el elemento conductor— constituyen los pilares fundamentales para entender y analizar la estructura de los semigrupos numéricos. Estos conceptos no solo ofrecen una descripción teórica precisa, sino que también facilitan aplicaciones prácticas en la resolución de problemas clásicos en teoría de números, optimización y combinatoria.

1.3. Importancia de la estructura arbórea de los semigrupos

Los semigrupos numéricos, a pesar de su definición elemental como subconjuntos de \mathbb{N} cerrados bajo la suma y con complemento finito, ocupan un lugar destacado en diversas áreas de las matemáticas. Su estudio no solo revela profundas conexiones entre teoría de números, álgebra conmutativa y combinatoria, sino que también abre puertas a aplicaciones prácticas en problemas de optimización y algoritmos computacionales. A continuación, se expone un análisis detallado sobre su relevancia en el ámbito matemático.

1.3.1. Conexiones con la teoría de números

La naturaleza discreta de los semigrupos numéricos los vincula intrínsecamente con la teoría de números:

- **Ecuaciones diofánticas:** La caracterización de un semigrupo numérico a través de la representabilidad de enteros como combinaciones lineales no negativas de un conjunto finito de generadores permite re-interpretar y resolver ecuaciones diofánticas lineales. Este enfoque es esencial en la resolución de problemas clásicos como el problema del cambio de monedas, donde se determina el mayor número no representable (número de Frobenius).

- **Invariantes aritméticos:** Conceptos como el número de Frobenius, el género y el conductor ofrecen medidas cuantitativas de la “deficiencia” de representabilidad en un semigrupo. Estos invariantes son análogos a otros invariantes aritméticos y geométricos, lo que permite establecer puentes entre diferentes ramas de la matemática.

1.3.2. Impacto en el álgebra conmutativa y la geometría algebraica

En el campo del álgebra conmutativa, los semigrupos numéricos se relacionan con la teoría de anillos y módulos:

- **Anillos semigraduados:** Los semigrupos numéricos sirven como ejemplos fundamentales en el estudio de anillos semigraduados, en los cuales la estructura del semigrupo refleja propiedades como la Noetherianidad y la finitud de ciertos invariantes.
- **curvas algebraicas:** En geometría algebraica, los semigrupos se asocian a puntos singulares en curvas proyectivas. El conjunto de huecos de un semigrupo puede interpretarse como la medida de “irregularidad” o de defecto en una curva, lo que influye en la clasificación y estudio de las singularidades.

1.3.3. Aplicaciones en combinatoria y optimización

El análisis combinatorio se beneficia significativamente del estudio de los semigrupos numéricos:

- **Enumeración y particiones:** La estructura de un semigrupo numérico facilita la formulación de problemas de conteo y partición, ya que la representación de enteros a través de un conjunto finito de generadores se puede analizar mediante técnicas combinatorias.
- **Problemas de optimización:** En programación entera y en algoritmos de optimización, los semigrupos ofrecen un marco teórico para determinar soluciones óptimas en contextos discretos. El conocimiento de invariantes como el número de Frobenius permite establecer límites y restricciones en problemas prácticos.

1.3.4. Ventajas en la computación y el desarrollo de algoritmos

La estructura finitamente generada de los semigrupos numéricos permite el desarrollo de algoritmos eficientes:

- **Algoritmos de cómputo:** La representación de un semigrupo en términos de sus generadores minimiza la complejidad computacional, lo que facilita la implementación de algoritmos para calcular invariantes como el género o el número de Frobenius.
- **Aplicaciones computacionales:** Estas propiedades han sido aprovechadas en áreas como la criptografía, donde la estructura discreta y los problemas de representabilidad ofrecen escenarios propicios para el diseño de sistemas seguros.

1.3.5. Análisis y perspectivas futuras

En síntesis, los semigrupos numéricos constituyen un puente entre diversas disciplinas matemáticas. Su estudio ha permitido:

- Profundizar en la comprensión de problemas clásicos de la teoría de números y la resolución de ecuaciones diofánticas.
- Establecer conexiones importantes con la teoría de anillos, módulos y geometría algebraica, especialmente en la clasificación de singularidades.
- Desarrollar herramientas y algoritmos aplicables en combinatoria, optimización y teoría computacional.

El análisis de estos conjuntos discretos continúa siendo un campo de investigación activo, donde nuevas técnicas y enfoques prometen ampliar aún más su aplicación en problemas tanto teóricos como prácticos. La intersección de teoría y aplicación práctica convierte a los semigrupos numéricos en una herramienta poderosa y versátil en el arsenal matemático.

1.4. Aplicaciones enfocadas a la informática

Los semigrupos numéricos han encontrado aplicaciones notables en el ámbito informático y en áreas computacionales, aportando herramientas y enfoques teóricos que facilitan el desarrollo de algoritmos eficientes y la optimización de procesos. A continuación, se presenta un análisis detallado sobre su importancia en diversas áreas de la informática.

1.4.1. Codificación y teoría de la información

La estructura de los semigrupos numéricos permite la formulación de esquemas de codificación que pueden mejorar la compresión y el procesamiento de datos. Algunas de las aplicaciones específicas incluyen:

- **Compresión de datos:** Al aprovechar la representación única de enteros a partir de un conjunto finito de generadores, se pueden diseñar algoritmos que reduzcan la redundancia y optimicen el almacenamiento.
- **Construcción de códigos:** Los semigrupos ofrecen marcos teóricos para la creación de códigos numéricos que garanticen propiedades deseables como la detección y corrección de errores, mejorando la fiabilidad en la transmisión de información.

1.4.2. Optimización y algoritmos

En el campo de la optimización, la naturaleza discreta y la representación finita de los semigrupos numéricos permiten:

- **Programación lineal entera:** Los problemas de optimización que involucran variables enteras se benefician de la estructura de los semigrupos, permitiendo desarrollar métodos y algoritmos que explotan la finitud de los generadores para encontrar soluciones óptimas.
- **Diseño de algoritmos:** La capacidad de describir conjuntos de soluciones mediante combinaciones lineales facilita la construcción de algoritmos eficientes en tiempo y espacio, esenciales en la resolución de problemas complejos en ambientes computacionales.

1.4.3. Criptografía

La criptografía se apoya en problemas de optimización y estructuras algebraicas para asegurar la protección de la información:

- **Criptografía de clave pública:** Algunos esquemas criptográficos exploran la dificultad de ciertos problemas asociados a los semigrupos numéricos, tales como la representación mínima y la resolución de ecuaciones diofánticas, para construir sistemas seguros.
- **Seguridad basada en problemas discretos:** La naturaleza discreta y la complejidad combinatoria inherente a los semigrupos ofrecen un terreno fértil para desarrollar protocolos criptográficos robustos, en los que la dificultad computacional se traduce en mayor seguridad.

1.4.4. Lenguajes formales y teoría de autómatas

La teoría de autómatas y los lenguajes formales se benefician del estudio de semigrupos, en particular de los semigrupos finitos:

- **Modelos algebraicos de autómatas:** Los semigrupos finitos representan el comportamiento de ciertos tipos de autómatas, permitiendo la caracterización de la sintaxis y la semántica de lenguajes formales.

- **Optimización de compiladores:** La interpretación algebraica de las operaciones sobre cadenas y palabras facilita el diseño de algoritmos para el análisis léxico y sintáctico en compiladores, optimizando la interpretación de lenguajes de programación.

1.4.5. Análisis de complejidad

El estudio de la complejidad computacional se ve enriquecido por la aplicación de conceptos algebraicos:

- **Funciones de costo y estructuras algebraicas:** Los semigrupos numéricos proporcionan un marco para modelar y analizar funciones de costo en algoritmos, permitiendo establecer límites y estimar la eficiencia de procedimientos computacionales.
- **Evaluación de algoritmos:** La relación entre la representación de números y la eficiencia algorítmica ayuda a identificar puntos críticos en los procesos de cómputo, optimizando la toma de decisiones en problemas de gran escala.

1.4.6. Visión informática

La aplicación de los semigrupos numéricos en informática es un ejemplo paradigmático de cómo conceptos teóricos pueden traducirse en herramientas prácticas y efectivas. Desde la mejora en la compresión y codificación de datos hasta el desarrollo de algoritmos de optimización y sistemas criptográficos, la estructura algebraica de estos semigrupos permite abordar problemas complejos mediante enfoques discretos y combinatorios. Este intercambio entre teoría y práctica no solo refuerza la importancia de los semigrupos en la matemática pura, sino que también abre caminos para innovaciones en el ámbito computacional, consolidando su papel como un componente esencial en el diseño y análisis de sistemas informáticos modernos.

1.5. Ejemplo del problema de la mochila

El problema de la mochila (o *knapsack problem*) es un clásico en optimización y teoría de algoritmos, y presenta una interesante relación con los semigrupos numéricos. En este análisis detallado se expone cómo la estructura de un semigrupo numérico puede ser aplicada para comprender y resolver variantes del problema de la mochila, sobre todo en aquellas formulaciones donde se busca representar enteros como combinaciones de pesos o valores.

1.5.1. Contextualización del problema de la mochila

El problema de la mochila se formula generalmente de la siguiente manera: dada una mochila con capacidad limitada y un conjunto de objetos, cada

uno con un peso y un valor, se desea maximizar el valor total sin exceder la capacidad de la mochila. Una variante simplificada se centra en la **representación de enteros** como combinaciones de ciertos valores fijos. Aquí es donde los semigrupos numéricos entran en juego.

1.5.2. Relación con semigrupos numéricos

Consideremos un conjunto finito de números positivos $\{a_1, a_2, \dots, a_n\}$ que representan, por ejemplo, los pesos o valores de objetos disponibles. El conjunto de todos los enteros que se pueden obtener como combinación lineal no negativa de estos números forma un semigrupo numérico:

$$S = \{a_1x_1 + a_2x_2 + \dots + a_nx_n \mid x_i \in \mathbb{N}\}.$$

Esta construcción es fundamental para el problema de la mochila en la siguiente forma:

- **Representabilidad de sumas:** Determinar si es posible alcanzar un cierto entero b (por ejemplo, un peso o valor objetivo) equivale a verificar si b pertenece al semigrupo S . Si $b \in S$, existe una combinación de los objetos que suma b .
- **Número de Frobenius:** El mayor entero que no se puede obtener mediante estas combinaciones es conocido como el número de Frobenius $F(S)$. En el contexto de la mochila, $F(S)$ indica el límite máximo de “inaccesibilidad” en la representación de enteros.
- **Conductor:** Existe un entero mínimo $c(S)$ a partir del cual todos los números naturales son representables. Esto implica que, para capacidades mayores que $c(S)$, siempre se encontrará una solución al problema.

1.5.3. Análisis del ejemplo: mochila y semigrupo

Supongamos que tenemos una colección de objetos cuyos pesos son $\{3, 5, 7\}$. Estos pesos generan el semigrupo:

$$S = \{3x + 5y + 7z \mid x, y, z \in \mathbb{N}\}.$$

En este escenario, analizar el semigrupo permite:

1. **Verificar la representabilidad:** Se puede determinar si un peso objetivo, digamos 16, se puede formar como combinación de 3, 5 y 7. Por ejemplo, una solución es $16 = 3 \cdot 2 + 5 \cdot 2 + 7 \cdot 0$, lo que demuestra que $16 \in S$.

2. **Determinar el número de Frobenius:** El número de Frobenius $F(S)$ sería el mayor entero que no se puede obtener. Aunque el cálculo exacto puede requerir técnicas algorítmicas o teóricas, conocer $F(S)$ permite al diseñador del algoritmo identificar rangos donde la solución no es factible y ajustar la estrategia de búsqueda.
3. **Identificar el conductor:** Una vez superado el conductor $c(S)$, cualquier peso mayor a este valor es representable. Esto es especialmente útil en aplicaciones computacionales, pues para capacidades superiores al conductor se puede garantizar la existencia de una solución, lo que simplifica la búsqueda en algoritmos de optimización.

1.5.4. Implicaciones en la resolución del problema

El enfoque basado en semigrupos numéricos aporta varias ventajas prácticas al problema de la mochila:

- **Eficiencia en la búsqueda:** Conocer los invariantes del semigrupo (como $F(S)$ y $c(S)$) permite delimitar el espacio de búsqueda. Por ejemplo, si se sabe que todo entero mayor que $c(S)$ es representable, el algoritmo puede centrarse en casos críticos en el intervalo $[0, c(S)]$.
- **Algoritmos de decisión:** La representación de la mochila en términos de un semigrupo permite el uso de técnicas algebraicas para determinar la factibilidad de una suma, lo cual se traduce en algoritmos de decisión que pueden operar de manera más eficiente que métodos puramente combinatorios.
- **Optimización y heurísticas:** En aplicaciones de programación entera, el análisis del semigrupo asociado ayuda a construir heurísticas basadas en la estructura interna del problema, mejorando la aproximación a soluciones óptimas en tiempos razonables.

1.5.5. Conclusión

El ejemplo del problema de la mochila ilustra cómo los semigrupos numéricos no solo tienen una relevancia teórica, sino que también ofrecen herramientas prácticas para abordar problemas complejos de optimización. Al traducir el problema en términos de representabilidad de enteros, se pueden explotar invariantes fundamentales como el número de Frobenius y el conductor, facilitando el desarrollo de algoritmos eficientes y robustos. Este enfoque interdisciplinario resalta la capacidad de los semigrupos para conectar la teoría algebraica con aplicaciones prácticas en la informática, mostrando su potencial en el diseño y análisis de sistemas computacionales avanzados.

Capítulo 2

Semigrupos numéricos

Los semigrupos numéricos son un concepto relevante a la hora de su estudio en el área de las matemáticas. Estas estructuras, formadas por subconjuntos de los enteros no negativos cerrados bajo la suma, proporcionan un marco sencillo pero poderoso para analizar propiedades aritméticas y algebraicas profundas. Su estudio no solo arroja luz sobre patrones y relaciones intrínsecos en el conjunto de los enteros, sino que también facilita la formulación e investigación de invariantes fundamentales, como el número de Frobenius, el género, el conductor, la serie de Hilbert, el sistema minimal de generadores, los conjuntos de Apéry, entre otros.

Al profundizar en estos invariantes se abren caminos para resolver problemas complejos en álgebra conmutativa, teoría de números y geometría algebraica. En este sentido, los semigrupos numéricos se revelan como una herramienta clave para conectar teoría y aplicaciones, ofreciendo nuevos enfoques y perspectivas en el estudio de la estructura interna de los conjuntos numéricos.

A continuación se explicará en detalle las aplicaciones mostradas de los semigrupos numéricos en el área de las matemáticas, en donde se han realizado programas informáticos optimizados para el cálculo correcto de los diversos conceptos.

2.1. Número de Frobenius

Dado un conjunto de números enteros positivos primos relativos entre sí, $A = \{a_1, a_2, \dots, a_k\}$, el problema de Frobenius consiste en determinar cuál es el número más grande que no puede expresarse como una combinación lineal no negativa de elementos del conjunto. Este número, conocido como **número de Frobenius**, se denota comúnmente como $F(A)$ y cumple con la condición (suponiendo que el máximo común divisor de los elementos de A es 1):

$$F(A) = \max \left\{ n \in \mathbb{N} \mid \nexists (c_1, \dots, c_k) \in \mathbb{N}_0^k \text{ con } n = \sum_{i=1}^k c_i a_i \right\},$$

El número de Frobenius tiene una fórmula explícita en el caso en que el conjunto contiene solo dos elementos primos relativos, $A = \{a_1, a_2\}$, y está dado por:

$$F(A) = a_1 a_2 - a_1 - a_2.$$

Para conjuntos que contienen más de dos elementos ($k > 2$), el cálculo del número de Frobenius se vuelve más complejo y no existe una fórmula general explícita. En estos casos, se requiere del uso de algoritmos computacionales avanzados para encontrar $F(A)$, siendo un problema relevante en el campo de la optimización y la teoría de números.

El número de Frobenius es de gran importancia en la teoría de los **se-migrupos numéricos** al ser conjuntos de números enteros cerrados bajo la suma y cuyo complemento en los números naturales es finito. En este contexto, el número de Frobenius representa el mayor número que no pertenece al semigrupo y delimita el conjunto de huecos, definido como:

$$H(S) = \{n \in \mathbb{N} : n \notin S\}.$$

Además, el número de Frobenius está directamente relacionado con el género del semigrupo, que es la cantidad de huecos en el conjunto. Por ejemplo, dado un conjunto $A = \{3, 5\}$, el semigrupo generado por A incluye los números $0, 3, 5, 6, 8, 9, \dots$, pero el número de Frobenius es:

$$F(A) = 3 \cdot 5 - 3 - 5 = 7,$$

y los huecos serían $H(S) = \{1, 2, 4, 7\}$.

A lo largo del tiempo, el problema de Frobenius ha sido estudiado por matemáticos como James Joseph Sylvester y Ferdinand Georg Frobenius, y desde entonces ha encontrado aplicaciones en áreas como criptografía, optimización combinatoria y análisis algebraico. Su relevancia radica en su capacidad para modelar problemas donde se deben determinar límites o restricciones en sistemas de producción, distribución o empaquetamiento, entre otros.

2.1.1. Propiedades

El número de Frobenius posee propiedades interesantes que se pueden abordar desde diversas perspectivas, cada una aportando herramientas y conceptos que enriquecen su estudio. A continuación se presenta un análisis detallado y conciso de las principales aproximaciones:

1. Métodos algebraicos basados en la teoría de bases de Gröbner.

La utilización de bases de Gröbner en el estudio del número de Frobenius permite transformar el problema en uno algebraico, donde se consideran ideales en anillos de polinomios. Mediante este enfoque se pueden determinar relaciones entre los generadores del semigrupo numérico, lo que posibilita el análisis de las condiciones de representabilidad de enteros. La clave de este método se encuentra en la capacidad para eliminar variables y simplificar sistemas de ecuaciones polinómicas, permitiendo identificar estructuras y patrones algebraicos que subyacen en el problema. Esta técnica, aunque teóricamente potente, puede implicar una alta complejidad computacional para conjuntos con múltiples elementos, lo que limita su aplicabilidad práctica en ciertos casos.

2. Enfoque combinatorio basado en el conteo de representaciones.

Si se va al punto de vista combinatorio, el número de Frobenius se analiza en función de las formas en que un número puede representarse como suma de múltiplos de los elementos del conjunto A . Este método consiste en estudiar la cantidad y estructura de las representaciones posibles, lo que ayuda a identificar patrones y regularidades en la distribución de los números alcanzables. Al contar sistemáticamente las representaciones, se puede inferir el comportamiento asintótico del sistema y delimitar el máximo entero no representable. Esta aproximación es particularmente útil para obtener intuiciones sobre la densidad y la estructura de los semigrupos numéricos involucrados.

3. Métodos computacionales basados en algoritmos constructivos.

Los métodos computacionales están basados en la construcción continua del conjunto de números representables. Por tanto, existen algoritmos que generan de forma iterativa y acumulativa estos números, se logra identificar de manera directa el mayor entero que no puede expresarse como combinación lineal de los elementos de A . Este enfoque, a pesar de ser exhaustivo, se ve beneficiado por técnicas de optimización como la programación dinámica, que evitan hacer repetidamente cálculos redundantes de soluciones intermedias. El método computacional destaca por su capacidad de adaptarse a distintos tamaños de problemas, lo que proporciona un balance entre exactitud y eficiencia, aunque su coste computacional puede incrementarse en instancias de alta dimensión.

En conjunto, estas aproximaciones resaltan la riqueza y la complejidad del problema del número de Frobenius. Los métodos algebraicos ofrecen una

perspectiva estructural y teórica, la combinatoria permite una visión intuitiva basada en el conteo y la organización de las representaciones, y los métodos computacionales brindan herramientas prácticas para obtener soluciones exactas o aproximadas. Cada enfoque tiene sus ventajas y limitaciones, lo que ha motivado a la comunidad científica a desarrollar estrategias híbridas que combinen lo mejor de cada uno para enfrentar este desafío en la teoría de números y la optimización.

2.1.2. Ejemplos

Ejemplo 1: $A = \{5, 7\}$

Si $A = \{5, 7\}$, los números que se pueden formar con combinaciones lineales no negativas de 5 y 7 son:

$$0, 5, 7, 10, 12, 14, 15, 17, 19, 20, 21, \dots$$

El mayor número que no se puede generar es 23, ya que para 24 en adelante, todas las combinaciones son posibles. El número de Frobenius se calcula con la fórmula cerrada, válida cuando el conjunto tiene exactamente dos elementos a_1 y a_2 :

$$F(a_1, a_2) = a_1 a_2 - a_1 - a_2.$$

Aplicando la fórmula en este caso:

$$F(5, 7) = 5 \cdot 7 - 5 - 7 = 23.$$

Ejemplo 2: $A = \{3, 7, 9\}$

Para el conjunto $A = \{3, 7, 9\}$, no existe una fórmula cerrada para calcular el número de Frobenius. Por tanto, se determinará comprobando las posibles combinaciones. Una estrategia consiste en observar que, si a partir de cierto número se pueden generar consecutivamente tantos números como el mínimo elemento del conjunto, entonces todos los números posteriores pueden obtenerse mediante combinaciones lineales no negativas.

Teniendo lo anterior en cuenta, en este ejemplo vemos que el máximo elemento es 9 y el mínimo es 3. Por tanto, si en algún momento dado se pueden generar tantos elementos como el mínimo del conjunto A , entonces sabemos que a partir de ese punto se pueden generar todos los números usando los generadores del semigrupo numérico. Se puede ver que:

- 10 se puede generar como $7 + 3$.
- No existe combinación válida que genere 11.
- 12 se puede generar como $3 \cdot 4$.
- Además, se comprueba que $13 = 2 \cdot 3 + 7$ y $14 = 7 + 7$.

Dado que 11 es el mayor número que no se puede generar, concluimos que el número de Frobenius es:

$$F(\{3, 7, 9\}) = 11.$$

Esto ilustra que, para conjuntos con $k \geq 3$ elementos, el cálculo del número de Frobenius puede volverse computacionalmente complejo y, a menudo, requiere métodos o reglas muy concretas para optimizar los cálculos.

2.1.3. Bases de Gröbner

Una alternativa para determinar el **número de Frobenius** es el uso de las Bases de Gröbner, una herramienta poderosa en el álgebra computacional. Estas bases permiten analizar sistemas polinomiales e ideales asociados para verificar si ciertos números pueden ser generados mediante combinaciones de los elementos de un conjunto dado. En el caso del cálculo del número de Frobenius, las Bases de Gröbner se aplican de la siguiente manera:

- Se construye un ideal polinomial I que representa el conjunto de combinaciones posibles generadas por los elementos del conjunto $A = \{a_1, a_2, \dots, a_k\}$.
- Se calcula una base de Gröbner para este ideal, lo que facilita el análisis computacional del sistema.
- Se verifica si un número N pertenece o no al conjunto de números generables por A al determinar si x^N se anula en la forma normal respecto a la base de Gröbner.

Por ejemplo, dado el conjunto $A = \{5, 7, 9\}$, se puede modelar el sistema de combinaciones posibles mediante el ideal polinomial:

$$I = \langle y_1 - x^5, y_2 - x^7, y_3 - x^9 \rangle,$$

donde cada polinomio relaciona las variables generadoras con las potencias de x . El objetivo es analizar si el polinomio x^N pertenece al ideal I utilizando la base de Gröbner. Si la forma normal de x^N es igual a cero, significa que N puede ser generado mediante combinaciones de los elementos de A . Si no, entonces N es un hueco del semigrupo numérico.

Para calcular el **número de Frobenius**, se busca el mayor número entero que no puede ser generado por combinaciones lineales no negativas de los elementos de A . En el ejemplo anterior, al hacer uso de las Bases de Gröbner, se demuestra que el número $N = 13$ es el mayor número no generable. Por lo tanto, el número de Frobenius para este conjunto es 13.

La utilidad de las Bases de Gröbner en este contexto se basa en su capacidad para manejar eficientemente sistemas polinomiales complejos, dando un

enfoque algebraico para calcular el número de Frobenius. Esto es particularmente importante en casos donde el conjunto tiene más de dos elementos y el problema se vuelve más complejo de resolver con métodos o técnicas tradicionales. Además, esta herramienta conecta la teoría de semigrupos numéricos con el álgebra computacional, donde se pueden encontrar aplicaciones prácticas de optimización, criptografía y matemáticas discretas.

2.1.4. Algoritmos y Métodos

El cálculo del número de Frobenius se ha vuelto crucial en ciertos algoritmos matemáticos y métodos para mejorar el rendimiento de diversos programas. Recordemos que su finalidad es encontrar el mayor número que no se puede generar, es decir, aquel hueco mayor de un semigrupo numérico.

Existen diversos enfoques para abordar este problema:

1. **Búsqueda exhaustiva.**

Consiste en generar todas las combinaciones posibles de los números dados para determinar aquellos enteros que pueden ser representados además de identificar el mayor entero no representable. No obstante, este método conlleva una complejidad exponencial que lo hace inviable para conjuntos de tamaño moderado o para valores elevados.

2. **Programación dinámica.**

Este método permite la reutilización de soluciones a subproblemas, lo cual permite evitar cálculos redundantes. En el contexto del número de Frobenius, la programación dinámica se utiliza para construir de forma iterativa una tabla que marca los enteros alcanzables mediante combinaciones de los números dados. A partir de esta estructura, se puede deducir el número de Frobenius identificando el mayor entero no alcanzado. Aunque este enfoque puede demandar un uso considerable de memoria en casos complejos, resulta significativamente más eficiente que la búsqueda exhaustiva al reducir la cantidad de operaciones necesarias.

3. **Algoritmos heurísticos y aproximaciones.**

Dado que no se dispone de un algoritmo polinomial conocido para $k \geq 3$, se han desarrollado métodos heurísticos que, si bien no garantizan encontrar siempre el número exacto, permiten obtener estimaciones razonables en tiempos de cómputo aceptables. Estos métodos son especialmente útiles en aplicaciones donde una aproximación del número de Frobenius es suficiente para modelar o resolver problemas prácticos.

La importancia del número de Frobenius en el ámbito de los algoritmos y métodos radica en que representa un ejemplo paradigmático de problemas

combinatorios difíciles, ilustrando conceptos fundamentales de la teoría de la complejidad computacional, como la NP-dificultad. El desafío que supone hallar este número ha impulsado el desarrollo de técnicas de optimización y estrategias algorítmicas que tienen aplicaciones en diversas áreas de la matemática y la informática.

En este contexto, la programación dinámica se destaca por su capacidad para descomponer el problema en subproblemas más pequeños y manejables, almacenando los resultados intermedios para evitar cálculos repetitivos. Esta estrategia no solo reduce el tiempo de ejecución en comparación con los métodos de fuerza bruta, sino que también permite una mejor comprensión de la estructura interna del problema, facilitando la identificación de patrones y la posibilidad de futuras mejoras en los algoritmos. Por ello, la programación dinámica se presenta como una herramienta esencial para enfrentar el reto computacional que implica calcular el número de Frobenius, ofreciendo un equilibrio entre exactitud y eficiencia computacional.

2.1.5. Aplicaciones

El cálculo del número de Frobenius tiene numerosas aplicaciones en diversos campos, abarcando tanto áreas teóricas como prácticas. A continuación, se detallan algunas de las aplicaciones más relevantes:

- **Matemática Discreta y Combinatoria.**

El número de Frobenius desempeña un papel importante en problemas de conteo y partición de números. En este contexto, se utiliza para analizar las formas en que los enteros pueden representarse como combinaciones lineales de un conjunto dado. Este análisis permite explorar la estructura de los semigrupos numéricos, contribuyendo al estudio de particiones, combinaciones y otros aspectos fundamentales de la matemática discreta.

- **Optimización y Teoría de Juegos.**

En aplicaciones de optimización, el número de Frobenius se relaciona con la minimización de residuos y la maximización del uso eficiente de recursos. Por ejemplo, en problemas de asignación o planificación, se busca determinar configuraciones óptimas que reduzcan el desperdicio de materiales o tiempo. En la teoría de juegos, estos conceptos se trasladan a estrategias de decisión, donde identificar límites óptimos (como el máximo entero no representable) puede ser determinante para el desarrollo de tácticas competitivas y eficientes.

- **Teoría de la Moneda.**

Una de las aplicaciones más conocidas es el problema del “cambio de monedas”, en el que se intenta determinar el mayor monto que no puede obtenerse utilizando un conjunto fijo de denominaciones. Este

problema no solo sirve como ejercicio clásico en cursos de algoritmos y teoría de números, sino que también tiene implicaciones prácticas en el diseño de sistemas monetarios, en la optimización de transacciones y en la planificación financiera, donde la eficiencia en la combinación de denominaciones es crucial.

- **Criptografía y Seguridad Computacional.**

En el ámbito de la criptografía, ciertas variantes del problema del número de Frobenius están vinculadas con problemas complejos de aritmética modular y estructuras algebraicas subyacentes en los sistemas criptográficos. La dificultad inherente en la resolución de estos problemas puede ser aprovechada para diseñar algoritmos seguros, ya que la complejidad computacional asociada se traduce en resistencia ante ataques, lo que es fundamental para la protección de datos y la seguridad en comunicaciones.

En conjunto, estas aplicaciones demuestran que el estudio del número de Frobenius va más allá de un problema teórico. Su análisis y cálculo aportan herramientas esenciales para resolver problemas prácticos en optimización, teoría de juegos, finanzas y seguridad computacional. La transversalidad de este problema en diversas áreas resalta su importancia y motiva la búsqueda de métodos y algoritmos cada vez más eficientes para su resolución.

2.1.6. Programa de cálculo del número de Frobenius

Se ha creado un programa que determina el número de Frobenius dado un semigrupo numérico. Para ello, debemos de tener en cuenta todas las propiedades descritas anteriormente.

El usuario proporcionará un conjunto de números enteros los cuales deben de formar un semigrupo numérico (máximo común divisor = 1). Para ello se han creado las siguiente funciones:

- **mcd()**: método en donde se ha implementado el algoritmo de Euclides para calcular el máximo común divisor (MCD) de dos números.
- **reducemcd()**: método en donde se inserta el conjunto de elementos del semigrupo dado por el usuario. El parámetro que se le pasa a la función es un array de dichos elementos. Se va comprobando iterativamente el máximo común divisor combinando los elementos dos a dos.
- **nFrob()**: es el método donde se procederá el cálculo de número de Frobenius.

Solo se procederá a hacer el cálculo si todos los elementos del conjunto son números enteros y cumplen con las restricciones de presentar un máximo común divisor de 1. En el caso de que se introduzca el

elemento 1, el programa devolverá automáticamente el valor de -1. Recordemos que el 0 siempre está incluido en el semigrupo y dado que 1 puede generar todos los números a partir de combinaciones del mismo, el mayor número que no se puede generar es -1.

Se hace un preprocesamiento previo al cálculo, en donde se ordenará el conjunto de menor a mayor para facilitar el cálculo del número máximo y las combinaciones subsiguientes. Se determina una cota superior donde se calcula el producto de los dos números más grandes del conjunto para limitar el rango de búsqueda, dado que cualquier número mayor a este podrá ser generado como una combinación de los números más pequeños del conjunto. Además, se ha creado un array de booleanos el cual nos indicará si un número puede ser representado como combinación de los elementos del conjunto.

Para determinar cuál es el mayor número que no se puede generar, se ha seguido la filosofía de la programación dinámica para marcar aquellos números que sí son alcanzables. Se recorren todos los números desde 1 hasta el valor máximo, verificando si se pueden generar restando uno de los números del conjunto. Si se encuentra que un valor es alcanzable, se rompe el bucle interno para evitar cálculos innecesarios.

Una vez se marcan los números que sí se pueden generar se procede a buscar el mayor número que no se puede generar, recorriendo el array de forma inversa y devolviendo el primer número que no sea alcanzable.

```

1: procedure NFROB( $S$ )
2:   if  $1 \in S$  then
3:     return -1
4:   end if
5:   if  $|S| < 2$  or  $\text{mcd}(S) \neq 1$  then
6:     return "No se puede calcular"
7:   end if
8:   Ordenar  $S$  en orden creciente
9:    $\text{valorMax} \leftarrow S[|S| - 1] \times S[|S| - 2]$ 
10:  Crear arreglo booleano  $\text{alcanzable}[0 \dots \text{valorMax}]$  y asignar
    false a todas sus posiciones
11:   $\text{alcanzable}[0] \leftarrow \text{true}$ 
12:  for  $i \leftarrow 1$  to  $\text{valorMax}$  do
13:    for each  $a \in S$  do
14:      if  $(i - a) \geq 0$  and  $\text{alcanzable}[i - a] = \text{true}$  then
15:         $\text{alcanzable}[i] \leftarrow \text{true}$ 
16:        break ▷ Salir del bucle interno
17:      end if
18:    end for
19:  end for

```

```

20:   for  $i \leftarrow valorMax$  downto 0 do
21:       if  $alcanzable[i] = false$  then
22:           return  $i$ 
23:       end if
24:   end for
25:   return "No se puede calcular"
26: end procedure

```

El diseño de este algoritmo garantiza que sea eficiente, limitando el rango de búsqueda y evitando cálculos redundantes.

2.2. Género

Sea S un semigrupo numérico, es decir, un subconjunto de \mathbb{N}_0 (o \mathbb{N} , según la convención, en el que usualmente se incluye el 0) cerrado bajo suma y con complemento finito en \mathbb{N}_0 . El *género* de S , denotado por $g(S)$, se define como la cantidad de números naturales (o enteros no negativos) que no pertenecen a S . Formalmente, se tiene:

$$g(S) = |\mathbb{N}_0 \setminus S|.$$

Esta cantidad cuenta el número de *huecos* o *gaps* en S , es decir, los números que no se pueden expresar como combinaciones lineales no negativas de los generadores de S .

Aunque el género y el *número de Frobenius* son invariantes relacionados, no son lo mismo. El número de Frobenius $F(S)$ es el máximo entero que no pertenece a S (es decir, el mayor gap). En muchos estudios se observa una estrecha relación entre ambas cantidades, en especial en semigrupos monogéneros o en contextos donde se busca estimar uno a partir del otro. Sin embargo, es importante recalcar que:

- El género es el número total de huecos.
- El número de Frobenius es únicamente el mayor hueco.

El género es una invariante fundamental en la teoría de semigrupos numéricos, ya que permite clasificar y distinguir diferentes semigrupos en función de la “densidad” de sus elementos. Algunas propiedades y puntos de interés son:

- **Cota y crecimiento:** Dado un conjunto de generadores, existen fórmulas y cotas que relacionan la multiplicidad (el mínimo generador no nulo), el número de generadores y el género. Estas relaciones son objeto de estudio en problemas de optimización y en la teoría de códigos.

- **Conexión con aritmética algebraica:** En el estudio de curvas algebraicas, el género de un semigrupo numérico asociado a una curva (por ejemplo, en la clasificación de singularidades) tiene implicaciones en la geometría y la aritmética de la curva.
- **Ejemplo:** Un ejemplo clásico es el semigrupo generado por 3 y 5, $S = \langle 3, 5 \rangle$. En este caso, se puede demostrar que $g(S) = 4$ (ya que los huecos son $\{1, 2, 4, 7\}$) y $F(S) = 7$.

Se introducen además los *semigrupos numéricos equilibrados*. Estos son aquellos semigrupos cuyo género es par y en los que el número de huecos pares es igual al número de huecos impares. Esta simetría en la distribución de los gaps puede tener implicaciones en el estudio de las propiedades aritméticas y combinatorias del semigrupo, siendo de interés tanto teórico como en aplicaciones prácticas.

Se puede ver entonces que el género es una medida crucial de la complejidad de un semigrupo numérico, ya que indica cuántos enteros no pueden generarse a partir de los elementos del semigrupo. Su estudio, junto con otros invariantes como el número de Frobenius y la multiplicidad, permite una caracterización detallada de la estructura interna del semigrupo, facilitando tanto su clasificación como la exploración de aplicaciones en diversas áreas de las matemáticas.

2.2.1. Propiedades

El género $g(S)$ de un semigrupo numérico S posee diversas propiedades que permiten entender y clasificar la estructura de S . A continuación, se presenta un análisis detallado y conciso de sus propiedades:

- **Finitud y caracterización.**

Por definición, $S \subset \mathbb{N}_0$ tiene complemento finito, lo que implica que $g(S) = |\mathbb{N}_0 \setminus S|$ es un número natural finito. Además, se cumple que:

$$g(S) = 0 \iff S = \mathbb{N}_0.$$

- **Relación con el número de Frobenius.**

En ciertos semigrupos, especialmente en los simétricos¹ y pseudo-simétricos², existe una relación directa entre el género y el número de Frobenius $F(S)$. Para un semigrupo simétrico se tiene:

$$g(S) = \frac{F(S) + 1}{2},$$

¹un semigrupo numérico es simétrico si $F(S)$ es impar y además no se puede poner como intersección de dos semigrupos numéricos que lo contengan propiamente.

²un semigrupo numérico es pseudo-simétrico si $F(S)$ es par y no se puede poner como intersección de dos semigrupos numéricos que lo contengan propiamente.

mientras que en un semigrupo pseudo-simétrico se cumple:

$$g(S) = \frac{F(S) + 2}{2}.$$

Estas fórmulas reflejan la simetría en la distribución de los huecos respecto al mayor número ausente.

■ **Cotas y límites.**

Existen cotas que relacionan el género con otros invariantes del semigrupo, como la multiplicidad m (el menor elemento de $S \setminus \{0\}$). Por ejemplo, para un semigrupo no trivial se tiene la cota:

$$g(S) \geq m - 1.$$

Esta desigualdad ayuda a estimar la "densidad" de S en \mathbb{N}_0 .

■ **Relación con el conjunto de Apéry.**

Sea $\text{Ap}(S, m) = \{w \in S \mid w - m \notin S\}$ el conjunto de Apéry relativo a la multiplicidad m . Existe una fórmula que relaciona este conjunto con el género:

$$g(S) = \frac{1}{m} \left(\sum_{w \in \text{Ap}(S, m)} w \right) - \frac{m-1}{2}.$$

Este resultado no solo facilita el cálculo del género, sino que también resalta la importancia de la estructura modular del semigrupo.

■ **Clasificación y recursividad.**

El género es un parámetro esencial en la enumeración y clasificación de semigrupos numéricos. Algoritmos de backtracking y recorridos en árbol utilizan el género para generar recursivamente semigrupos con un número fijo de huecos, permitiendo identificar patrones y estructuras comunes entre ellos.

■ **Simplicidad en casos especiales.**

En semigrupos simétricos y pseudo-simétricos, la relación directa entre $g(S)$ y $F(S)$ simplifica su análisis y verificación. Esta propiedad resulta útil para identificar semigrupos con distribución de huecos simétrica, lo que tiene implicaciones en problemas de optimización y teoría de códigos.

El estudio de las propiedades del género da información sobre la completitud de un semigrupo numérico, además de conectar con varios campos de las matemáticas, como el álgebra conmutativa, la geometría algebraica y la combinatoria, permitiendo así una clasificación y análisis profundos de estas estructuras.

2.2.2. Semigrupos Equilibrados

Un semigrupo numérico se denomina de **equilibrado** si cumple con dos condiciones fundamentales:

1. **El género debe ser par:** esto significa que la cantidad total de huecos, que representan los números no generables mediante combinaciones lineales no negativas de los elementos del semigrupo, debe ser un número par.
2. **Igual cantidad de huecos pares que impares:** dentro de los huecos del semigrupo, la cantidad de números pares que no pertenecen al semigrupo debe ser igual a la cantidad de números impares que no pertenecen a él.

La propiedad de equilibrio de los semigrupos numéricos se utiliza para clasificar y estudiar su estructura, siendo un concepto clave en la teoría de semigrupos numéricos. El equilibrio del género asegura una simetría en la distribución de huecos, lo que a menudo está relacionado con propiedades algebraicas más profundas del semigrupo.

Por ejemplo, consideremos un semigrupo con género $g = 4$, donde los huecos son $H = \{2, 3, 6, 7\}$. En este caso, los huecos pares son $\{2, 6\}$ y los huecos impares son $\{3, 7\}$. Dado que la cantidad de huecos pares e impares es igual (dos de cada tipo) y el género es par, este semigrupo cumple con las condiciones de equilibrio y se clasifica como un semigrupo de género equilibrado.

El estudio de los semigrupos de género equilibrado tiene aplicaciones en áreas como combinatoria, álgebra y optimización, y conecta la teoría numérica con patrones de simetría y regularidad.

2.2.3. Ejemplos

Consideremos dos semigrupos numéricos generados por distintos conjuntos:

Ejemplo 1: $S = \langle 3, 5, 7 \rangle$

Los enteros que no se pueden obtener como combinaciones lineales no negativas de 3, 5 y 7 son:

$$\{1, 2, 4\}.$$

Como existen 3 elementos en este conjunto, se tiene:

$$g(S) = 3.$$

Al ser impar, el semigrupo no es equilibrado.

Ejemplo 2: $S = \langle 8, 9, 11, 13 \rangle$

Los enteros que no se pueden generar son:

$$\{1, 2, 3, 4, 5, 6, 7, 10, 12, 14, 15, 23\}.$$

Por lo tanto, el género es:

$$g(S) = 12.$$

Además, al dividir los huecos en pares e impares se obtiene que hay 6 de cada uno, lo que implica que el semigrupo es equilibrado.

2.2.4. Algoritmos y Métodos

Existen varios métodos y algoritmos para determinar el género de un semigrupo numérico, cada uno con sus ventajas según la complejidad del semigrupo y la información disponible. A continuación, se detallan los principales enfoques:

■ **Enumeración directa de huecos.**

Este método consiste en generar los elementos del semigrupo S hasta alcanzar el número de Frobenius (el mayor número ausente) y, posteriormente, identificar y contar los números que no pertenecen a S .

- **Ventaja:** es conceptualmente sencillo y se implementa de forma directa.
- **Desventaja:** puede resultar ineficiente para semigrupos con gran número de huecos o cuando el número de Frobenius es elevado, ya que se requiere verificar la pertenencia de muchos números a S .

■ **Método basado en el conjunto de Apéry.**

Dado un semigrupo numérico S y su *multiplicidad* m (el menor elemento de $S \setminus \{0\}$), se define el conjunto de Apéry de S respecto a m como:

$$\text{Ap}(S, m) = \{w \in S \mid w - m \notin S\}.$$

Se recuerda la fórmula:

$$g(S) = \frac{1}{m} \left(\sum_{w \in \text{Ap}(S, m)} w \right) - \frac{m-1}{2}.$$

- **Ventaja:** permite calcular el género de eficientemente dado a que el conjunto de Apéry contiene exactamente m elementos, lo que reduce significativamente la cantidad de cálculos.

- **Desventaja:** requiere el conocimiento o la computación previa de la multiplicidad m y del conjunto de Apéry, aunque esto suele ser factible para la mayoría de los semigrupos.
- **Algoritmos de recorrido y Backtracking.**
En estudios combinatorios, sobre todo cuando se desea enumerar todos los semigrupos numéricos de un cierto género, se emplean algoritmos de recorrido (o *backtracking*). Estos algoritmos exploran la estructura en forma de árbol de los semigrupos, utilizando invariantes como el género y el número de Frobenius para podar ramas y reducir la búsqueda.
 - **Ventaja:** son muy útiles en la clasificación y generación de semigrupos con propiedades particulares.
 - **Desventaja:** pueden ser computacionalmente intensivos para géneros elevados o cuando se busca una enumeración completa.
- **Métodos basados en funciones generadoras.**
Otra aproximación utiliza funciones generadoras (o series de Hilbert) asociadas al semigrupo. Estas funciones codifican la información combinatoria del semigrupo, y a través de técnicas analíticas es posible extraer el número de huecos, es decir, el género.
 - **Ventaja:** conecta la teoría de semigrupos con herramientas de álgebra conmutativa y análisis combinatorio.
 - **Desventaja:** su implementación requiere conocimientos avanzados y puede ser menos directa que los métodos anteriores.

En resumen, la elección del método para calcular el género de un semigrupo numérico dependerá de la naturaleza del semigrupo y de los recursos computacionales disponibles. El método basado en el conjunto de Apéry es especialmente popular por su eficiencia, mientras que la enumeración directa y los algoritmos de backtracking resultan útiles en contextos exploratorios y de clasificación.

2.2.5. Aplicaciones

Se ha visto que el estudio del género en semigrupos numéricos tiene varias aplicaciones en teoría y en problemas prácticos. Ahora, se pasa a detallar algunas de las áreas donde esta invariante desempeña un papel fundamental:

- **Clasificación y estructura de semigrupos.**
El género actúa como una medida de la “completitud” o “densidad” de un semigrupo. Al cuantificar los huecos en el conjunto, permite diferenciar semigrupos con estructuras similares y facilita la clasificación en familias según la complejidad de su complemento en \mathbb{N}_0 .

- **Geometría algebraica.**

La relación entre semigrupos numéricos y curvas algebraicas, en particular a través de los semigrupos de Weierstrass asociados a puntos en una curva, hace que el género sea esencial para entender la geometría de la curva. Este vínculo permite deducir propiedades de la curva (como su género en el sentido geométrico) a partir del análisis del semigrupo y viceversa.

- **Optimización y particiones de números:**

En problemas de optimización, especialmente aquellos relacionados con particiones y representaciones de números, el género ofrece una medida de cuán “completo” es un sistema de generadores. Esto se utiliza, por ejemplo, en algoritmos que buscan minimizar o maximizar ciertos parámetros en problemas de combinación y factorización numérica.

- **Aplicaciones en codificación:**

La teoría de códigos y la criptografía han encontrado en los semigrupos numéricos una herramienta útil, ya que la estructura de los huecos y la distribución del género pueden relacionarse con la eficiencia y la robustez de ciertos códigos. En este contexto, el género puede ser interpretado como un parámetro que afecta la corrección de errores y la capacidad de detección.

- **Estudios combinatorios:**

El análisis del género es central en problemas combinatorios relacionados con la generación de conjuntos numéricos y la estructura de sus complementos. La forma en que se distribuyen los huecos (por ejemplo, en semigrupos equilibrados) permite desarrollar teoremas de simetría y dualidad que tienen repercusiones en diversas ramas de la matemática discreta.

En síntesis, el género no solo es un indicador de la complejidad interna de un semigrupo numérico, sino que también actúa como puente entre diversas áreas matemáticas, enriqueciendo tanto la teoría como las aplicaciones prácticas en optimización, geometría, teoría de códigos y combinatoria.

2.2.6. Programa de cálculo de género de un semigrupo numérico

Para calcular el género correctamente será necesario tener en cuenta el máximo común divisor del semigrupo numérico dado ($\text{MCD} = 1$). Además, tenemos en cuenta el número de Frobenius ya que nos dará información de cual es el mayor número que no se puede generar, por lo que no será necesario tener en cuenta aquellos números que sean mayores a este. Estos métodos ya fueron explicados en la sección 2.1.6.

El usuario escribirá un conjunto de números enteros los cuales formarán un semigrupo numérico. Como consecuencia, el programa le mostrará el género del conjunto proporcionado. Se han implementado las siguientes funciones para llevar a cabo esta tarea.

- **formaExpandida()**: este método está basado en la serie de Hilbert expandida el cual se explicará mas detalladamente en la sección 2.4. Lo que hace es generar una lista de números que pueden ser creados como combinaciones lineales no negativas de los elementos en el conjunto. Esta lista estará limitada a un número de términos ordenados de forma ascendente. El objetivo de la función es comprobar aquellos números que se pueden generar a partir del semigrupo dado.

Debemos de tener en cuenta un conjunto que represente todos los números que pueden generarse a partir de los elementos del conjunto y el menor valor del conjunto para determinar un límite superior razonable en los cálculos.

Se ha creado un bucle que ejecuta un número elevado de iteraciones para comprobar las combinaciones posibles de los elementos del conjunto. Si la suma $t+a$ es menor o igual a la cota superior planteada, se agregará al conjunto nuevo *nuevosTerminos*. Finalmente, se añaden al conjunto *terms* para que almacene los valores que son posibles de generar.

Finalmente, debemos de convertir el conjunto *terms* en un array para posteriormente ordenarlos, devolviendo sólo la cantidad concreta que se ha especificado en *numTerms*.

- **calcularGeneroYHuecos()**: se tiene como objetivo el cálculo de 3 valores fundamentales para mostrar correctamente el género del semigrupo. Estos son el género (número de enteros no representables con los elementos del conjunto), la lista de huecos (números enteros que no pueden generarse a partir de combinaciones lineales no negativas de los elementos del conjunto) y el número de Frobenius (número entero más grande que no puede ser generado del conjunto).

Lo primero de todo es calcular la forma expandida de la serie de Hilbert para obtener un conjunto ampliado de los números que pueden generarse a partir del semigrupo.

Luego, determinaremos el número de Frobenius para comprobar cual es el mayor número no generable. De esta manera tenemos en cuenta que a partir de dicho número, el resto pueden generarse.

Con la serie expandida de Hilbert se podrá determinar qué números enteros no pueden ser generados a partir de los elementos del conjunto. Para ello, si un número no se encuentra en la serie expandida de Hilbert, entonces se considerará hueco.

Para calcular el género, tendremos en cuenta el número total de huecos que se han encontrado.

```

1: function CALCULARGENEROYHUECOS( $S$ )
2:    $hilbertExpandida \leftarrow \text{formaExpandida}(S)$ 
3:    $numFrobenius \leftarrow nFrob(S)$ 
4:    $maxNumber \leftarrow \text{máx}(hilbertExpandida)$ 
5:    $huecos \leftarrow \emptyset$ 
6:   for  $i \leftarrow 0$  to  $maxNumber$  do
7:     if  $i \notin hilbertExpandida$  then
8:        $huecos \leftarrow huecos \cup \{i\}$ 
9:     end if
10:  end for
11:   $genero \leftarrow \text{card}(huecos)$ 
12:  return  $\{genero, huecos, numFrobenius\}$ 
13: end function

```

- **verificarEquilibrioSemigrupo()**: con esta función se quiere determinar si un semigrupo numérico es equilibrado o no. Para ello, evaluaremos las propiedades del género, huecos y el número de Frobenius del conjunto para comprobar si cumple las condiciones de equilibrio. Recordemos que para que un semigrupo numérico sea equilibrado debe de haber la misma cantidad de huecos con números pares que impares.

En el caso de que el género sea impar, entonces, directamente el semigrupo no podrá ser equilibrado.

En el caso de que el género sea par, se tendrá que evaluar si hay la misma cantidad de huecos con números pares que impares. Si esto es así, entonces el género es equilibrado.

2.3. Conductor

El *conductor* de un semigrupo numérico es un concepto fundamental en la teoría de semigrupos numéricos y tiene importantes implicaciones en la estructura y propiedades de estos conjuntos. Sea $S \subset \mathbb{N}$ (donde \mathbb{N} incluye al 0) un semigrupo numérico, es decir, un conjunto cerrado bajo la suma y con complemento finito en \mathbb{N} . El conductor de S , denotado por $c(S)$ o simplemente c , se define formalmente como

$$c = \min\{n \in \mathbb{N} \mid \forall m \geq n, m \in S\}.$$

Este número representa el menor entero a partir del cual todos los números naturales son elementos del semigrupo S .

2.3.1. Propiedades

El conductor de un semigrupo numérico posee una serie de propiedades fundamentales que facilitan el análisis y la aplicación de estos semigrupos en diversos contextos. A continuación, se presenta un análisis detallado de sus propiedades:

- **Umbral de inclusión.** El conductor, denotado generalmente como c , es el entero mínimo a partir del cual todos los números naturales pertenecen al semigrupo. Esto significa que para todo $n \geq c$, se tiene $n \in S$. Esta propiedad es esencial, ya que delimita el intervalo en el que se deben buscar los “huecos” o elementos ausentes del semigrupo.
- **Relación con el número de Frobenius.** El número de Frobenius es el mayor entero que no se encuentra en el semigrupo, y se ubica estrictamente por debajo del conductor. Es decir, si g es el número de Frobenius, entonces se cumple $g < c$. Esta relación es crucial en algoritmos que buscan optimizar el cálculo del número de Frobenius.
- **Estabilidad de la estructura.** Una vez alcanzado el conductor, la estructura del semigrupo se estabiliza, ya que la inclusión de todos los enteros a partir de c implica que las operaciones de suma o combinaciones lineales no generan nuevos “huecos”. Esto permite inferir propiedades sobre la densidad y la completitud del semigrupo en el intervalo $[c, \infty)$.
- **Utilidad en algoritmos de enumeración.** El conocimiento del conductor facilita la implementación de algoritmos de enumeración de semigrupos, ya que permite definir límites precisos en el proceso de generación y poda de ramas en estructuras jerárquicas. Al establecer c como punto de corte, se evitan cálculos innecesarios en la porción del semigrupo que ya se conoce por completo.
- **Conexión con invariantes del semigrupo.** El conductor se utiliza para determinar otros invariantes importantes, como la multiplicidad y el tipo del semigrupo. La existencia de un umbral garantiza que dichos invariantes puedan ser calculados de forma local, concentrándose en el intervalo previo a c donde se presentan los huecos, lo que mejora la eficiencia de los métodos de cómputo.
- **Simplificación del análisis combinatorio.** Dado que el comportamiento del semigrupo se vuelve predecible a partir del conductor, se simplifica el análisis combinatorio de sus elementos. Esto permite aplicar técnicas de optimización y descomposición en subproblemas, facilitando la búsqueda de patrones y la implementación de métodos heurísticos en problemas complejos.

2.3.2. Ejemplos

Consideremos dos semigrupos numéricos generados por distintos conjuntos, y analicemos el comportamiento del **conductor** en cada caso.

Ejemplo 1: $S = \langle 3, 5 \rangle$

Los enteros que no se pueden obtener como combinaciones lineales no negativas de 3 y 5 son:

$$\{1, 2, 4, 7\}.$$

El **conductor** es el menor entero c tal que para todo $n \geq c$ se tiene $n \in S$. En este semigrupo, se observa que a partir de 8 todos los enteros son alcanzables, por lo que:

$$c(S) = 8.$$

Ejemplo 2: $S = \langle 3, 5, 7 \rangle$

Los enteros que no se pueden generar como combinaciones lineales no negativas de 3, 5 y 7 son:

$$\{1, 2, 4\}.$$

Analizando la secuencia se determina que a partir de 5 se tiene una secuencia continua de enteros generables, es decir:

$$c(S) = 5.$$

2.3.3. Algoritmos y Métodos

El uso del conductor en semigrupos numéricos permite delimitar el umbral a partir del cual todos los enteros forman parte del semigrupo, lo que resulta esencial para optimizar la resolución de diversos problemas algorítmicos. Los enfoques que incorporan este concepto son los siguientes:

1. **Búsqueda exhaustiva.**

Se genera de forma sistemática el conjunto de enteros hasta alcanzar el valor del conductor. Este método permite identificar de manera directa los huecos y la estructura del semigrupo, ya que, a partir del conductor, se garantiza la inclusión de todos los números. Sin embargo, su complejidad es elevada, pues implica examinar exhaustivamente todas las combinaciones en el intervalo crítico, lo cual puede resultar inviable para semigrupos con parámetros de gran magnitud.

2. **Programación dinámica.**

La programación dinámica utiliza la propiedad del conductor para construir de forma iterativa una tabla que registra los enteros alcanzables. Al conocer el límite a partir del cual la pertenencia es automática,

es posible evitar cálculos redundantes y focalizar los esfuerzos computacionales en el análisis de los enteros anteriores a dicho umbral. Esta estrategia reduce significativamente la complejidad computacional y el consumo de recursos, permitiendo resolver problemas de mayor escala.

3. Algoritmos heurísticos y aproximaciones.

Dado que la determinación exacta de ciertos invariantes de un semigrupo puede ser computacionalmente costosa, se han desarrollado métodos heurísticos que se apoyan en el conocimiento del conductor para estimar estos parámetros. Estos enfoques, aunque no siempre garantizan la exactitud total, ofrecen resultados razonablemente precisos en tiempos de cómputo aceptables, facilitando la aplicación práctica en contextos donde una aproximación es suficiente para el modelado o la resolución de problemas.

La integración del conductor en estos algoritmos y métodos destaca su importancia en el estudio de semigrupos numéricos, pues permite delimitar el espacio de búsqueda y simplificar el análisis de su estructura. La programación dinámica, en particular, se beneficia de esta propiedad al descomponer el problema en subproblemas más manejables y evitar la evaluación innecesaria de combinaciones, lo que contribuye a un equilibrio adecuado entre exactitud y eficiencia computacional.

2.3.4. Aplicaciones

El conductor de un semigrupo numérico desempeña un papel crucial en diversas áreas de las matemáticas y sus aplicaciones. Las aplicaciones que más destacan se van a detallar a continuación.

El primer ámbito en el que el conductor presenta relevancia es en la **teoría de números**. En este contexto, se relaciona directamente con el problema de Frobenius, que consiste en determinar el mayor número entero que no puede representarse como combinación lineal no negativa de ciertos enteros. Para semigrupos generados por dos números coprimos, existe una fórmula explícita que involucra el conductor, lo cual permite establecer límites precisos en problemas de representabilidad y, por ende, contribuye a una mejor comprensión de la distribución de los números representables.

En el campo de la **geometría algebraica**, los semigrupos numéricos surgen al estudiar las singularidades de curvas algebraicas. El conductor se relaciona con la desingularización de estas curvas, ya que puede interpretarse como un indicador de la “completitud” de la curva en términos de sus invariantes. Concretamente, el valor del conductor se conecta con el número de Milnor y otros invariantes que caracterizan la complejidad de una singularidad, permitiendo clasificar y entender mejor la estructura local y global de

las curvas.

Otra aplicación importante se encuentra en el área de la **teoría de códigos y la criptografía**. La estructura de los semigrupos numéricos y, en particular, el conocimiento del conductor, es aprovechado en algoritmos que buscan optimizar la generación de códigos con ciertas propiedades. La capacidad para calcular el conductor de manera eficiente se traduce en mejoras en los métodos de codificación y en la optimización de algoritmos relacionados con la detección y corrección de errores.

Por último, el estudio del conductor tiene implicaciones en **problemas computacionales y de optimización combinatoria**. El hecho de que el semigrupo se establezca a partir del conductor hace que el desarrollo de algoritmos reduzcan la complejidad computacional al limitar la búsqueda a un conjunto finito de valores (los huecos), facilitando la resolución de problemas en teoría de grafos, programación y otros campos donde se manejan combinaciones lineales de números enteros.

Se puede ver entonces que el análisis del conductor no solo enriquece la teoría subyacente de los semigrupos numéricos, sino que también tiene un impacto directo en aplicaciones prácticas que van desde la resolución de problemas clásicos en teoría de números hasta el diseño de algoritmos en la informática y la optimización en diversas áreas científicas.

2.3.5. Programa de cálculo del género de un semigrupo numérico

Para calcular el conductor de un semigrupo numérico correctamente será necesario tener en cuenta el máximo común divisor del semigrupo numérico dado ($\text{MCD} = 1$). Además, tenemos en cuenta el número de Frobenius ya que nos dará información de cual es el mayor número que no se puede generar y, por tanto, a partir del siguiente valor todos los elementos podrán ser generados a partir del semigrupo numérico proporcionado. Estos métodos ya fueron explicados en la sección 2.1.6.

Para que el cálculo sea computacionalmente más eficiente, lo único que hará este programa es calcular el número de Frobenius y sumarle 1 lo cual determina el valor del conductor.

```

1: numFrob  $\leftarrow$  nFrob(numeros)
2: if numFrob = "No se puede calcular" then
3:   resultadoConductor.textContent  $\leftarrow$  "No se puede calcular"
4: else
5:   conductor  $\leftarrow$  numFrob + 1
6:   resultadoConductor.textContent  $\leftarrow$  conductor

```


7: end if

2.4. Serie de Hilbert

Sea S un semigrupo numérico, es decir, un subconjunto de \mathbb{N}_0 cerrado bajo suma y generado por un conjunto finito $A = \{a_1, a_2, \dots, a_k\}$. La *serie de Hilbert* de S , denotada por $H_S(t)$, es una herramienta algebraica que describe el crecimiento de los elementos de S mediante una serie formal. Formalmente, se define como

$$H_S(t) = \sum_{s \in S} t^s.$$

La serie de Hilbert presenta propiedades fundamentales que la hace bastante útil en álgebra conmutativa y en la teoría de invariantes. Entre ellas, destacan:

- **Forma racional:** Si S es finitamente generado, la serie de Hilbert puede expresarse como una fracción racional. Una de las representaciones clásicas es:

$$H_S(t) = \frac{P(t)}{\prod_{i=1}^k (1 - t^{a_i})},$$

donde $P(t)$ es un polinomio de coeficientes enteros que recoge la información de los huecos y de la estructura interna del semigrupo. Por otra parte, utilizando las propiedades particulares de los semigrupos numéricos, se puede obtener una forma cerrada alternativa basada en el conjunto de Apéry: sea $m = \min(S \setminus \{0\})$ la multiplicidad y

$$\text{Ap}(S, m) = \{w_0, w_1, \dots, w_{m-1}\} \quad \text{con} \quad w_r = \min\{s \in S \mid s \equiv r \pmod{m}\},$$

entonces se tiene

$$H_S(t) = \frac{\sum_{r=0}^{m-1} t^{w_r}}{1 - t^m}.$$

Esta última forma es válida para *cualquier* semigrupo numérico y permite expresar la serie de Hilbert en una forma cerrada en función de la multiplicidad y de los elementos mínimos en cada clase residual.

Las dos formas que se han visto son equivalentes y cada una resulta útil en varios contextos: la primera está conectada con técnicas de álgebra computacional y la segunda es especialmente práctica para el estudio de la estructura modular y los huecos del semigrupo.

- **Relación con el número de Frobenius:** En el caso particular de semigrupos generados por dos números coprimos, la forma cerrada de la serie de Hilbert permite identificar el número de Frobenius $F(S)$ (el mayor entero que no pertenece a S). Aunque en general el polinomio $P(t)$ depende de la estructura de los huecos, en el caso simétrico suele simplificarse a $P(t) = 1 - x^c$ con $c = F(S) + 1$. Para semigrupos no simétricos, sin embargo, la representación mediante el conjunto de Apéry es más precisa y general.

La serie de Hilbert es una herramienta poderosa para estudiar la distribución y el crecimiento de los elementos de un semigrupo numérico. Su representación racional ya sea en la forma que involucra el producto sobre los generadores o en la obtenida mediante el conjunto de Apéry y su conexión con invariantes algebraicos la convierten en un objeto de gran interés tanto en el análisis teórico como en aplicaciones prácticas en matemáticas.

2.4.1. Propiedades

La serie de Hilbert de un semigrupo numérico posee propiedades fundamentales que facilitan tanto el análisis combinatorio como la aplicación de métodos algebraicos para estudiar la estructura de S . Entre ellas se destacan:

- **Función generatriz.**

La serie de Hilbert

$$H_S(t) = \sum_{s \in S} t^s = \sum_{n \geq 0} \gamma(n) t^n,$$

actúa como función generatriz de S , donde $\gamma(n)$ indica la cardinalidad del conjunto de elementos de S en n . Esto traduce problemas de enumeración en un análisis de series formales y permite aplicar técnicas combinatorias y algebraicas al estudio del crecimiento y la estructura interna del semigrupo.

- **Conexión con invariantes algebraicos.**

La serie de Hilbert cuantifica el crecimiento de S y se relaciona estrechamente con invariantes del anillo de semigrupo $K[S]$. Por ejemplo, en álgebra conmutativa, $H_S(t)$ coincide con la serie de Hilbert del anillo $K[S]$, dando información sobre la multiplicidad, la dimensión, el tipo

y la regularidad del espacio graduado, lo que permite el uso de técnicas de homología y teoría de invariantes en el análisis combinatorio y algebraico del semigrupo.

■ **Estabilidad y eventual periodicidad.**

Dado que los semigrupos numéricos son co-finitos en \mathbb{N}_0 , existe un conductor a partir del cual todos los enteros pertenecen a S . Esta propiedad se refleja en $H_S(t)$, ya que para grados suficientemente altos los coeficientes se estabilizan y presentan una periodicidad eventual. Esta estabilidad es clave para inferir el comportamiento asintótico de S y simplificar tanto análisis teóricos como cálculos computacionales.

■ **Utilidad en algoritmos enumerativos.**

Al representar el recuento de elementos mediante una función generatriz, se pueden desarrollar algoritmos eficientes para enumerar los elementos del semigrupo. Tanto la forma expandida como la cerrada permiten detectar patrones en la secuencia de coeficientes, lo que resulta útil para optimizar cálculos de invariantes (como el número de Frobenius) y para implementar métodos de búsqueda y poda en algoritmos combinatorios.

■ **Aplicabilidad transversal.**

Las propiedades formales de $H_S(t)$ trascienden el estudio de semigrupos numéricos y se extienden a áreas tales como la geometría algebraica, la teoría de invariantes y el análisis de anillos graduados. Esta transversalidad posibilita la transferencia de técnicas y resultados entre distintos campos, enriqueciendo el marco teórico y proporcionando herramientas poderosas para abordar problemas complejos en diversas áreas de las matemáticas.

Podemos ver que estas propiedades evidencian la relevancia de la serie de Hilbert como herramienta fundamental para el análisis del crecimiento, la estructura y la complejidad de los semigrupos numéricos, permitiendo abordar problemas combinatorios y algebraicos con bases teóricas sólidas y métodos computacionales eficientes.

2.4.2. Ejemplos

Ejemplo 1: $S = \langle 5, 7 \rangle$

Si $S = \langle 5, 7 \rangle$, los números que se pueden formar con combinaciones lineales no negativas de 5 y 7 son:

$$0, 5, 7, 10, 12, 14, 15, 17, 19, 20, 21, \dots$$

El mayor número que no se puede generar es 23; es decir, para 24 en adelante todas las combinaciones son posibles. El número de Frobenius se calcula mediante la fórmula:

$$F(5, 7) = 5 \cdot 7 - 5 - 7 = 23.$$

La serie de Hilbert del semigrupo $S = \langle 5, 7 \rangle$ se expresa de forma expandida como:

$$H_S(t) = 1 + t^5 + t^7 + t^{10} + t^{12} + t^{14} + t^{15} + t^{17} + \dots$$

Y para obtener su forma cerrada usamos el conjunto de Apéry. Sea $m = \min(S \setminus \{0\}) = 5$. Definimos el conjunto

$$\text{Ap}(S, 5) = \{w_0, w_1, w_2, w_3, w_4\},$$

donde

$$\begin{aligned} w_0 &= \min\{s \in S \mid s \equiv 0 \pmod{5}\} = 0, \\ w_1 &= \min\{s \in S \mid s \equiv 1 \pmod{5}\} = 21, \\ w_2 &= \min\{s \in S \mid s \equiv 2 \pmod{5}\} = 7, \\ w_3 &= \min\{s \in S \mid s \equiv 3 \pmod{5}\} = 28, \\ w_4 &= \min\{s \in S \mid s \equiv 4 \pmod{5}\} = 14. \end{aligned}$$

Con ello, la forma cerrada queda:

$$H_S(t) = \frac{t^0 + t^{21} + t^7 + t^{28} + t^{14}}{1 - t^5}.$$

Ejemplo 2: $S = \langle 3, 7, 9 \rangle$

Para el semigrupo $S = \langle 3, 7, 9 \rangle$ no existe una fórmula elemental para el número de Frobenius; utilizando un algoritmo se obtiene:

$$F(\{3, 7, 9\}) = 11.$$

La serie de Hilbert expandida es:

$$H_S(t) = 1 + t^3 + t^6 + t^7 + t^9 + t^{10} + t^{12} + t^{13} + \dots$$

Con $m = \min(S \setminus \{0\}) = 3$, definimos

$$\text{Ap}(S, 3) = \{w_0, w_1, w_2\},$$

donde

$$\begin{aligned}w_0 &= \min\{s \in S \mid s \equiv 0 \pmod{3}\} = 0, \\w_1 &= \min\{s \in S \mid s \equiv 1 \pmod{3}\} = 7, \\w_2 &= \min\{s \in S \mid s \equiv 2 \pmod{3}\} = 14.\end{aligned}$$

Por lo tanto, la forma cerrada es:

$$H_S(t) = \frac{t^0 + t^7 + t^{14}}{1 - t^3}.$$

Ejemplo 3: $S = \langle 5, 7, 9 \rangle$

Para $S = \langle 5, 7, 9 \rangle$, se obtiene:

$$F(\{5, 7, 9\}) = 13.$$

La serie de Hilbert expandida es:

$$H_S(t) = 1 + t^5 + t^7 + t^9 + t^{10} + t^{12} + t^{14} + t^{15} + \dots$$

Con $m = \min(S \setminus \{0\}) = 5$, definimos

$$\text{Ap}(S, 5) = \{w_0, w_1, w_2, w_3, w_4\},$$

donde

$$\begin{aligned}w_0 &= \min\{s \in S \mid s \equiv 0 \pmod{5}\} = 0, \\w_1 &= \min\{s \in S \mid s \equiv 1 \pmod{5}\} = 16, \\w_2 &= \min\{s \in S \mid s \equiv 2 \pmod{5}\} = 7, \\w_3 &= \min\{s \in S \mid s \equiv 3 \pmod{5}\} = 18, \\w_4 &= \min\{s \in S \mid s \equiv 4 \pmod{5}\} = 9.\end{aligned}$$

La forma cerrada es:

$$H_S(t) = \frac{t^0 + t^{16} + t^7 + t^{18} + t^9}{1 - t^5}.$$

2.4.3. Algoritmos y Métodos

El cálculo de la serie de Hilbert es fundamental en álgebra conmutativa y en el estudio de semigrupos numéricos, pues proporciona una visión compacta sobre el crecimiento y la distribución de sus elementos. La serie de Hilbert de un semigrupo finitamente generado resulta ser una función racional, lo que permite representar de forma eficiente la sucesión de coeficientes que indican la presencia de elementos en cada grado.

Existen diversos enfoques para construir algoritmos que aprovechan esta serie, entre los que destacan:

1. Método de la forma expandida.

Este enfoque consiste en generar explícitamente todos los elementos del semigrupo hasta alcanzar cierto límite —usualmente relacionado con el conductor (el umbral a partir del cual todos los enteros son alcanzables)— mediante la exploración de combinaciones lineales no negativas de los generadores. Se utiliza, por ejemplo, un algoritmo basado en búsqueda en anchura (BFS) para recorrer las combinaciones. Así se obtiene la representación formal:

$$H_S(t) = \sum_{s \in S} t^s.$$

Este método es muy intuitivo, ya que se aprecia directamente la aparición de cada término; sin embargo, su complejidad crece notablemente para semigrupos con altos conductores o con muchos generadores, dado que implica examinar exhaustivamente muchas combinaciones posibles.

2. Método de la forma cerrada.

Apoyándose en resultados teóricos, se sabe que la serie de Hilbert de un semigrupo finitamente generado es una función racional. Tradicionalmente, para un conjunto de generadores

$$S = \{a_1, a_2, \dots, a_k\},$$

se escribe:

$$H_S(t) = \frac{P(t)}{\prod_{i=1}^k (1 - t^{a_i})},$$

donde $P(t)$ es un polinomio de coeficientes enteros que recoge la información de los huecos y de la estructura interna del semigrupo. Sin embargo, para poder abordar la generalidad de los semigrupos numéricos (no sólo los simétricos), se suele utilizar la *forma cerrada mediante el conjunto de Apéry*. Sea $m = \min(S \setminus \{0\})$ la multiplicidad y definamos

$$\text{Ap}(S, m) = \{w_0, w_1, \dots, w_{m-1}\}, \quad \text{donde} \quad w_r = \min\{s \in S : s \equiv r \pmod{m}\}.$$

Entonces se tiene la representación:

$$H_S(t) = \frac{\sum_{r=0}^{m-1} t^{w_r}}{1 - t^m}.$$

Este enfoque es especialmente útil porque es válido para *cualquier* semigrupo numérico y conecta directamente con invariantes algebraicos importantes, aunque en algunos casos la determinación del conjunto de Apéry pueda implicar un análisis computacional cuidadoso.

3. Enfoques heurísticos y aproximaciones.

Debido a que el cálculo exacto de la serie de Hilbert en ciertos semigrupos puede ser computacionalmente intensivo, se han desarrollado algoritmos heurísticos que combinan búsqueda exhaustiva con estrategias de programación dinámica. Típicamente, estos métodos:

- Emplean una estructura (como un arreglo booleano o una tabla) para marcar los grados alcanzables mediante combinaciones de los generadores.
- Aprovechan la propiedad del conductor para delimitar el rango en el que es necesario calcular de forma precisa los coeficientes, y a partir de ahí extrapolan o aproximan el comportamiento de la serie.

Aunque la precisión exacta de estos métodos puede no ser garantizada en todos los casos, son de gran utilidad en aplicaciones prácticas (por ejemplo, en optimización de problemas en teoría de códigos o en el análisis de espacios graduados) donde se requiere una solución en tiempos razonables.

La integración de estos métodos subraya la versatilidad del concepto de la serie de Hilbert. Por un lado, la representación expandida permite visualizar de manera directa la estructura del semigrupo; por otro, la forma cerrada (especialmente la obtenida mediante el conjunto de Apéry) conecta con teorías algebraicas profundas a través de una expresión racional. Además, los enfoques heurísticos potencian el cálculo para semigrupos complejos o de gran escala, facilitando tanto el análisis teórico como las aplicaciones prácticas.

En conjunto, estos algoritmos y técnicas no solo robustecen el análisis de espacios graduados y semigrupos numéricos, sino que también impulsan aplicaciones en diversas áreas de las matemáticas, demostrando la relevancia de la serie de Hilbert como herramienta fundamental en el estudio del crecimiento y la estructura algebraica.

2.4.4. Aplicaciones

La serie de Hilbert asociada a un semigrupo numérico desempeña un papel crucial en diversas áreas de las matemáticas, al ofrecer una ventana compacta sobre el crecimiento y la distribución de los elementos generados. A continuación, se exponen algunas de las aplicaciones más destacadas:

■ **Teoría de números.**

En este ámbito, la función generatriz que representa la serie de Hilbert permite analizar el comportamiento asintótico y la distribución de enteros obtenidos mediante combinaciones lineales no negativas. Esto se vincula fuertemente con problemas clásicos como el del número de Frobenius, puesto que la estructura de la serie (ya sea en su forma expandida o a través de su representación cerrada mediante el conjunto de Apéry) ayuda a identificar patrones, regularidades y huecos en los números representables.

■ **Geometría algebraica.**

La serie de Hilbert es indispensable en el estudio de anillos graduados y variedades proyectivas. La representación racional

$$H_S(t) = \frac{P(t)}{\prod_{i=1}^k (1 - t^{a_i})},$$

donde $P(t)$ codifica la información relativa a los huecos y la estructura interna del semigrupo, permite determinar invariantes esenciales como la dimensión, el grado y la regularidad de objetos algebraicos. En particular, la estabilización asintótica de la serie conduce al *polinomio de Hilbert*, herramienta fundamental en la clasificación de curvas, superficies y otras variedades.

■ **Teoría de códigos y criptografía.**

La capacidad de la serie de Hilbert para condensar información combinatoria y algebraica se aprovecha en el análisis de anillos y módulos, lo cual tiene aplicaciones directas en el diseño de códigos de corrección de errores y en la seguridad de sistemas criptográficos basados en estructuras algebraicas. El conocimiento de la distribución de los coeficientes e invariantes derivado de la serie facilita la optimización de parámetros críticos en estos sistemas.

■ **Problemas computacionales y optimización Combinatoria.**

La representación en forma cerrada, especialmente la obtenida mediante el conjunto de Apéry,

$$H_S(t) = \frac{\sum_{r=0}^{m-1} t^{w_r}}{1 - t^m},$$

junto con la eventual periodicidad de sus coeficientes, permite desarrollar algoritmos eficientes para la enumeración de elementos y para

la optimización en espacios de soluciones. Este enfoque resulta particularmente útil en problemas de programación entera, teoría de grafos y otros ámbitos donde se analizan combinaciones lineales de enteros, reduciendo significativamente la complejidad computacional.

En consecuencia, la serie de Hilbert no solo enriquece la teoría de semigrupos y anillos graduados con una herramienta analítica potente, sino que también impulsa aplicaciones prácticas en áreas tan diversas como la teoría de números, la geometría algebraica, la teoría de códigos y la optimización computacional.

2.4.5. Programa de cálculo de la serie de Hilbert

Para calcular la serie expandida y cerrada de Hilbert debemos de tener en cuenta que el conjunto sea un semigrupo numérico, por lo que será necesario tener en cuenta el máximo común divisor de dicho semigrupo para saber si es 1. Para ello, se usan los algoritmos `mcd` y `reducemcd` descritos en la sección 2.1.6.

- **Forma expandida de Hilbert.** Debemos de tener presente la fórmula descrita previamente sobre la forma expandida de Hilbert. El código está hecho para que muestre los primeros 20 términos de esta serie con el objetivo de que no muestre muchos elementos.

```

1: function FORMAEXPANDIDA(S, numTerminos)
2:   if numTerminos no está definido then
3:     numTerminos  $\leftarrow$  20
4:   end if
5:   terminos  $\leftarrow$  {0}
6:   minS  $\leftarrow$  mín(S)
7:   for i  $\leftarrow$  0 to (numTerminos  $\times$  minS) - 1 do
8:     nuevosTerminos  $\leftarrow$   $\emptyset$ 
9:     for all t  $\in$  terminos do
10:      for all a  $\in$  S do
11:        if t + a  $\leq$  numTerminos  $\times$  minS then
12:          Agregar t + a a nuevosTerminos
13:        end if
14:      end for
15:    end for
16:    for all t  $\in$  nuevosTerminos do
17:      Agregar t a terminos
18:    end for
19:  end for
20:  resultado  $\leftarrow$  Los primeros numTerminos elementos de terminos ordenados en forma ascendente

```

```

21:   return Concatena los elementos de resultado separados por
      “,” seguido de “...”
22: end function

```

- **Forma cerrada de Hilbert.** Para calcular esta forma debemos de realizar dos funciones las cuales se encargaran de determinar la expansión del semigrupo numérico dado hasta un límite razonable (en este caso se puede ver que el límite puede ser la suma del conductor del semigrupo y la multiplicidad, esto es, el mínimo generador positivo) y la función que determina la forma cerrada de Hilbert usando el conjunto de Apéry. Esta forma es válida para cualquier semigrupo numérico que se proporcione. A continuación se muestra el pseudocódigo de la forma cerrada de Hilbert.

```

1: function FORMACERRADA(S)
2:   if  $1 \in S$  then
3:     return " $\frac{1}{1-x}$ "
4:   end if
5:    $m \leftarrow \min(S)$ 
6:    $nFrobenius \leftarrow nFrob(S)$ 
7:   if  $nFrobenius$  es un número y  $nFrobenius \neq -1$  then
8:      $conductor \leftarrow nFrobenius + 1$ 
9:   else
10:     $conductor \leftarrow m \times 10$ 
11:   end if
12:    $limite \leftarrow conductor + m$ 
13:    $semigrupo \leftarrow \text{expansionSemigrupo}(S, limite)$ 
14:   Inicializar arreglo apery de tamaño  $m$  con valor -Infinity- en
      cada posición
15:   for all  $s \in semigrupo$  do
16:      $r \leftarrow s \bmod m$ 
17:     if  $s < apery[r]$  then
18:        $apery[r] \leftarrow s$ 
19:     end if
20:   end for
21:   Construir numerador concatenando, para cada  $s \in apery$ , la
      cadena " $x^{\{s\}}$ " separadas por - "
22:    $denominador \leftarrow (1 - x^m)$ 
23:   return " $\frac{\text{numerador}}{\text{denominador}}$ "
24: end function

```

2.5. Sistema minimal de generadores

Sea S un semigrupo numérico, es decir, un subconjunto de \mathbb{N}_0 cerrado bajo suma y generado por un conjunto finito $A = \{a_1, a_2, \dots, a_k\}$. El conjunto A se denomina *sistema de generadores* de S . Cuando ningún elemento de A puede ser expresado como combinación lineal no negativa de los restantes, se dice que A es un *sistema minimal de generadores*. En este caso, se verifica que

$$S = \langle A \rangle \quad \text{y} \quad \forall a_i \in A, \quad a_i \notin \langle A \setminus \{a_i\} \rangle.$$

El sistema minimal de generadores de un semigrupo numérico posee propiedades fundamentales que lo convierten en un objeto central dentro de la teoría de semigrupos. Entre ellas, destacan:

- **Unicidad y existencia:** Se debe de tener en cuenta que todo semigrupo numérico tiene un único sistema minimal de generadores. Esta unicidad permite definir de forma canónica la *dimensión de inmersión* del semigrupo, que básicamente es el cardinal de dicho sistema. Es decir,

$$e(S) = |A|.$$

Esta dimensión mide cuántos generadores irreducibles son necesarios para construir S , y es un invariante importante en la clasificación de semigrupos.

- **Relación con los huecos:** La estructura del sistema minimal condiciona directamente la distribución de los enteros que no pertenecen a S , llamados *huecos*. El conjunto de huecos de S , denotado por $\mathbb{N}_0 \setminus S$, es finito, y su cardinal (denotado por $g(S)$) depende de cómo interactúan entre sí los generadores mínimos. Semigrupos con un sistema minimal de generadores de mayor tamaño tienden a tener más huecos y una estructura más enriquecida.
- **Influencia sobre los invariantes clásicos:** Invariantes como la *multiplicidad* $m = \min(S \setminus \{0\})$, el *número de Frobenius* $F(S) = \max(\mathbb{N}_0 \setminus S)$ o el *género* $g(S)$ se ven afectados por el conjunto minimal. Por ejemplo, cuando $e(S) = 2$, se puede calcular $F(S)$ explícitamente como

$$F(S) = a_1 a_2 - a_1 - a_2,$$

pero para valores de $e(S) > 2$, esta fórmula deja de ser válida y la relación entre los generadores se vuelve más compleja, dependiendo esencialmente de su minimalidad.

- **Conexión con el conjunto de Apéry:** Sea $m = \min(S \setminus \{0\})$, es decir, la multiplicidad de S . El conjunto de Apéry respecto a m , denotado por $\text{Ap}(S, m)$, está formado por los elementos más pequeños de S en cada clase módulo m , y se define como

$$\text{Ap}(S, m) = \{w_0, w_1, \dots, w_{m-1}\}, \quad w_r = \min\{s \in S \mid s \equiv r \pmod{m}\}.$$

La estructura del conjunto de Apéry refleja de manera precisa cómo los generadores minimales tienen influencia en la cobertura modular del semigrupo. Su conocimiento permite, por ejemplo, describir series generadoras o desarrollar algoritmos de factorización.

- **Conexión con el álgebra conmutativa:** En términos algebraicos, el sistema minimal de generadores de S coincide con los exponentes de las variables en el monomorfismo de $K[x_1, \dots, x_k] \rightarrow K[t]$, definido por $x_i \mapsto t^{a_i}$. La minimalidad de A implica que el ideal de relaciones del anillo del semigrupo $K[S]$ es generado por relaciones mínimas, lo cual es esencial para estudiar su resolución libre, su regularidad y su dimensión homológica.

El sistema minimal de generadores no solo permite describir de manera eficiente todos los elementos de un semigrupo numérico, sino que también determina sus propiedades combinatorias y algebraicas más profundas. Su unicidad, su influencia en invariantes como la multiplicidad, el número de Frobenius o el conjunto de Apéry, y su conexión con la estructura de anillos hacen de él una herramienta esencial tanto en el análisis estructural como en aplicaciones de la teoría de semigrupos numéricos en álgebra conmutativa, teoría de códigos y problemas de optimización discreta.

2.5.1. Propiedades

Sea S un semigrupo numérico finitamente generado y sea $A = \{a_1, a_2, \dots, a_k\}$ su sistema minimal de generadores, es decir, ningún elemento de A puede ser omitido sin perder la propiedad de que $S = \langle A \rangle$. A continuación, se detallan las propiedades fundamentales que caracterizan a estos sistemas:

- **Unicidad:** El sistema minimal de generadores de un semigrupo numérico es único. Si A y B son dos sistemas minimales para S , se tiene que $A = B$. Esta propiedad permite definir de manera canónica el *dimensión de inmersión* de S como

$$\text{edim}(S) = |A|.$$

La unicidad es muy relevante para la clasificación y comparación de semigrupos numéricos.

- **Irredundancia:** Por definición, cada elemento $a_i \in A$ es indispensable, ya que se cumple

$$a_i \notin \langle A \setminus \{a_i\} \rangle.$$

Esta irredundancia garantiza que el sistema minimal contiene únicamente aquellos generadores que aportan información esencial para la estructura del semigrupo.

- **Influencia en invariantes combinatorios y algebraicos:** El sistema minimal de generadores determina varios invariantes importantes del semigrupo:

- *Multiplicidad:* El elemento mínimo de $S \setminus \{0\}$, denotado por m , siempre pertenece a A y desempeña un papel clave en la estructura modular del semigrupo.
- *Número de Frobenius:* En el caso de dos generadores coprimos, el número de Frobenius se calcula con

$$F(a_1, a_2) = a_1 a_2 - a_1 - a_2.$$

Para sistemas con más generadores, la relación entre los generadores afecta directamente el cálculo de $F(S)$.

- *Género:* La cantidad de huecos (enteros que no pertenecen a S) depende de la interacción y distribución de los elementos de A .
- **Conexión con el conjunto de Apéry:** Como se ha visto previamente, la estructura de este conjunto refleja directamente la eficiencia del sistema minimal, ya que permite expresar la serie de Hilbert del semigrupo de forma cerrada:

$$H_S(t) = \frac{\sum_{r=0}^{m-1} t^{w_r}}{1 - t^m}.$$

Así, el conjunto de Apéry es una manifestación concreta de cómo el sistema minimal organiza la distribución modular de los elementos de S .

- **Relación con las presentaciones y relaciones mínimas:** El sistema minimal de generadores no solo define a S , sino que también induce un conjunto de relaciones mínimas entre los generadores. Estas relaciones forman el *ideal de relaciones* del semigrupo y son fundamentales para estudiar la estructura del anillo del semigrupo $K[S]$, su resolución libre y sus propiedades homológicas.

- **Estabilidad en representaciones aritméticas:** La minimalidad de A implica que cada elemento de S tiene una representación, en cierto sentido, única como combinación lineal no negativa de los elementos de A . Esta propiedad es especialmente relevante en la determinación de invariantes combinatorios y facilita el diseño de algoritmos para la factorización y el análisis de la estructura interna de S .

Las propiedades del sistema minimal de generadores, ya sea su unicidad, irredundancia, y la influencia en invariantes clave como la multiplicidad, el número de Frobenius, el género y el conjunto de Apéry son fundamentales para entender la estructura y el comportamiento de los semigrupos numéricos. Estas propiedades no solo permiten una clasificación canónica de los semigrupos, sino que también establecen una sólida conexión entre la teoría combinatoria y el álgebra conmutativa.

2.5.2. Ejemplos

Ejemplo 1: $S = \langle 5, 7 \rangle$

Sea $S = \langle 5, 7 \rangle$. En este caso, el conjunto generador $\{5, 7\}$ es minimal, pues ninguno de sus elementos se puede expresar como combinación lineal no negativa del otro. Los elementos de S se obtienen como combinaciones de 5 y 7:

$$0, 5, 7, 10, 12, 14, 15, 17, 19, 20, 21, 22, 24, 25, \dots$$

Ejemplo 2: $S = \langle 5, 8, 10 \rangle$

Consideremos el semigrupo $S = \langle 5, 8, 10 \rangle$. En este caso, el conjunto generador no es minimal ya que el elemento 10 es redundante:

$$10 = 5 + 5.$$

Por lo tanto, el sistema minimal de generadores de S es en realidad $\{5, 8\}$.

Los elementos de S (tomando $\{5, 8\}$ como generadores) son:

$$0, 5, 8, 10, 13, 15, 16, 18, 20, 21, 23, 24, 25, 26, 28, \dots$$

Ejemplo 3: $S = \langle 3, 7, 11 \rangle$

Sea $S = \langle 3, 7, 11 \rangle$. En este caso, el conjunto generador $\{3, 7, 11\}$ es minimal, ya que ninguno de sus elementos puede ser omitido sin perder la capacidad de generar S . Los elementos de S se generan mediante combinaciones lineales no negativas de 3, 7 y 11:

$$0, 3, 6, 7, 9, 10, 11, 12, 13, 14, 15, \dots$$

En estos ejemplos se puede ver claramente la diferencia entre un semigrupo generado por un conjunto minimal (Ejemplo 1 y Ejemplo 3) y uno en el que el conjunto de generadores presenta redundancia (Ejemplo 2), en donde la eliminación del elemento superfluo conduce al sistema minimal real.

2.5.3. Algoritmos y Métodos

El estudio de los sistemas minimales de generadores en semigrupos numéricos no solo es relevante desde un punto de vista teórico, sino que también ha impulsado el desarrollo de diversos algoritmos y métodos computacionales. Estos algoritmos tienen como objetivos fundamentales determinar la minimalidad de un conjunto generador, calcular invariantes asociados y construir presentaciones eficientes del semigrupo. A continuación, se describen algunos de los enfoques más destacados:

- **Algoritmo de verificación de minimalidad:** Para un conjunto generador $A = \{a_1, a_2, \dots, a_k\}$, un algoritmo básico consiste en comprobar, para cada $a_i \in A$, si

$$a_i \in \langle A \setminus \{a_i\} \rangle.$$

Esta verificación se realiza mediante técnicas de factorización y búsqueda en el semigrupo, descartando aquellos elementos que puedan ser expresados como combinaciones lineales no negativas de los demás. La eliminación de estos elementos redundantes garantiza que el conjunto resultante sea minimal.

- **Métodos basados en el conjunto de Apéry:** El conjunto de Apéry, definido para la multiplicidad $m = \min(S \setminus \{0\})$ como

$$\text{Ap}(S, m) = \{w_0, w_1, \dots, w_{m-1}\}, \quad w_r = \min\{s \in S \mid s \equiv r \pmod{m}\},$$

es una herramienta poderosa para investigar la estructura interna de S . Algoritmos que calculan este conjunto permiten no solo determinar invariantes como el género o el número de Frobenius, sino también verificar la eficiencia del sistema minimal de generadores, ya que la forma cerrada de la serie de Hilbert se expresa directamente en términos del conjunto de Apéry.

- **Algoritmos de optimización y búsqueda exhaustiva:** En casos donde el semigrupo se genera por un conjunto grande o cuando se estudian familias de semigrupos, se emplean técnicas de búsqueda exhaustiva combinadas con estrategias de poda (pruning). Estos algoritmos

exploran sistemáticamente todos los subconjuntos del conjunto generador candidato, eliminando aquellos que no cumplen la condición de minimalidad. El uso de técnicas heurísticas y algoritmos de optimización (como la programación dinámica) mejora la eficiencia, permitiendo el tratamiento de instancias con alta complejidad.

- **Implementación en software algebraico:** Existen diversos paquetes computacionales y sistemas de álgebra computacional (por ejemplo, GAP, Singular y Macaulay2) que incluyen implementaciones especializadas para el estudio de semigrupos numéricos. Estas herramientas no solo automatizan la verificación de la minimalidad de un conjunto generador, sino que también permiten calcular la serie de Hilbert, el conjunto de Apéry y otros invariantes algebraicos asociados. La integración de estos métodos en software facilita la experimentación y la aplicación de los conceptos teóricos en problemas prácticos.
- **Análisis de complejidad:** La complejidad de los algoritmos para determinar sistemas minimales de generadores varía en función del tamaño del conjunto generador y de la estructura aritmética del semigrupo. Aunque los algoritmos de verificación directa pueden ser eficientes para semigrupos con pocos generadores, el problema se vuelve más complejo a medida que aumenta la dimensión de inmersión. Por ello, la investigación en esta área se centra en desarrollar métodos que optimicen el proceso de eliminación de redundancias y que aprovechen propiedades estructurales específicas de los semigrupos numéricos.

Vemos en consecuencia que estos algoritmos y métodos para el estudio de los sistemas minimales de generadores combinan técnicas de verificación directa, cálculos basados en el conjunto de Apéry, y estrategias de optimización computacional. Estos enfoques permiten no solo identificar de manera eficaz el sistema minimal, sino también explorar y cuantificar las propiedades algebraicas y combinatorias del semigrupo, aportando herramientas fundamentales tanto en el análisis teórico como en aplicaciones prácticas dentro del álgebra conmutativa y la teoría de invariantes.

2.5.4. Aplicaciones

Los sistemas minimales de generadores de semigrupos numéricos no solo son objetos de estudio teórico en el ámbito del álgebra y la teoría de números, sino que también poseen aplicaciones relevantes en diversas áreas. A continuación, se presenta un análisis detallado de algunas de estas aplicaciones:

- **Cálculo de invariantes y resolución de problemas clásicos:** El sistema minimal de generadores es fundamental para determinar invariantes clave de un semigrupo, como la *multiplicidad*, el *número de*

Frobenius y la *género*. Por ejemplo, en el caso de semigrupos generados por dos elementos coprimos, la fórmula clásica

$$F(a_1, a_2) = a_1 a_2 - a_1 - a_2$$

se deduce directamente de la estructura minimal. En semigrupos con mayor número de generadores, conocer el sistema minimal permite establecer métodos para la estimación y cálculo de estos invariantes, esenciales en problemas de factorización y optimización discreta.

- **Teoría de códigos y criptografía:** La estructura de los semigrupos numéricos y, en particular, la información encapsulada en su sistema minimal, se utiliza en la construcción de códigos y en la criptografía. La capacidad de describir de forma concisa el conjunto de combinaciones lineales (mediante los generadores mínimos) facilita la codificación de información y el diseño de algoritmos robustos en sistemas criptográficos, en donde la dificultad de la factorización y la estructura modular desempeñan un papel clave.
- **Optimización y problemas combinatorios:** Los algoritmos que se basan en la identificación del sistema minimal de generadores se aplican en problemas de optimización combinatoria. Por ejemplo, en la determinación del número de Frobenius o en la resolución de problemas de cobertura y partición en teoría de números, la representación minimal ayuda a reducir la complejidad computacional, permitiendo un análisis más eficiente de las posibles soluciones.
- **Análisis de anillos de semigrupos y resolución de ideales:** En álgebra conmutativa, el anillo del semigrupo $K[S]$ está íntimamente relacionado con el sistema minimal de generadores de S . La correspondencia

$$x_i \mapsto t^{a_i}$$

asocia cada generador a una variable del anillo de polinomios, y la estructura del ideal de relaciones entre estos generadores es crucial para determinar propiedades homológicas, como la regularidad y la resolución libre del anillo. Esto tiene aplicaciones directas en el estudio de singularidades y en la clasificación de variedades algebraicas.

- **Modelado y simulación en ciencias aplicadas:** La capacidad de describir de manera compacta un semigrupo mediante su sistema minimal tiene implicaciones en la modelación de procesos discretos y en la simulación de sistemas donde las combinaciones lineales juegan un papel central. Esto incluye aplicaciones en economía (modelos de optimización de recursos), en ingeniería (análisis de redes y procesos de fabricación) y en ciencias de la computación (problemas de scheduling y asignación).

- **Estudio de invariantes topológicos y geométricos:** La conexión entre la serie de Hilbert, que se expresa en función de los generadores mínimos, y los invariantes geométricos del espacio afín asociado al semigrupo permite trasladar métodos combinatorios a contextos geométricos. Este puente entre álgebra y geometría abre la puerta al estudio de variedades tóricas y a la aplicación de técnicas geométricas en la resolución de problemas combinatorios.

Se tiene entonces que el análisis y la identificación del sistema minimal de generadores no solo simplifican la descripción de los semigrupos numéricos, sino que también facilitan el desarrollo de métodos algorítmicos y teóricos que tienen un amplio espectro de aplicaciones. Desde el cálculo de invariantes clásicos y la optimización combinatoria, hasta su impacto en la teoría de códigos, la criptografía y el análisis algebraico-geométrico. Los sistemas minimales son herramientas fundamentales que permiten abordar y resolver problemas complejos en diversas áreas de las matemáticas y las ciencias aplicadas.

2.5.5. Programa de cálculo de sistemas minimales de generadores

Vamos a generar un programa en el cual el usuario inserte un conjunto generador de un semigrupo numérico y determinar si este es minimal o no. Para ello, debemos de verificar inicialmente que sea un semigrupo numérico ($\text{MCD} = 1$), por lo cual deberemos de usar los métodos *mcd* y *reducemcd* descritos en la sección 2.1.6. Además, debemos de tener en cuenta la forma expandida de Hilbert ya que nos da información de los elementos que puede generar el semigrupo numérico. Este método se explicó en la sección 2.4.5 y nos ayudará para ver si dado un semigrupo numérico existe algún elemento que se pueda generar a partir de combinaciones de otros (si esto sucede, el conjunto generador no será minimal). Además, el programa hará, en caso de que el conjunto generador introducido no sea minimal, se le indicará al usuario cual es el sistema minimal de generadores en base a los elementos que ha proporcionado. Esto se hará eliminando aquellos elementos que puedan ser generados por otros a partir de combinaciones.

```

1: function ESMINIMAL(S)
2:   if  $|S| = 1$  y  $S[0] = 1$  then
3:     return true
4:   end if
5:   elementosEliminar  $\leftarrow \emptyset$ 
6:    $\min S \leftarrow \min(S)$ 
7:   hilbertExpandida  $\leftarrow \text{formaExpandida}(S, 10 \times \min S)$ 
8:   consecutivos  $\leftarrow 0$ 

```

```

9:   prev  $\leftarrow$  -1
10:  for  $i \leftarrow 0$  to  $\text{length}(\text{hilbertExpandida}) - 1$  do
11:    if  $\text{hilbertExpandida}[i] \neq \text{prev} + 1$  then
12:      consecutivos  $\leftarrow$  1
13:    else
14:      consecutivos  $\leftarrow$  consecutivos + 1
15:      if consecutivos  $\geq \text{min}S$  then
16:        break
17:      end if
18:    end if
19:    prev  $\leftarrow$   $\text{hilbertExpandida}[i]$ 
20:  end for
21:  if consecutivos  $\geq \text{min}S$  then
22:    for all num in S do
23:      if num  $\notin$   $\text{hilbertExpandida}$  then
24:        Agregar num a elementosEliminar
25:      end if
26:    end for
27:  end if
28:  for all num in S do
29:    subConjunto  $\leftarrow S \setminus \{\text{num}\}$ 
30:    expandida  $\leftarrow \text{formaExpandida}(\text{subConjunto}, \text{num} + 1)$ 
31:    if num  $\in$  expandida then
32:      Agregar num a elementosEliminar
33:    end if
34:  end for
35:  if elementosEliminar  $\neq \emptyset$  then
36:    return  $\{x \in S \mid x \notin \text{elementosEliminar}\}$ 
37:  else
38:    return true
39:  end if
40: end function

```

2.6. Sistema de generadores minimal infinito

Partiendo de la noción ya conocida de semigrupos numéricos, en este documento nos centraremos en los **semigrupos numéricos infinitos** con énfasis en el criterio de *sistema de generadores minimal infinito*. La idea es que se verifique que, a partir del mayor generador M , es posible obtener de forma consecutiva los próximos m números, siendo m el menor generador del conjunto minimal.

Teniendo en cuenta el criterio de minimalidad infinita, sea S un sistema

minimal de generadores

$$A = \{a_1, a_2, \dots, a_k\} \quad \text{con} \quad a_1 < a_2 < \dots < a_k.$$

Definimos:

$$M = a_k, \quad m = a_1.$$

El semigrupo S se dice que tiene un *sistema de generadores minimal infinito* si se cumple la siguiente condición:

$$\forall i \in \{1, 2, \dots, m\}, \quad M + i \in S.$$

Es decir, cada uno de los m números inmediatamente mayores que M debe poder obtenerse como combinación lineal (con coeficientes en \mathbb{N}) de los elementos de A . Además, se exige la irredundancia en A , es decir, ningún elemento puede expresarse mediante una combinación de los otros.

2.6.1. Propiedades

Las propiedades que caracterizan a estos sistemas son:

1. Cobertura de bloque consecutivo.

La inclusión de los enteros $M + 1, M + 2, \dots, M + m$ en S garantiza que, a partir del mayor generador, se genere un bloque sin huecos. Esta propiedad asegura una densidad local inmediata y es crucial para la transición al comportamiento global del semigrupo.

2. Transición a la conclusión global.

Dado que un semigrupo numérico contiene a todos los enteros a partir de un cierto conductor, la generación consecutiva del bloque inmediato posterior a M confirma que la propiedad de completitud se activa de forma inmediata, facilitando la extensión a la totalidad de los números naturales.

3. Irredundancia del conjunto de generadores.

La condición de sistema de generadores minimal infinito se fundamenta en que el conjunto A sea minimal en el sentido habitual: ningún generador puede expresarse como combinación lineal de los otros. Esta irredundancia es indispensable para que cada elemento contribuya de forma esencial a la generación del bloque consecutivo y, en consecuencia, a la estructura del semigrupo.

4. Influencia en la estructura del conjunto de Apéry.

La condición del bloque consecutivo impacta directamente en la configuración del conjunto de Apéry, definido como

$$\text{Ap}(S, m) = \{s \in S : s - m \notin S\}.$$

Esta interacción reduce la distribución de residuos módulo m , ayudando en el análisis del género y el número de Frobenius del semigrupo (para más detalles ir a la sección 2.7).

Estas propiedades en conjunto permiten identificar y caracterizar aquellos semigrupos que, siendo generados de manera minimal, aseguran una generación inmediata y consecutiva a partir del mayor elemento, eliminando interrupciones y garantizando la cohesión estructural del sistema.

2.6.2. Ejemplos

Ejemplo 1: $S = \langle 3, 5, 7 \rangle$

Consideramos el semigrupo numérico generado por

$$A = \{3, 5, 7\}.$$

Este conjunto genera el semigrupo de los números que se pueden escribir como combinación lineal de los tres generadores con coeficientes naturales. Dado que el mayor generador es 7 y el menor es 3, se debe comprobar que se puedan generar los siguientes números:

$$7 + 1 = 8, \quad 7 + 2 = 9, \quad 7 + 3 = 10.$$

En este caso se observa que:

$$5 + 3 = 8, \quad 3 + 3 + 3 = 9, \quad 5 + 5 = 10.$$

Como es posible generar los 3 números consecutivos inmediatamente posteriores a 7 (equivalentes al valor de $m = 3$), concluimos que este sistema es *minimal infinito*.

Ejemplo 2: $S = \langle 5, 8, 9, 11 \rangle$

Consideremos el semigrupo numérico generado por

$$A = \{5, 8, 9, 11\}.$$

Aquí, 11 es el mayor generador y 5 es el menor. La condición requiere que se puedan generar los números:

$$11 + 1 = 12, \quad 11 + 2 = 13, \quad 11 + 3 = 14, \quad 11 + 4 = 15, \quad 11 + 5 = 16.$$

Sin embargo, se puede verificar que el número 12 no es alcanzable mediante ninguna combinación lineal de los elementos del conjunto. Por lo tanto, este sistema no cumple la condición de ser minimal infinito.

Ejemplo 3: $S = \langle 3, 4, 5, 19 \rangle$

Consideremos el semigrupo numérico generado por

$$A = \{3, 4, 5, 19\}.$$

Con 19 como mayor generador y 3 como menor, se requiere que los números:

$$19 + 1 = 20, \quad 19 + 2 = 21, \quad 19 + 3 = 22,$$

se puedan generar a partir de los elementos de S . Aunque dichos números pueden generarse, se observa que existe una redundancia en la representación, puesto que

$$5 \times 2 + 3 \times 3 = 19.$$

La posibilidad de expresar el mayor generador 19 como combinación lineal de los otros valores implica que el conjunto no es minimal, lo que invalida la condición de sistema de generadores minimal infinito.

Ejemplo 4: $S = \langle 4, 7 \rangle$

Consideremos el conjunto

$$A = \{4, 7\}$$

y calculemos el número de Frobenius:

$$\text{Número de Frobenius} = 4 \times 7 - 4 - 7 = 17.$$

Dado que el número de Frobenius es mayor que cualquiera de los elementos de A , se concluye que el sistema no es minimal infinito.

2.6.3. Algoritmos y Métodos

Tras las explicaciones previas y con dicha metodología, podemos hallar diversos algoritmos y métodos interesantes donde se ve la utilidad de los semigrupos numéricos.

Verificación directa por fuerza bruta

El método más sencillo consiste en enumerar las combinaciones lineales (con coeficientes en \mathbb{N}) de los elementos de A y comprobar si cada uno de los enteros $M+1, \dots, M+m$ es representable. Un esquema básico del algoritmo es el siguiente:

Algorithm 1 Verificación por fuerza bruta del sistema de generadores minimal infinito

```

1: procedure MINIMALINFINITODIRECTO( $A$ )
2:    $m \leftarrow \min(A)$  ▷ Menor generador
3:    $M \leftarrow \max(A)$  ▷ Mayor generador
4:   for  $i \leftarrow 1$  to  $m$  do
5:      $x \leftarrow M + i$ 
6:     if NOREPRESENTABLE( $x, A$ ) then
7:       return False
8:     end if
9:   end for
10:  return True
11: end procedure

```

La función auxiliar *NoRepresentable* comprueba, mediante búsqueda (recursiva o iterativa), si existe alguna combinación de los generadores que sume x . Aunque intuitivo, este método puede resultar ineficiente cuando la cantidad de generadores o los coeficientes necesarios son altos, ya que la complejidad crece de forma exponencial.

Programación dinámica

Para evitar la enumeración completa de combinaciones, se puede emplear programación dinámica. La idea es construir un arreglo booleano T que indique si cada número hasta $M + m$ es alcanzable:

Algorithm 2: Verificación por Programación Dinámica

```

1: procedure MINIMALINFINITODP( $A$ )
2:    $m \leftarrow \min(A)$ ,  $M \leftarrow \max(A)$ 
3:   Sea  $T[0 \dots (M + m)]$  un arreglo de booleanos, inicializado en False
4:    $T[0] \leftarrow \mathbf{True}$  ▷ El 0 siempre es alcanzable
5:   for  $x \leftarrow 0$  to  $(M + m)$  do
6:     if  $T[x]$  then
7:       for cada  $a \in A$  do
8:         if  $x + a \leq M + m$  then
9:            $T[x + a] \leftarrow \mathbf{True}$ 
10:        end if
11:      end for
12:    end if
13:  end for
14:  for  $i \leftarrow 1$  to  $m$  do
15:    if  $T[M + i] = \mathbf{False}$  then
16:      return False

```

```

17:     end if
18:   end for
19:   return True
20: end procedure

```

Este método es más eficiente que la fuerza bruta al reutilizar resultados previos, resultando especialmente adecuado cuando la suma $M + m$ no es excesivamente grande. Su complejidad es pseudo-polinomial, lo que lo hace práctico para instancias moderadas del problema.

Uso del conjunto de Apéry

Aunque se va a explicar con mayor detenimiento este concepto en la sección 2.7, el conjunto de Apéry de un semigrupo S respecto al menor generador m se define como:

$$\text{Ap}(S, m) = \{s \in S : s - m \notin S\}.$$

Dado que este conjunto contiene exactamente m elementos, resulta útil para analizar la estructura residual de S . Un método basado en el conjunto de Apéry incluye los siguientes pasos:

- Calcular $\text{Ap}(S, m)$ de forma iterativa o mediante programación dinámica.
- Verificar si los elementos del bloque $\{M + 1, M + 2, \dots, M + m\}$ se pueden obtener mediante sumas de elementos de $\text{Ap}(S, m)$ combinados con múltiplos de m .

Este enfoque aprovecha propiedades invariantes del semigrupo para simplificar la verificación de la condición minimal infinito.

Optimización e irredundancia

Además de verificar la condición de sistema de generadores minimal infinito, es crucial comprobar la irredundancia del conjunto de generadores, es decir, asegurarse de que ningún generador pueda expresarse como combinación lineal de los otros. Para ello, se pueden incorporar módulos de optimización que:

- Eliminan generadores redundantes mediante técnicas de reducción (por ejemplo, resolviendo sistemas de ecuaciones diofánticas).
- Utilizan heurísticas basadas en la factorización y en la estructura del semigrupo, para evitar realizar cálculos costosos sobre combinaciones que no alteran el resultado final.

Estas estrategias de optimización son esenciales en semigrupos complejos, ya que mejoran la velocidad y eficiencia del algoritmo global, garantizando al mismo tiempo que la estructura minimal del sistema se mantenga intacta.

Análisis de los métodos

Podemos observar como la detección y construcción de sistemas minimal infinitos en semigrupos numéricos se puede abordar mediante diversos enfoques:

- La verificación directa por fuerza bruta, que es intuitiva pero potencialmente costosa computacionalmente.
- Algoritmos basados en programación dinámica, que permiten reutilizar resultados previos para optimizar la búsqueda.
- El uso del conjunto de Apéry, que simplifica el problema al aprovechar propiedades estructurales del semigrupo.
- Procedimientos de optimización que verifican la irredundancia y reducen la complejidad del conjunto generador.

Cada uno de estos métodos tiene sus ventajas y limitaciones, y su aplicación dependerá del contexto específico y de la complejidad del semigrupo estudiado.

2.6.4. Aplicaciones

El estudio de los sistemas minimal infinitos en semigrupos numéricos no solo aporta elementos de interés teórico, sino que también tiene aplicaciones prácticas en diversas áreas, entre las cuales se destacan:

1. Teoría de números y problema de la moneda.

Una aplicación central se da en el famoso Problema del Frobenius (o problema de la moneda). La propiedad minimal infinito, al garantizar un bloque consecutivo de m números a partir del mayor generador M , permite determinar de forma más precisa el número de Frobenius y el género del semigrupo. En este contexto, se utiliza la condición para identificar cuál es el máximo entero no representable como combinación lineal de los generadores, así como para analizar la distribución de huecos en la representación de los números naturales. Esto facilita el desarrollo de fórmulas y algoritmos que optimizan la resolución de este tipo de problemas en teoría de números.

2. Optimización y algoritmos combinatorios.

La estructura inherente a un sistema de generadores minimal infinito resulta muy útil en el diseño de algoritmos eficientes. La garantía de que, a partir

de M , existe un bloque consecutivo de m números, permite implementar técnicas de programación dinámica que evitan búsquedas exhaustivas en la representación de enteros. Esta propiedad se traduce en una reducción de la complejidad computacional de problemas combinatorios donde se requiere verificar la alcanzabilidad de ciertos valores mediante sumas de elementos predefinidos, abriendo así aplicaciones en asignación de recursos y partición de enteros.

3. Análisis de invariantes y estructuras algebraicas.

El sistema de generadores minimal infinito se relaciona estrechamente con el conjunto de Apéry y otros invariantes algebraicos del semigrupo. Estos invariantes permiten estudiar la irredundancia del conjunto de generadores y analizar la factorización intrínseca de los semigrupos numéricos. Dichos estudios tienen repercusiones en ramas de la álgebra conmutativa y en la teoría de anillos, donde es fundamental entender cómo se distribuyen y rellenan los huecos en la secuencia de enteros representables, proporcionando además herramientas para el estudio de singularidades en geometría algebraica.

4. Aplicaciones en criptografía y teoría de códigos.

Aunque es un ámbito de investigación emergente, se ha observado que las propiedades estructurales de los semigrupos numéricos, y en particular la condición del sistema de generadores minimal infinito, pueden ser aprovechadas en contextos criptográficos y en la teoría de códigos. La capacidad de controlar la suma y representabilidad de enteros a partir de un conjunto mínimo de generadores resulta útil en la generación de claves y en la construcción de códigos con propiedades aditivas específicas. La regularidad y predecibilidad que ofrece esta propiedad son aspectos atractivos para el desarrollo de algoritmos criptográficos seguros y eficientes.

Estas aplicaciones muestran que la propiedad del sistema de generadores minimal infinito impulsa el estudio teórico del semigrupo, proporcionando herramientas para resolver problemas complejos en optimización, análisis algebraico y sistemas criptográficos, y abriendo nuevas vías de investigación en la interfaz entre la teoría de números y la informática.

5. Modelos en producción industrial y optimización de recursos.

Los sistemas minimales infinitos pueden aplicarse en la optimización de procesos industriales, tales como el corte de materiales y la planificación de la producción. Al garantizar la generación de un bloque consecutivo de números, se pueden definir combinaciones óptimas que minimicen el desperdicio y mejoren la asignación de recursos, lo que es crucial en problemas como el de corte o el empaclado.

6. Optimización en sistemas de información y bases de datos.

En informática, los semigrupos numéricos se utilizan para modelar estruc-

turas de datos y relaciones entre entidades. La propiedad de un sistema minimal infinito puede aplicarse en la optimización y normalización de bases de datos, ayudando a detectar y eliminar redundancias. Esto se traduce en la mejora del rendimiento en la ejecución de consultas y en la integridad de la información almacenada.

2.6.5. Programa de cálculo de sistemas minimales infinitos

Se ha realizado un programa en donde se puede ver el funcionamiento de, a partir de un semigrupo numérico dado, ver si presenta un sistema de generadores minimal infinito. Para ello, debemos de tener en cuenta que los sistemas de generadores minimales infinitos es un subconjunto de los semigrupos numéricos con un sistema de generadores minimal, dado que debe de cumplir sus propiedades además de las explicadas en esta sección.

El pseudocódigo del programa realizado es el siguiente:

```

1: procedure EsMinimalInfinito( $S$ )
2:   Entrada: Conjunto  $S$  de enteros positivos (generadores mínimos)
3:   Salida: Verdadero si  $S$  es minimal infinito, Falso en caso contrario
                                     ▷ Validar la entrada

4:   if  $S = \emptyset$  o  $\exists x \in S$  tal que  $x \leq 0$  then
5:     return Error: Conjunto  $S$  inválido
6:   end if
7:   if  $\text{MCD}(S) \neq 1$  then
8:     return Error: El conjunto  $S$  no forma un semigrupo nu-
        mérico
9:   end if
        ▷ Ordenar  $S$  y definir el menor generador  $m$  y el mayor generador
         $M$ 
10:  Ordenar  $S$  en orden ascendente
11:   $m \leftarrow \min(S)$ 
12:   $M \leftarrow \max(S)$ 
        ▷ Verificar la generación del bloque consecutivo: de  $M + 1$  a  $M + m$ 
13:  Inicializar arreglo booleano  $\text{alcanzable}[0 \dots M + m]$  con Falso
14:   $\text{alcanzable}[0] \leftarrow \text{Verdadero}$ 
15:  for  $i \leftarrow 1$  hasta  $M + m$  do
16:    for  $a \in S$  do
17:      if  $i - a \geq 0$  y  $\text{alcanzable}[i - a] = \text{Verdadero}$  then
18:         $\text{alcanzable}[i] \leftarrow \text{Verdadero}$ 
19:        break                                     ▷ Se encontró una combinación para  $i$ 
20:      end if
21:    end for
22:  end for
23:  for  $i \leftarrow M + 1$  hasta  $M + m$  do

```

```

24:      if alcanzable[i] = Falso then
25:          return Falso                ▷ S no genera el bloque requerido
26:      end if
27:  end for
    ▷ Verificar irredundancia: ningún generador debe generarse a partir
    de los otros
28:  for cada  $g \in S$  do
29:      if  $g$  puede ser generado a partir de  $S \setminus \{g\}$  then
30:          return Falso                ▷ Se encontró redundancia en  $g$ 
31:      end if
32:  end for
33:  return Verdadero
34: end procedure

```

Explicación General:

1. Se valida que la entrada forme un semigrupo numérico (en particular, que el máximo común divisor de S sea 1).
2. Se ordena S y se definen el menor generador m y el mayor generador M .
3. Programaremos un algoritmo de programación dinámica para determinar si cada entero desde $M + 1$ hasta $M + m$ es alcanzable mediante combinaciones lineales de los elementos de S .
4. Se verifica que no exista redundancia, es decir, que ningún generador se pueda obtener como combinación de los otros elementos de dicho generador.
5. Si ambas condiciones se cumplen, se concluye que S es un sistema minimal infinito.

2.7. Conjuntos de Apéry

En la teoría de semigrupos numéricos, el estudio de los conjuntos de Apéry representa un aspecto fundamental para comprender la estructura interna y modular de estos conjuntos. Sea S un semigrupo numérico, es decir, un subconjunto de \mathbb{N}_0 cerrado bajo suma y generado por un conjunto finito. Un elemento central en el análisis de S es su *multiplicidad* m , definida como el mínimo elemento de $S \setminus \{0\}$. El conjunto de Apéry de S respecto a m , denotado por $\text{Ap}(S, m)$, se define de la siguiente manera:

$$\text{Ap}(S, m) = \{w_0, w_1, \dots, w_{m-1}\}, \quad \text{donde} \quad w_r = \min\{s \in S \mid s \equiv r \pmod{m}\}.$$

Esta construcción permite capturar, para cada clase residual módulo m , el menor representante que pertenece al semigrupo S . La relevancia del conjunto de Apéry radica en su capacidad para codificar de forma compacta información sobre la distribución modular de los elementos de S . Al hacerlo, se establecen vínculos profundos entre la aritmética modular y la estructura combinatoria del semigrupo, lo que facilita el estudio de invariantes como el número de Frobenius, el género y otros parámetros que caracterizan a S .

Por tanto, el conjunto de Apéry en el estudio de semigrupos numéricos presenta un marco conceptual y operativo para abordar problemas fundamentales en la teoría de números y en aplicaciones algebraicas y combinatorias. Un estudio detallado de este abre la puerta a una entendimiento mayor de cómo se organizan los elementos en S y de cómo interactúan las propiedades modulares con los invariantes clásicos de estos semigrupos.

2.7.1. Propiedades

El conjunto de Apéry posee propiedades interesantes que se pueden abordar desde diversas perspectivas, cada una aportando herramientas y conceptos que enriquecen su estudio. A continuación se presenta un análisis detallado y conciso de las principales aproximaciones:

1. Propiedades algebraicas y estructurales.

El conjunto de Apéry encapsula la estructura modular del semigrupo numérico al identificar, para cada clase residual módulo m , el representante mínimo. Esto permite expresar de forma única cualquier elemento $s \in S$ como

$$s = w_r + qm, \quad 0 \leq r < m \text{ y } q \in \mathbb{N}_0.$$

Esta descomposición es fundamental para determinar invariantes del semigrupo, tales como el número de Frobenius, el género y la multiplicidad, y refleja la conexión intrínseca entre la aritmética modular y la estructura algebraica del semigrupo. Además, se relaciona estrechamente con la teoría de ideales en el anillo del semigrupo $K[S]$, proporcionando un puente entre la teoría de semigrupos numéricos y el álgebra conmutativa.

2. Enfoque combinatorio en la distribución de residuos.

Desde una perspectiva combinatoria, el conjunto de Apéry ofrece una ventana para analizar la distribución de los residuos de los elementos de S módulo m . Al garantizar la existencia de un único representante mínimo por clase, se facilita el estudio del crecimiento y la densidad del semigrupo, así como el conteo de las formas en que los números pueden expresarse mediante los generadores. Este enfoque no solo ayuda a identificar patrones en la suma de múltiplos, sino que también

proporciona una intuición clara sobre la regularidad en la organización de los elementos del semigrupo.

3. Métodos computacionales y aplicaciones algorítmicas.

El cálculo efectivo del conjunto de Apéry es una herramienta central en el diseño de algoritmos para semigrupos numéricos. Mediante métodos computacionales, es posible generar de forma iterativa los elementos de S y, a partir de ellos, extraer el conjunto $\text{Ap}(S, m)$. Esta aproximación permite no solo confirmar la finitud y unicidad del conjunto, sino también optimizar algoritmos para la factorización de elementos, el cálculo de series generadoras y la resolución de ecuaciones diofánticas. La aplicación de estas técnicas computacionales es especialmente relevante en instancias de alta dimensión, donde la complejidad combinatoria del semigrupo se incrementa significativamente.

Podemos ver la importancia de tener en consideración el conjunto de Apéry en la teoría de semigrupos numéricos. La perspectiva algebraica proporciona un marco estructural sólido, el enfoque combinatorio ofrece intuiciones valiosas sobre la organización interna del semigrupo, y los métodos computacionales permiten aplicar estos conceptos en contextos prácticos y de investigación, impulsando avances tanto teóricos como aplicados en la teoría de números y la optimización.

2.7.2. Ejemplos

Ejemplo 1: $S = \langle 5, 7 \rangle$

Sea $S = \langle 5, 7 \rangle$ y note que su multiplicidad es $m = 5$ (el menor elemento de $S \setminus \{0\}$). El conjunto de Apéry de S respecto a m se define por

$$\text{Ap}(S, 5) = \{w_0, w_1, w_2, w_3, w_4\},$$

donde

$$w_r = \min\{s \in S \mid s \equiv r \pmod{5}\}.$$

Se tiene:

- $w_0 = 0$.
- $w_1 = 21$, ya que 21 es el menor elemento en S congruente con 1 módulo 5.
- $w_2 = 7$, pues $7 \equiv 2 \pmod{5}$.
- $w_3 = 28$, siendo el menor elemento con residuo 3 módulo 5.
- $w_4 = 14$, siendo el menor elemento de S que satisface $s \equiv 4 \pmod{5}$.

Ejemplo 2: $S = \langle 3, 7, 9 \rangle$

Para el semigrupo $S = \langle 3, 7, 9 \rangle$, se tiene $m = 3$. Así, el conjunto de Apéry respecto a m es

$$\text{Ap}(S, 3) = \{w_0, w_1, w_2\},$$

donde:

- $w_0 = 0$.
- $w_1 = 7$, puesto que $7 \equiv 1$ módulo 3 y es el menor elemento de S con dicho residuo.
- $w_2 = 14$, el menor número en S con $14 \equiv 2$ módulo 3.

Ejemplo 3: $S = \langle 4, 6, 11 \rangle$

Consideremos $S = \langle 4, 6, 11 \rangle$ con multiplicidad $m = 4$. El conjunto de Apéry respecto a m se expresa como

$$\text{Ap}(S, 4) = \{w_0, w_1, w_2, w_3\},$$

donde:

- $w_0 = 0$.
- $w_1 = 17$, ya que es el menor elemento en S que satisface $s \equiv 1 \pmod{4}$.
- $w_2 = 6$, dado que $6 \equiv 2$ módulo 4 y es el mínimo con ese residuo.
- $w_3 = 11$, el mínimo elemento con $s \equiv 3$ módulo 4.

2.7.3. Algoritmos y Métodos para los Conjuntos de Apéry en Semigrupos Numéricos

El cálculo y análisis del conjunto de Apéry es un componente esencial en el estudio computacional de semigrupos numéricos. Diversos algoritmos y métodos han sido desarrollados para obtener $\text{Ap}(S, m)$ de forma eficiente, lo que a su vez facilita el cálculo de invariantes clave como el número de Frobenius, el género y otros parámetros. Los algoritmos y métodos más destacados son:

1. Métodos iterativos y de enumeración:

Este enfoque se basa en generar de forma sistemática los elementos del semigrupo S a partir de sus generadores. Una vez obtenido un conjunto suficientemente amplio de elementos, se procede a clasificar estos números según sus residuos módulo m . El algoritmo identifica, para cada clase residual r , el elemento mínimo w_r , lo que permite construir directamente $\text{Ap}(S, m)$. Aunque este método puede resultar

computacionalmente intensivo para semigrupos con alta densidad o generadores numerosos, su simplicidad y claridad lo hacen adecuado para casos de baja complejidad.

2. Programación dinámica:

La programación dinámica se utiliza para evitar el recálculo redundante de soluciones intermedias al generar elementos de S . Este método aprovecha la propiedad de descomposición única de cada elemento $s \in S$ en la forma

$$s = w_r + qm, \quad 0 \leq r < m, \quad q \in \mathbb{N}_0,$$

lo que permite almacenar y reutilizar resultados parciales en una tabla o estructura de datos similar. Este enfoque reduce significativamente el coste computacional y es especialmente útil para semigrupos con parámetros de alta dimensión o cuando se requiere explorar grandes intervalos numéricos.

3. Algoritmos basados en técnicas de optimización y búsqueda:

Otra aproximación consiste en utilizar algoritmos de búsqueda optimizada, tales como algoritmos voraces o de ramificación y poda, para identificar de forma directa los elementos mínimos de cada clase residual. Estos métodos se valen de criterios heurísticos para descartar caminos que no conducen a la mínima representación, mejorando la eficiencia global del proceso. En algunos casos, la incorporación de técnicas de paralelización permite explorar múltiples clases residuales de manera simultánea, acelerando la construcción del conjunto de Apéry.

La variedad de métodos disponibles para el cálculo del conjunto de Apéry refleja la complejidad y la riqueza de la teoría de semigrupos numéricos. Cada aproximación ofrece ventajas específicas: los métodos iterativos son conceptualmente directos, la programación dinámica optimiza el rendimiento en problemas de mayor escala, y los algoritmos de búsqueda optimizada proporcionan soluciones eficientes en contextos con restricciones computacionales. La elección del método adecuado dependerá de las características del semigrupo en estudio y de los requisitos de precisión y eficiencia en la aplicación concreta.

2.7.4. Aplicaciones

Los conjuntos de Apéry son herramientas esenciales en el estudio de semigrupos numéricos y encuentran aplicaciones en diversas áreas de la teoría de números y el álgebra conmutativa. Su estructura y propiedades permiten abordar problemas tanto teóricos como prácticos, facilitando el análisis y el cómputo de invariantes importantes. A continuación, se describen en detalle algunas de las principales aplicaciones:

1. Cálculo de Invariantes:

El conjunto de Apéry es clave para determinar invariantes clásicos de un semigrupo numérico. Gracias a la descomposición única de cada elemento $s \in S$ en la forma

$$s = w_r + qm, \quad 0 \leq r < m, \quad q \in \mathbb{N}_0,$$

es posible calcular el *número de Frobenius* $F(S)$ mediante la relación

$$F(S) = \max\{\text{Ap}(S, m)\} - m.$$

Además, la suma de los elementos en $\text{Ap}(S, m)$ se utiliza para determinar el *género* $g(S)$, que es el número total de huecos en S . Estos invariantes son fundamentales para clasificar y estudiar la estructura de los semigrupos numéricos.

2. Series Generadoras y Análisis Combinatorio:

La representación de S mediante el conjunto de Apéry permite construir series generadoras de la forma

$$\sum_{s \in S} x^s = \sum_{r=0}^{m-1} x^{w_r} \left(\sum_{q \geq 0} x^{qm} \right).$$

Esta expresión no solo facilita el estudio analítico del semigrupo, sino que también ofrece una vía para analizar su crecimiento y densidad. El enfoque combinatorio derivado de la estructura modular de $\text{Ap}(S, m)$ ayuda a identificar patrones en la distribución de las representaciones de los enteros y a establecer conexiones con otras áreas de la combinatoria.

3. Aplicaciones en Álgebra Conmutativa:

En el contexto del álgebra conmutativa, el conjunto de Apéry juega un papel crucial en el estudio de los anillos de semigrupos $K[S]$. La descomposición $s = w_r + qm$ se refleja en la estructura del ideal de relaciones del anillo, permitiendo la obtención de una resolución libre mínima y el análisis de invariantes homológicos. Esta conexión es esencial para investigar propiedades como la regularidad, la profundidad y la dimensión de $K[S]$, estableciendo un puente entre la teoría combinatoria de semigrupos y la geometría algebraica.

4. Diseño de Algoritmos y Resolución de Problemas Diofánticos:

La estructura explícita de $\text{Ap}(S, m)$ facilita el desarrollo de algoritmos eficientes para la factorización de elementos en S y la solución de ecuaciones diofánticas lineales. Al reducir problemas potencialmente

complejos a la búsqueda del representante mínimo en cada clase residual, se optimizan métodos de programación dinámica y técnicas de búsqueda, lo que resulta en algoritmos aplicables a instancias de alta dimensión o con gran densidad de generadores.

Podemos resumir que las aplicaciones del conjunto de Apéry en semigrupos numéricos abarcan tanto el cálculo de invariantes fundamentales como el desarrollo de herramientas analíticas y algorítmicas. Estas aplicaciones han impulsado avances significativos en la comprensión de la estructura de los semigrupos, fortaleciendo la conexión entre la teoría de números, la combinatoria y el álgebra conmutativa.

2.7.5. Programa de cálculo de conjuntos de Apéry

Dado un semigrupo numérico, vamos a calcular el conjunto de Apéry. Para ello, deberemos de comprobar primero que el conjunto proporcionado es un semigrupo numérico ($\text{MCD} = 1$), por lo que será necesario hacer uso de un método que calcule el máximo común divisor del conjunto de números dado. Para ello, se hará uso de los métodos *mcd* y *reducemcd* explicados en la sección 2.1.6.

Haremos uso también de la forma expandida de Hilbert para almacenar aquellos elementos mínimos que se puedan generar que cumplan la condición de formar parte de los elementos del conjunto de Apéry de ese semigrupo numérico. Este método fue explicado en la sección 2.4.5 llamado *formaExpandida*.

El método *calculaConjuntoApery* va a permitir calcular el conjunto de Apéry de un semigrupo numérico. Se procesará este conjunto para determinar los elementos más pequeños que pueden ser generados por el semigrupo numérico que tiene cada residuo posible módulo d . Recordemos que se hace módulo con el menor número introducido en el semigrupo numérico sin contar el 0.

```

1: function CALCULACONJUNTOAPERY
2:   entrada  $\leftarrow$  Obtener el valor de inputNumeros
3:   numeros  $\leftarrow$  Entrada por ‘,’ , convertir a enteros y filtrar valores válidos
4:    $d \leftarrow \text{mín}(\text{numeros})$ 
5:   if longitud de numeros = 0 or  $d \leq 0$  then
6:     Mostrar mensaje: Tienes que introducir números que sean válidos.
7:     return
8:   end if
9:   if reducemcd(numeros)  $\neq 1$  then
10:    Mostrar mensaje: El conjunto de números que has metido no es
    un semigrupo numérico.
11:    vaciarResultados()
12:    return
```

```
13:   end if
14:   serieExpandida  $\leftarrow$  formaExpandida(numeros)
15:   conjuntoApery  $\leftarrow \emptyset$ 
16:   modulos  $\leftarrow \emptyset$ 
17:   for  $r \leftarrow 0$  to  $d - 1$  do
18:       min  $\leftarrow$  1er elemento de serieExpandida que cumple  $s \bmod d = r$ 
19:       Agregar min a conjuntoApery
20:       Agregar Residuo  $r$ : min a modulos
21:   end for
22:   Actualizar elemento Ap con: Conjunto de Apéry para  $d$ : {conjuntoApery}
23:   Actualizar elemento Modulos con lista de modulos
24: end function
```


Capítulo 3

Semigrupos numéricos basados en informática

Los semigrupos numéricos son un concepto matemático fundamental que, al aplicarse en el ámbito de la informática, abre la puerta a soluciones algorítmicas innovadoras. Diversos programas, desde autómatas hasta la seguridad en la generación de contraseñas, el problema de la mochila y el cálculo de semigrupos a partir de un género fijo, se puede realizar gracias al poder combinatorio que ofrecen los semigrupos numéricos.

En el diseño de autómatas, los semigrupos permiten representar transiciones de estados de forma eficiente, facilitando el análisis y simulación de sistemas complejos. En la seguridad, su aplicación aumenta la entropía para mejorar la robustez contra ataques, mientras que en problemas combinatorios como el de la mochila, estos conceptos ayudan a optimizar algoritmos y a reducir la complejidad computacional. Asimismo, el cálculo de semigrupos a partir de un género fijo constituye un enfoque innovador para generar nuevas estructuras algorítmicas, fortaleciendo la conexión entre la teoría numérica y sus aplicaciones prácticas en informática.

A continuación, se procederá a detallar los diversos programas informáticos realizados que dan un enfoque alternativo a diversos problemas en el área de la informática que pueden ser resueltos gracias a la estructura que presenta los semigrupos numéricos.

3.1. Autómatas

Un autómata es un modelo matemático empleado para describir sistemas de cómputo que procesan secuencias de símbolos. Su propósito es representar, de forma abstracta, todo tipo de procesos secuenciales. En su forma clásica, un Autómata Finito Determinista (AFD) se define formalmente como una 5-tupla

$$A = (Q, \Sigma, \delta, q_0, F),$$

donde:

- Q es un conjunto finito de estados.
- Σ es el alfabeto, es decir, un conjunto finito de símbolos de entrada.
- $\delta : Q \times \Sigma \rightarrow Q$ es la función de transición, que determina, para cada estado y símbolo, a qué estado se debe mover el autómata.
- $q_0 \in Q$ es el estado inicial.
- $F \subseteq Q$ es el conjunto de estados de aceptación o finales.

Esta definición no solo aplica a autómatas deterministas, sino que puede extenderse a autómatas no deterministas, con pila o incluso a modelos más complejos como las máquinas de Turing.

El estudio de los autómatas es fundamental en la teoría de lenguajes formales y la computación. Permiten clasificar problemas en función de la complejidad y de los recursos computacionales necesarios para resolverlos. Para poner un ejemplo podemos ver que los autómatas finitos se utilizan para reconocer lenguajes regulares, que son esenciales en el procesamiento de texto, la implementación de analizadores léxicos en compiladores y el reconocimiento de patrones en secuencias.

Además, el análisis de un autómata implica considerar la minimización, es decir, la reducción del número de estados sin alterar el lenguaje reconocido. Esta optimización es clave para mejorar la eficiencia en aplicaciones prácticas, como la simulación de sistemas o el diseño de circuitos digitales.

En ámbitos más avanzados, el estudio de autómatas no deterministas, autómatas con pila y máquinas de Turing permite abordar problemas de decisión y complejidad computacional. La integración de conceptos algebraicos, como los semigrupos y monoides, en la teoría de autómatas enriquece la comprensión de la estructura de los lenguajes formales y proporciona herramientas para la transformación y clasificación de estos lenguajes.

Los autómatas ofrecen un marco unificador que conecta teoría y práctica. Su aplicación se extiende desde la validación de protocolos de comunicación hasta la implementación de algoritmos de seguridad, permitiendo analizar y optimizar sistemas complejos de manera sistemática y rigurosa.

Podemos ver entonces que los autómatas guardan cierta relación con los semigrupos numéricos. Aunque surgen en contextos teóricos diferentes, comparten una estructura algebraica subyacente que permite conectar el análisis de procesos secuenciales con propiedades aritméticas. En concreto, la función de transición de un autómata que determina el cambio de estado ante la lectura de un símbolo que puede interpretarse como una transformación.

La composición de estas transformaciones induce un semigrupo de funciones sobre el conjunto de estados, estableciendo así un vínculo natural entre la teoría de autómatas y los semigrupos numéricos.

Esta conexión se torna especialmente relevante cuando se abordan aplicaciones prácticas en informática. Por ejemplo, en programas de generación de contraseñas o en la optimización de algoritmos para el problema de la mochila, la estructura numérica de ciertos semigrupos definidos como subconjuntos de los enteros no negativos cerrados bajo la suma nos ofrece un marco ideal para modelar y analizar patrones emergentes. Podemos ver aspectos como el género, el conductor o el número de Frobenius de un semigrupo numérico pueden traducirse en parámetros que revelan la complejidad y eficiencia de las transiciones en un autómata.

Asimismo, la relación se profundiza al aplicar técnicas comunes en ambos campos. La minimización de estados en un autómata se asemeja a la búsqueda de un sistema minimal de generadores en un semigrupo, y el análisis de series asociadas, similar a la serie de Hilbert, permite estudiar propiedades de crecimiento y repetición. De esta manera, la transferencia de conceptos entre la teoría de autómatas y la de semigrupos numéricos no solo enriquece ambos campos, sino que también abre nuevas posibilidades para diseñar algoritmos más óptimos y robustos.

3.1.1. Propiedades

Los autómatas son modelos formales poderosos para representar y analizar sistemas de cómputo discretos, mientras que los semigrupos numéricos, definidos como subconjuntos de los enteros no negativos cerrados bajo la suma, ofrecen un marco algebraico para estudiar estructuras y patrones aritméticos. La intersección de estos dos campos revela propiedades interesantes y útiles, las cuales vamos a explicar en detalle.

1. Representación de estados y elementos del semigrupo

En un autómata diseñado para modelar un semigrupo numérico, cada estado puede interpretarse como un elemento específico del semigrupo. El estado inicial, generalmente representado por el número 0, actúa como la base a partir de la cual se generan otros elementos mediante la suma de generadores. Esta correspondencia establece un mapeo natural entre la estructura del autómata y el conjunto de números alcanzables, permitiendo visualizar la evolución aritmética del semigrupo como un recorrido por estados.

2. Transiciones deterministas y cierre operacional

La operación de suma que define un semigrupo numérico cumple la propiedad de cierre, es decir, la combinación de dos elementos produce otro elemento del mismo semigrupo. En el autómata, cada transición—representada por la adición de un generador—conduce de mane-

ra determinista de un estado a otro. Esto garantiza que el autómata opere de forma predecible y única, reflejando la estructura aditiva y asociativa característica de los semigrupos.

3. Minimización y generadores mínimos

Un semigrupo numérico se puede caracterizar mediante un conjunto mínimo de generadores, análogo al proceso de minimizar un autómata para eliminar estados redundantes. La búsqueda del conjunto generador mínimo permite optimizar el proceso de validación y enumeración de elementos, tal como la minimización de estados en un autómata optimiza el reconocimiento de un lenguaje. Esta propiedad facilita el análisis tanto teórico como computacional del semigrupo.

4. Condiciones de aceptación: conductor y estados finales

En la teoría de autómatas se definen estados de aceptación para indicar cuándo una entrada es reconocida. Paralelamente, en un semigrupo numérico es común considerar el concepto de conductor, es decir, el menor número a partir del cual todos los enteros mayores pertenecen al semigrupo. Esta idea se asemeja a la noción de estados de aceptación: una vez superado el conductor, cualquier transición (o suma) adicional produce un elemento “aceptado” dentro del semigrupo.

5. Función de transición y composición de operaciones

La función de transición de un autómata, al componer sus operaciones secuencialmente, forma un semigrupo de transformaciones sobre el conjunto de estados. De manera análoga, la suma en un semigrupo numérico es una operación asociativa que permite generar nuevos elementos a partir de combinaciones de generadores. Esta similitud estructural refuerza la conexión entre ambos conceptos, evidenciando cómo la composición de transiciones en el autómata refleja la estructura algebraica inherente a los semigrupos.

6. Aplicaciones algorítmicas y reconocimiento de patrones

La utilización de autómatas para explorar semigrupos numéricos facilita el desarrollo de algoritmos eficientes para la verificación y generación de elementos. En contextos como la seguridad, la optimización y la resolución de problemas combinatorios, el autómata actúa como un mecanismo de reconocimiento en tiempo lineal de secuencias o patrones numéricos, aprovechando la propiedad determinista de sus transiciones para garantizar resultados precisos y optimizados.

Se puede observar en consecuencia que estas propiedades demuestran cómo la unión entre autómatas y semigrupos numéricos nos da un mayor entendimiento de ambos campos, proporcionando un enfoque robusto y versátil para enfrentar problemas complejos de manera estructurada y eficiente.

3.1.2. Ejemplos

Los ejemplos que se van a ofrecer son a partir del programa realizado en la sección 3.1.5, en donde se comprueba, dado un semigrupos numérico, si los números que inserta el usuario pueden ser generados o no bajo dicho semigrupo.

Ejemplo 1: $S = \langle 3, 5 \rangle$

Supongamos que introducimos el semigrupo numérico $\langle 3, 5 \rangle$. Este conjunto presenta un grupo de generadores que se pueden llegar a conseguir a partir de este semigrupo. Sabiendo que el número de Frobenius es 7, si se introduce en el programa del autómata un número mayor a este, se devuelve que puede ser generado por el conjunto descrito. En el caso de que no se pueda generar cierto número bajo dicho conjunto (en este caso, uno de ellos es el 4), el autómata indicará que no se puede generar bajo el conjunto descrito.

Ejemplo 2: $S = \langle 10, 12, 13, 15 \rangle$

Si introducimos el semigrupo numérico $\langle 10, 12, 13, 15 \rangle$, se generará un conjunto que presenta un grupo de generadores alcanzables a partir de este semigrupo. Sabiendo que el número de Frobenius es 31, si se introduce en el programa del autómata un número mayor a este, se devuelve que puede ser generado por el conjunto descrito. En el caso de que no se pueda generar cierto número bajo dicho conjunto (por ejemplo, el 14 o el 29), el autómata indicará que no se puede generar bajo el conjunto descrito.

Ejemplo 3: $S = \langle 1 \rangle$

Si introducimos el semigrupo numérico $\langle 1 \rangle$, se generará un conjunto que presenta un grupo de generadores derivado de este semigrupo. Sabiendo que el número de Frobenius es -1 , si se introduce en el programa del autómata un número mayor a este, se devuelve que puede ser generado por el conjunto descrito. En este caso, todos los números naturales pueden ser generados mediante combinaciones de $\{1\}$, es decir, todos los números pueden ser generados.

Estos ejemplos nos muestran como el autómata debe de tener en cuenta el número de Frobenius (sección 2.1) y el género (sección 2.2) para determinar qué números pueden ser generados o no bajo el semigrupo numérico dado. Las propiedades que nos proporcionan los autómatas nos permite determinar si un semigrupo numérico puede cumplir con ciertas condiciones, reduciendo

y optimizando la combinatoria para determinar si a través de combinaciones lineales no negativas se pueden generar ciertos números a partir de este.

3.1.3. Utilidades

El uso de autómatas con semigrupos numéricos permite aprovechar la estructura algebraica de estos para diseñar modelos computacionales robustos y de gran eficiencia. Las utilidades que presentan son diversas.

En primer lugar, los semigrupos numéricos, definidos como subconjuntos de los enteros no negativos cerrados bajo la suma, presentan propiedades algebraicas que se muestran directamente en la construcción del autómata. Cada elemento generable se identifica naturalmente con un *estado* en el autómata, mientras que la operación de suma se modela como una *transición determinista*. Esta correspondencia permite diseñar autómatas que simulan de manera exacta la evolución de un semigrupo numérico, facilitando la representación y análisis de procesos secuenciales en ámbitos tan variados como la teoría de números y la optimización algorítmica.

Otra utilidad crucial es la capacidad para automatizar la verificación y generación de elementos del semigrupo. Por medio del autómata se puede determinar, de forma rápida y determinista, si un número dado es *generable* a partir de un conjunto de generadores. Esto es especialmente útil en aplicaciones como la validación de contraseñas, la compresión de datos o el procesamiento de lenguaje natural, donde es fundamental establecer criterios claros de aceptación basados en propiedades numéricas (por ejemplo, mediante el uso del número de Frobenius y la serie de Hilbert).

Además, el enfoque de autómatas basado en semigrupos permite optimizar algoritmos al reducir la complejidad de operaciones repetitivas. La representación de la suma como una función de transición optimizada refuerza la capacidad del sistema para manejar grandes volúmenes de datos en problemas combinatorios o de optimización, pues se pueden minimizar los estados redundantes y mejorar la eficiencia computacional mediante técnicas de minimización y agrupamiento de generadores.

Por otra parte, la relación entre autómatas y semigrupos numéricos favorece el desarrollo de soluciones híbridas en las que se integran métodos algebraicos y computacionales. Esta combinación resulta muy valiosa en el análisis de problemas como el cálculo del número de Frobenius, la determinación del conductor o la generación de la serie expandida de Hilbert. El autómata actúa como un mecanismo para explorar sistemáticamente el espacio de soluciones, identificando patrones y ofreciendo perspectivas tanto teóricas como prácticas para la resolución de problemas matemáticos y de

ingeniería.

La utilidad de estos autómatas se extiende a variados campos, desde la seguridad y la criptografía hasta la investigación en teoría de lenguajes y sistemas de control. La capacidad para reconocer patrones, validar condiciones de aceptación y automatizar la generación de elementos bajo reglas algebraicas rígidas abre nuevas posibilidades en la creación de algoritmos predictivos, la simulación de sistemas discretos y el análisis de estructuras numéricas complejas, aportando así herramientas de gran valor tanto en la investigación matemática como en aplicaciones tecnológicas avanzadas.

3.1.4. Aplicaciones

Si hablamos de aplicaciones a la hora de integrar los conceptos de los autómatas con semigrupos numéricos vemos que hay un beneficio mutuo entre la teoría matemática, y las aplicaciones prácticas en diversos campos como lo son la informática, la criptografía, la optimización y el análisis de lenguajes formales. Esta sinergia se basa en la capacidad de modelar propiedades algebraicas (como el cierre bajo la suma, los generadores mínimos o el número de Frobenius) en un marco computacional mediante autómatas deterministas. Se va a ver algunas de las aplicaciones más relevantes.

1. Optimización de algoritmos y reconocimiento de patrones.

El uso de autómatas para modelar semigrupos numéricos permite construir algoritmos eficientes para el reconocimiento y la validación de secuencias numéricas. Esta técnica es esencial en problemas combinatorios, en los que la detección de patrones y la verificación de la generabilidad de números, a partir de un conjunto de generadores, se traducen en mejoras significativas en términos de velocidad y ahorro de recursos computacionales.

2. Aplicaciones en criptografía y seguridad informática.

La generación y verificación de números mediante procesos autómatas, basados en propiedades de semigrupos numéricos, ofrece una novedosa aproximación a la creación de claves criptográficas. Al aprovechar propiedades como el número de Frobenius y la serie de Hilbert, es posible diseñar sistemas de generación de contraseñas o algoritmos de encriptación que incorporen la complejidad inherente de la estructura numérica, aumentando así la robustez y la seguridad de los sistemas.

3. Teoría de lenguajes y procesamiento de datos.

Los autómatas son herramientas fundamentales en la teoría de lenguajes formales. Al relacionarlos con semigrupos numéricos, se pueden diseñar modelos que no solo reconozcan patrones de símbolos, sino que también integren propiedades aritméticas en la evaluación de dichos

lenguajes. Esto resulta útil en la compilación y análisis sintáctico, donde ciertos lenguajes requieren restricciones numéricas específicas para su validación.

4. Análisis y resolución de problemas combinatorios.

En problemas como el cálculo del número de Frobenius o la determinación de la serie expandida de Hilbert, los autómatas permiten explorar secuencialmente el espacio de posibles soluciones. Esta capacidad de simular transiciones de estado (a partir de la suma de generadores) facilita una aproximación sistemática a problemas combinatorios complejos, lo que se traduce en métodos de solución más eficientes y en una mayor comprensión estructural del problema.

5. Modelado de sistemas discretos y automatización.

La representación de los estados numéricos y de las transiciones mediante autómatas proporciona una plataforma versátil para modelar sistemas discretos. En contextos de automatización industrial o sistemas de control, esta metodología permite el diseño de sistemas predictivos y adaptativos que optimizan procesos, basándose en la evolución y transición de estados definidos algebraicamente.

6. Aplicaciones híbridas y multidisciplinarias.

La unión de métodos algebraicos, combinatorios y computacionales a través de autómatas basados en semigrupos numéricos promueve el desarrollo de estrategias híbridas. Estas se aplican en áreas emergentes como la inteligencia artificial y la ciencia de datos, donde la detección de patrones y la optimización de procesos requieren modelos que integren información estructural y operativa de manera simultánea.

Podemos ver que el uso de autómatas combinados con semigrupos numéricos no solo facilita el análisis matemático y computacional, sino que potencia una variedad de aplicaciones prácticas. Desde la criptografía hasta la resolución de problemas combinatorios, estas herramientas ofrecen soluciones robustas y versátiles que aprovechan la riqueza estructural de los semigrupos para abordar desafíos complejos en entornos reales.

3.1.5. Programa de cálculo de un autómata

Existen multitud de programas que se pueden realizar para unir la teoría de autómatas con los semigrupos numéricos. En nuestro caso, vamos a ver un programa cuya finalidad es comprobar que, dado un semigrupo numérico, verificar si los números que inserta el usuario pueden ser generados o no bajo dicho semigrupo. Para ello se ha usado el número de Frobenius para determinar que aquellos números que sean mayores a este pueden ser generados bajo combinaciones del conjunto. En caso de que se hayan introducido números

que son menores al número de Frobenius, entonces el autómata genera la serie expandida de Hilbert para determinar si ese número puede ser generado bajo combinaciones del conjunto.

Vamos a usar algunos métodos que hemos explicado anteriormente, como la validación de semigrupos numéricos (`mcd` y `reducemcd` explicados en la sección 2.1.6), el número de Frobenius (también explicado en la sección 2.1.6), y la forma expandida de Hilbert, la cual usaremos en caso de tener que generar las combinaciones lineales para comprobar si los números proporcionados se pueden generar o no con el semigrupo insertado (método descrito en la sección 2.4.5). Tendremos dos variables globales las cuales serán aquellas donde se almacene el número de Frobenius y la serie expandida de Hilbert. Luego, en un método se llamará a las funciones que nos calcula el número de Frobenius y la serie expandida de Hilbert almacenándolos en las variables globales, para posteriormente considerar si los elementos insertados por el usuario se encuentran o no en el semigrupo numérico que él nos ha proporcionado.

Una vez hemos calculado los datos necesarios para comprobar el funcionamiento del autómata, vamos a verificar si dado un número o varios, estos se encuentran o no bajo el semigrupo numérico previamente proporcionado.

Este método tiene la finalidad de comprobar que el número o números que ha insertado el usuario están o no en el semigrupo numérico dado. Para ello, hace diversas técnicas de validación a partir de un bloque condicional en donde se comprueban distintas validaciones:

1. Si el número es 0, entonces siempre se encuentra en cualquier semigrupo numérico que se nos proporcione.
2. En el caso de que el número sea mayor que el de Frobenius, sabemos que va a pertenecer al semigrupo (recordemos que el número de Frobenius es el mayor elemento que no puede ser generado bajo un semigrupo numérico.)
3. En caso de que, al calcular la serie expandida de Hilbert, ese número se encuentre en dicha serie, entonces pertenecerá al semigrupo.
4. En cualquier otro caso, ese número no se puede generar bajo combinaciones lineales del semigrupo numérico dado.

Se ha hecho el programa de tal manera que si se valida en un bloque condicional donde se tiene en cuenta esta enumeración, se optimiza el tiempo de cómputo del programa. Esto es dado a que si el número es 0 o mayor al número de Frobenius, podemos indicar sin comprobar combinaciones de que

ese número se encuentra presente y, en caso de que el número sea menor al Frobenius y distinto de 0, se genera la serie expandida de tal forma que compruebe en ese rango si el número se puede generar. Esto nos optimiza el programa de manera considerable al tener en consideración otras alternativas antes de generar combinaciones lineales. Además, recordemos que el usuario puede insertar varios números (no tiene por qué ser uno solo), en lo cual el programa se encargará de comprobar si pueden ser generados o no cada uno de ellos bajo el semigrupo numérico descrito.

3.2. Seguridad

La seguridad informática es fundamental para proteger datos sensibles y sistemas críticos. En este contexto, las contraseñas ultra seguras desempeñan un papel esencial, ya que actúan como la primera línea de defensa contra accesos indebidos y ataques cibernéticos.

Una contraseña ultra segura debe cumplir con las siguientes características:

- **Longitud y complejidad:** debe ser suficientemente larga y combinar letras mayúsculas, minúsculas, números y caracteres especiales para aumentar la entropía.
- **Única y aleatoria:** no debe reutilizarse en diferentes servicios, y es recomendable que sea generada de forma aleatoria para evitar patrones predecibles.
- **Protección contra ataques:** la combinación de alta complejidad y aleatoriedad reduce la vulnerabilidad frente a ataques de fuerza bruta o basados en diccionarios.

En el ámbito de la programación, es crucial implementar técnicas que aseguren el almacenamiento y manejo adecuado de contraseñas:

- **Hashing seguro:** el uso de algoritmos de hashing fuertes, como `bcrypt` o `Argon2`, que convierten las contraseñas en valores irreversibles. Esto garantiza que, incluso si se compromete la base de datos, las contraseñas permanezcan protegidas.
- **Autenticación multifactor (MFA):** complementar la verificación basada en contraseñas con métodos adicionales (por ejemplo, códigos temporales, autenticación biométrica o dispositivos físicos) para añadir capas extra de seguridad.
- **Gestores de contraseñas:** emplear administradores de contraseñas que generen y almacenen claves complejas, facilitando la gestión y evitando prácticas inseguras, como la reutilización de contraseñas.

Por tanto, combinar contraseñas ultra seguras con técnicas de hashing avanzado y autenticación multifactor es esencial para fortalecer la seguridad de los programas informáticos y salvaguardar la integridad de los datos en la era digital. Sin embargo, nosotros nos vamos a enfocar en el uso de semigrupos numéricos para generar estas contraseñas ultra seguras de manera que sea casi inhackeable. Nos vamos a aprovechar de sus propiedades algebraicas y combinatorias para construir secuencias numéricas con alta entropía. Por tanto, a partir de un subconjunto de los enteros no negativos cerrado bajo la suma nos permite generar una secuencia de números a partir de un conjunto de *generadores*, lo cual se puede usar para la generación de contraseñas ultra seguras.

3.2.1. Propiedades

El uso de semigrupos numéricos para generar contraseñas ultra seguras aporta un dinamismo y una robustez extra a los mecanismos de seguridad. En este enfoque, las contraseñas se derivan a partir de combinaciones de generadores de un semigrupo numérico, y se actualizan de forma periódica al cambiar las combinaciones, incrementando así la impredecibilidad. A continuación, se detallan las propiedades fundamentales de este método:

1. **Alta entropía.** La capacidad combinatoria inherente a un semigrupo numérico permite generar una gran cantidad de combinaciones posibles a partir de un conjunto de generadores. Esto incrementa la entropía de la contraseña, dificultando la predicción y reduciendo la efectividad de ataques de fuerza bruta o basados en diccionarios.
2. **Dinamismo y actualización periódica.** El mecanismo que permite cambiar las combinaciones del semigrupo de forma periódica introduce dinamismo en la generación de contraseñas. Al actualizarse la estructura combinatoria en intervalos definidos, la contraseña generada varía con el tiempo, lo que refuerza su seguridad al hacerla transitoria y menos vulnerable a ataques prolongados.
3. **Impredecibilidad.** La naturaleza algebraica y combinatoria de los semigrupos numéricos proporciona una base no lineal para la generación de contraseñas. Esto genera secuencias difíciles de predecir, ya que incluso cambios sutiles en el conjunto de generadores pueden producir salidas radicalmente diferentes.
4. **Robustez matemática.** El uso de conceptos como el número de Frobenius, la serie de Hilbert y los conjuntos de Apéry en la construcción del semigrupo añade una capa de rigor matemático. Esta solidez teórica contribuye a la fiabilidad del mecanismo de generación, ofreciendo una estructura compleja y resistente a técnicas de análisis predictivo.

5. **Adaptabilidad y flexibilidad.** La metodología permite mapear los números generados a una amplia gama de caracteres (alfanuméricos y símbolos especiales) mediante funciones de conversión personalizadas. Esto hace posible adaptar el método a distintos requisitos de seguridad y políticas de contraseñas, lo que lo convierte en una herramienta versátil para diversas aplicaciones informáticas.
6. **Resistencia a ataques.** La combinación de alta entropía, actualización dinámica y una estructura combinatoria no predecible protege contra ataques convencionales. Cada nueva combinación del semigrupo actúa como una “nueva contraseña”, lo que disminuye en gran medida la ventana de oportunidad para que un atacante pueda explotar vulnerabilidades basadas en contraseñas fijas.

La integración de semigrupos numéricos en la generación de contraseñas no solo aporta una alta seguridad a través de una gran entropía y resistencia a ataques, sino que además facilita la implementación de sistemas dinámicos en los que las contraseñas se renuevan automáticamente. Esto mejora significativamente la protección de sistemas y datos críticos en entornos informáticos.

3.2.2. Ejemplos

Los ejemplos vienen descritos a partir del programa que se explicará en la sección 3.2.5. Se trata de un código el cual, dado un semigrupo numérico de generadores minimales forma de manera dinámica una contraseña ultra segura a partir de una combinación válida usando los generadores del semigrupo. Esta contraseña irá cambiando cada 10 segundos, en donde se generará otra combinación lineal diferente dado por el mismo semigrupo numérico de generadores minimales.

Ejemplo 1: $S = \langle 3, 5 \rangle$

Supongamos que se introduce el sistema minimal de generadores $\langle 3, 5 \rangle$. Las combinaciones que se pueden hacer con este semigrupo son

$$\{0, 3, 5, 6, 8, 9, 10, 11, 12, \dots\}.$$

Este semigrupo produce números combinando 3 y 5 mediante sumas y productos. Las contraseñas generadas son únicas y cambian cada 10 segundos.

Ejemplo de contraseña inicial: !@#D5wZ1&aQdm

Contraseña tras 10 segundos: %Ez7v^KNLm4

Ejemplo 2: $S = \langle 4, 7 \rangle$

Supongamos que se introduce el sistema minimal de generadores $\langle 4, 7 \rangle$. Las combinaciones que se pueden hacer con este semigrupo son

$$\{0, 4, 7, 8, 11, 12, 14, 15, 16, 18, \dots\}.$$

En este caso, el semigrupo combina 4 y 7, produciendo números con una estructura equilibrada. Las contraseñas generadas son únicas y cambian cada 10 segundos.

Ejemplo de contraseña inicial: \$W@gKl9&3dX4!

Contraseña tras 10 segundos: Qr#7v@22Ym&E^

Ejemplo 3: $S = \langle 6, 10, 15 \rangle$

Supongamos que se introduce un semigrupo generado por $\langle 6, 10, 15 \rangle$. Las combinaciones que se pueden hacer con este semigrupo son

$$\{0, 6, 10, 12, 15, 16, 18, 20, \dots\}.$$

Este semigrupo creará contraseñas ultra seguras a partir de combinaciones lineales no negativas de los elementos del semigrupo.

Ejemplo de contraseña inicial: X#9v&Jw0e3^2z

Contraseña tras 10 segundos: AqL8%@6mZtYD!

Ejemplo 4: $S = \langle 2, 4 \rangle$

En el caso de que se introduzcan números que no formen un semigrupo numérico, como $\langle 2, 4 \rangle$, el conjunto no cumple con la condición de generar un semigrupo numérico (ya que el MCD no es 1). En este escenario, el programa notifica al usuario que el semigrupo es inválido y no genera contraseñas dinámicas hasta que se ingrese un semigrupo válido.

3.2.3. Utilidades

La creación de contraseñas ultra seguras mediante semigrupos numéricos combina fundamentos matemáticos rigurosos con técnicas computacionales avanzadas. La estructura algebraica de un semigrupo definido como un conjunto de enteros no negativos cerrado bajo la suma—permite generar secuencias numéricas complejas que, al transformarse en cadenas de caracteres, ofrecen diversas utilidades fundamentales para la seguridad informática. Algunas de las utilidades más relevantes son.

1. **Derivación determinista de llaves maestras.**

Una utilidad es la generación automática a partir de un único “semigrupo numérico secreto” y otros parámetros (ID, fecha, usuario), llaves maestras independientes. De esta forma cada aplicación o servicio recibe su propia clave, con lo que basta con conocer el semigrupo numérico y los metadatos para generar la clave.

2. **Tokens de un solo uso (OTP/TOTP) basados en semigrupos numéricos**

Se puede sustituir el clásico HMAC-SHA1 de las OTP por un algoritmo que tome como semilla un semigrupo, donde en cada intervalo de tiempo se combinen distintos generadores para producir códigos con varios dígitos. Esto permite reducir ataques a esquemas tradicionales a la hora de introducir una estructura algebraica subyacente no lineal.

3. **Segmentación de accesos y permisos por parámetros de generadores.**

Se puede llegar a definir varios semigrupos numéricos parametrizados (por departamento, rol, proyecto, etc) en donde cada uno de estos crea credenciales con derechos limitados a cada ámbito. El servidor de autorización valida la contraseña consultando los generadores que se aplican y consiguiendo un control de acceso de grado fino sin bases de datos extras.

4. **Auditoría y trazabilidad criptográfica.**

Se puede llegar a registrar un log inmutable (blockchain ligero o Merkle tree) cuando se actualiza el semigrupo numérico o se generan credenciales.

5. **Integración con hardware seguro y tokens físicos.**

La incorporación del semigrupo numérico en un HSM, TPM o tarjeta inteligente puede ser útil dado a que los generadores del semigrupo quedan “firmados” en el módulo y solo este puede ejecutar las combinaciones. El ordenador, móvil, u otro dispositivo nunca va a ver la semilla completa, recibiendo solamente la contraseña generada resultante, haciendo que el proceso de generación de credenciales quede protegido contra malware o agentes maliciosos en el cliente.

La generación de contraseñas ultra seguras mediante semigrupos numéricos presenta una solución innovadora y creativa, altamente adaptable y efectiva para proteger sistemas informáticos y datos sensibles. La combinación de alta entropía, actualización dinámica y rigor matemático representa un gran avance en el ámbito de la seguridad digital.

3.2.4. Aplicaciones

El uso de semigrupos numéricos para generar contraseñas ultra seguras tiene un amplio rango de aplicaciones prácticas, gracias a su capacidad para combinar dinamismo, robustez matemática y alta entropía. A continuación, se presentan algunas de las aplicaciones más destacadas:

1. Seguridad en sistemas bancarios y financieros.

En el ámbito financiero, donde la protección de datos sensibles es crítica, las contraseñas dinámicas basadas en semigrupos numéricos ofrecen una solución robusta. La capacidad de actualizar automáticamente las contraseñas en intervalos regulares reduce significativamente el riesgo de ataques prolongados, como el phishing o la fuerza bruta.

2. Autenticación en infraestructuras críticas.

En sectores como la energía, la salud o el transporte, donde la seguridad de los sistemas es esencial, las contraseñas generadas mediante semigrupos numéricos pueden integrarse en sistemas de autenticación multifactor. Esto refuerza la protección contra accesos no autorizados y asegura la continuidad operativa.

3. Protección de datos en la nube.

Con el creciente uso de servicios en la nube, la generación de contraseñas dinámicas y altamente seguras es clave para proteger la información almacenada. Los semigrupos numéricos permiten crear contraseñas únicas y complejas que dificultan los intentos de acceso no autorizado.

4. Aplicaciones en criptografía:

Las propiedades algebraicas de los semigrupos numéricos pueden integrarse en algoritmos criptográficos para generar claves seguras. Estas claves, derivadas de combinaciones numéricas complejas, son ideales para cifrar datos y proteger comunicaciones sensibles.

5. Seguridad en dispositivos IoT.

En el Internet de las Cosas (IoT), donde los dispositivos suelen ser vulnerables a ataques, las contraseñas dinámicas basadas en semigrupos numéricos pueden proporcionar una capa adicional de seguridad. Esto es especialmente útil en dispositivos que requieren autenticación constante para interactuar con redes o sistemas.

6. Acceso seguro a redes y sistemas empresariales.

En entornos corporativos, las contraseñas generadas dinámicamente pueden utilizarse para proteger el acceso a redes internas, bases de datos y sistemas críticos. La actualización periódica de las contraseñas minimiza el riesgo de comprometer credenciales.

7. Aplicaciones en juegos y plataformas digitales.

En plataformas de videojuegos y servicios digitales, donde los usuarios manejan cuentas con datos personales y financieros, las contraseñas dinámicas ofrecen una experiencia segura y confiable. Además, el dinamismo de las contraseñas puede integrarse como una característica innovadora para atraer a usuarios preocupados por la seguridad.

8. Sistemas de control de acceso físico.

En sistemas de control de acceso a instalaciones físicas, como oficinas, laboratorios o centros de datos, las contraseñas basadas en semigrupos numéricos pueden utilizarse para generar códigos de acceso temporales. Esto asegura que los códigos sean únicos y difíciles de predecir.

Podemos ver en consecuencia que las contraseñas ultra seguras basadas en semigrupos numéricos tienen aplicaciones en una amplia variedad de sectores, desde la protección de datos personales hasta la seguridad de infraestructuras críticas. Su capacidad para generar contraseñas dinámicas, únicas y altamente seguras las convierte en una herramienta versátil y efectiva para enfrentar los desafíos de seguridad en la era digital.

3.2.5. Programa de cálculo sobre contraseñas ultra seguras

Se ha realizado un programa el cual muestra el funcionamiento de los semigrupos numéricos a la hora de generar contraseñas ultra seguras. La idea es que, dado un sistema minimal de generadores de un semigrupo numérico, se pueda generar contraseñas a partir de la vinculación de caracteres a los distintos elementos que se pueden formar a partir de dicho semigrupo. Estos semigrupos tienen propiedades únicas que los hacen ideales para producir combinaciones numéricas potentes y complejas para generar contraseñas ultra seguras con el objetivo de poder cifrar nuestros programas.

Se tienen en cuenta las combinaciones lineales posibles que se pueden realizar con un semigrupo numérico, por lo que podemos formar múltiples contraseñas a partir de un semigrupo numérico de generadores minimales. Además, esta contraseña cambiará de manera dinámica cada 10 segundos (es decir, se establece otra combinación lineal a partir de los generadores del semigrupo), lo cual hace casi impredecible conocer dicha contraseña.

La idea es que esta implementación se pueda incrustar dentro de programas como bases de datos, datos protegidos de bancos, contraseñas internas de cuentas, entre otros, para hacer más difícil la accesibilidad de la información privada a los hackers. Al ser contraseñas que se generan dinámicamente y que cambien cada 10 segundos gracias a las diversas combinaciones que tiene un semigrupo numérico, podemos hacer que estos datos sean casi inac-

cesibles.

Vamos a usar 3 variables globales:

- **mapDinamico**: es el mapa dinámico para la contraseña.
- **contraGenerada**: almacenará la contraseña que se genera dinámicamente.
- **interContrasenia**: controla el intervalo de generación a tener en cuenta cuando se genera la contraseña a partir del semigrupo numérico (inicialmente a nulo).

Lo primero de todo es la validación de que estamos trabajando con semigrupos numéricos, por lo que usaremos los métodos *mcd* y *reduce mcd* los cuales se explicaron en la sección 2.1.6. También usaremos la forma expandida de Hilbert, en donde en este caso vamos a generar una cantidad razonable de elementos que viene determinada por los elementos del semigrupo numérico (si tiene más valores o valores más altos, se generan más), el límite predefinido (se establece la cantidad máxima de términos que pueden generarse), y se tiene la condición de que solo se incluyen aquellos términos que cumplen el criterio de estar dentro del límite y que no hayan sido generados previamente.

El código de la serie expandida de Hilbert ha sido modificado ligeramente para tener en cuenta más cantidad de términos y tener un límite altamente razonable para poder hacer múltiples combinaciones dado el semigrupo numérico.

Debemos de tener en cuenta que el semigrupo numérico sea minimal, por lo que usaremos el método *esMinimal* descrito en la sección 2.5.5 para validar la entrada comprobando que el semigrupo numérico presenta un sistema minimal de generadores.

Ahora, para generar contraseñas ultra seguras y dinámicas a partir de un sistema minimal de generadores de un semigrupo numérico, vamos a configurar un sistema que genera y actualiza una contraseña dinámica de manera continua, garantizando que dicha contraseña se genere lo antes posible y que el sistema permanezca sincronizado con nuevas contraseñas cada 10 segundos. Además, nos debemos de asegurar de que las contraseñas se generen a partir del semigrupo numérico proporcionado, por lo que previamente debemos de generar aquellos términos que pueden ser formados a partir del semigrupo (esto lo hacemos gracias al método de la forma expandida de Hilbert).

Para asignar caracteres a los números generados por combinaciones lineales del semigrupo numérico de generadores minimales, vamos a usar el código ASCII en el rango de 33-126. Este rango corresponde a:

- **33-47:** Símbolos de puntuación y caracteres especiales como !, ", #, \$, %, etc.
- **48-57:** Los dígitos numéricos del 0 al 9.
- **58-64:** Más caracteres especiales como :, ;, <, =, >, ?, @.
- **65-90:** Letras mayúsculas del alfabeto inglés (A a Z).
- **91-96:** Símbolos como [, \,], ^, _, '.
- **97-122:** Letras minúsculas del alfabeto inglés (a a z).
- **123-126:** Más caracteres especiales como {, |, }, ~.

De esta forma nos podemos asegurar de que la contraseña que se genere sea ultra segura (usaremos a partir de los elementos del semigrupo numérico combinaciones lineales no negativas para poder formar la contraseña a partir de estos caracteres). Aunque se usen los elementos del semigrupo, la combinación que se haga será aleatoria, aunque siempre teniendo presente que se forme a partir de los elementos que se puedan generar a partir del semigrupo en cuestión. Esta contraseña estará comprendida en 16 caracteres.

Por otro lado, debemos de tener una forma de verificar que la contraseña sea correcta. Para ello se ha creado un método para que se pueda verificar la contraseña generada a partir del semigrupo numérico usado. No obstante, debemos de ser conscientes de que esta contraseña solo puede ser verificada en un rango de 10 segundos ya que al presentar dinamismo la contraseña sería invalida cada dicho tiempo (se generará otra contraseña diferente a partir de otra combinación del semigrupo).

En conclusión, podemos observar el potencial que tiene el dinamismo de generar contraseñas ultra seguras a partir de generaciones aleatorias de elementos que pueden ser formados a partir de semigrupos numéricos. El gran uso del dinamismo y la aleatoriedad de uso de combinaciones a partir del rango descrito del código ASCII hacen que la verificación de contraseñas sea más seguro y complejo de descifrar, haciéndolo más desafiante para los hackers que intenten adentrarse en cualquier programa que use esta filosofía.

Esta generación de contraseñas dinámicas se debe de hacer de manera interna en los programas, en donde el mismo pueda hacer comunicaciones a partir de estas contraseñas a la hora del transporte de datos. El uso complejo

de los semigrupos numéricos y la alta aleatoriedad a partir de las combinaciones de estos añade un extra de complejidad a la hora de descifrar dichas contraseñas en el paso de comunicación.

3.3. Problema de la mochila

El problema de la mochila es un clásico problema de optimización combinatoria. Se plantea como el desafío de seleccionar un subconjunto de elementos, cada uno con un peso y un valor asignado, de manera que se maximice el valor total sin exceder una capacidad predefinida. Entre sus variantes, destaca la versión 0-1, donde cada elemento puede ser tomado una única vez, y la versión fraccionaria, en la que se permite dividir elementos. Debido a su importancia teórica y sus numerosas aplicaciones prácticas, desde la logística hasta la toma de decisiones financieras, el problema de la mochila ha generado una amplia variedad de algoritmos y técnicas, como la programación dinámica y los métodos voraces, que continúan siendo áreas activas de investigación.

La integración del problema de la mochila con la teoría de semigrupos numéricos propone un enfoque novedoso para la optimización de combinaciones de objetos. En esta implementación, se consideran los pesos de los objetos como generadores de un semigrupo numérico, lo que permite construir, mediante sumas iterativas, el conjunto de todos los valores alcanzables. De esta manera, se pueden explorar sistemáticamente las combinaciones de pesos que no excedan la capacidad máxima de la mochila.

La metodología consiste en definir el conjunto de pesos disponibles y calcular las sumas combinatorias resultantes, las cuales representan los posibles totales acumulables. A partir de este conjunto, se selecciona la combinación óptima que maximiza el valor o beneficio sin superar la restricción de peso. Este enfoque permite aprovechar propiedades algebraicas inherentes al semigrupo para reducir la complejidad en la búsqueda de la solución, ofreciendo una alternativa potencialmente más eficiente frente a métodos tradicionales.

3.3.1. Propiedades

La unión del problema de la mochila con la teoría de los semigrupos numéricos ofrece un marco analítico innovador para abordar la optimización combinatoria. Al modelar el conjunto de combinaciones de pesos (o valores) como un semigrupo numérico, se obtienen propiedades algebraicas que permiten explorar y reducir el espacio de soluciones. A continuación, se detallan las principales propiedades y ventajas de esta integración.

1. Representación de soluciones como sumas cerradas.

En el problema de la mochila, el objetivo es seleccionar un subconjunto de elementos cuyos pesos se sumen a un valor que no exceda la capacidad. Al interpretar los pesos como generadores de un semigrupo numérico, se obtiene un conjunto de todos los posibles totales alcanzables mediante sumas. Esta representación cerrada permite visualizar la totalidad del espacio factible y facilita la identificación de combinaciones óptimas.

2. Aplicación del número de Frobenius y el conductor.

Conceptos propios de los semigrupos, como el número de Frobenius (el mayor entero no representable) y el conductor (el mínimo entero a partir del cual todos los números son alcanzables), ofrecen información valiosa acerca de la estructura del problema. En el contexto de la mochila, estos parámetros ayudan a delimitar la región en la que las combinaciones pueden ser inalcanzables o, en forma inversa, garantizan la completitud de la suma a partir de cierto umbral.

3. Reducción y optimización del espacio de búsqueda.

La propiedad de cierre bajo la suma en un semigrupo implica que, una vez alcanzado un determinado valor, todas las combinaciones superiores se pueden generar. Esto permite implementar criterios de poda en algoritmos de optimización, reduciendo la necesidad de explorar exhaustivamente todas las combinaciones y acelerando la búsqueda de la solución óptima en el problema de la mochila.

4. Estructura algebraica y periodicidad.

La estructura algebraica de los semigrupos numéricos brinda herramientas teóricas para analizar la periodicidad y repetición de combinaciones de sumas. Este análisis permite diseñar algoritmos que aprovechen patrones repetitivos en la distribución de los pesos, facilitando la implementación de técnicas de programación dinámica o métodos heurísticos que mejoran tanto la precisión como la eficiencia.

5. Flexibilidad para problemas híbridos.

Al integrar conceptos de semigrupos con la formulación clásica del problema de la mochila, se abren las puertas a métodos híbridos que combinan análisis algebraico con algoritmos convencionales. Esto es especialmente útil en aplicaciones con restricciones complejas o en escenarios donde se requiere tanto exactitud matemática como agilidad computacional.

6. Enfoque sistemático para validar combinaciones.

La implementación de la teoría de semigrupos permite validar de forma sistemática si un número determinado (correspondiente a un total de pesos) es alcanzable a partir de un conjunto de generadores. Esta propiedad se traduce en mecanismos de verificación que aseguran la

factibilidad de la solución, evitando la exploración ineficaz de combinaciones que no cumplen con los requisitos del problema.

En resumen, la unión del problema de la mochila con los semigrupos numéricos aporta un enfoque robusto que:

- Permite una representación completa y cerrada del espacio de soluciones mediante sumas.
- Utiliza propiedades algebraicas (como el número de Frobenius y el conductor) para delimitar y optimizar la búsqueda.
- Facilita el desarrollo de algoritmos híbridos que combinan el rigor matemático con la eficiencia computacional.

Esta combinación del problema de la mochila con el uso de semigrupos numéricos mejora la capacidad de resolver problemas combinatorios complejos y ofrece nuevas perspectivas en el diseño de algoritmos de optimización.

3.3.2. Ejemplos

Hemos hecho un programa el cual a partir de un semigrupo numérico y la capacidad de la mochila dada, el programa buscará una combinación en donde se pueda ver si la mochila se puede llenar a partir de los elementos del semigrupo y qué elementos son aquellos que se usan. La implementación de este programa se explicará con más detalle en la sección 3.3.5

Ejemplo 1

Supongamos que $S = \langle 5, 7 \rangle$ y la capacidad de la mochila es 38, el programa buscará combinaciones de estos números que maximicen la capacidad. Una posible combinación sería:

$$(2 \times 5) + (4 \times 7)$$

alcanzando un valor máximo de 38.

Ejemplo 2

Supongamos que $S = \langle 3, 4, 7 \rangle$ y la capacidad es 24, la combinación utilizada sería:

$$(8 \times 3)$$

logrando un valor máximo de 24.

Ejemplo 3

Supongamos que $S = \langle 2, 19, 21 \rangle$ y la capacidad es 17, el programa encontrará que la mejor solución es:

$$(8 \times 2)$$

con un valor máximo de 16 (menor dado que no existe ninguna combinación con ese semigrupo que nos de 17).

Ejemplo 4

Supongamos que $S = \langle 50, 57, 64 \rangle$ y la capacidad es 783, el programa encontrará que la mejor solución es:

$$(3 \times 50) + (1 \times 57) + (9 \times 64)$$

logrando un valor máximo de 783.

Ejemplo 5

Supongamos que $S = \langle 7, 9 \rangle$ y la capacidad es 5, el programa encontrará que el valor máximo alcanzable es

$$0$$

esto se debe a que la capacidad de la mochila es menor a los elementos del semigrupo, por lo que no existe ninguna combinación válida.

3.3.3. Utilidades

Hemos podido comprobar que la unión del problema de la mochila con la teoría de semigrupos numéricos ofrece un enfoque innovador para abordar problemas de optimización combinatoria. Este método no sólo enriquece el análisis teórico, sino que también provee ventajas prácticas en la implementación de algoritmos y en la reducción de la complejidad computacional. Las utilidades son las siguientes.

1. Representación completa del espacio de soluciones.

Al modelar los pesos (o incluso los valores) de los objetos mediante un conjunto de generadores, se forma un semigrupo numérico que abarca todas las sumas alcanzables. Esta representación algebraica permite visualizar todas las combinaciones posibles de una manera sistemática, lo que facilita la detección de la solución óptima en el problema de la mochila.

2. Determinación de límites y cotas.

La aplicación de conceptos como el número de Frobenius y el conductor en un semigrupo numérico proporciona límites naturales. Estas medidas permiten establecer cotas inferiores y superiores para las sumas posibles, reduciendo el espacio de búsqueda. De este modo, se pueden descartar combinaciones que claramente exceden la capacidad de la mochila o que no son posibles, optimizando el algoritmo de búsqueda.

3. Estructura y propiedad de cierre.

La propiedad de cierre bajo la suma, inherente a los semigrupos, garantiza que una vez alcanzado un cierto umbral, todas las combinaciones superiores son generables. Esto ayuda a identificar rápidamente rangos en los que la solución es completa y, a partir de dichos rangos, se puede simplificar la exploración de soluciones frente a búsquedas exhaustivas.

4. Reducción de la complejidad computacional.

Al traducir el problema a un marco algebraico basado en semigrupos, se pueden emplear técnicas de poda y métodos de programación dinámica que aprovechan la estructura regular de las combinaciones numéricas. Esto permite disminuir la cantidad de operaciones necesarias para encontrar la solución óptima, haciendo que el proceso sea más eficiente en términos computacionales.

5. Análisis estructural y validación sistemática.

El uso de semigrupos facilita el análisis de la factibilidad de las combinaciones obtenidas. Al disponer de un marco teórico robusto, es posible validar si un número (o combinación) es alcanzable mediante la suma de los generadores. Esta verificación sistemática resulta especialmente útil en instancias complejas y asegura que las soluciones generadas cumplan con todos los requisitos impuestos por el problema de la mochila.

6. Aplicaciones en algoritmos híbridos y multidisciplinarios.

La integración de técnicas algebraicas con métodos tradicionales de optimización abre la puerta a algoritmos híbridos que combinan lo mejor de ambos mundos. Esto es particularmente provechoso en áreas como la planificación logística, la asignación de recursos y problemas financieros, donde se requiere tanto la robustez teórica como la eficiencia práctica en la resolución de problemas.

7. Perspectiva innovadora y nuevas estrategias.

Este enfoque no sólo aporta ventajas prácticas, sino que también fomenta el desarrollo de nuevas estrategias algorítmicas. Al tratar el problema de la mochila desde la óptica de los semigrupos, se abren oportunidades para investigar soluciones basadas en propiedades algebraicas

que, en muchos casos, pueden ofrecer mejoras significativas frente a los métodos convencionales.

En resumen, combinar el problema de la mochila con semigrupos numéricos permite:

- Explorar el espacio de soluciones de forma sistemática y completa.
- Establecer límites claros y reducir el espacio de búsqueda.
- Aprovechar la estructura inherente de las sumas para implementar algoritmos más eficientes y robustos.
- Desarrollar soluciones híbridas que integren rigor matemático y eficiencia computacional.

La optimización combinatoria y la teoría algebraica no solo mejora el rendimiento en la práctica, sino que también enriquece la comprensión teórica de los problemas complejos, ofreciendo nuevas perspectivas en el diseño y análisis de algoritmos de optimización.

3.3.4. Aplicaciones

La combinación de los semigrupos numéricos con el clásico problema de la mochila ofrece un enfoque innovador que une conceptos algebraicos y de optimización combinatoria. Esta integración no solo enriquece la base teórica de ambos temas, sino que también habilita nuevas aplicaciones prácticas en diversos campos. A continuación vamos a ver las principales aplicaciones que pueden derivarse de esta fusión.

1. Optimización de recursos y toma de decisiones.

En problemas donde se debe maximizar el beneficio sin superar una capacidad determinada, la representación de todas las combinaciones posibles a través de un semigrupo numérico permite identificar rápidamente los subconjuntos viables. Esto es útil en logística, planificación de inventarios y asignación de recursos, ya que se puede analizar el espacio de soluciones de manera sistemática para encontrar la combinación óptima.

2. Análisis y reducción del espacio de búsqueda.

La estructura cerrada de un semigrupo numérico —donde a partir de un conjunto de generadores se obtienen todas las sumas alcanzables— ayuda a establecer límites naturales (como el conductor o el número de Frobenius) que permiten reducir el espacio de búsqueda en el problema de la mochila. Esta propiedad permite descartar de manera temprana combinaciones imposibles o irrelevantes, lo que resulta en algoritmos más eficientes y en una mejora de la escalabilidad.

3. Desarrollo de algoritmos híbridos.

La integración de técnicas algebraicas con métodos clásicos de optimización abre la posibilidad de diseñar algoritmos híbridos que aprovechen los puntos fuertes de ambos enfoques. Por ejemplo, se pueden utilizar las propiedades de los semigrupos para precomputar subconjuntos alcanzables y luego aplicar técnicas de programación dinámica o estrategias voraces para seleccionar la mejor combinación, optimizando así la solución del problema de la mochila.

4. Aplicaciones criptográficas y de seguridad.

La generación de combinaciones numéricas complejas mediante semigrupos es una herramienta útil en la creación de claves y contraseñas seguras. Tradicionalmente, se han estudiado sistemas criptográficos basados en el problema de la mochila. Al combinar esta problemática con la teoría de semigrupos, se puede obtener un mayor grado de entropía y dinamismo en las claves generadas, reforzando así la seguridad en aplicaciones sensibles.

5. Resolución de problemas enteros y programación lineal.

El mecanismo de representar números alcanzables mediante combinaciones lineales de generadores se alinea con técnicas utilizadas en la programación entera y lineal. Este enfoque permite determinar la factibilidad de determinadas soluciones y establecer criterios de optimalidad que pueden aplicarse en contextos de optimización matemática y análisis de sistemas de control.

6. Estrategias para la innovación en investigación operativa.

La unión de estos dos paradigmas fomenta nuevas formas de abordar problemas NP-hard. La perspectiva algebraica ofrecida por los semigrupos aporta herramientas conceptuales adicionales para explorar propiedades de completitud y redundancia, lo que puede traducirse en mejoras teóricas y prácticas en la formulación y solución de problemas de optimización combinatoria compleja.

El uso de semigrupos numéricos unido con el problema de la mochila no solo permite modelar de forma efectiva el espacio completo de soluciones, sino que también brinda una serie de ventajas en términos de eficiencia algorítmica, reducción de complejidad y ampliación de los posibles campos de aplicación —desde la seguridad informática hasta la optimización en la gestión de recursos. Esta sinergia representa, en definitiva, un valioso aporte tanto en el ámbito teórico como en el desarrollo de soluciones prácticas para problemas reales.

3.3.5. Programa de cálculo del problema de la mochila con semigrupos numéricos

Se ha realizado un programa el cual a partir de un semigrupo numérico dado y la capacidad de la mochila, poder determinar una combinación eficiente para llenar la mochila hasta su capacidad máxima (en caso de ser posible a partir de las generaciones de elementos del semigrupo).

Para ello, lo primero de todo será validar que la entrada proporcionada sea un semigrupo numérico, lo cual lo hacemos a partir de los métodos *mcd* y *reduce mcd* descritos en la sección 2.1.6.

El problema del cálculo de la mochila se ha hecho teniendo en cuenta las combinaciones posibles las cuales pueden usarse para llenar lo máximo posible la mochila (teniendo en cuenta siempre la capacidad mencionada). Para ello debemos de tener una tabla dinámica que guarde el valor máximo que puede conseguirse para cada capacidad parcial desde 0 hasta la capacidad final, y una tabla de combinaciones para ver aquellas combinaciones posibles que generen el valor máximo limitado por la capacidad. Una vez tengamos todas las combinaciones posibles, se devolverá una única combinación óptima de elementos que suman los más cercanos posible a la capacidad descrita, además del valor máximo alcanzable con dicha combinación.

```

1: procedure PROBLEMAMOCHILA(numeros, capacidad)
2:   tablaDinamica  $\leftarrow$  array de tamaño (capacidad + 1) inicializado con
   0
3:   combinaciones  $\leftarrow$  array de tamaño (capacidad + 1) inicializado con
   listas vacías
4:   for  $i \leftarrow 0$  to capacidad do
5:     for all  $n \in \text{numeros}$  do
6:       if  $i - n \geq 0$  and tablaDinamica[ $i - n$ ] +  $n >$  tablaDinamica[ $i$ ] then
7:         tablaDinamica[ $i$ ]  $\leftarrow$  tablaDinamica[ $i - n$ ] +  $n$ 
8:         combinaciones[ $i$ ]  $\leftarrow$  concatenar(combinaciones[ $i - n$ ],
           $n$ )
9:       end if
10:    end for
11:  end for
12:  return { valorMaximo: tablaDinamica[capacidad], combinaciones:
    combinaciones[capacidad] }
13: end procedure

```

Una forma de poder ver la combinación usada es que a partir de dicha combinación, hacer un conteo de cuántas veces se ha usado cada elemento del semigrupo numérico para alcanzar la máxima capacidad posible (conta-

mos las ocurrencias que se ha usado de cada elemento del semigrupo). En nuestro caso, podemos mostrarlo como suma de productos (tal y como se ha mostrado en los ejemplos de 3.3.2).

Este programa nos ofrece una versión de cómo combinar los semigrupos numéricos para resolver el problema de la mochila a partir de la generación de elementos del semigrupo, en donde nos quedamos con una combinación óptima que llega al valor máximo que se pueda a partir del semigrupo, siempre teniendo en cuenta la capacidad máxima que permite la mochila.

3.4. Semigrupos numéricos de género fijo

La investigación de los semigrupos numéricos que existen con un determinado género en concreto es un concepto desafiante tanto en el área de las matemáticas como de la informática. Dado un conjunto de números enteros positivos primos relativos entre sí,

$$A = \{a_1, a_2, \dots, a_k\},$$

se define un semigrupo numérico como el conjunto formado por todas las combinaciones lineales no negativas de los elementos de A , es decir, todos los números que pueden expresarse como

$$n_1 a_1 + n_2 a_2 + \dots + n_k a_k,$$

con n_i enteros no negativos.

El género de un semigrupo se corresponde con la cantidad de enteros positivos (los huecos) que no se pueden obtener de dichas combinaciones, hasta llegar al conductor, que es el primer número a partir del cual todos los enteros son alcanzables (explicado con mas detalle en la sección de género y conductor).

Así, estudiar los semigrupos numéricos de género fijo consiste en clasificar y analizar aquellos semigrupos que presentan un número determinado de huecos, lo cual aporta claridad en el conocimiento de su estructura y en sus aplicaciones en teoría de números y problemas algebraicos. Por tanto, se deberá de crear un programa el cual dado un número (el género), sea capaz de encontrar todos los semigrupos numéricos que tengan dicho género. Los semigrupos numéricos tienen una estructura de árbol en donde se presentan 2 tipos de semigrupos numéricos a tener en cuenta.

- **Semigrupos numéricos internos:** son aquellos nodos del árbol que poseen descendientes. En otras palabras, un semigrupo interno es aquel a partir del cual es posible obtener al menos un semigrupo de género

$g+1$ (u otro mayor) mediante la eliminación de uno de sus generadores mínimos sin perder la propiedad de ser un semigrupo numérico. Estos nodos indican puntos de posible bifurcación en la estructura del árbol y suelen tener más de una opción de extensión.

- **Semigrupos numéricos hoja:** son aquellos que, además de cumplir con las propiedades de un semigrupo numérico, tienen un número de Frobenius (el mayor entero positivo que no pertenece al semigrupo) que es mayor que el máximo generador del conjunto. Esto los convierte en semigrupos con una estructura más dispersa, donde los huecos y el conductor juegan un papel clave en su clasificación. Los semigrupos hoja son útiles para explorar configuraciones extremas dentro de un género fijo.

Por tanto, podemos ver que son los nodos terminales del árbol, aquellos que no admiten más extensiones o, dicho de otra forma, cuya única posible extensión ya ha sido previamente contabilizada. Dichos semigrupos representan los casos “finales” dentro de la clasificación para un género dado, y su estudio es crucial para comprender la distribución total de semigrupos numéricos de género g .

3.4.1. Propiedades

La importancia de conocer y clasificar los semigrupos numéricos en internos y hojas, para un género fijo tiene bastantes propiedades a tener en consideración.

1. Enumeración y crecimiento.

Conocer cuántos semigrupos numéricos existen para un género dado es un problema central en la teoría. La clasificación en internos y hojas permite estudiar la tasa de crecimiento de esta cantidad, reflejada en la secuencia $N(g)$, que es el número total de semigrupos numéricos para género g . Además, el análisis de la estructura del árbol facilita la generación de recursiones y fórmulas aproximadas para estimar $N(g)$ en función del género g .

2. Estructura del árbol de semigrupos.

La división en internos y hojas proporciona una visión clara de la estructura jerárquica y ramificada del árbol de semigrupos. Los semigrupos internos, al generar múltiples descendientes, reflejan una mayor complejidad combinatoria, mientras que los de hoja indican rutas terminales en esa evolución. Esto permite, además, identificar patrones y simetrías dentro del árbol.

3. Optimización algorítmica.

En la búsqueda y enumeración de semigrupos numéricos de un género

fijo, distinguir entre nodos internos y hojas resulta esencial para optimizar algoritmos. Si se sabe que un nodo es hoja, el algoritmo puede evitar explorar ramas adicionales, reduciendo de manera significativa el espacio de búsqueda y la complejidad computacional.

4. Estudio de invariantes y parámetros.

La clasificación en internos y hojas facilita el análisis de invariantes relevante ya sea la multiplicidad, la dimensión de inmersión y el conductor. Estos parámetros, a su vez, ayudan a caracterizar las propiedades algebraicas y aritméticas de los semigrupos, ofreciendo herramientas para comparar y hacer una clasificación de semigrupos con el mismo género.

5. Aplicaciones en teoría y combinatoria.

La comprensión detallada de la distribución de semigrupos (internos versus hojas) tiene implicaciones en problemas de teoría de números y en la combinatoria algebraica. Por ejemplo, estudiar la forma en que nuevos generadores pueden o no extender un semigrupo permite desarrollar técnicas para la construcción inductiva de semigrupos numéricos y contribuye a la búsqueda de patrones generales en la estructura de estos conjuntos.

Conocer los semigrupos numéricos internos y hojas para un género fijo fortalece la comprensión teórica de la estructura de estos objetos, además de aportar importantes ventajas prácticas en términos de eficiencia algorítmica y análisis combinatorio al área de la informática. Este enfoque integral es bastante relevante para avanzar en la enumeración, clasificación y aplicación de los semigrupos numéricos en distintas áreas de la matemática y la informática.

3.4.2. Ejemplos

Se van a proporcionar ejemplos donde se indica, dado un el valor del género, algunos semigrupos numéricos internos u hojas con ese género fijo.

Ejemplo 1 (Género 1)

Si el género fijo es 1, el único semigrupo interno que hay es $\langle 2, 3 \rangle$. En este caso, el único hueco es 1 y, a partir de 2, todas las combinaciones lineales no negativas forman el semigrupo numérico.

Ejemplo 2 (Género 2)

Para género fijo 2, un semigrupo numérico interno es $\langle 3, 4, 5 \rangle$. Aquí, los únicos huecos son 1 y 2; a partir de 3 se pueden obtener todos los números, lo que define el género como 2.

Ejemplo 3 (Género 3)

Si el género fijo es 3, el único semigrupo numérico hoja resultante es $\langle 3, 4 \rangle$. Los huecos resultantes son $\{1, 2, 5\}$ y el número de Frobenius es 5, mayor que el máximo generador 4.

Ejemplo 4 (Género 4)

Un semigrupo numérico hoja para género fijo 4 es $\langle 3, 5 \rangle$. En este caso, los huecos son $\{1, 2, 4, 7\}$ de acuerdo con las combinaciones posibles, estableciendo el género como 4. Sin embargo, al ser el número de Frobenius mayor que todos los elementos del semigrupo ($n\text{Frob} = 7$), se considera hoja.

Ejemplo 5 (Género 5)

Para género fijo 5, se puede considerar el semigrupo numérico interno $\langle 5, 6, 7, 9 \rangle$. Aquí, tras calcular las combinaciones lineales se obtienen los huecos $\{1, 2, 3, 4, 8\}$, lo que confirma que el género es 5.

3.4.3. Utilidades

Los semigrupos numéricos de género fijo, definidos por el número de enteros que permanecen fuera del semigrupo, constituyen un objeto de estudio fundamental en teoría de números y combinatoria algebraica. Al delimitar la clasificación mediante un género fijo, se agrupan tanto los semigrupos *internos* (aquellos que generan ramas adicionales en el árbol de semigrupos) como los *hoja* (los nodos terminales que no permiten extensiones sin salir del ámbito de género dado). Esta organización estructural permite derivar múltiples utilidades, tanto en el ámbito teórico como en aplicaciones computacionales y algorítmicas. Podemos destacar las siguientes utilidades principales.

1. Enumeración y clasificación estructurada.

El estudio de semigrupos numéricos de género fijo permite enumerar y clasificar de forma sistemática todas las posibles configuraciones que presentan exactamente g huecos. La distinción entre semigrupos internos y de hoja no solo aporta claridad al mapa completo de la estructura, sino que también facilita la formulación de fórmulas recursivas y la obtención de cotas asintóticas para el crecimiento de $N(g)$, el cual se recuerda que es el número total de semigrupos para género g .

2. Modelado y análisis del árbol de semigrupos.

La representación jerárquica en forma de árbol, en la cual los nodos internos corresponden a semigrupos que se pueden extender y las hojas a aquellos terminales, ofrece una manera natural de visualizar la evolución y generación de semigrupos. Este enfoque facilita el análisis de propiedades invariantes (como el número de Frobenius, el conductor

y la multiplicidad) y permite identificar rutas óptimas o restricciones inherentes a la formación de nuevos semigrupos.

3. Optimización de algoritmos de enumeración.

La división en semigrupos internos y hoja es esencial para optimizar algoritmos que enumeran semigrupos numéricos de un género fijo. Al reconocer que los nodos hoja representan caminos terminales, se pueden implementar estrategias de poda en algoritmos recursivos o de programación dinámica, reduciendo significativamente la complejidad computacional necesaria para explorar el espacio de soluciones.

4. Desarrollo de métodos recursivos e inductivos.

La estructura modular que surge de clasificar semigrupos por género facilita el uso de técnicas inductivas. Cada semigrupo de género g puede analizarse en función de sus “predecesores” en el árbol, lo que permite formular teoremas y conjeturas sobre la relación entre los invariantes de semigrupos de géneros consecutivos. Este método inductivo tiene aplicaciones en la prueba de propiedades generales y en la construcción de algoritmos iterativos.

5. Aplicaciones en teoría y combinatoria algebraica.

La comprensión fina de los semigrupos internos y hojas para un género fijo aporta herramientas para estudiar invariantes algebraicos y combinatorios en estructuras numéricas. Esto tiene repercusiones en problemas de representabilidad y en la codificación de información, donde conocer todas las configuraciones posibles (para un determinado “error” o hueco) resulta crucial para la formulación de códigos óptimos.

6. Intersección con otras áreas de las matemáticas.

La clasificación de semigrupos por género fijo se relaciona estrechamente con otros problemas clásicos, como los de los anillos de factores, variedades algebraicas y la teoría de la optimización entera. Estas interrelaciones permiten que los métodos y resultados obtenidos en el estudio de semigrupos numéricos se apliquen a la resolución de problemas en geometría algebraica, teoría de singularidades y optimización combinatoria.

En definitiva, el análisis y la clasificación de semigrupos numéricos de género fijo, distinguiendo entre internos y hojas, profundizan en la base teórica del campo y nos proporcionan herramientas prácticas para el desarrollo de algoritmos eficientes, el estudio de invariantes algebraicos y la resolución de problemas en diversas áreas de la matemática y la informática.

3.4.4. Aplicaciones

Los semigrupos numéricos, definidos como subconjuntos de \mathbb{N} que contienen el cero y son cerrados bajo la suma, se caracterizan mediante invariantes fundamentales; entre ellos, el género, que corresponde al número de huecos (enteros que no aparecen en el semigrupo). Estudiar los semigrupos numéricos de género fijo tiene importantes aplicaciones tanto en el campo teórico como en el práctico. A continuación, se presenta un análisis detallado de las principales aplicaciones de estos objetos.

1. Aplicaciones en teoría de números y combinatoria

- **Enumeración y clasificación.**

Dado que, para un género fijo g , el conjunto de semigrupos numéricos es finito, la clasificación y enumeración de estos objetos se convierten en un problema central de la combinatoria. Los resultados obtenidos permiten establecer relaciones recursivas y estimar el crecimiento de $N(g)$, el número de semigrupos de género g . Esto contribuye a la formulación de conjeturas y teoremas sobre la estructura de estos conjuntos.

- **Estudio de invariantes.**

Los semigrupos de género fijo tienen invariantes relevantes, como el conductor, la multiplicidad y el número de Frobenius. Comprender cómo estos invariantes se comportan y se relacionan entre sí para semigrupos de un mismo género proporciona información importante sobre la distribución de los huecos y la estructura interna del semigrupo.

- **Conexiones con la teoría de particiones.**

La enumeración de semigrupos numéricos de género fijo se relaciona con problemas de partición de enteros. Estas conexiones permiten trasladar técnicas y resultados entre ambas áreas, enriqueciendo el estudio combinatorio y ofreciendo nuevas perspectivas para abordar problemas clásicos.

2. Aplicaciones en geometría algebraica y teoría de curvas

- **Semigrupos de Weierstrass.**

En la geometría algebraica, los semigrupos numéricos aparecen como semigrupos de Weierstrass asociados a puntos de curvas algebraicas. El género del semigrupo coincide con el género de la curva, y analizar los semigrupos de género fijo ayuda a clasificar los posibles comportamientos de las singularidades y a estudiar propiedades de la curva en torno a sus puntos especiales.

- **Singularidades y resolución.**

Los invariantes de los semigrupos numéricos ofrecen herramientas para

investigar las singularidades de curvas. Conocer la estructura de los semigrupos asociados permite formular métodos para la resolución de singularidades y para entender la localización de ciertos invariantes geométricos.

3. Aplicaciones en optimización y algoritmos

- **Optimización combinatoria e implementación algorítmica.**

La estructura de los semigrupos numéricos de género fijo, organizados en forma de árbol (con nodos internos y hojas), facilita la poda y la simplificación de algoritmos de búsqueda y enumeración. Esto es especialmente relevante en problemas de optimización combinatoria, donde la eficiencia computacional depende en gran medida de reducir el espacio de búsqueda.

- **Desarrollo de métodos recursivos:** La naturaleza finita y bien estructurada de los semigrupos de género fijo permite el desarrollo de algoritmos recursivos o inductivos para construir todos los semigrupos de un cierto género. Tales métodos son aplicables en el diseño de algoritmos eficientes para problemas en teoría de números y programación matemática.

4. Aplicaciones en criptografía y codificación

- **Generación de claves y contraseñas.** Algunas técnicas criptográficas se basan en la dificultad de ciertos problemas combinatorios. La estructura compleja y la variabilidad de los semigrupos numéricos de género fijo pueden utilizarse para diseñar sistemas de generación de claves o contraseñas, donde la impredecibilidad y el alto número de configuraciones aseguran una mayor robustez.
- **Teoría de códigos.** Los semigrupos numéricos tienen aplicaciones en la teoría de códigos, especialmente en la construcción de códigos de corrección de errores. La información extraída de la estructura de los semigrupos (como la distribución de huecos) puede emplearse para diseñar códigos eficientes, aprovechando la correspondencia entre ciertos parámetros algebraicos y propiedades de decodificación.

5. Aportaciones a la investigación en el área matemática

- **Desarrollo teórico y conjeturas.**

El estudio de semigrupos numéricos de género fijo ha impulsado el desarrollo de numerosas conjeturas y teoremas en teoría de números. Los trabajos en este campo han motivado investigaciones en la estructura combinatoria, la teoría de invariantes y la optimización, generando

bastantes conocimientos interconectados. En este trabajo se establecen 3 conjeturas, como se explicará en la sección 3.4.5, que son:

Conjetura 1

- Sea k un número natural, entonces la cantidad de semigrupos numéricos internos de género k es menor o igual que el número de semigrupos numéricos internos de género $k+1$.

$$\#y(\text{gen} = k) \leq \#y(\text{gen} = k + 1).$$

donde y es el conjunto de los semigrupos internos.

Conjetura 2

- Sea k un número natural, entonces la cantidad de semigrupos numéricos hojas de género k es menor o igual que el número de semigrupos numéricos hojas de género $k+1$.

$$\#L(\text{gen} = k) \leq \#L(\text{gen} = k + 1).$$

donde L es el conjunto de los semigrupos hojas.

Conjetura 3

- En el árbol de los semigrupos también se observa que la cantidad de semigrupos numéricos hojas de género k es menor o igual a la cantidad de semigrupos numéricos internos de género k .

$$\#L(\text{gen} = k) \leq \#y(\text{gen} = k).$$

■ Interacción multidisciplinar.

La versatilidad de los semigrupos numéricos se refleja en su aplicación a problemas de diversas ramas matemáticas y computacionales. Esta interacción multidisciplinaria facilita el intercambio de técnicas y conceptos entre áreas como la geometría algebraica, la combinatoria, la teoría de códigos y la optimización, promoviendo un enfoque integrado en la resolución de problemas complejos.

Por tanto, el análisis de semigrupos numéricos de género fijo, considerando tanto los semigrupos internos (que permiten extensiones) como los semigrupos hoja (los nodos terminales), ofrece un marco integral con aplicaciones significativas en varios campos de la matemática y la informática. Este enfoque no solo contribuye a la comprensión profunda de la estructura algebraica y combinatoria inherente a estos semigrupos, sino que también impulsa la creación de algoritmos más eficientes y el desarrollo de nuevos métodos en áreas tan diversas como la geometría algebraica, la teoría de códigos y la criptografía.

3.4.5. Programa de cálculo de semigrupos numéricos de género fijo

Una variedad de Frobenius es una familia no vacía V de semigrupos numéricos que cumplen estas condiciones:

1. Si $\{S, T\} \subseteq V$, entonces $S \cap T \in V$.
2. Si $S \in V$ y $S \neq \mathbb{N}$, entonces $SU\{F(S)\} \in V$.

En donde $F(S)$ es el número de Frobenius del semigrupo numérico S (explicado en la sección 2.1). Para la realización del programa vamos a tener en cuenta el siguiente lema y proposición.

- **Lema:** Si S y T son semigrupos numéricos, entonces $S \cap T$ es también un semigrupo numérico y $F(S \cap T) = \max\{F(S), F(T)\}$
- **Proposición:** y es el conjunto de los semigrupos numéricos internos que tienen estructura de variedad de Frobenius.

También es importante puntualizar que si S es un semigrupo numérico, entonces se denota por $\mu(S) = \#\{x \in \text{msg}(S) \text{ tq } x > F(S)\}$. Un semigrupo numérico S es interno si $\mu(S) \neq 0$.

Si $\{S, T\} \subseteq y$, entonces por el lema anterior sabemos que $S \cap T$ es un semigrupo numérico y $F(S \cap T) = \max\{F(S), F(T)\}$. Podemos suponer sin pérdida de generalidad que $F(S \cap T) = F(T)$. Por tanto, si un valor del semigrupo numérico $x \in \text{msg}(T)$ y $x > F(T)$ entonces $x \in \text{msg}(S \cap T)$ y $x > F(S \cap T)$. Como $\mu(T) \neq 0$, entonces $\mu(S \cap T) \neq 0$ y, por tanto, $S \cap T \in y$ (tenga en cuenta que msg se corresponde al sistema minimal de generadores descrito en la sección 2.5).

Si $S \in y$ y $S \neq \mathbb{N}$, y sabiendo que si S es un semigrupo numérico y $x \in S$, entonces:

1. Si $S \neq \mathbb{N}$, entonces $SU\{F(S)\}$ es un semigrupo numérico.
2. $S \setminus \{x\}$ es semigrupo numérico si y solo si $x \in \text{msg}(S)$

podemos deducir que $SU\{F(S)\}$ es un semigrupo numérico. A partir de estas reglas, debemos de tener en cuenta que $F(S) \in \text{msg}(SU\{F(S)\})$ y $F(S) > F(SU\{F(S)\})$. Por tanto, podemos ver $\mu(SU\{F(S)\}) \neq 0$ y en consecuencia tenemos que $SU\{F(S)\} \in y$

Para la realización del código y poder obtener todos los semigrupos numéricos internos y hojas dado un género fijo, nos basaremos en este algoritmo.

Algorithm 3 Semigrupos numéricos de género fijo

Require: Un entero no negativo g .

Ensure: $y(\text{gen} = g)$.

```

1:  $A \leftarrow \{\mathbb{N}\}$ 
2:  $i \leftarrow 0$ 
3: while  $i \neq g$  do
4:   for all  $S \in A$  do
5:     Calcular  $\alpha(S) \leftarrow \{x \in \text{msg}(S) \text{ tq } x > F(S) \text{ y } \mu(S \setminus \{x\}) \neq 0\}$ 
6:   end for
7:    $A \leftarrow \bigcup_{S \in A} \{S \setminus \{x\} \text{ tq } x \in \alpha(S)\}$ 
8:    $i \leftarrow i + 1$ 
9: end while
10: return  $A$ 

```

Este algoritmo se ha diseñado inicialmente en el lenguaje de programación **C++** cuya finalidad es, dado un número entero no negativo k , encontrar y clasificar todos los semigrupos numéricos de ese género k , dividiéndolos en semigrupos numéricos internos u hojas. Finalmente, se muestran los semigrupos numéricos de cada tipo, comparando si hay más semigrupos numéricos internos u hojas.

Para optimizar el rendimiento del programa de manera notoria, se han creado dos variables globales, las cuales son tablas de memoización que sirven para guardar los resultados ya calculados para evitar cálculos repetidos.

El motivo por el que son variables globales es para hacer más fácil el acceso desde cualquier función además de permitir que permanezcan activas durante toda la ejecución del programa para que cualquier valor ya calculado previamente esté disponible para el resto de cálculos posteriores.

A continuación, se procederá a explicar la filosofía tomada para realizar el programa basandonos en el pseudocódigo previamente mencionado.

- **maxCD**: lo primero de todo será tener una función que calcule el máximo común divisor de dos enteros mediante el uso del algoritmo de Euclides por resto iterativo y con el método *swap*.
- **mcdEsUno**: debemos de comprobar que el máximo común divisor de cada semigrupo numérico S sea 1 o lo que es lo mismo, que el máximo común divisor de todos los elementos de S sea 1.

- **clave**: para optimizar mediante memoización se convierte el conjunto S en una cadena canónica. Se clona S , se ordena y se concatenan sus elementos, separados por comas, obteniendo de esta manera una firma única que se usa como índice en las tablas. Esto permite recuperar resultados de forma segura, acelerando considerablemente el programa al eliminar trabajo redundante.
- **calculaConductor**: debemos de comprobar cuál es el conductor de cada semigrupo numérico S (explicado el funcionamiento del conductor en la sección 2.3). Iniciamos un array hasta el valor máximo de la suma del semigrupo S , marcando a *true* los valores que se pueden generar a partir de combinaciones lineales no negativas del semigrupo numérico S . Cuando consigamos encontrar aquel elemento que sea el menor entero a partir de donde todos los elementos pueden generarse a partir de dichas combinaciones (debemos de poder generar consecutivamente el número establecido por el mínimo valor de $S \setminus \{0\}$), ya se habrá encontrado el conductor.
- **calculaGenero**: tal y como se vió en la sección 2.2, esta función calculará los huecos que hay antes del conductor para cada semigrupo numérico S (se usará el conductor para ver dichos huecos). Se utilizará también programación dinámica para ver los valores que se pueden generar a partir de los generadores de un semigrupo numérico S y se verá cuántos valores menores que el conductor están marcados como *false*. Finalmente, memoizamos el resultado para guardarlo para próximos cálculos.
- **calculaFrobenius**: tal y como se vió en la sección 2.1, se debe de encontrar el mayor entero que no se puede generar a partir de combinaciones lineales de los generadores de un semigrupo numérico S . Se usará también programación dinámica y se hará un recorrido "hacia atrás", buscando el primer elemento que esté a *false*. Finalmente, memoizamos el resultado.
- **esMinimalHilbert**: explicado en las secciones 2.4 y 2.5, se ha realizado un método que nos indica a partir de la serie expandida de Hilbert si cada uno de los semigrupos numéricos S son minimales o no teniendo como límite el valor del $conductor + \min(S \setminus \{0\})$. Una vez generada la expansión de Hilbert de todos los enteros $\leq conductor + \min(S \setminus \{0\})$, se comprobará que en esa serie haya al menos $\min(S \setminus \{0\})$ valores consecutivos para asegurar que a partir del conductor se puedan generar todos los elementos (todos los valores generados deben de pertenecer a la serie expandida de Hilbert del semigrupo numérico S).
- **esHoja**: con este método determinaremos si un semigrupo numérico S es hoja. Para ello debemos de tener en cuenta el género y el número

de Frobenius de dicho semigrupo. Si el número de Frobenius es mayor que el máximo de $msg(S)$, entonces el semigrupo numérico es hoja.

- **generaCombinaciones**: esta función se ha hecho recursiva en la que dado un rango de números y un tamaño concretos, se pueda crear todos los subconjuntos de ese tamaño. Solo existirán algunos subconjuntos que sean semigrupos numéricos válidos para posteriormente mostrarlos por pantalla.
- **comparaCantidades**: este método será el encargado de comparar si hay más semigrupos numéricos internos que hojas, si hay más semigrupos numéricos hoja que internos, o si hay la misma cantidad.
- **encontrarSemigruposYHojas**: este método es el encargado de encontrar todos los semigrupos numéricos internos y hojas que hay dado un género fijo. El método comprobará cuáles son los semigrupos numéricos que cumplen con el género descrito usando las funciones explicadas anteriormente. A partir del filtro, se deberá distinguir aquellos semigrupos numéricos que sean internos o hojas.

Por otra parte, como sabemos que dado un género cualquiera g existe un semigrupo numérico interno generado por $\langle g+1, g+2, \dots, 2 \cdot g+1 \rangle$. Los huecos de este semigrupo interno van a ser todos los números consecutivos desde $\{1, 2, \dots, \text{numFrobenius}\}$.

Finalmente, muestra aquellos semigrupos numéricos internos y hojas y se pasan las listas al método *comparaCantidades*.

```

1: procedure ENCONTRARSEMIGRUPOSYHOJAS(genero)
2:   internos  $\leftarrow$  lista vacía de subconjuntos
3:   hojas  $\leftarrow$  lista vacía de subconjuntos
4:   limite  $\leftarrow$  genero  $\times$  5
5:   numeros  $\leftarrow$  lista de enteros desde 2 hasta limite
6:   for tamano  $\leftarrow$  2 to genero do
7:     subconjuntos  $\leftarrow$  lista vacía
8:     combinacion  $\leftarrow$  lista vacía
9:     GENERACOMBINACIONES(subconjuntos, combinacion, numeros, 0, tamano)
10:    for all subconjunto in subconjuntos do
11:      if no MCDESUNO(subconjunto) then
12:        continue
13:      end if
14:      if CALCULAGENERO(subconjunto)  $\neq$  genero then
15:        continue
16:      end if
17:      if no ESMINIMALHILBERT(subconjunto) then
18:        continue

```

```

19:         end if
20:         if ESHOJA(subconjunto, genero) then
21:             agregar subconjunto a hojas
22:         else
23:             agregar subconjunto a internos
24:         end if
25:     end for
26: end for
27: semigrupoExtra ← lista vacía
28: for  $i \leftarrow genero+1$  to  $2 \times genero + 1$  do
29:     agregar  $i$  a semigrupoExtra
30: end for
31: agregar semigrupoExtra a internos
32: imprimir Semigrupos numéricos internos:
33: for all  $s$  in internos do
34:     imprimir  $< + unir(s, ",") + >$ 
35: end for
36: imprimir Semigrupos numéricos hoja:
37: for all  $h$  in hojas do
38:     imprimir  $< + unir(h, ",") + >$ 
39: end for
40: COMPARACANTIDADES(internos, hojas)
41: end procedure

```

- **main:** la lógica del método principal será solicitar al usuario el género fijo, el cual deberá de ser mayor o igual a 0. Si el usuario introduce el número 0, se harán los cálculos por defecto (si el género es 0 el único semigrupo numérico interno válido es \mathbb{N} , sin tener ningún semigrupo numérico hoja). Si la entrada es ≥ 1 , comienza la ejecución del programa llamando al método *encontrarSemigruposYHojas*. Además, se medirá el tiempo para aquellos géneros mayores o iguales a 1.

Finalmente, se ha realizado una tabla de los resultados obtenidos a partir de un género fijo dado. Podemos observar como los semigrupos numéricos internos y hojas crecen exponencialmente conforme el género es mayor, mientras que el tiempo también crece de manera notoria a partir de género ≥ 7 . Recordar que el algoritmo empleado se ha optimizado para reducir la búsqueda en donde se sabe que no existen semigrupos numéricos que cumplan con el género pasado por parámetro. La tabla es la siguiente.

Género	Internos	Hojas	Tiempo de cálculo (s)
0	$\langle 1 \rangle$	\emptyset	< 1
1	$\langle 2, 3 \rangle$	\emptyset	< 1
2	$\langle 2, 5 \rangle \langle 3, 4, 5 \rangle$	\emptyset	< 1
3	$\langle 2, 7 \rangle \langle 3, 5, 7 \rangle \langle 4, 5, 6, 7 \rangle$	$\langle 3, 4 \rangle$	< 1
4	$\langle 2, 9 \rangle \langle 3, 7, 8 \rangle \langle 4, 5, 7 \rangle$ $\langle 4, 6, 7, 9 \rangle \langle 5, 6, 7, 8, 9 \rangle$	$\langle 3, 5 \rangle \langle 4, 5, 6 \rangle$	< 1
5	$\langle 2, 11 \rangle \langle 3, 7, 11 \rangle \langle 3, 8, 10 \rangle$ $\langle 4, 5, 11 \rangle \langle 4, 6, 9, 11 \rangle$ $\langle 4, 7, 9, 10 \rangle \langle 5, 6, 7, 9 \rangle$ $\langle 5, 6, 8, 9 \rangle \langle 5, 7, 8, 9, 11 \rangle$ $\langle 6, 7, 8, 9, 10, 11 \rangle$	$\langle 4, 6, 7 \rangle \langle 5, 6, 7, 8 \rangle$	< 1
6	$\langle 2, 13 \rangle \langle 3, 8, 13 \rangle \langle 3, 10, 11 \rangle$ $\langle 4, 6, 11, 13 \rangle \langle 4, 7, 10, 13 \rangle$ $\langle 4, 9, 10, 11 \rangle \langle 5, 6, 9, 13 \rangle$ $\langle 5, 7, 8, 11 \rangle \langle 5, 7, 9, 11, 13 \rangle$ $\langle 5, 8, 9, 11, 12 \rangle \langle 6, 7, 8, 9, 11 \rangle$ $\langle 6, 7, 8, 10, 11 \rangle \langle 6, 7, 9, 10, 11 \rangle$ $\langle 6, 8, 9, 10, 11, 13 \rangle$ $\langle 7, 8, 9, 10, 11, 12, 13 \rangle$	$\langle 3, 7 \rangle \langle 4, 5 \rangle$ $\langle 4, 6, 9 \rangle \langle 4, 7, 9 \rangle$ $\langle 5, 6, 7 \rangle \langle 5, 6, 8 \rangle$ $\langle 5, 7, 8, 9 \rangle$ $\langle 6, 7, 8, 9, 10 \rangle$	14
7	$\langle 2, 15 \rangle \langle 3, 10, 14 \rangle \langle 3, 11, 13 \rangle$ $\langle 4, 7, 13 \rangle \langle 4, 6, 13, 15 \rangle$ $\langle 4, 9, 10, 15 \rangle \langle 4, 9, 11, 14 \rangle$ $\langle 4, 10, 11, 13 \rangle \langle 5, 6, 13, 14 \rangle$ $\langle 5, 7, 9, 13 \rangle \langle 5, 7, 11, 13 \rangle$ $\langle 5, 8, 9, 12 \rangle \langle 6, 7, 8, 11 \rangle$ $\langle 6, 7, 9, 11 \rangle \langle 5, 8, 11, 12, 14 \rangle$ $\langle 5, 9, 11, 12, 13 \rangle$ $\langle 6, 7, 10, 11, 15 \rangle \langle 6, 8, 9, 10, 13 \rangle$ $\langle 6, 8, 9, 11, 13 \rangle$ $\langle 6, 8, 10, 11, 13, 15 \rangle$ $\langle 6, 9, 10, 11, 13, 14 \rangle$ $\langle 7, 8, 9, 10, 11, 13 \rangle$ $\langle 7, 8, 9, 10, 12, 13 \rangle$ $\langle 7, 8, 9, 11, 12, 13 \rangle$ $\langle 7, 8, 10, 11, 12, 13 \rangle$ $\langle 7, 9, 10, 11, 12, 13, 15 \rangle$ $\langle 8, 9, 10, 11, 12, 13, 14, 15 \rangle$	$\langle 3, 8 \rangle \langle 4, 6, 11 \rangle$ $\langle 4, 7, 10 \rangle \langle 5, 6, 9 \rangle$ $\langle 5, 7, 8 \rangle$ $\langle 5, 7, 9, 11 \rangle$ $\langle 5, 8, 9, 11 \rangle$ $\langle 6, 7, 8, 9 \rangle$ $\langle 6, 7, 8, 10 \rangle$ $\langle 6, 7, 9, 10 \rangle$ $\langle 6, 8, 9, 10, 11 \rangle$ $\langle 7, 8, 9, 10, 11, 12 \rangle$	207

Género	Internos	Hojas	Tiempo de cálculo (s)
8	$\langle 2, 17 \rangle \langle 3, 10, 17 \rangle \langle 3, 11, 16 \rangle \langle 3, 13, 14 \rangle \langle 4, 7, 17 \rangle$ $\langle 5, 6, 14 \rangle \langle 4, 6, 15, 17 \rangle \langle 4, 9, 14, 15 \rangle \langle 4, 10, 11, 17 \rangle$ $\langle 4, 10, 13, 15 \rangle \langle 4, 11, 13, 14 \rangle \langle 5, 7, 13, 16 \rangle$ $\langle 5, 8, 12, 14 \rangle \langle 6, 7, 8, 17 \rangle \langle 6, 7, 9, 17 \rangle \langle 6, 7, 10, 15 \rangle$ $\langle 6, 8, 9, 13 \rangle \langle 5, 8, 11, 14, 17 \rangle \langle 5, 9, 11, 13, 17 \rangle$ $\langle 5, 9, 12, 13, 16 \rangle \langle 5, 11, 12, 13, 14 \rangle$ $\langle 6, 7, 11, 15, 16 \rangle \langle 6, 8, 10, 11, 15 \rangle \langle 6, 8, 11, 13, 15 \rangle$ $\langle 6, 9, 10, 11, 14 \rangle \langle 7, 8, 9, 10, 13 \rangle \langle 7, 8, 9, 11, 13 \rangle$ $\langle 7, 8, 9, 12, 13 \rangle \langle 7, 8, 10, 11, 13 \rangle \langle 7, 8, 10, 12, 13 \rangle$ $\langle 6, 8, 10, 13, 15, 17 \rangle \langle 6, 9, 10, 13, 14, 17 \rangle$ $\langle 6, 9, 11, 13, 14, 16 \rangle \langle 6, 10, 11, 13, 14, 15 \rangle$ $\langle 7, 8, 11, 12, 13, 17 \rangle \langle 7, 9, 10, 11, 12, 15 \rangle$ $\langle 7, 9, 10, 11, 13, 15 \rangle \langle 7, 9, 10, 12, 13, 15 \rangle$ $\langle 7, 9, 11, 12, 13, 15, 17 \rangle \langle 7, 10, 11, 12, 13, 15, 16 \rangle$ $\langle 8, 9, 10, 11, 12, 13, 15 \rangle \langle 8, 9, 10, 11, 12, 14, 15 \rangle$ $\langle 8, 9, 10, 11, 13, 14, 15 \rangle \langle 8, 9, 10, 12, 13, 14, 15 \rangle$ $\langle 8, 9, 11, 12, 13, 14, 15 \rangle$ $\langle 8, 10, 11, 12, 13, 14, 15, 17 \rangle$ $\langle 9, 10, 11, 12, 13, 14, 15, 16, 17 \rangle$	$\langle 4, 6, 13 \rangle \langle 4, 9, 10 \rangle$ $\langle 4, 9, 11 \rangle \langle 5, 6, 13 \rangle$ $\langle 5, 7, 9 \rangle \langle 5, 7, 11 \rangle$ $\langle 5, 8, 9 \rangle \langle 5, 8, 11, 12 \rangle$ $\langle 5, 9, 11, 12 \rangle$ $\langle 6, 7, 10, 11 \rangle \langle 6, 8, 9, 10 \rangle$ $\langle 6, 8, 9, 11 \rangle$ $\langle 6, 8, 10, 11, 13 \rangle$ $\langle 6, 9, 10, 11, 13 \rangle$ $\langle 7, 8, 9, 10, 11 \rangle$ $\langle 7, 8, 9, 10, 12 \rangle$ $\langle 7, 8, 9, 11, 12 \rangle$ $\langle 7, 8, 10, 11, 12 \rangle$ $\langle 7, 9, 10, 11, 12, 13 \rangle$ $\langle 8, 9, 10, 11, 12, 13, 14 \rangle$	3345

Cuadro 3.1: Tabla de cálculo de semigrupos numéricos de género fijo con el género correspondiente, sus semigrupos numéricos (internos y hojas), así como el tiempo de cálculo.

3.5. Semigrupos numéricos con número de Frobenius fijo

Dado que los conceptos fundamentales de semigrupo numérico ya han sido explicados en el capítulo 3, vamos a centrarnos en el estudio de aquellos semigrupos que se caracterizan por tener un *número de Frobenius fijo*, es decir, el máximo entero que no pertenece al semigrupo es un número preestablecido F . Este parámetro hay que tenerlo en cuenta dado que se relaciona directamente con el género y el conductor de los semigrupos.

Para los semigrupos numéricos con número de Frobenius fijo se tendrá en cuenta la estructura de árbol de todos los semigrupos (internos y hojas), pero ahora se considerará aquellos que tienen un número de Frobenius fijo.

La implementación de este enfoque nos ayuda con la tarea de enumerar semigrupos con un número de Frobenius dado, además de abrir nuevas formas y puntos de vista para el desarrollo de algoritmos combinatorios y la

formulación de conjeturas en la teoría de semigrupos. La estructura interna revelada por el análisis en árbol ofrece una perspectiva amplia que vincula parámetros fundamentales y patrones de exclusión inherentes a estos conjuntos numéricos.

3.5.1. Propiedades

Podemos ver que existen diversas propiedades fundamentales de los semigrupos numéricos con un número de Frobenius fijo, de las cuales tenemos que tener en cuenta las siguientes:

1. Generadores mínimos y transformaciones estructurales.

El sistema minimal de generadores de un semigrupo determina su organización y estructura. En el estudio con F fijo, cualquier transformación como la eliminación o sustitución de un generador debe preservar la propiedad de tener F como el número de Frobenius. Este control de cambios se refleja en la forma de un árbol, en el cual:

- Los **nodos internos** corresponden a aquellos semigrupos que permiten transformaciones manteniendo F constante.
- Las **hojas** son los casos límite que, al no permitir más transformaciones sin violar la condición sobre F , representan los semigrupos terminales en la estructura.

2. Representación en árbol y propagación de la estructura.

La representación mediante un árbol es una herramienta valiosa para entender cómo se propaga la estructura de los semigrupos con F fijo. Cada nodo del árbol se asocia a un semigrupo y las ramas indican la transición a nuevos semigrupos obtenidos mediante modificaciones en el conjunto de generadores, manteniendo invariable el número de Frobenius. Esta visión permite identificar patrones y relaciones de inclusión entre diferentes semigrupos.

3. Aspectos combinatorios y algorítmicos.

Fijar F reduce el espacio de búsqueda, lo que favorece el desarrollo de algoritmos específicos para enumerar y clasificar los semigrupos correspondientes. Los métodos algorítmicos basados en la eliminación controlada de generadores mínimos han permitido obtener resultados precisos sobre la cantidad de semigrupos con un determinado número de Frobenius, aportando información relevante sobre el comportamiento asintótico de dicho conteo conforme varía F .

4. Implicaciones en la clasificación teórica.

El análisis de los semigrupos numéricos que comparten un mismo número de Frobenius ofrece nuevas perspectivas para su clasificación. Al

interrelacionar parámetros como el género, el conductor y la dimensión de embebimiento (número de generadores mínimos), se pueden formular y comprobar conjeturas que enriquecen el estudio teórico y estructural de estos objetos algebraicos.

Este enfoque, basado en la representación en árbol y la exploración de transformaciones en el conjunto generador, no solo facilita la enumeración de semigrupos con F fijo, sino que también abre nuevas vías para investigar fenómenos combinatorios y algebraicos de gran interés.

3.5.2. Ejemplos

Ejemplo 1 (Número de Frobenius 1)

Si el número de Frobenius fijo es 1, el único semigrupo interno que existe es

$$\langle 2, 3 \rangle.$$

En este caso, el único hueco es 1 (el propio número de Frobenius) y, a partir de 2, todas las combinaciones lineales no negativas forman el semigrupo numérico.

Ejemplo 2 (Número de Frobenius 2)

Para un número de Frobenius fijo 2, el único semigrupo numérico interno es

$$\langle 3, 4, 5 \rangle.$$

Aquí, a partir del 3 en adelante se pueden generar todas las combinaciones posibles a partir de dichos generadores.

Ejemplo 3 (Número de Frobenius 3)

Si el número de Frobenius fijo es 3, un semigrupo numérico es

$$\langle 2, 5 \rangle.$$

A partir del número de Frobenius, se generan el resto de combinaciones lineales empleando los generadores del semigrupo.

Ejemplo 4 (Número de Frobenius 7)

Un semigrupo hoja para un número de Frobenius fijo 7 es

$$\langle 4, 5, 6 \rangle.$$

En este caso, con ese conjunto de generadores, ninguna combinación lineal permite generar el número de Frobenius fijo. Como este es el mayor de los enteros ausentes y se presentan todos los elementos del conjunto de generadores que son menores al número de Frobenius, se considera que el semigrupo es hoja.

Ejemplo 5 (Número de Frobenius 11)

Para un número de Frobenius fijo 11, se puede considerar el semigrupo numérico hoja

$$\langle 5, 7, 8, 9 \rangle.$$

Aquí, al calcular las combinaciones lineales, se observa que el mayor hueco no generable es $\{11\}$, que corresponde al número de Frobenius fijo. Al ser mayor que todos los elementos del conjunto de generadores, este semigrupo se clasifica como hoja.

3.5.3. Utilidades

Debemos de comprender las mejoras que ha proporcionado el conocimiento de los semigrupos numéricos que comparten un número de Frobenius fijo. Para ello, vamos a ver aquellos aspectos teóricos como aplicaciones prácticas en distintas áreas de las matemáticas y la informática.

1. Aplicación al problema de la moneda.

Los semigrupos numéricos se usan para modelar el clásico problema de Frobenius, que consiste en determinar el mayor entero (el número de Frobenius) que no puede expresarse como combinación lineal de un conjunto dado de enteros positivos. Fijar este número ayuda a reducir el espacio de búsqueda y a establecer comparaciones entre diferentes conjuntos generadores, facilitando la resolución y el análisis de este problema.

2. Contribución a la combinatoria algebraica.

La restricción mediante un número de Frobenius fijo optimiza los métodos de enumeración y el desarrollo de algoritmos combinatorios. Al delimitar el espacio de estudio, se pueden identificar patrones estructurales y propiedades regulares en la familia de semigrupos, lo que resulta clave para formular conjeturas y establecer resultados en combinatoria algebraica.

3. Implicaciones en la teoría de la factorización.

Los semigrupos numéricos son esenciales en el estudio de la factorización en dominios no únicos. La condición de tener un número de Frobenius fijo influye en la manera en que se pueden descomponer los

enteros en sus factores, ofreciendo un marco teórico para abordar cuestiones de unicidad y sobre la estructura de las factorizaciones.

4. Aplicaciones en geometría algebraica.

En el estudio de curvas proyectivas y discriminantes, los semigrupos numéricos aparecen en forma de semigrupos de Weierstrass, donde su estructura está vinculada a la singularidad de puntos en curvas. El hecho de fijar el número de Frobenius permite relacionar propiedades combinatorias del semigrupo con invariantes geométricos, como la multiplicidad y el índice del punto singular, enriqueciendo el análisis en geometría algebraica.

5. Desarrollo de algoritmos y herramientas computacionales.

Al establecer el número de Frobenius como un parámetro fijo, se reduce la complejidad del espacio de búsqueda, lo que resulta en algoritmos más eficientes para la enumeración y clasificación de semigrupos. Estos algoritmos no solo son útiles en la investigación matemática teórica, sino que también tienen aplicaciones en la generación de datos experimentales que permiten estudiar el comportamiento asintótico y la distribución de dichos semigrupos.

6. Posibles aplicaciones en teoría de códigos y criptografía.

Aunque aún es un área en desarrollo, la regularidad y la estructura controlada de los semigrupos numéricos con parámetros fijos pueden favorecer el diseño de ciertos códigos algebraicos. La claridad en la estructura combinatoria y los métodos de transformación en el conjunto de generadores pueden resultar en aplicaciones innovadoras en contextos criptográficos y en la teoría de códigos.

7. Fomento de nuevos enfoques teóricos.

Estudiar semigrupos numéricos con un número de Frobenius fijo impulsa la integración de diferentes ramas matemáticas, ya que combina técnicas de teoría de números, combinatoria y álgebra computacional. Esta intersección multidisciplinaria promueve la formulación de nuevas conjeturas y estrategias metodológicas que pueden repercutir en el estudio de objetos algebraicos y la resolución de problemas complejos.

Cada uno de estos aspectos nos da a entender cómo la fijación del número de Frobenius no solo simplifica y optimiza el análisis estructural de los semigrupos numéricos, sino que también potencia diversas aplicaciones tanto en ámbitos teóricos como prácticos. El estudio de estos semigrupos ofrece herramientas esenciales para la investigación en áreas tan disímiles como la geometría algebraica o la teoría de códigos, demostrando la versatilidad y la riqueza de estos objetos matemáticos.

3.5.4. Aplicaciones

El estudio de semigrupos numéricos con un número de Frobenius fijo ha generado múltiples aplicaciones en diversas ramas de las matemáticas y de la informática. Vamos a ver algunas aplicaciones teóricas y prácticas donde el uso de estos semigrupos son relevantes.

1. Modelado de problemas monetarios.

Como hemos visto antes, el problema de la moneda es uno de los ejemplos paradigmáticos. Aquí, los semigrupos numéricos representan las cantidades alcanzables mediante combinaciones lineales de monedas de determinados valores. Fijando el número de Frobenius se delimita el mayor monto inalcanzable, lo cual no solo tiene interés teórico, sino que también orienta estrategias en sistemas de cambio y en la modelación de sistemas económicos.

2. Avances en combinatoria algebraica.

La restricción del número de Frobenius permite reducir el espacio de búsqueda durante la enumeración y clasificación de semigrupos. Esta propiedad es crucial para identificar patrones estructurales y formular conjeturas en combinatoria algebraica, posibilitando el desarrollo de algoritmos enumerativos con aplicaciones en diversas ramas teóricas.

3. Estudio de la factorización en dominios no únicos.

Los semigrupos numéricos son herramientas fundamentales en el análisis de la unicidad (o su fallo) en la factorización de enteros, especialmente en dominios que no poseen factorización única. Fijar el número de Frobenius establece un marco para estudiar cómo se puede, o no, descomponer un entero en función de un conjunto fijo de generadores, proporcionando información sobre invariantes aritméticos y sobre la estructura de la factorización.

4. Aplicaciones en geometría algebraica.

En el estudio de curvas proyectivas y en la caracterización de puntos de Weierstrass, los semigrupos numéricos aparecen de manera natural. Al imponer un número de Frobenius fijo, se pueden establecer vínculos entre las propiedades combinatorias del semigrupo y ciertos invariantes geométricos (por ejemplo, la multiplicidad de un punto singular), lo que permite el análisis de fenómenos en la geometría algebraica.

5. Desarrollo de algoritmos computacionales.

La condición de tener un número de Frobenius fijo no solo simplifica la estructura teórica, sino que también reduce la complejidad computacional al enumerar semigrupos numéricos. Esta propiedad favorece el diseño de algoritmos más eficientes para clasificar y explorar estos

objetos, siendo de utilidad en investigaciones que requieren el procesamiento de grandes volúmenes de datos combinatorios.

6. Nuevas aplicaciones en programación entera y optimización combinatoria.

Una aplicación emergente y prometedora es la incorporación de semigrupos numéricos con número de Frobenius fijo en problemas de programación entera. En muchos modelos de optimización, las restricciones de factibilidad se pueden interpretar mediante combinaciones lineales de variables enteras, similar a los generadores de un semigrupo. Al conocer un umbral fijo (el número de Frobenius), es posible establecer límites precisos para la viabilidad de determinadas soluciones y reducir el espacio de búsqueda de manera significativa. Esta aproximación ha mostrado potencial en la optimización de redes de transporte, la planificación de recursos y en problemas logísticos, donde identificar rápidamente la factibilidad de soluciones enteras resulta crucial. Mediante el análisis estructural de estos semigrupos se pueden desarrollar heurísticas y algoritmos exactos que mejoren la eficiencia y la precisión en la resolución de problemas combinatorios complejos.

Vemos por tanto que la capacidad de fijar el número de Frobenius en los semigrupos numéricos permite una amplia variedad de aplicaciones, desde abordar problemas monetarios clásicos hasta ofrecer herramientas innovadoras en la optimización y programación entera. Esta versatilidad demuestra la profunda conexión entre estructuras algebraicas y problemas prácticos, abriendo nuevas vías de investigación tanto en teoría como en aplicaciones tecnológicas.

3.5.5. Programa para calcular los semigrupos numéricos dado un número de Frobenius fijo

Se ha realizado un programa en donde, dado un número de Frobenius fijo, calcularemos todos los semigrupos numéricos (tanto internos como hojas) que cumplan la condición de número de Frobenius dado. Tendremos en cuenta el número de semigrupos numéricos internos en comparación con los semigrupos numéricos hojas.

A continuación se presenta el pseudocódigo que se ha realizado para el programa.

```
1: Entrada: Entero positivo  $F$  ▷ Número de Frobenius fijo
2: if  $F \leq 0$  then
3:   Imprimir: "F debe ser un entero positivo."
4:   return
5: end if
```

```

6: vistos  $\leftarrow$  conjunto vacío de cadenas
7: resultado  $\leftarrow$  lista vacía de semigrupos
8:      $\triangleright$  Genera el semigrupo inicial  $S_0 = \{F + 1, F + 2, \dots, 2F + 1\}$ 
9:  $S_0 \leftarrow$  semigrupoInicial( $F$ )
10:  $S_0 \leftarrow$  minimizarGeneradores( $S_0$ )
11: Agregar  $S_0$  a resultado
12: Insertar semigrupoACadena( $S_0$ ) en vistos
13: Imprimir: semigrupoACadena( $S_0$ ) , aperyACadena( $S_0, F + 1$ )
14: actual  $\leftarrow$  lista que contiene  $S_0$ 
15: while actual no está vacía do
16:     siguienteNivel  $\leftarrow$  lista vacía
17:     for cada semigrupo  $S$  en actual do
18:         candidatos  $\leftarrow$  obtenerCandidatos( $S, F$ )
19:         for cada candidato  $c$  en candidatos do
20:             if semigrupoACadena( $c.semigrupo$ ) no está en vistos then
21:                 Insertar semigrupoACadena( $c.semigrupo$ ) en vistos
22:                 Agregar  $c.semigrupo$  a resultado
23:                 Agregar  $c.semigrupo$  a siguienteNivel
24:                 Imprimir: semigrupoACadena( $c.semigrupo$ ) , aperyACadena( $c.semigrupo, F$ 
25:             1)
26:             end if
27:         end for
28:     actual  $\leftarrow$  siguienteNivel
29: end while
30:
31:      $\triangleright$  Clasificación de semigrupos en internos y hojas
32: internos  $\leftarrow$  lista vacía; hojas  $\leftarrow$  lista vacía
33: for cada semigrupo  $S$  en resultado do
34:     todosMenores  $\leftarrow$  verdadero
35:     for cada generador  $g$  en  $S$  do
36:         if  $g \geq F$  then
37:             todosMenores  $\leftarrow$  falso
38:             salir del ciclo
39:         end if
40:     end for
41:     if todosMenores then
42:         Agregar  $S$  a hojas
43:     else
44:         Agregar  $S$  a internos
45:     end if
46: end for
47:
48: Imprimir: "Semigrupos numéricos internos:"

```

```

49: for cada semigrupo  $S$  en internos do
50:   Imprimir: semigrupoACadena( $S$ )
51: end for
52: Imprimir: "Semigrupos numéricos hoja:"
53: for cada semigrupo  $S$  en hojas do
54:   Imprimir: semigrupoACadena( $S$ )
55: end for
56: Imprimir: "Total internos: - longitud(internos) + "Total hojas: - longitud(hojas)

```

A continuación se presentan las funciones auxiliares utilizadas en el pseudocódigo:

- **semigrupoACadena(generadores):** convierte la lista de generadores **generadores** en una cadena tipo $\langle g_1, g_2, \dots, g_n \rangle$.
- **esRepresentable(x, generadores):** retorna verdadero si x es representable como combinación lineal no negativa de los elementos en **generadores**.
- **minimizarGeneradores(generadores):** elimina de **generadores** los generadores redundantes.
- **frobeniusEsValido(generadores, F):** retorna verdadero si F no es representable y $F + 1$ sí lo es para el conjunto **generadores**.
- **semigrupoInicial(F):** genera el semigrupo inicial $S_0 = \langle F+1, \dots, 2F+1 \rangle$.
- **obtenerCandidatos(S, F):** genera candidatos $x \in [2, m) \setminus \{F\}$ que, al agregarlos a S (y minimizar), conservan F como número de Frobenius.
- **aperyConjunto(generadores, periodo):** calcula el conjunto de Apéry para **generadores** y un período dado.
- **aperyACadena(generadores, periodo):** retorna la representación en cadena del conjunto de Apéry.

El pseudocódigo implementa la generación y clasificación de semigrupos numéricos cuyo número de Frobenius es fijo (denotado por F). A grandes rasgos, se estructura de la siguiente forma:

1. **Entrada y validación:** se solicita un entero positivo F ; si $F \leq 0$, el programa muestra un mensaje de error y detiene la ejecución.
2. **Generación del semigrupo inicial:** se construye el semigrupo inicial $S_0 = \langle F + 1, F + 2, \dots, 2F + 1 \rangle$ y se procede a optimizar el conjunto de generadores eliminando aquellos que resultan redundantes.

3. **Búsqueda en amplitud:** usando S_0 como punto de partida, se exploran de forma sistemática (con una búsqueda en amplitud) todos los semigrupos posibles que mantienen el número de Frobenius igual a F . Durante esta búsqueda, se generan nuevos candidatos (por la adición controlada de generadores) y se verifica, mediante minimización y validación, que se cumpla la propiedad requerida. Los semigrupos ya generados se almacenan y se evitan duplicados con el uso de una estructura de control.
4. **Cálculo del conjunto de Apéry:** para cada semigrupo generado, se calcula el conjunto de Apéry, que consiste en obtener, para cada residuo (módulo un período determinado), el elemento mínimo representable. Este conjunto aporta información acerca de la estructura combinatoria del semigrupo.
5. **Clasificación y salida:** por último, clasificamos cada semigrupo en dos categorías:
 - **Internos:** semigrupos numéricos que contienen al menos un generador mayor o igual a F , lo que permite derivar nuevos semigrupos.
 - **Hojas:** semigrupos numéricos terminales que no admiten más transformaciones sin romper la condición del número de Frobenius.

Se imprime la lista de semigrupos en cada categoría, junto con un conteo total de semigrupos internos y hojas.

Por tanto podemos ver que el pseudocódigo se encarga de generar, validar, clasificar y presentar semigrupos numéricos manteniendo un número de Frobenius fijo, utilizando técnicas de búsqueda en amplitud, minimización de generadores y cálculo del conjunto de Apéry. Estas operaciones permiten obtener una representación única y ordenada de cada semigrupo, facilitando de esta manera su estudio y análisis.

Número Frobenius	Internos	Hojas	Tiempo de cálculo (s)
1	$\langle 2, 3 \rangle$	\emptyset	<1
2	$\langle 3, 4, 5 \rangle$	\emptyset	<1
3	$\langle 4, 5, 6, 7 \rangle \langle 2, 5 \rangle$	\emptyset	<1
4	$\langle 5, 6, 7, 8, 9 \rangle \langle 3, 5, 7 \rangle$	\emptyset	<1
5	$\langle 6, 7, 8, 9, 10, 11 \rangle \langle 2, 7 \rangle \langle 3, 7, 8 \rangle \langle 4, 6, 7, 9 \rangle$	$\langle 3, 4 \rangle$	<1

Número Frobenius	Internos	Hojas	Tiempo de cálculo (s)
6	$\langle 7, 8, 9, 10, 11, 12, 13 \rangle \langle 4, 7, 9, 10 \rangle \langle 5, 7, 8, 9, 11 \rangle$ $\langle 4, 5, 7 \rangle$	\emptyset	<1
7	$\langle 8, 9, 10, 11, 12, 13, 14, 15 \rangle \langle 2, 9 \rangle \langle 3, 8, 10 \rangle$ $\langle 4, 9, 10, 11 \rangle \langle 5, 8, 9, 11, 12 \rangle \langle 6, 8, 9, 10, 11, 13 \rangle$ $\langle 4, 5, 11 \rangle \langle 4, 6, 9, 11 \rangle \langle 5, 6, 8, 9 \rangle$	$\langle 3, 5 \rangle \langle 4, 5, 6 \rangle$	<1
8	$\langle 9, 10, 11, 12, 13, 14, 15, 16, 17 \rangle \langle 3, 10, 11 \rangle$ $\langle 5, 9, 11, 12, 13 \rangle \langle 6, 9, 10, 11, 13, 14 \rangle$ $\langle 7, 9, 10, 11, 12, 13, 15 \rangle \langle 5, 6, 9, 13 \rangle \langle 3, 7, 11 \rangle$ $\langle 5, 7, 9, 11, 13 \rangle \langle 6, 7, 9, 10, 11 \rangle \langle 5, 6, 7, 9 \rangle$	\emptyset	<1
9	$\langle 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 \rangle \langle 2, 11 \rangle$ $\langle 4, 10, 11, 13 \rangle \langle 5, 11, 12, 13, 14 \rangle \langle 6, 10, 11, 13, 14, 15 \rangle$ $\langle 7, 10, 11, 12, 13, 15, 16 \rangle \langle 8, 10, 11, 12, 13, 14, 15, 17 \rangle$ $\langle 4, 6, 11, 13 \rangle \langle 5, 6, 13, 14 \rangle \langle 4, 7, 10, 13 \rangle \langle 5, 7, 11, 13 \rangle$ $\langle 6, 7, 10, 11, 15 \rangle \langle 5, 8, 11, 12, 14 \rangle \langle 6, 8, 10, 11, 13, 15 \rangle$ $\langle 7, 8, 10, 11, 12, 13 \rangle \langle 5, 7, 8, 11 \rangle \langle 6, 7, 8, 10, 11 \rangle$	$\langle 4, 6, 7 \rangle \langle 5, 6, 7 \rangle$ $\langle 5, 6, 8 \rangle \langle 5, 6, 7, 8 \rangle$	<1
10	$\langle 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 \rangle \langle 3, 11, 13 \rangle$ $\langle 4, 11, 13, 14 \rangle \langle 6, 11, 13, 14, 15, 16 \rangle$ $\langle 7, 11, 12, 13, 15, 16, 17 \rangle \langle 8, 11, 12, 13, 14, 15, 17, 18 \rangle$ $\langle 9, 11, 12, 13, 14, 15, 16, 17, 19 \rangle \langle 4, 7, 13 \rangle$ $\langle 6, 7, 11, 15, 16 \rangle \langle 3, 8, 13 \rangle \langle 6, 8, 11, 13, 15 \rangle$ $\langle 7, 8, 11, 12, 13, 17 \rangle \langle 4, 9, 11, 14 \rangle \langle 6, 9, 11, 13, 14, 16 \rangle$ $\langle 7, 9, 11, 12, 13, 15, 17 \rangle \langle 8, 9, 11, 12, 13, 14, 15 \rangle$ $\langle 6, 7, 8, 11 \rangle \langle 6, 7, 9, 11 \rangle \langle 6, 8, 9, 11, 13 \rangle$ $\langle 7, 8, 9, 11, 12, 13 \rangle \langle 6, 7, 8, 9, 11 \rangle$	$\langle 4, 7, 9 \rangle$	<1
11	$\langle 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 \rangle \langle 2, 13 \rangle$ $\langle 3, 13, 14 \rangle \langle 4, 13, 14, 15 \rangle \langle 5, 12, 13, 14, 16 \rangle$ $\langle 6, 13, 14, 15, 16, 17 \rangle \langle 7, 12, 13, 15, 16, 17, 18 \rangle$ $\langle 8, 12, 13, 14, 15, 17, 18, 19 \rangle$ $\langle 9, 12, 13, 14, 15, 16, 17, 19, 20 \rangle$ $\langle 10, 12, 13, 14, 15, 16, 17, 18, 19, 21 \rangle \langle 4, 6, 13, 15 \rangle$ $\langle 5, 7, 13, 16 \rangle \langle 6, 7, 15, 16, 17 \rangle \langle 5, 8, 12, 14 \rangle$ $\langle 6, 8, 13, 15, 17 \rangle \langle 7, 8, 12, 13, 17, 18 \rangle \langle 4, 9, 14, 15 \rangle$ $\langle 5, 9, 12, 13, 16 \rangle \langle 6, 9, 13, 14, 16, 17 \rangle$ $\langle 7, 9, 12, 13, 15, 17 \rangle \langle 8, 9, 12, 13, 14, 15, 19 \rangle$ $\langle 3, 10, 14 \rangle \langle 4, 10, 13, 15 \rangle \langle 6, 10, 13, 14, 15, 17 \rangle$ $\langle 7, 10, 12, 13, 15, 16, 18 \rangle \langle 8, 10, 12, 13, 14, 15, 17, 19 \rangle$ $\langle 9, 10, 12, 13, 14, 15, 16, 17 \rangle \langle 6, 7, 8, 17 \rangle \langle 5, 7, 9, 13 \rangle$ $\langle 6, 7, 9, 17 \rangle \langle 5, 8, 9, 12 \rangle \langle 6, 8, 9, 13 \rangle \langle 7, 8, 9, 12, 13 \rangle$ $\langle 6, 7, 10, 15 \rangle \langle 6, 8, 10, 13, 15, 17 \rangle \langle 7, 8, 10, 12, 13 \rangle$ $\langle 4, 9, 10, 15 \rangle \langle 6, 9, 10, 13, 14, 17 \rangle \langle 7, 9, 10, 12, 13, 15 \rangle$ $\langle 8, 9, 10, 12, 13, 14, 15 \rangle \langle 6, 8, 9, 10, 13 \rangle$ $\langle 7, 8, 9, 10, 12, 13 \rangle$	$\langle 4, 5 \rangle \langle 3, 7 \rangle$ $\langle 5, 7, 8 \rangle \langle 4, 6, 9 \rangle$ $\langle 5, 7, 8, 9 \rangle$ $\langle 6, 7, 8, 9 \rangle$ $\langle 6, 7, 8, 10 \rangle$ $\langle 6, 7, 9, 10 \rangle$ $\langle 6, 7, 8, 9, 10 \rangle$	<1

Cuadro 3.2: Tabla de cálculo de semigrupos numéricos con número de Frobenius fijo, clasificándolos en semigrupos numéricos internos u hojas, así como el tiempo de cálculo.

Capítulo 4

Sitio web de estudio y análisis

Hemos explicado bastantes conceptos de los semigrupos numéricos tanto desde el ámbito matemático como sus aplicaciones en la informática (secciones 2 y 3). No obstante, con la intención de hacer más interactivo el aprendizaje de estos conceptos, se ha creado un sitio web desde cero donde se ha introducido las descripciones más relevantes de todos los ámbitos de los semigrupos numéricos, así como sus aplicaciones en el amplio campo de la informática. Para ello, se ha creado el sitio web con el lenguaje HTML (HyperText Markup Language) y se ha realizado un diseño atractivo para el usuario con CSS (Cascading Style Sheets). El enlace al sitio web es <https://mariocp10.github.io/semigruposnumericos.github.io/>

Lo primero de todo será diseñar un esqueleto que sea similar para todas las páginas, por lo que todas las páginas constarán de un menú principal en la parte superior donde se encontrarán los siguientes elementos:

- **Inicio:** se dará una breve descripción del proyecto que se ha realizado y una explicación básica del concepto.
- **Visión matemática:** se explicarán todos los conceptos teóricos de los semigrupos numéricos. Se tendrá un menú en la parte superior izquierda donde se explican todos los detalles a tener en cuenta de los semigrupos numéricos.
- **Visión informática:** se explicará los conceptos prácticos de los semigrupos numéricos. Como, con la teoría mostrada en la *Visión matemática*, podemos ver la cantidad de problemas reales que se pueden abordar con la realización de algoritmos basados en semigrupos. Se tendrá además un menú en la parte superior derecha explicando diversas áreas en donde podemos aplicar los conceptos estudiados de los semigrupos numéricos.

- **Utilidades:** contendrá de manera general las utilidades que nos presenta el estudio de los semigrupos numéricos tanto en las matemáticas, la informática, y cómo podemos aplicarlo a problemas reales. Además, se analizarán las perspectivas mostradas de estos. Como se hablan tanto del campo matemático como informático, presentará ambos menuses explicados previamente, uno en la parte superior izquierda y otro en la parte superior derecha.
- **Información:** presentará varios libros de interés con sus respectivos datos (Título, Autor/es, Editorial, Año de Publicación, Número de páginas, Idioma y Enlace a compra). También se encontrarán los menuses de los campos estudiados de las matemáticas y sus aplicaciones en informática en la parte superior izquierda y derecha.
- **Experiencias:** se comentará la experiencia personal de la realización del proyecto y cómo se ha llevado a cabo. También se presenta los dos menuses de los campos estudiados de las matemáticas y sus aplicaciones en informática en la parte superior izquierda y derecha.

Ahora pasaremos a ver cómo se han realizado los menús de la visión matemática y de la visión informática.

- **Visión matemática:** el menú corresponde a todos los conceptos teóricos matemáticos necesarios para llevar a cabo realizaciones prácticas de los algoritmos informáticos para hacer diversas funcionalidades. Entre los conceptos tenemos:

1. Número de Frobenius.
2. Género.
3. Conductor.
4. Serie de Hilbert.
5. Sistema Minimal.
6. Conjuntos de Apéry.
7. Sistema Minimal Infinito.

Estos conceptos ya han sido estudiados con detalle en la sección 2. La estructura de las páginas realizadas para cada uno de los campos es la misma, donde se tendrá una breve descripción del concepto, las propiedades que presenta, varios ejemplos para su entendimiento y las utilidades. Además, para que haya una mayor comprensión de cada uno de estos conceptos, se tendrá dos secciones adicionales en donde el usuario podrá interactuar con un programa para que vea el funcionamiento de cada uno de estos campos de los semigrupos numéricos en

un entorno dinámico. En otras palabras, para cada apartado donde se explica cada concepto, se ha realizado un programa donde el usuario podrá ver los resultados correspondientes a sus entradas relacionadas con los semigrupos numéricos. De esta manera, se fomenta el estudio de los semigrupos numéricos desde un punto de vista práctico, insertando semigrupos numéricos y mostrándole los resultados correspondientes dependiendo del campo teórico que se esté estudiando de este.

- **Visión informática:** el menú corresponde a conceptos donde la teoría matemática de los semigrupos numéricos puede aplicarse a la informática, presentando soluciones innovadoras para la realización de algoritmos. Entre los conceptos tenemos:

1. Autómatas
2. Seguridad.
3. Mochila.
4. Género Fijo.
5. Número de Frobenius Fijo.

Estos conceptos ya han sido estudiados con detalle en la sección 3. La estructura de las páginas realizadas para cada uno de los campos es la misma, donde se tendrá una breve descripción del problema a resolver, las propiedades que presenta, varios ejemplos para su entendimiento y las utilidades vistas para la realización del programa. Además, para que haya una mayor comprensión de cada uno de estos conceptos, se tendrá dos secciones adicionales en donde el usuario podrá interactuar con un programa para ver el funcionamiento de estos conceptos en un entorno dinámico. El usuario podrá observar de manera detallada los problemas que se abordan en la visión informática, donde el estudio de los semigrupos numéricos y la realización de algoritmos eficientes han sido clave para desarrollar esta visión.

Este sitio web tiene como finalidad mostrar el estudio realizado en el campo de los semigrupos numéricos, así como abordar tanto de manera teórica como práctica todos los conceptos estudiados en la realización de este trabajo fin de grado. Además, se ha pensado en la utilidad que es tener un sitio donde se explica de manera clara y concisa todos los datos necesarios para entender a la perfección este concepto, así como las utilidades plasmadas en el área de la informática, dando puntos de vista diferentes a la solución de diversos problemas relacionados con la seguridad, autómatas y algoritmos. Además, hemos podido observar como la eficiencia de estos programas es alta, dando soluciones correctas en tiempos relativamente bajos, haciendo que se caracterize la funcionalidad de estos.

Capítulo 5

Conclusiones

Hemos podido observar a lo largo de este proyecto la importancia de los semigrupos numéricos tanto en el ámbito matemático como informático. Gracias a la teoría matemática estudiada hemos podido diseñar algoritmos eficaces y óptimos para resolver diversos problemas en ambas áreas, viendo como este concepto presenta una solución ingeniosa para resolver diversos problemas. Además, hemos podido concluir la importancia de los semigrupos numéricos para diseñar algoritmos para optimizar programas, como pueden ser autómatas, seguridad e incluso problemas clásicos, dando un enfoque de la versatilidad de este concepto.

Como se ha analizado una teoría matemática avanzada, la creación de un sitio web diseñado específicamente para los semigrupos numéricos abre una puerta a un entendimiento sencillo y dinámico donde los usuarios interesados podrán estudiar, analizar y probar cada una de las funcionalidades que ofrecen los semigrupos numéricos tanto en la teoría matemática como en la informática. Además, los programas realizados se han dividido en base a su fundamento, en donde aquellos que son más teóricos se han insertado en la parte de *Visión matemática*, y los que abordan problemas informáticos se han metido en *Visión informática*. Cada uno de los apartados de estos se explica con detalle para que el usuario comprenda los fundamentos y los problemas a abordar, en donde todas esas páginas tienen una estructura similar (breve descripción, propiedades, ejemplos, utilidades y programa de cálculo). Con esto se ha querido hacer un entorno dinámico donde se plasme una parte considerable del estudio realizado en este trabajo.

Por último, gracias al estudio realizado de los semigrupos numéricos se ha podido observar como conceptos matemáticos pueden tener un gran impacto en ciertas áreas de la informática, como teoría de números o seguridad. La correcta implementación de estos nos podrá ayudar a la realización de algoritmos robustos y eficientes donde la fusión de las matemáticas y de la

informática desempeñan un papel fundamental.

Bibliografía

- [1] Tres sesiones con semigrupos numéricos. Disponible en <https://www.ugr.es/~pedro/minicurso-jarandilla.pdf>.
- [2] Los dobles de un semigrupo numérico. Disponible en <https://www.ugr.es/~arobles/Files/VIjmda-RPRV.pdf>.
- [3] The Frobenius Problem and Its Generalizations. Disponible en <https://cs.uwaterloo.ca/~shallit/Talks/frob14.pdf>.
- [4] Un algoritmo para resolver el problema de Frobenius usando bases de Gröbner. Disponible en <https://www.redalyc.org/pdf/468/46816207.pdf>.
- [5] Aplicaciones de las bases de Gröbner. Disponible en <https://www.ugr.es/~anillos/textos/pdf/2020/Aplicaciones.pdf>.
- [6] J.C. Rosales y P.A. García-Sánchez, *Numerical Semigroups*, Springer, 2009. Disponible en: <https://link.springer.com/book/10.1007/978-1-4419-0160-6>.
- [7] A. Assi, M. D'Anna y P.A. García-Sánchez, *Numerical Semigroups and Applications*, Springer, 2020. Disponible en: <https://link.springer.com/book/10.1007/978-3-030-54943-5>.
- [8] S. Lang y H.F. Trotter, *Frobenius Distributions in GL₂-Extensions*, Springer Berlin, Heidelberg, 1976. Disponible en: <https://link.springer.com/book/10.1007/BFb0082087>.
- [9] J.M. Mazón, *Nonlinear Semigroups*, ResearchGate, 2016. Disponible en: https://www.researchgate.net/publication/299680331_Nonlinear_semigroups.
- [10] W.C. Holland y J. Martinez (eds.), *Ordered Algebraic Structures*, Springer, 1997. Disponible en: <https://link.springer.com/book/10.1007/978-94-011-5640-0>.