COMUNICACIÓN EN REDES CODIFICADAS LINEALMENTE

Universidad de Valladolid

Universidad de Valladolid

autor: Ángel Kirilov Naldzhiev
tutor: Umberto Martínez Peñas

28 de septiembre de 2023

Índice general

Resumen

En el presente escrito se tratará la comunicación en redes lineales mediante el Network Coding, o codificación en redes (lineal). Esta teoría es relativamente nueva, ya que se empezó a desarrollar el siglo pasado, junto con el auge de las nuevas tecnologías y la teoría de la información. Se verá la existencia de una solución al problema de la transmisión de datos con una fuente y varios sumideros. Para ello, será necesaria la teoría de grafos, redes de flujo, cuerpos finitos y bases de Gröebner. Algunos resultados fundamentales son el teorema de máximo flujo y mínimo corte y la cota de Schwartz-Zippel, los cuales utilizaremos para resolver el problema del Network Coding.
La codificación lineal en redes tiene un gran uso en la actualidad, como por ejemplo, en los sectores de la industria, las telecomunicaciones y la medicina. Por esta razón representa un campo innovador y su estudio y desarrollo son de gran importancia.

Abstract

The present document will address communication in linear networks through linear Network Coding. This theory is relatively new, as it began to develop in the last century, alongside the rise of new technologies and information theory. We will explore the existence of a solution to the coding problem with a single source and multiple sinks. To do this, we will need graph theory, flow networks, finite fields, and Gröebner bases. The fundamental results include the max-flow min-cut theorem and the Schwartz-Zippel bound, which we will use to solve the Network Coding problem.
Linear network coding has widespread use nowadays, such as in the industry, telecommunications, and medicine sectors. This is why its study and development are of great importance, and it is also an innovative field.

Introducción

Esta memoria se centra en el tema del Network Coding (o codificación en redes en español), con un enfoque específico en la optimización del flujo de información en una red de comunicaciones determinada. Para abordar este objetivo, se hará uso de la teoría de grafos para establecer el contexto, y la cota de Schwartz-Zippel para garantizar la existencia de una solución.

En los apartados preliminares introduciremos los conceptos y resultados necesarios para desarrollar el tema que trataremos. Esto permitirá abordar la teoría posterior con mayor fluidez.

El orden de los capítulos introductorios no importa. Hablaremos sobre la teoría de grafos, ya que está directamente relacionada con lo que veremos posteriormente; y las bases de Groebner se utilizarán para probar un resultado que nos ayudará a la solución de dicho problema. En realidad, el lema de Schwartz-Zippel no requiere de bases de Gröbner para su demostración (hay demostraciones más elementales), pero usando bases de Gröbner obtenemos una cota un poco más fuerte, la de Footprint.

El primer capítulo tratará sobre grafos y redes de flujo, demostrando un resultado muy notorio: el teorema de max-flow min-cut. También se dará un algoritmo para obtener un flujo máximo: el algoritmo de Ford Fulkerson, y se dará un ejemplo detallado. Finalmente, se hablará sobre el problema de multidifusión. Se ha usado como referencia principal [1].
En el segundo capítulo veremos los cuerpos finitos y algunos conceptos vistos en asignaturas optativas del grado (álgebra conmutativa y criptografía). El resultado principal es la cota de Schwarz Zippel, la cual demostraremos utilizando bases de Gröbner para justificar la cota de footprint. La bibliografía utilizada en esta sección son los libros clásicos [2] y [3]. También ha sido necesaria una sección sobre cuerpos finitos, en la que se ha utilizado [5].
En el tercer y último capítulo se expondrá el Network Coding de una forma técnica y detallada. Nos apoyaremos en los dos capítulos anteriores para su desarrollo, en particular, serán indispensables el teorema de max-flow min-cut y el lema de Schwartz-Zippel. Para la demostración constructiva y las definiciones técnicas se ha utilizado [6]. También ha sido de ayuda el artículo

[4], el cual estudia el Network Coding utilizando geometría algebraica.

Capítulo 1

Grafos y Codificación en redes

En este capítulo introduciremos conceptos básicos sobre la teoría de grafos. Los problemas de codificación en redes se plantean y resuelven sobre grafos de diversos tipos, que repesentan el problema que se intenta resolver. Veremos el teorema del max-flow min-cut y acabaremos enunciando el problema del Network Coding sobre la red mariposa, y ofreceremos además una solución.
En este capítulo se ha usado como referencia [1].

1.1. Teorema de max flow-min-cut

Se expondrán los conceptos que utilizaremos en este apartado, que serán mayoritariamente sobre teoría de grafos y flujos sobre ellos. Acabaremos enunciando uno de los resultados principales de este trabajo: el teorema de max-flow min-cut (del inglés, teorema de máximo flujo y mínimo corte), y después veremos un algoritmo para obtener un flujo máximo con un ejemplo detallado.

Definición 1.1.1. Un ***grafo dirigido*** G es un par (V, E) donde V es el conjunto de vértices o nodos del grafo y $E = \{(u, v) \mid u, v \in V\}$ es una relación binaria entre los vértices (denominado conjunto de aristas).

Se habla de grafos dirigidos porque es donde tiene sentido considerar redes de flujo, las cuales tienen una dirección (es decir, van en un sentido, de un punto a otro/s). Para obtener una representación completa del problema se necesitan ciertos conceptos.

Definición 1.1.2. Si $G = (V, E)$ es un grafo dirigido, entonces $in(v)$ es el conjunto de las aristas que *llegan* al vértice $v \in V$, y $out(v)$ es el conjunto de las aristas que *salen* del vértice v.

Normalmente, $in(v) \neq \varnothing$ y $out(v) \neq \varnothing$, pero puede no ocurrir.
Cuando $in(v) = \varnothing$ y $out(v) \neq \varnothing$ (solamente hay aristas salientes), se dice que v es una ***fuente*** o ***salida***.
En cambio, cuando $in(v) \neq \varnothing$ y $out(v) = \varnothing$ (hay aristas entrantes pero no salientes), se dice que v es una ***llegada*** o ***sumidero***.
Las aristas (dirigidas) representan las conexiones posibles sobre los vértices del grafo, e incluso podremos asociar una cantidad de flujo máximo por cada arista. En comunicación de redes, las aristas suelen denominarse ***canales***, y los vértices del grafo, ***nodos***.
Más adelante, se verá que en el escenario *multicast*, se tendrá solamente una fuente y varias llegadas.
En la práctica se suelen dar estas situaciones en muchas ocasiones, como por ejemplo, en las redes de distribución (reparto de recursos a diferentes ciudades) o en las comunicaciones móviles (envío de mensajes o reuniones grupales).

La siguiente definición asentará las bases sobre las que se trabajarán los problemas de network coding.

Definición 1.1.3. Una ***red de flujo*** es una 4-upla (G,s,t,c) donde:

- $G = (V, E)$ es un grafo dirigido.
- s es el *nodo de salida* o *fuente* ($in(s) = \varnothing$ y $out(s) \neq \varnothing$).
- t es el *nodo de llegada* o *sumidero* ($in(t) \neq \varnothing$ y $out(t) = \varnothing$).
- $c : E \to \mathbb{R}_+$ es la *función capacidad*.

A partir de ahora se fijará una una red de flujo $\mathbb{G} = (G, s, t, c)$ siendo $G = (V, E)$ un grafo dirigido.
La función de capacidad nos indica la cantidad máxima de información que puede pasar por cada arista. Notemos que esta función solo puede tener valores no negativos. Es conveniente que cada vértice se encuentre en algún camino desde la fuente hasta el sumidero, así trabajaremos sobre un grafo **conexo** (esto es, que existe un camino desde cualesquiera dos vértices del grafo).
En realidad una red de flujo puede representar la distribución de bienes sobre un conjunto de fábricas situadas en múltiples localidades, una red de comunicación entre múltiples dispositivos, el flujo en una red de tráfico, etc. Veamos cómo podríamos aplicar a un problema la definición de red de flujo.

Ejemplo 1.1.1 (commodity flow)**.** Supongamos que debemos enviar una mercancía de una ciudad (fuente) a otra (sumidero) a través de un conjunto

de ciudades (nodos intermedios), es decir, el viaje no es directo y es imperativo pasar por ellas. Suponemos que ya existen vehículos de transporte de una ciudad a otra, donde cada vehículo admite cierta capacidad: ¿cómo podemos representar esta situación? Es posible hacerlo con una red de flujo. Este ejemplo sencillo es una posible solución (puede haber más de una solución en las redes de flujo, de hecho, suele ser lo habitual). Se observa con

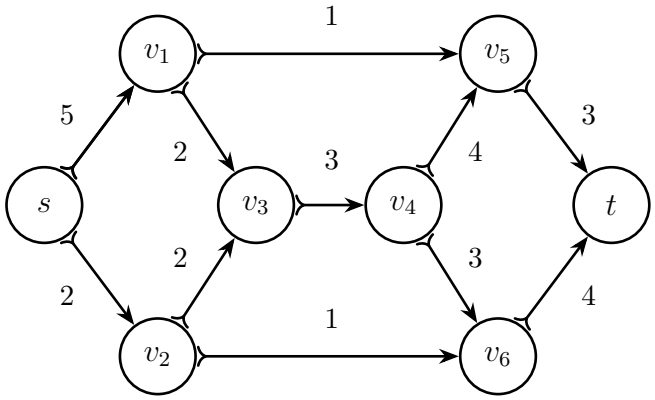


Figura 1.1: Una red de flujo

claridad la fuente y el sumidero, y los números sobre cada arista representan la capacidad de cada vehículo.

Es lógico plantearse el problema de maximizar el flujo de información de una red dada, ya que su solución podría aplicarse a la práctica y resolver una gran cantidad de problemas. En general, no es una tarea sencilla.

Definición 1.1.4. Un ***flujo*** sobre un grafo dirigido $G = (V, E)$ es una función $f : E \to \mathbb{R}_+$.

Un flujo asocia a cada arista de G un número real positivo, y nos señala la cantidad de información que fluye por dicha arista.
Es claro que se deben exceptuar los casos en los que la cantidad de flujo supera la capacidad de flujo de la arista (esta nos viene dada en una red de flujo por la función capacidad c). Por el otro lado, no queremos que se pierda información, es decir, sería conveniente que la información enviada sea igual a la información recibida en cada nodo del grafo. Si un flujo cumple estas condiciones se denomina flujo factible:

Definición 1.1.5. Un flujo $f : E \to \mathbb{R}_+$ sobre una red de flujo (G, s, t, c) es ***factible*** si se cumplen las siguientes condiciones:

- *Límite de la capacidad*: $f(e) \leq c(e), \forall e \in E$.
- *Conservación del flujo*:

$$\sum_{e \in in(v)} f(e) = \sum_{e \in out(v)} f(e), \qquad \forall v \in V.$$

La segunda condición significa que la información que entra es igual a la información que sale en cada vértice de la red de flujo. Es decir, que la suma de la cantidad de información de las aristas que entran en el vértice sea igual a la suma de la de las aristas que salen de dicho vértice. Veamos un posible flujo factible sobre la red de flujo del ejemplo 1.1.1:

Ejemplo 1.1.2. Dada la red de flujo dicha, un flujo factible sobre ella sería aquel que cumple las condiciones dichas en la definición 1.1.5. Un buen ejemplo sería la siguiente figura:

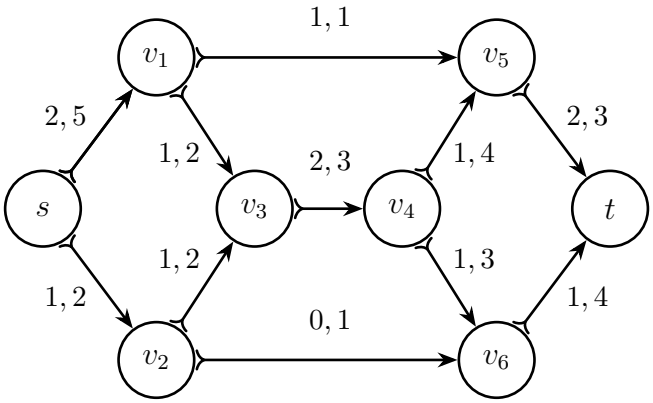


Figura 1.2: Un flujo factible sobre una red de flujo

Notemos que se respeta tanto el límite de la capacidad como la conservación del flujo. Este es un flujo cualquiera propuesto como ejemplo, a priori no tiene por qué ser óptimo pero, ¿cómo podemos saber si realmente lo es? Lo veremos más adelante, en el ejemplo 1.1.4.

Definición 1.1.6. Dada una red de flujo factible, definimos el ***flujo total*** como:

$$|f| = \sum_{e \in out(s)} f(e)$$

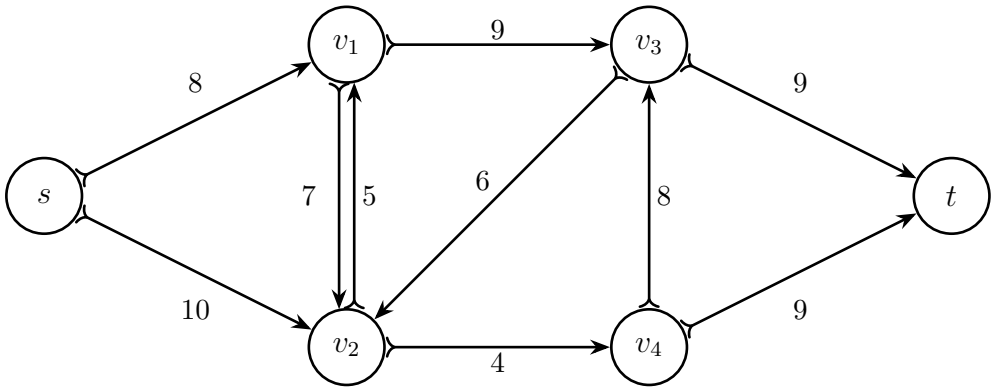
Dicho en palabras, el flujo total de una red de flujo es la suma de la información de todas las aristas salientes de la fuente.

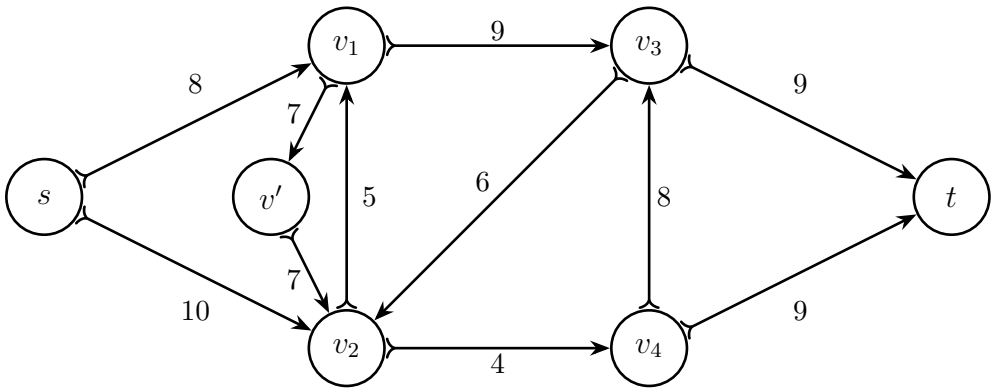
Observación. Podemos observar que en un flujo factible, el flujo total es la cantidad total de información que se desplazará en toda la red (debido a que no hay pérdida de información). En el ejemplo 1.1.2, el flujo total es $|f| = 2 + 1 = 3$.

En la práctica pueden presentarse flujos complejos con varias aristas. Queremos una generalización para resolver ese tipo de flujos, y nos apoyaremos en las aristas inversas para su resolución.

Definición 1.1.7. Sean $v_1, v_2 \in V$ dos vértices. Entonces las aristas $(v_1, v_2) \in E$ y $(v_2, v_1) \in E$ son ***antiparalelas***.

Es posible que en el grafo de partida ya aparezcan este tipo de aristas. En ese caso, se intentará transformar el grafo existente en uno equivalente que no contenga aristas antiparalelas. Veamos un ejemplo:





En la figura se puede ver que se ha solucionado el problema en los vértices v_1 y v_2 añadiendo otro vértice v', convirtiendo el grafo en uno equivalente sin aristas antiparalelas.

Más adelante demostraremos el siguiente lema.

Lema 1.1.1. *Sea (G, s, t, c) una red de flujo con $f(e) \leq c(e)$ para todo $e \in E$. Si el flujo es factible entonces*

$$\sum_{e \in out(s)} f(e) = \sum_{e \in in(t)} f(e).$$

Observación. Es natural preguntarse si en el lema anterior 1.1.1 el recíproco es cierto. En realidad no tiene por qué darse.
Podría existir una red donde el flujo total de salida de s y de llegada a t sean iguales, pero que no se cumpla la conservación del flujo en los nodos intermedios. Lo que hace falta es una red en la que haya aristas con capacidad nula que no salgan de la fuente ni lleguen al sumidero. Entonces se tiene un flujo tal que el flujo total que sale de la fuente es igual al flujo total que llega al sumidero, pero en aristas que no sean incidentes ni con la fuente ni con el sumidero el flujo es 0. Ese flujo cumpliría la cota superior $f(e) \leq c(e)$, pero no sería un flujo factible porque no se conserva el flujo en los nodos que se conectan a la fuente o al sumidero.

Para enviar la máxima cantidad de información en un flujo, vamos a querer maximizar esa cantidad respetando los límites de cada arista.¿Sería posible hacerlo sobre una red de flujo? Es obvio porder encontrar un flujo que maximice el flujo total: por una parte, el máximo está acotado por la suma de capacidades de las aristas que salen de la fuente, y por tanto existe un flujo máximo.El teorema de max-flow min-cut nos aclarará ese aspecto. Para enunciarlo, necesitamos otra definición.

Definición 1.1.8. Un ***corte*** *de un grafo G* es una partición (S, T) del conjunto de vértices V donde $s \in S$ y $t \in T$.

Los cortes nos interesan para saber la zona del grafo por la que menos información puede pasar, ese será el flujo máximo (de no ser así, no se respetaría el límite de capacidad). Esto se llama *capacidad del corte.*

Definición 1.1.9. Si $E(S, T) = \{(u, v) \mid u \in S, v \in T\} \subset E$ es el conjunto de las aristas que unen los vértices de S con los vértices de T, entonces la ***capacidad del corte*** (S, T) es la suma de la capacidad de las aristas de $E(S, T)$:

$$c(S, T) = \sum_{e \in E(S,T)} c(e).$$

Observación. La capacidad de un corte se puede ver análogamente como:

$$c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$$

Teniendo en cuenta las aristas antiparalelas antes mencionadas, generalizamos el flujo a través de un corte. Gráficamente representa la diferencia de flujo entre cada parte de la partición formada a partir del corte.

Definición 1.1.10. Sea f un flujo. El **flujo neto a través del corte** (S, T) se define como:

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u)$$

El corte de un grafo se puede ver de forma ilustrativa. Si tenemos un grafo cualquiera y hacemos un corte cualquiera en el grafo, tendremos una partición del mismo. La capacidad del corte es la suma de las capacidades de las aristas que se cortan. También se puede ver como la máxima cantidad de información que se puede transmitir de una parte a la otra.
Ejemplos de cortes extremos pero completamente válidos serían $S = \{s\}$ y $T = E \setminus S$ o, análogamente, $T = \{t\}$ y $S = E \setminus T$.
Veamos sobre el ejemplo 1.1.1 algunos cortes:

Ejemplo 1.1.3. En este caso podemos considerar muchas particiones, y como podemos intuir, será de interés el corte mínimo. Esto es porque esta será la máxima cantidad de flujo posible que podrá circular dentro del grafo, o dicho de otra manera, el flujo máximo (se puede ver intuitivamente, aunque lo demostraremos al final de esta sección). En la figura 1.1.3 se pueden ver algunos ejemplos. En todos los casos se está considrando una partición de vértices como se ha dicho anteriormente en las definiciones vistas.

1. En el grafo de arriba a la izquierda se ha considerado la partición $S_1 = \{s, v_1\}$ y $T_1 = E \setminus S = \{v_2, v_3, v_4, v_5, v_6, t\}$.
   La capacidad del corte (S_1, T_1) es $c(S_1, T_1) = 1 + 2 + 2 = 5$.

2. En el grafo de arriba a la derecha se ha considerado la partición $S_2 = \{s, v_1, v_2, v_3\}$ y $T_2 = E \setminus S = \{v_4, v_5, v_6, t\}$.
   La capacidad del corte (S_2, T_2) es $c(S_2, T_2) = 1 + 3 + 1 = 5$.

3. En el grafo de abajo a la izquierda se tiene la partición $S_3 = \{s, v_1, v_2\}$ y $T_3 = E \setminus S_3$.
   La capacidad del corte (S_3, T_3) es $c(S_3, T_3) = 1 + 2 + 2 + 1 = 6$.

4. En el grafo restante se tiene la partición $S_4 = \{s, v_2, v_6\}$ y $T_4 = E \setminus S_4$.
   La capacidad del corte (S_4, T_4) es $c(S_4, T_4) = 5 + 2 + 3 + 4 = 14$.

Recordemos que un corte mínimo es aquel cuya capacidad es mínima sobre todos los cortes posibles de las redes y que el flujo neto de un corte es

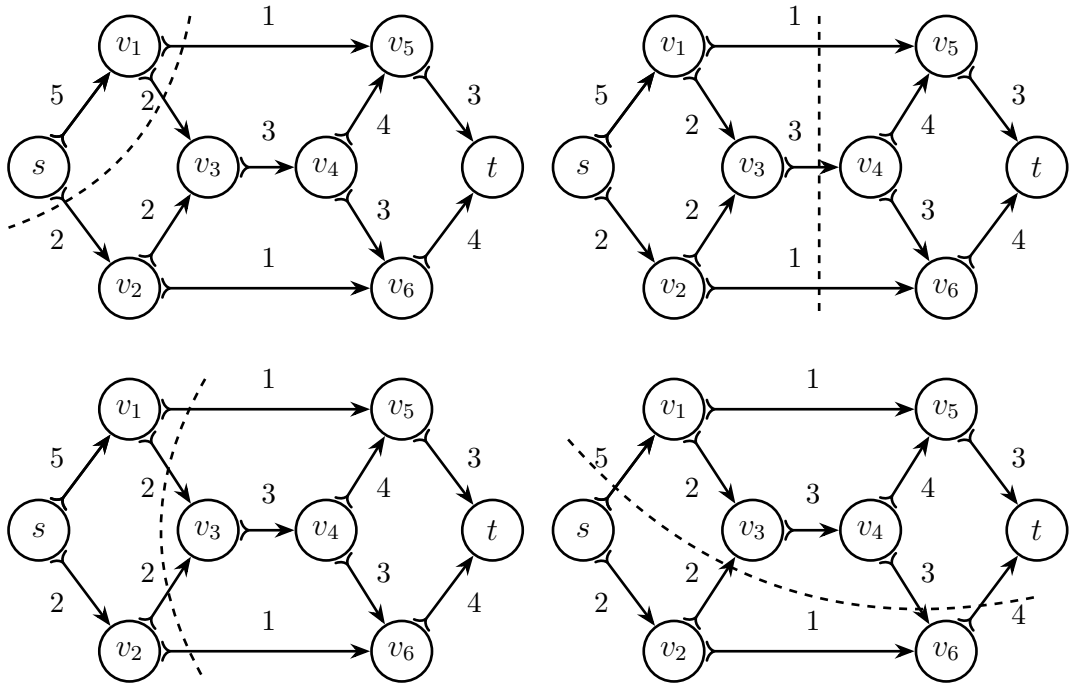


Figura 1.3: Cortes sobre una red de flujo

el flujo que va de una parte a la otra íntegramente.
¿Cuál es la diferencia entre flujo neto y capacidad de corte? El flujo neto tiene en cuenta las aristas antiparalelas, mientras que la capacidad del corte solamente considera las que van de S a T.

Observación. Un corte mínimo no es único, de hecho puede haber varios cortes mínimos. No obstante, su valor sí que es único, y veremos además que coincide con el máximo flujo que puede circular.

Necesitaremos introducir las redes residuales para explicar cómo conseguir el flujo máximo.

Definición 1.1.11. Dado un flujo factible f, se define la **capacidad residual** en $V \times V$ como:

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & si \quad (u,v) \in E \\ f(u,v) & si \quad (v,u) \in E \\ 0 & si \quad \text{en otro caso} \end{cases}$$

Podríamos entender la capacidad residual como la cantidad de flujo disponible en una red de flujo por la que ya pasa un flujo factible.

Definición 1.1.12. Dada una red de flujo $\mathbb{G} = (G, s, t, c)$ con $G = (V, E)$, se define la ***red residual*** $\mathbb{G}_f = (G_f, s, t, c_f)$ con $G_f = (V, E_f)$, donde $E_f = \{(u, v) \in V \times V \mid c_f(u, v) > 0\}$ es el conjunto de aristas residuales.

Para el planteamiento del problema se considerará primero la red de flujo, y esta no puede tener aristas antiparalelas. Después, se plantea la red residual, donde esta sí puede contener aristas antiparalelas, ya que en alguna iteración del algoritmo es posible que nos convenga reducir el flujo en alguna arista de la red principal.
En la red residual se escogen las aristas donde la capacidad residual es positiva. Esto es útil porque sobre esta red se optimizará el resultado de forma iterativa mediante el algoritmo de Ford Fulkerson, que se verá al final de esta sección.

Observación. Notemos que las aristas residuales E_f las forman las aristas de E o sus inversas, por lo tanto se tiene la desigualdad $|E_f| \leq 2|E|$ de forma trivial.

Definición 1.1.13. Sea f un flujo en G y f' un flujo en la red residual G_f. El **aumento del flujo** f por f' es la función definida en $V \times V$ tal que:

$$(f \uparrow f')(u, v) = \begin{cases} f(u, v) + f'(u, v) - f'(v, u) & si \quad (u, v) \in E \\ 0 & si \quad (u, v) \notin E \end{cases}$$

Se debe cumplir que $f \uparrow f'$ es un flujo sobre G, aunque no nos vale con cualquier flujo residual, este debe cumplir ciertas condiciones. Para que sea un flujo factible, debe darse $0 \leq f(u, v) + f'(u, v) - f'(v, u) \leq c(u, v)$ (condición del límite de capacidad) y la conservación del flujo. Lo veremos posteriormente.

Podemos ver el aumento de un flujo por otro flujo del residual como la adición de las cantidades a cada arista, descontando la información que fluye al revés (en el flujo residual existen aristas antiparalelas).
Con las redes residuales podemos aumentar o disminuir el flujo de las aristas de la red.

Lema 1.1.2. *Sea f un flujo en $\mathbb{G}$. Sea $\mathbb{G}_f$ el flujo residual de $\mathbb{G}$ inducido por f y f' un flujo en $\mathbb{G}_f$. Entonces se tiene:*

$$|f \uparrow f'| = |f| + |f'|$$

Demostración. La demostración se realiza viendo que el flujo $f \uparrow f'$ cumple las resticciones de capacidad y después se calcula la suma utilizando las definiciones, llegando a la tesis.
La prueba completa se encuentra en [1], y comienza en la página 718. □

Veamos lo que es un camino en un grafo.

Definición 1.1.14. Un **camino** l de longitud k de un vértice u a un vértice u' en un grafo $G = (V, E)$ es una sucesión $(v_0, v_1, v_2, \ldots, v_k)$ de vértices tales que $u = v_0, u' = v_k$, y $(v_{i-1}, v_i) \in E$ para $i = 1, 2, \ldots, k$. La longitud del camino s es el número de aristas en el camino. Se dice que el camino l **contiene** a los vértices $v_0, v_1, v_2, \ldots, v_k$ y a las aristas $(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$.

Observación. Aclaramos el convenio que utilizaremos. Algunos autores consideran que siempre existe un camino de longitud 0 en cada vértice del grafo, aunque no suele ser lo común. Nosotros consideraremos caminos de longitud al menos 1. Un camino de longitud exactamente 1 de u a u será un bucle, y normalmente no se tienen en cuenta grafos con bucles en el Network Coding (la explicación de ello se encuentra en el capítulo 3, y tiene que ver con el delay y el procesamiento en los nodos).

Definición 1.1.15. Un camino l es **simple** si todos los vértices en el camino son diferentes entre sí, esto es, $v_i \neq v_j$ para cada i, j tales que $i \neq j$ donde $1 \leq i \leq k$ y $1 \leq j \leq k$.

Un camino también puede contener caminos. A estos los llamamos subcaminos.

Definición 1.1.16. Un **subcamino** l' de un camino $l = \{v_0, v_1, \ldots, v_k\}$ es una subsucesión de vértices seguidos $\{v_i, v_{i+1}, \ldots, v_j\}$ donde $1 \leq i \leq j \leq k$,

Ahora trataremos los caminos de aumento. Estos son una parte clave en el algoritmo de Ford Fulkerson.

Definición 1.1.17. Un **camino de aumento** p de una red de flujo $\mathbb{G}$ es un camino simple de la fuente s al sumidero t a través del grafo residual G_f.

Observación. Un camino de aumento es un camino de G que también pasa por G_f.

Por la definición de red residual, mediante los caminos de aumento es posible aumentar el flujo de una arista sin romper el límite de la capacidad de ninguna arista de la red de flujo original.
La máxima capacidad por la que podemos aumentar el flujo de cada arista en un camino de aumento p se llama **capacidad residual de** p y se denota por:

$$c_f(p) = \text{mín}\{c_f(u_{i-1}, u_i) \mid 1 \leq i \leq k\}.$$

Lema 1.1.3. *Sea f un flujo en $\mathbb{G}$ y p un camino de aumento en $\mathbb{G}_f$. Se define la función $f_p : V \times V \to \mathbb{R}$*

$$f_p(u,v) := \begin{cases} c_f(p) & si \quad (u,v) \in p \\ 0 & si \quad (u,v) \notin p \end{cases}$$

Entonces la función f_p es un flujo en $\mathbb{G}_f$ con valor $|f_p| = c_f(p) > 0$

Demostración. Que es un flujo sobre la red residual es claro, y como los caminos de aumento son caminos que también existen en la red de flujo, se cumple también la igualdad. □

Ahora introduciremos un corolario que se cumple aplicando los dos lemas anteriores.

Corolario 1.1.1. *Sea f un flujo en $\mathbb{G}$, p un camino de aumento en $\mathbb{G}_f$ y la función*

$$f_p(u,v) = \begin{cases} c_f(p) & si \quad (u,v) \in p \\ 0 & si \quad (u,v) \notin p \end{cases}$$

Entonces, la función $f \uparrow f_p$ es un flujo en $\mathbb{G}$ con flujo total $|f \uparrow f_p| = |f| + |f_p| > |f|$.

El anterior corolario nos indica que si aumentamos el flujo por el flujo inducido por la capacidad residual del camino de aumento dado, entonces el flujo resultante se acerca más al máximo. Si realizamos este proceso varias veces, nos acercaremos cada vez más al máximo, y podríamos incluso construir un método iterativo para obtenerlo.
El método de Ford Fulkerson es iterativo y el teorema de máx-flow mín-cut nos indicará cuándo parar el algoritmo (esto será al haber alcanzado un flujo máximo), de hecho ocurrirá cuando la red residual ya no contenga más caminos de aumento.
El siguiente lema está relacionado con el lema 1.1.1 visto antes, y es de hecho, más fuerte.

Lema 1.1.4. *Sea f un flujo y (S,T) un corte cualquiera en la red. Entonces el flujo neto en (S,T) es $f(S,T) = |f|$*

Demostración. La condición de conservación del flujo, se puede reescribir como

$$\sum_{v \in V} f(u,v) - \sum_{v \in V} f(v,u) = 0$$

para cualquier vértice $u \in V$ que no sea la fuente o el sumidero.
Aplicando la fórmula del primer lema y sumando la expresión antes dicha para cada $u \in S \setminus \{s\}$ (es legítimo ya que iguala 0) se obtiene

$$|f| = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) + \sum_{u \in S \setminus \{s\}} (\sum_{v \in V} f(u,v) - \sum_{v \in V} f(v,u)).$$

Desarrollando el lado derecho de la igualdad y reagrupando términos se tiene

$$|f| = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) + \sum_{u \in S \setminus \{s\}} \sum_{v \in V} f(u,v) - \sum_{u \in S \setminus \{s\}} \sum_{v \in V} f(v,u)) =$$

$$= \sum_{v \in V} (f(s,v) + \sum_{u \in S \setminus \{s\}} f(u,v)) - \sum_{v \in V} (f(v,s) + \sum_{u \in S \setminus \{s\}} f(v,u)) =$$

$$= \sum_{v \in V} \sum_{u \in S} f(u,v) - \sum_{v \in V} \sum_{u \in S} f(v,u).$$

Como $V = S \cup T$ con $S \cap T = \varnothing$, podemos separar las sumas sobre S y T para obtener

$$|f| = \sum_{v \in S} \sum_{u \in S} f(u,v) + \sum_{v \in T} \sum_{u \in S} f(u,v) - \sum_{v \in S} \sum_{u \in S} f(v,u) - \sum_{v \in T} \sum_{u \in S} f(v,u) =$$

$$= \sum_{v \in T} \sum_{u \in S} f(u,v) - \sum_{v \in T} \sum_{u \in S} f(v,u) + (\sum_{v \in S} \sum_{u \in S} f(u,v) - \sum_{v \in S} \sum_{u \in S} f(v,u)).$$

Los sumandos entre paréntesis son los mismos, ya que para cada $x, y \in V$, el valor del flujo $f(x,y)$ aparece una única vez en cada sumando, por lo que finalmente

$$|f| = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) = f(S,T).$$

□

El siguiente corolario es bastante intuitivo pero ha sido necesario desarrollar bastante contenido para demostrarlo rigurosamente.

Corolario 1.1.2. *El flujo total $|f|$ de un flujo cualquiera f en una red de flujo $\mathbb{G}$ dada está acotado por la capacidad de cualquier corte de G:*

$$|f| \leq c(S,T), \text{ para cualquier corte } (S,T)$$

Demostración. Supongamos que (S,T) es un corte cualquiera de G y sea f un flujo. Por el lema anterior y la restricción de la capacidad, se tiene

$$|f| = f(S,T) = \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u) \leq$$

$$\leq \sum_{u\in S}\sum_{v\in T} f(u,v) \leq \sum_{u\in S}\sum_{v\in T} c(u,v) = c(S,T).$$

□

Es claro que la máxima cantidad de información que podrá fluir por una red no puede superar la capacidad de la misma, y esta viene determinada por el corte mínimo. En realidad, el flujo máximo igualará al corte mínimo, de ahí el nombre del teorema que resuelve dicho problema: max-flow min-cut. Se trata de encontrar el máximo flujo posible dentro de una red sujeto a la menor capacidad de la misma, ya que no puede superar el límite de la capacidad.
Como hemos indicado, pueden existir varios cortes. Una manera interesante de verlo es como el número de caminos (distintos) que se pueden hallar dentro de una red de flujo que van desde la salida hasta la llegada.

Llegamos al resultado culminante de esta sección:

Teorema 1.1.1 (max-flow min-cut, 1ª versión)**.** *Dada una red de flujo (G,s,t,c) y un flujo factible f sobre dicha red, son equivalentes las siguientes afirmaciones:*

1. *f es un flujo máximo en G.*
2. *La red residual $\mathbb{G}_f$ no contiene caminos de aumento.*
3. *Existe un corte (S,T) tal que $|f| = c(S,T)$ (corte mínimo).*

Demostración. (1) → (2) : Razonemos por reducción al absurdo, es decir, que f es un flujo máximo en G pero G_f tiene un camino de aumento p. Entonces por el corolario 1.1.1, el flujo obtenido de aumentar f por f_p, donde f_p viene dado por el lema 1.1.3, es un flujo en G cuyo valor es estrictamente mayor que $|f|$, contradiciendo la definición de flujo máximo.
(2) → (3) : Suponer que G_f no tiene caminos de aumento es lo mismo que suponer que no existe un camino de aumento en G de s a t, o lo que es lo mismo, no existen caminos en G_f. Definamos el conjunto:

$$S = \{v \in V | \text{existe un camino de s a v en } G_f\}$$

siendo $T = V \setminus S$. La partición (S,T) es un corte: tenemos $s \in S$ y $t \notin S$ porque no hay ningún camino de s a t en G_f por hipótesis. Consideremos ahora dos vértices $u \in S$ y $v \in T$. Si la arista $(u,v) \in E$, se debe tener $f(u,v) = c(u,v)$, ya que si fuera de otra manera, entonces se daría $(u,v) \in E_f$, lo cual implicaría $v \in S$. En cambio, si $(v,u) \in E$, se tendría $f(u,v) = 0$, porque de lo contrario, $c_f(u,v) = f(v,u)$ sería estrictamente positivo, y entonces $(u,v) \in E_f$, y vimos que eso no puede ser. Entonces, como $(u,v), (v,u) \notin E$, se tiene $f(u,v) = f(v,u) = 0$, y como consecuencia

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in T} \sum_{v \in S} f(v,u) =$$

$$= \sum_{u \in S} \sum_{v \in T} c(u,v) - \sum_{u \in T} \sum_{v \in S} 0 = c(S,T).$$

Por el lema 1.1.4, obtenemos $|f| = f(S,T) = c(S,T)$.
(3) $\rightarrow$ (1) : Como $|f| \leq c(S,T)$ para cada corte (S,T), la condición $|f| = c(S,T)$ significa que f es un flujo máximo. □

El teorema del max-flow min-cut ha quedado demostrado, no obstante, se puede reformular el teorema de forma equivalente al anterior:

Teorema 1.1.2 (max-flow min-cut, 2ª versión)**.** *Dada una red de flujo, se tiene que*

$$\text{máx}\{|f| : \ f \textit{ flujo factible}\} = \text{mín}\{c(S,T) : \ (S,T) \textit{ corte}\},$$

donde $|f|$ es el flujo total en la red donde f es un flujo factible y $c(S,T)$ es la capacidad del corte (S,T).

El corte mínimo nos indica el mínimo de aristas a eliminar para que se desconecte totalmente la fuente de la llegada.

1.1.1. Algoritmo de Ford Fulkerson

Este método incrementa iterativamente el valor del flujo hasta que no haya más caminos de aumento (es entonces cuando termina).
El algoritmo sería el siguiente:

En cada iteración dentro del bucle mientras, se encuentra un camino de aumento p y se usa para modificar el flujo f realizando $f \uparrow f_p$.
Se computa el flujo máximo en una red de flujo $\mathbb{G}$ aumentado el flujo en las aristas de E.
La función c_f varía con el tiempo y almacena la capacidad residual del camino

Algoritmo 1: Ford − Fulkerson

Input : (G, s, t, c)
Output: Un flujo factible máximo f sobre la red de flujo (G, s, t, c).

```
1 para (u, v) ∈ E hacer
2     f(u, v) := 0
3 mientras existe un camino de aumento p de s a t en la red residual
    G_f hacer
4     c_f(p) := mín{c_f(u, v) | (u.v) ∈ p}
5     para (u, v) ∈ p hacer
6         si (u, v) ∈ E entonces
              f(u, v) := f(u, v) + c_f(p)
7         en otro caso
              f(u, v) := f(v, u) − c_f(p)
8 devolver f
```

p, esto es, cuánto más podemos añadir hasta igualar el límite de capacidad. Cuando dejen de existir caminos de aumento, se acaba el bucle mientras y el flujo se convierte en un flujo máximo (esto se cumple por el teorema anterior, el cual decía que si no hay caminos de aumento, entonces el flujo es máximo), que será lo que el algoritmo devuelva.

Observación. Es posible que el bucle no acabe. Esto se puede dar cuando las capacidades de los ejes sean números irracionales. En esas situaciones, existirán caminos de aumento con capacidades residuales cada vez más pequeñas, pero nunca nulas. Aquí sería recomendable introducir un límite para el número de las iteraciones, para conseguir una capacidad residual tan pequeña como se quiera.

Cabe destacar que el algoritmo de Ford Fulkerson no es ni el único método ni el más eficiente para optimizar el flujo de una red. No obstante, es relativamente sencillo y eficaz.
Hay muchos algoritmos más eficientes que el que se acaba de exponer, aunque también más elaborados. Un ejemplo es el de Edmond-Karps, que se puede encontrar en [1] (págs. 728-730).

1.1.2. Aplicación práctica

Ahora veremos aplicado a la práctica lo dicho en la teoría. Dado que el funcionamiento del algoritmo ya está demostrado, nos limitaremos a aplicarlo. En la práctica resulta bastante mecánica su implementación.

Ejemplo 1.1.4. Intentaremos hallar un flujo máximo sobre la red de flujo que expusimos al principio en el ejemplo 1.1.1. Primero cogeremos un camino de aumento en la red. Suele dar igual el camino que se escoja, al final siempre se llegará a un flujo máximo (no siempre el mismo), pero normalmente escogeremos las aristas que tengan mayor capacidad para "quitárnoslas de encima"lo antes posible.
En este caso, escogeremos el camino de aumento marcado en la figura 1.1.4, y añadiremos el mismo flujo a cada arista, y esa cantidad será el mínimo de las capacidades del camino de aumento.

Por un lado tenemos el flujo sobre el grafo que se va actualizando en cada iteración, y por el otro las distintas capacidades residuales, que van disminuyendo como consecuencia del aumento del flujo.
Después de la segunda iteración, tenemos que las dos capacidades de salida son iguales, por lo que a priori parece lo mismo escoger cualquiera de los dos. No obstante, nos fijamos que después del vértice v_1 hemos agotado todas las capacidades, y estamos obligados a escoger la arista que lleva al vértice v_2, ya que es la única opción disponible.

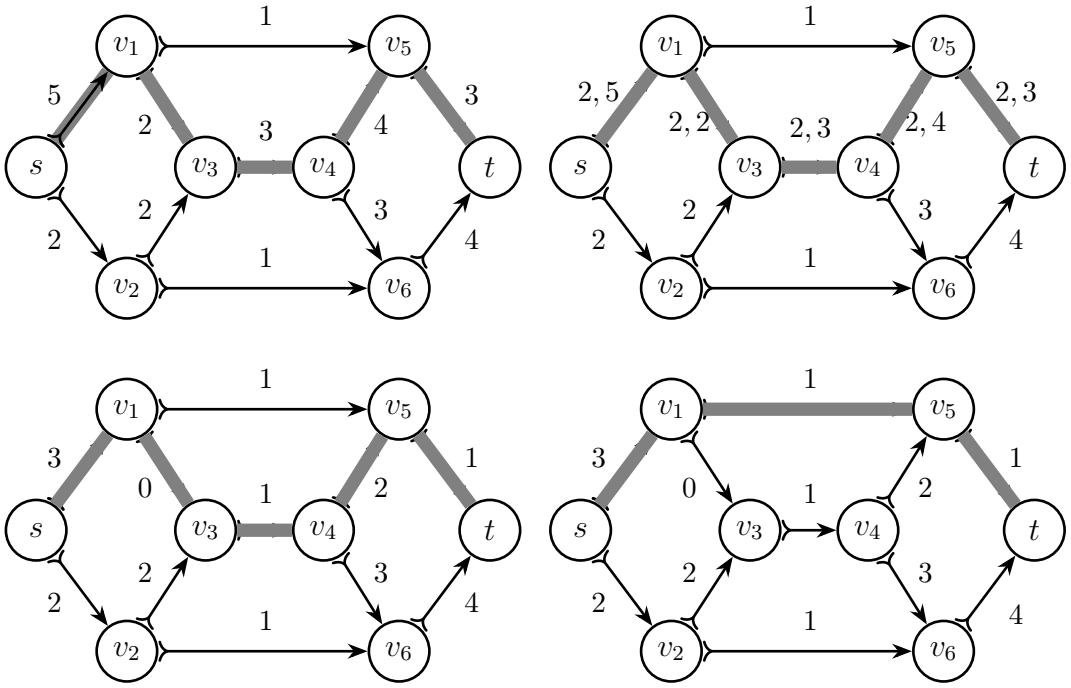


Figura 1.4: Algoritmo de Ford Fulkerson - 1

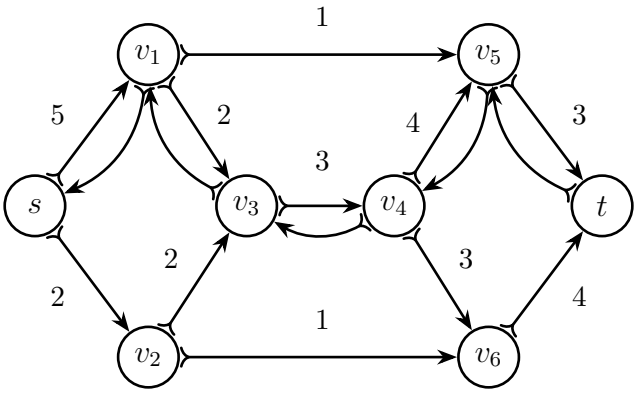


Figura 1.5: Grafo residual - Paso 1
Nota: El valor de las aristas que van en sentido contrario es 2

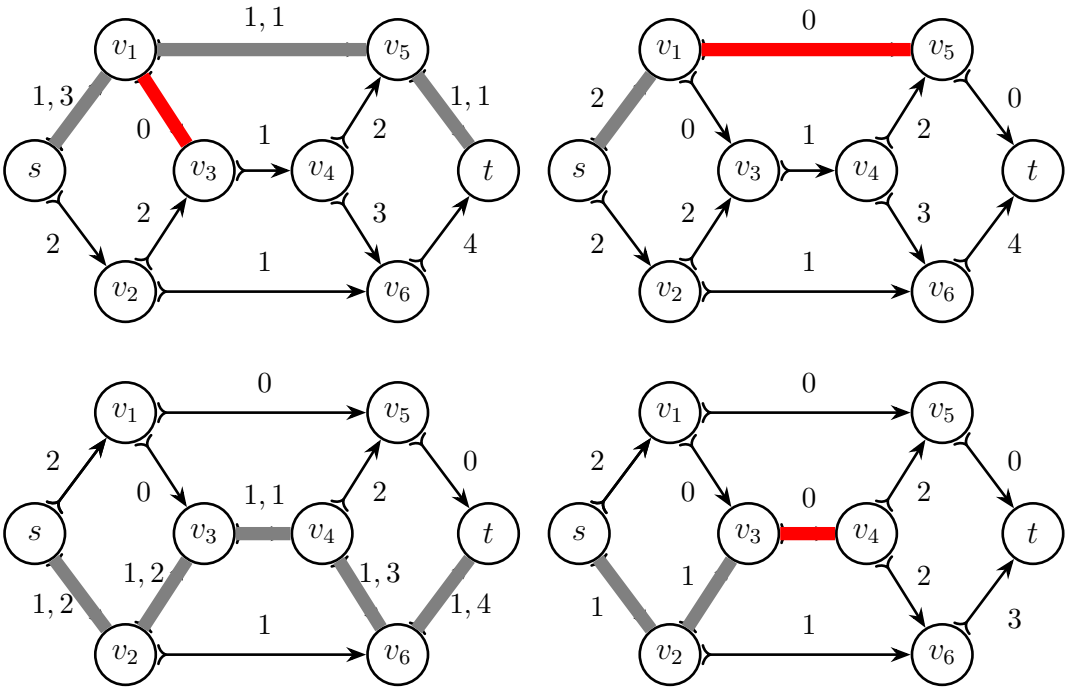


Figura 1.6: Algoritmo de Ford Fulkerson - 2

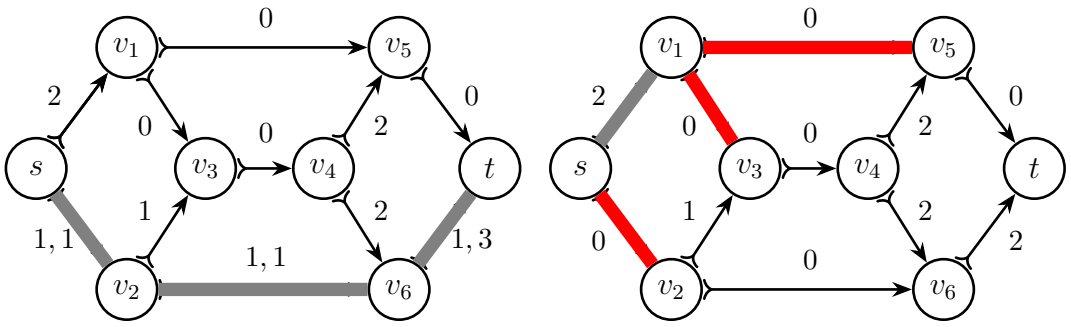


Figura 1.7: Algoritmo de Ford Fulkerson - 3

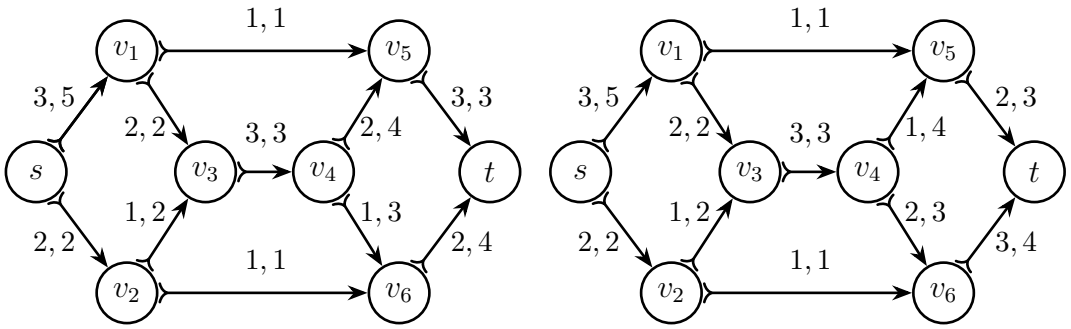


Figura 1.8: Flujos máximos sobre la red

Llegamos así al final hasta que no hay más caminos de aumento, como bien se puede apreciar en la figura 1.1.4.
El grafo residual obtenido en el primer paso sería el de la figura 1.1.5. En la imagen vemos cómo se deriva del grafo original, añadiendo las aristas antiparalelas en aquellas aristas que se encuentran en el camino de aumento escogido. El flujo en estas aristas es la capacidad mínima de las aristas del camino de aumento escogido. Pondremos la figura para el primer paso, para el resto de pasos se realiza de manera análoga. Podría parecer que se puede hacer directamente sin considerar el grafo residual, pero hay que recordar que este es un ejemplo sencillo, y que al computar casos más complejos sí que sería necesario. Aunque no se construya explícitamente se está teniendo en cuenta de una manera implícita.v
En el siguiente paso se coge el camino de aumento sobre el grafo habiendo quitado las capacidades de las aristas antiparalelas.
La red de flujo final, se obtiene sumando en cada arista todos los flujos que se han ido añadiendo acumulativamente. Vemos que el flujo no es el mismo que el del ejemplo 1.1.1, de hecho el flujo total de esa red es $3 < 5$.
Se podría haber optado por otras iteraciones, y se habría obtenido otro flujo diferente aunque también máximo. El corte mínimo de estos dos flujos debe tener el mismo valor: 5.
En la imagen 1.1.8 vemos a la izquierda el flujo máximo obtenido de la manera en la que se ha explicado, y a la derecha otro flujo máximo sobre la red.

Si se quiere ver otro ejemplo sobre una red de flujo diferente, se puede ver en [1] (págs. 726-727).

1.2. El caso de multidifusión

¿Es siempre aplicable el teorema de max-flow min-cut?¿Existen situaciones en las que no lo podemos aplicar?
En la sección anterior se hablaba de redes de flujo con una salida y una llegada, pero existen situaciones en las que queremos enviar varios mensajes a varios puntos de llegada de manera simultánea (es lo que se conoce como escenario **multicast** en telecomunicaciones o en teoría de la información, en castellano, multidifusión).
Para dar respuesta a esta pregunta, planteamos el problema de Network Coding en un ejemplo muy conocido: **the butterfly network** (en castellano, la red mariposa).
Tenemos una salida s, dos mensajes X e Y, y dos destinatarios o llegadas t_1 y t_2. Asumimos que cada arista solamente tiene capacidad para enviar un

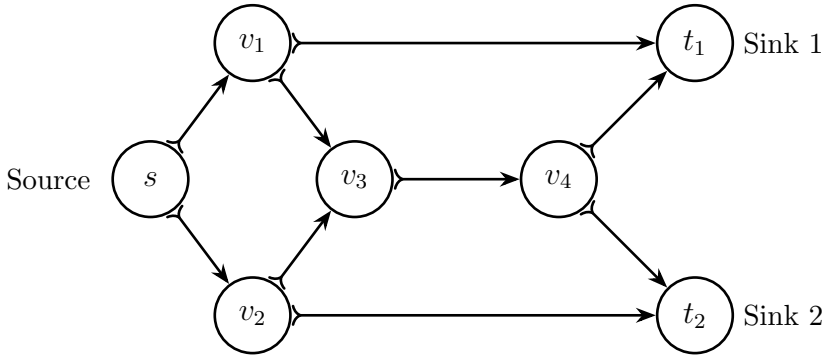


Figura 1.9: Problema multidifusión - Red mariposa

mensaje de manera simultánea.

Si cada canal tiene capacidad 1, entonces el mínimo corte de la red mariposa entre la fuente y cualquiera de las dos llegadas tiene capacidad 2, por lo que la fuente deberá ser capaz de enviar 2 mensajes tanto a una llegada como a la otra., aunque no se pueden enviar los dos mensajes a la vez a los dos sumideros. La razón es el nodo central v_3, que tiene que decidir entre enviar el mensaje a o el mensaje b. Se puede ver que tanto en un caso como

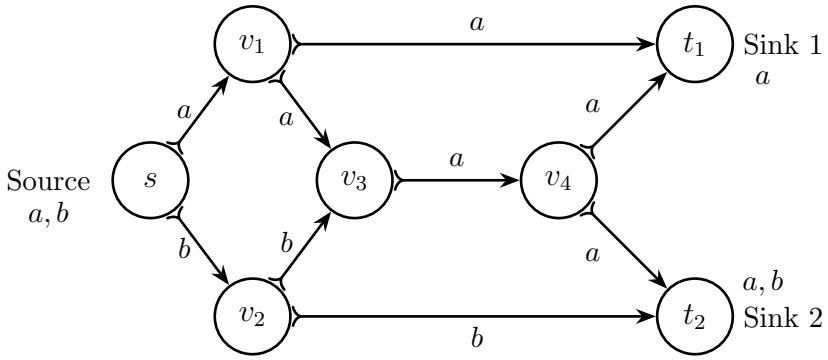


Figura 1.10: Problema multidifusión - Red mariposa - Caso 1

en el otro no hay manera de enviar los dos mensajes simultáneamente a los puntos de llegada.

Sin embargo, no todo está perdido, aún se puede ingeniar algo para que ambos destinatarios obtengan ambos mensajes, o al menos una combinación lineal de ellos, de tal forma que se puedan obtener ambos, uno a partir del otro. Esto es posible y asienta las bases de la codificación lineal en redes.

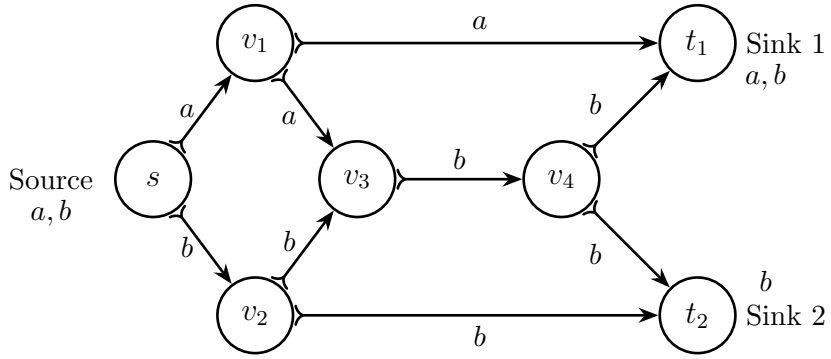


Figura 1.11: Problema multidifusión - Red mariposa - Caso 2

Si en lugar de enviar el mensaje a o b en el nodo central decidimos enviar el mensaje $a + b$ (este existe puesto que se trata de codificación lineal en redes), podremos obtener la combinación linealen forma de la suma de ambos mensajes en cada nodo. La manera de obtener a y b a partir de a o b y $a+b$ es sencilla: $b = (a + b) - a$ y $a = (a + b) - b$. Obtenemos así una matriz de codificación en cada llegada, en esto se basa el Network Coding.

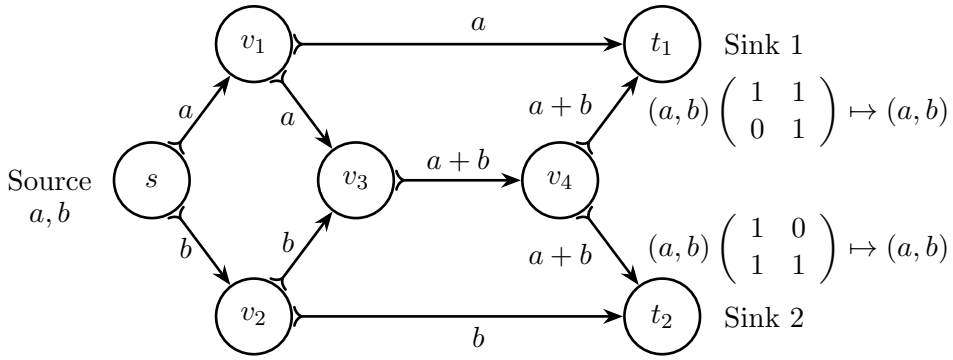


Figura 1.12: Problema multidifusión - Solución

Se ha resuelto un problema básico, y a partir de éste se puede abstraer y aplicar a problemas más complejos.
De hecho, se podría aplicar a un problema generalizado con n mensajes y j llegadas. Incluso también varias salidas, pero como hemos dicho, no consideraremos ese escenario.
El capítulo 3 de este escrito tratará con mayor profundidad este apartado y se resolverá el problema de forma general, pero con el uso de la cota de Schwarz-Zippel sobre el número de ceros de un polinomio. En el capítulo 2

daremos una cota más fuerte que la de Schwartz-Zippel, conocida como cota de footprint, para la cual necesitaremos bases de Gröbner.

Capítulo 2

Lema de Schwartz-Zippel

En este capítulo trataremos conceptos relativos a los anillos de polinomios en varias variables sobre un cuerpo $\mathbb{F}$ cualquiera: orden monomial, grado de un polinomio en varias variables, etc., y llegaremos a resultados importantes relacionados con las bases de Gröbner. La cota de Footprint y el lema de Schwartz-Zippel son acotaciones sobre el número de ceros de polinomios en varias variables que utilizan los datos sobre los monomios del polinomio (caso de Footprint), o simplemente sobre el grado y el tamaño del cuerpo (Schwartz-Zippel). Usaremos estos resultados para demostrar que existe una solución lineal al problema de Network Coding para cuerpos finitos suficientemente grandes.
Una referencia muy útil en este apartado es [2].
También se ha usado [3].

2.1. Ideales monomiales y bases de Gröbner

Primero debemos definir lo que es un orden monomial. La notación que utilizaremos será la siguiente: si $\alpha = (\alpha_1, \alpha_2, \dots \alpha_n) \in \mathbb{Z}_{>0}^n$ entonces $x^\alpha = {x_1}^{\alpha_1} \cdot {x_2}^{\alpha_2} \cdots {x_n}^{\alpha_n} \in \mathbb{F}[x_1, x_2, \dots, x_n]$. Hay varios tipos de órdenes monomiales, como el orden lexicográfico o el orden lexicográfico graduado. En general, todos deben cumplir la siguiente definición:

Definición 2.1.1. Un ***orden monomial*** $>$ sobre $\mathbb{F}[x_1, x_2, \dots, x_n]$ es una relación en $\mathbb{Z}_{>0}^n$, o equivalentemente en el conjunto de monomios x^α con $\alpha \in \mathbb{Z}_{>0}^n$, tal que:

- $>$ es un *orden total* en $\mathbb{Z}_{>0}^n$.
- si $\alpha > \beta$ y $\gamma \in \mathbb{Z}_{>0}^n$, entonces $\alpha + \gamma > \beta + \gamma$.

- $>$ cumple el *buen orden* en $\mathbb{Z}^n_{>0}$.

La última propiedad de la definición 2.1.1 indica que cada subconjunto no vacío en $\mathbb{Z}^n_{>0}$ tiene un elemento mínimo respecto al orden $>$. En realidad, se puede demostrar que esta última condición es superflua, y que las condiciones realmente importantes son las dos primeras.
Un ejemplo de orden monomial es el orden lexicográfico, aunque puede interesar utilizar otros órdenes monomiales, por ejemplo para mejorar la eficiencia de algunos algoritmos.

Definición 2.1.2 (Orden lexicográfico)**.** Sea $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\beta = (\beta_1, \beta_2, \ldots, \beta_n) \in \mathbb{Z}^n_{\geq 0}$. Entonces $\alpha >_{lex} \beta$ si la componente no nula situada más a la izquierda del vector diferencia $\alpha - \beta$ es positiva.

Observación. Se suele utilizar el abuso de notación $x^\alpha > x^\beta$ cuando $\alpha > \beta$.

Para llegar al resultado que nos interesa, es necesario introducir las siguientes definiciones:

Definición 2.1.3. Sea $f = \sum_\alpha a_\alpha x^\alpha \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ un polinomio no nulo y $>$ un orden monomial. Se definen:

1. El ***grado total*** de f: $MG(f) = \text{máx}\{\alpha \mid a_\alpha \neq 0\}$.
2. El ***coeficiente líder*** de f es el término que acompaña al monomio de máximo grado: $CL(f) = a_{MG(f)} \in \mathbb{F}$.
3. El ***monomio líder*** (o forma inicial) de f es $ML(f) = x^{MG(f)}$.
4. El ***término líder*** de f: $TL(f) = CL(f)ML(f)$.
5. El ***grado de un monomio*** $M = x_1^{a(1)} x_2^{a(2)} \cdots x_n^{a(n)} \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ se define como: $deg(M) = a(1) + a(2) + \cdots + a(n)$.
6. El ***grado de un polinomio*** $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ es el grado más alto de un monomio en f.

La siguiente definición es fundamental en esta sección porque lo que buscamos es dar una cota superior sobre el número de ceros de un polinomio de varias variables.

Definición 2.1.4. Sea I un ideal de $\mathbb{F}[x_1, x_2, \ldots, x_n]$. Se define $V(I)$ como el conjunto

$$V(I) = \{(a_1, a_2, \ldots, a_n) \in \mathbb{F}^n \mid f(a_1, a_2, \ldots, a_n) = 0, \forall f \in I\}.$$

El siguiente lema muestra los grados de los polinomios según las diferentes operaciones que se pueden realizar. La demostración es elemental.

Lema 2.1.1. *Sean f y g dos polinomios de $\mathbb{F}[x_1, x_2, \ldots, x_n]$. Entonces*

1. $MG(f \cdot g) = MG(f) + MG(g)$.
2. $MG(f + g) \leq \text{máx}\{MG(f), MG(g)\}$. *Se da la igualdad si* $MG(f) \neq MG(g)$.

La división es una de las operaciones elementales en matemáticas. Sabemos dividir números e incluso polinomios en una variable pero, ¿existe algún método para realizar divisiones de polinomios en varias variables? ¿Será el resultado siempre igual o variará según el orden monomial que se escoja? El siguiente teorema arroja luz a las cuestiones planteadas.

Teorema 2.1.1 (Algoritmo de división)**.** *Sea $>$ un orden monomial en $\mathbb{Z}^n_{>0}$ y $F = (f_1, \ldots, f_s)$ una s-upla de polinomios no nulos de $\mathbb{F}[x_1, x_2, \ldots, x_n]$. Si $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ entonces existen $q_i, r \in \mathbb{F}[x_1, x_2, \ldots, x_n], i = 1, 2, \ldots, s$ tales que*

$$f = q_1 f_1 + \ldots + q_s f_s + r,$$

donde o bien $r = 0$ o bien r no es divisible por $TL(f_j)$ para cada $j = 1, \ldots s$.

Cuando realizamos una división en un anillo de polinomios en varias variables, el resultado puede no ser único. Esto es, puede depender del orden que se siga. El algoritmo que se utiliza se encuentra en [2] (pág. 65).
En verdad, es posible caracterizar a los ideales que están generados por un monomio, y son de hecho bastante frecuentes en el Álgebra Conmutativa.

Definición 2.1.5. Un ideal $I \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ es *monomial* si existe un subconjunto A de $\mathbb{Z}^n_{>0}$ tal que $I = \langle x^\alpha \mid \alpha \in A \rangle$. Es decir, los elementos de I son combinaciones lineales finitas de la forma $\sum_{\alpha \in A} h_\alpha x^\alpha$ con $h_\alpha \in \mathbb{F}[x_1, x_2, \ldots, x_n]$.

Es posible caracterizar los monomios de un ideal monomial:

Lema 2.1.2. *Sea I un ideal monomial. El monomio x^β está en I si y solo si existe $\alpha \in A$ tal que x^α divide a x^β.*

Demostración. Supongamos que $x^\beta \in I$. Al ser I un ideal, podemos expresar x^β como $x^\beta = \sum_{i=1}^m p_i x^{\alpha_i}$ donde $p_i \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, $\alpha_i \in A \subset \mathbb{Z}^n_{>0}$ (cada coeficiente se escribiría $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in})$ para cada i). Entonces, cualquier término del lado derecho de la igualdad es divisible por algún x^{α_i}. Por tanto, en el lado izquierdo de la igualdad también se debe cumplir lo mismo. El recíproco es inmediato por la definición de ideal. □

Con el anterior lema, se pueden conocer los monomios pertenecientes a un ideal monomial para $n = 2$ identificando el punto $(a, b) \in \mathbb{Z}^2_{>0}$ con el monomio $x^a y^b \in \mathbb{F}[x, y]$ (y de hecho se pueden dibujar).

Observación. En las matemáticas discretas, los puntos como (a, b) se denominan *puntos reticulares*, y su noción se puede generalizar.

Veámoslo con un ejemplo sencillo pero ilustrativo.

Ejemplo 2.1.1. El punto $(2, 2)$ que se ve en la figura 2.1 se identifica con el monomio $x^2y^2 \in \mathbb{F}[x, y]$.

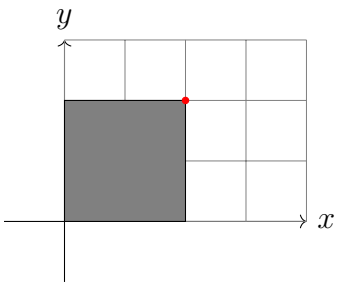


Figura 2.1: Identificación de punto reticular con monomio

El siguiente resultado indica que la pertenencia de un polinomio f en un ideal monomial I puede determinarse viendo los monomios de f.

Lema 2.1.3. *Sea I un ideal monomial y sea $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$. Entonces son equivalentes:*

1. *f está en I.*
2. *Cualquier término de f está en I.*
3. *f es una combinación $\mathbb{F}-$lineal de los monomios de I.*

Demostración. (1) $\rightarrow$ (2) : Si $f \in I$, podemos escribir

$$f = \sum_{i=1}^{s} g_i \cdot x^{\alpha(i)} \qquad g_i \in \mathbb{F}[x_1, x_2, \ldots, x_n],\ x^{\alpha(i)} \in I.$$

De la misma manera que en el lema 2.1.2, vemos que todo término de f es divisible por algún $\alpha(i)$, obteniendo (2).
(2) $\rightarrow$ (3) : Es inmediato.
(3) $\rightarrow$ (1) : Se cumple por definición. □

De la parte (3) del anterior resultado, se obtiene el siguiente corolario, que será de utilidad en la demostración del lema de Dickson:

Corolario 2.1.1. *Dos ideales monomiales I, J son iguales si y solamente si contienen los mismos monomios.*

Definición 2.1.6. Sea $I \subseteq k[x_1, \ldots, x_n]$ un ideal, y sea $>$ un orden monomial en $k[x_1, \ldots, x_n]$. Entonces:

1. Denotamos por $LT(I)$ al conjunto de términos líder de elementos (no nulos) de I, es decir:

$$LT(I) = \{c \cdot x^\alpha \mid \text{existe } f \in I \setminus \{0\} \text{ tal que } LT(f) = cx^\alpha\}.$$

2. Denotamos por $\langle LT(I)\rangle$ al ideal generado por los elementos de $LT(I)$.

Observación. Si $I = \langle f_1, f_2, \ldots, f_s\rangle$, entonces los ideales $\langle LT(f_1), LT(f_2), \ldots, LT(f_s)\rangle$ y $\langle LT(I)\rangle$ pueden ser diferentes. Se cumple que

$$\langle LT(f_1), LT(f_2), \ldots, LT(f_s)\rangle \subseteq \langle LT(I)\rangle,$$

ya que $LT(f_i) \in LT(I) \subseteq \langle LT(I)\rangle$ por definición. No obstante, $\langle LT(I)\rangle$ puede contenerlo estrictamente.
Por ejemplo, si $I = \langle f_1, f_2\rangle = \langle x^3 - 2xy, x^2y - 2y^2 + x\rangle$ y usamos el orden lexicográfico en $\mathbb{F}[x, y]$, tenemos que

$$x \cdot (x^2y - 2y^2 + x) - y \cdot (x^3 - 2xy) = x^2,$$

así que $x^2 \in I$, luego $x^2 = LT(x^2) \in \langle LT(I)\rangle$. Sin embargo, x^2 no es divisible por $LT(f_1) = x^3$, ni tampoco por $LT(f_2) = x^2y$, por lo que x^2 está en el ideal $\langle LT(f_1), LT(f_2)\rangle$ por el lema 2.1.2.

Es de suma importancia la siguiente definición.

Definición 2.1.7 (Base de Gröbner)**.** Fijado un orden monomial, sea $I \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ un ideal. Un subconjunto finito $G = \{g_1, \ldots, g_t\}$ de I es una **base de Gröbner** de I si se cumple

$$\langle LT(I)\rangle = \langle LT(g_1), \ldots, LT(g_t)\rangle.$$

Observación. Cuando $I = \{0\}$, por convenio $G = \varnothing$.

El resto r puede variar al cambiar el orden monomial o el orden de los polinomios f_i.
El proceso del algoritmo de división en varias variables es análogo al de

una variable, pero en este caso se debe fijar un orden monomial para ir dividiendo por los términos líderes de los polinomios y tener los monomios de cada polinomio con un cierto orden. Dependiendo del orden que se fije, tendremos resultados diferentes (la descomposición no es única pero el resto de la división sí que lo es si los polinomios f_i forman una base de Groebner.). Esto se ve más claro con el siguiente resultado:

Proposición 2.1.1. *Sea $I \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ un ideal y sea $G = \{g_1, g_2, \ldots, g_t\}$ una base de Gröbner de I. Entonces, para $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ existe un único polinomio $r \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ tal que:*

1. *Existe algún $g \in I$ tal que $f = g + r$.*
2. *Ningún término de r es divisible por cualquiera de los monomios $LT(g_1), LT(g_2), \ldots, LT(g_t)$.*

En particular, r es el resto de la división de f por G sin importar el orden de los elementos de G cuando se realiza el algoritmo de división.

Demostración. Por el algoritmo de división 2.1.1 se obtiene $f = q_1 \cdot g_1 + q_2 \cdot g_2 + \cdots + q_t \cdot g_t + r$, donde r cumple (1). También se cumple (2) si denotamos $g = q_1 \cdot g_1 + q_2 \cdot g_2 + \cdots + q_t \cdot g_t \in I$ (ya que G es una base de Gröbner de I). Queda así probada la existencia de r.
Ahora supongamos que existen $g, g' \in I$ y $r, r' \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ tales que $f = g + r = g' + r'$ y satisfacen (1) y (2). Entonces $r' - r = g - g' \in I$, así que si $r \neq r'$ entonces $LT(r' - r) \in \langle LT(I) \rangle = \langle LT(g_1), LT(g_2), \ldots, LT(g_t) \rangle$. Por el lema 2.1.2, se sigue que $LT(r' - r)$ es divisible por algún $LT(g_i)$, pero esto entra en contradicción con (1), ya que ningún término de r, r' es divisible por algún $LT(g_i)$. La única opción plausible es que $r' - r = 0$, y así la unicidad quedaría probada. □

Es común denominar al resto r la *forma inicial* de f.
Como corolario de la proposición anterior, se puede obtener un criterio (mediante bases de Gröbner) para determinar cuándo un polinomio dado f pertenece a un ideal I.

Corolario 2.1.2. *Sea $G = \{g_1, \ldots, g_t\}$ una base de Gröbner para un ideal $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$, y sea $f \in \mathbb{F}[x_1, \ldots, x_n]$. Entonces, $f \in I$ si y solo si el resto de la división de f por G es cero.*

Demostración. Si $r = 0$, entonces ya hemos vimos que $f \in I$.
Recíprocamente, supongamos que $f \in I$. Si se escribe $f = f + 0$, se tiene que f satisface las dos condiciones de la proposición anterior, y por tanto se deduce que 0 es el resto de la división de f por G. □

Por lo dicho anteriormente, queda claro que al realizar la división, siempre que la base sea de Gröbner y se haya fijado un orden monomial, el resto será único. Esto motiva la siguiente definición.

Definición 2.1.8. Fijado un orden monomial, si F es una base de Gröbner, decimos que r es el ***resto*** de la división de f por F, y se denota por $r = \overline{f}^F$.

El siguiente teorema afirma que es posible definir una cantidad finita de generadores en un ideal monomial.

Teorema 2.1.2 (Lema de Dickson)**.** *Sea $I = \langle x^\alpha \mid \alpha \in A\rangle \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$. Entonces I se puede escribir de la forma $I = \langle x^{\alpha(1)}, x^{\alpha(2)}, \ldots, x^{\alpha(s)}\rangle$ donde $\alpha(1), \alpha(2), \ldots, \alpha(s) \in A$. Es decir, todo ideal monomial tiene una base finita.*

Demostración. La prueba se realiza por inducción sobre el número de variables n.
Para $n = 1$, se tiene que I está generado por los monomios x_1^α, con $\alpha \in A \subset \mathbb{Z}_{\geq 0}$ un entero positivo. Sea β el elemento más pequeño de A. Entonces $\beta \leq \alpha$ para cualquier $\alpha \in A$, por lo que x_1^β divide a todos los posibles generadores x_1^α. De ello se sigue que $I = \langle x_1^\beta\rangle$.
Supongamos la propiedad cierta para $n-1$ y probémosla para $n > 1$ variables. Consideramos el anillo de polinomios $\mathbb{F}[x_1, x_2, \ldots, x_{n-1}, y]$, así tenemos elementos de la forma $\mathbf{x}^\alpha y^m$ con $\alpha \in \mathbb{Z}_{>0}^{n-1}$ y $m \in \mathbb{Z}_{>0}$.
Sea $I \subset \mathbb{F}[x_1, x_2, \ldots, x_{n-1}, y]$ un ideal monomial. Para encontrar generadores para I, consideramos el ideal 'proyección' de I sobre $\mathbb{F}[x_1, x_2, \ldots, x_{n-1}]$: $J = \langle x^\alpha \mid x^\alpha y^m \in I\rangle$, para algún$m > 0$, $m \in \mathbb{N}$. Por hipótesis de inducción, J está finitamente generado, y podemos decir que $J = \langle x^{\alpha(1)}, x^{\alpha(2)}, \ldots, x^{\alpha(s)}\rangle$, para algunos $\alpha(i) \in \mathbb{Z}_{\geq 0}^{n-1}$.
Podemos ver por la definición de J que $x^{\alpha(i)} y^{m_i} \in I$ para cada $i = 1, 2, \ldots, s$. Sea $m = \max\{m_i \mid 1 \leq i \leq s\}$. Ahora consideramos los m ideales $J_l = \langle x^\beta \mid x^\beta y^l \in I\rangle$ para $l = 0, 1 \ldots, m-1$. (los ideales J_l se pueden ver como la 'rebanada' de I que contienen exactamente a los monomios y^l). Aplicando otra vez la hipótesis de inducción, tenemos que los ideales J_l están finitamente generados, es decir, para cada l existen $\alpha_l(j) \in \mathbb{Z}_{\geq 0}^{n-1}$ tales que $J_l = \langle x^{\alpha_l(1)}, x^{\alpha_l(2)}, \ldots x^{\alpha_l(s_l)}\rangle$.
Veamos que I está generado por los monomios que generan los ideales
$J, J_0, J_1, \ldots, J_{m-1}$.
Cualquier polinomio de I es divisible por alguno de los elementos que generan estos ideales:
Sea $x^\alpha y^p$ un elemento genérico de I. Si $p < m$ entonces algún elemento generador $x^{\alpha_p(j)} y^p$ del ideal J_p divide a $x^\alpha y^p$ (por el algoritmo de división). Si $p \geq m$, entonces algún generador $x^{\alpha(i)} y^m$ del ideal J divide al monomio $x^\alpha y^p$

escogido por la construcción de J.
Por el lema 2.1.2, los monomios anteriores escogidos generan un ideal que tiene los mismos monomios que I, y por el corolario 2.1.1, concluimos que estos son iguales. Para completar la demostración, necesitamos mostrar que el conjunto finito de generadores puede ser elegido a partir de un conjunto dado de generadores para el ideal. Si volvemos a escribir las variables como $x_1, x_2, \ldots, x_n$, entonces nuestro ideal monomial es $I = \langle x^\alpha \,|\, \alpha \in A\rangle \subseteq \mathbb{F}_q[x_1, x_2, \ldots, x_n]$. Debemos demostrar que I está generado por un número finito de los x^α, donde $\alpha \in A$. Según el párrafo anterior, sabemos que $I = \langle x^{\beta(1)}, x^{\beta(2)}, \ldots, x^{\beta(s)}\rangle$ para algunos monomios $x^{\beta(i)}$ en I. Como $x^{\beta(i)} \in I = \langle x^\alpha : \alpha \in A\rangle$, por el lema 2.1.2 deducimos que $x^{\beta(i)}$ es divisible por $x^{\alpha(i)}$ para algún $\alpha(i) \in A$. A partir de aquí, es fácil concluir que $I = \langle x^{\alpha(1)}, x^{\alpha(2)}, \ldots, x^{\alpha(s)}\rangle$. □

Para comprender mejor cómo funciona la demostración del lema de Dickson, apliquémoslo al ideal $I = \langle x^4y^2, x^3y^4, x^2y^5\rangle$. A partir de la representación de los exponentes, podemos observar que la "proyección"del ideal es $J = \langle x^2\rangle \subseteq k[x]$. Como $x^2y^5 \in I$, se tiene $m = 5$. Luego obtenemos las rebanadas"J_i, $0 \leq i \leq 4 = m-1$, generadas por monomios que contienen y^i:
$J_0 = J_1 = \{0\}$, $J_2 = J_3 = \langle x^4\rangle$, $J_4 = \langle x^3\rangle$.

Estas rebanadas"se pueden identificar utilizando la representación de los exponentes. Luego, la demostración del lema de Dickson nos brinda $I = \langle x^2y^5, x^4y^2, x^4y^3, x^3y^4\rangle$.

El lema de Dickson resuelve el problema de la descripción de ideales para ideales monomiales, ya que establece que dicho ideal tiene una base finita. Esto a su vez, nos permite resolver el problema de pertenencia en ideales monomiales. Es decir, si $I = \langle x^{\alpha(1)}, x^{\alpha(2)} \ldots, x^{\alpha(s)}\rangle$, entonces se puede ver que si un polinomio f pertenece a I si y solo si el resto de la división de f por $x^{\alpha(1)}, x^{\alpha(2)}, \ldots, x^{\alpha(s)}$ es cero.

Podemos ver también que todo ideal tiene una base de Gröbner finita.

Proposición 2.1.2. *Supongamos que $I \subseteq k[x_1, \ldots, x_n]$ es un ideal diferente de $\{0\}$. Entonces:*

1. *$LT(I)$ es un ideal monomial.*
2. *Existen $g_1, g_2, \ldots, g_t \in I$ tales que $LT(I) = \langle LT(g_1), LT(g_2), \ldots, LT(g_t)\rangle$.*

Demostración. Veamos (1): Los monomios líderes $LM(g)$ de los elementos $g \in I \setminus \{0\}$ generan el ideal monomial $\langle LM(g) \mid g \in I \setminus \{0\}\rangle$. Como $LM(g)$ y $LT(g)$ difieren por una constante no nula, general el mismo ideal, y se tiene que $\langle LT(g) \mid g \in I \setminus \{0\}\rangle = \langle LT(I)\rangle$. Por tanto, el ideal $\langle LT(I)\rangle$ es monomial.

Para ver (2), aplicaremos el lema de Dickson a (1): existen $g_1, g_2, \ldots, g_t \in I$ tales que $\langle LT(I)\rangle = \langle LM(g_1), LM(g_2), \ldots, LM(g_t)\rangle$. Como $LM(g_i) = \lambda LT(g_i)$ con $\lambda \neq 0$ un escalar de $\mathbb{F}$, se sigue como antes que $\langle LT(I)\rangle = \langle LT(g_1), LT(g_2), \ldots, LT(g_t)\rangle$. □

Apoyándonos en esta proposición demostraremos de manera constructiva que todo ideal tiene un conjunto finito de generadores. Este resultado se conoce como el teorema de la base de Hilbert:

Teorema 2.1.3 (Teorema de la base de Hilbert)**.** *Todo ideal $I \subseteq k[x_1, \ldots, x_n]$ tiene un conjunto finito de generadores, es decir, existen $g_1, g_2, \ldots, g_t \in I$ tales que $I = \langle g_1, g_2, \ldots, g_t\rangle$.*

Demostración. Si $I = \{0\}$, se cumple trivialmente. Supongamos que $I \neq \{0\}$. Fijado un orden monomial $>$, el ideal I tiene un conjunto de términos líderes $LT(I)$. Según la proposición anterior, existen $g_1, g_2, \ldots, g_t \in I$ tales que $\langle LT(I)\rangle = \langle LT(g_1), LT(g_2), \ldots, LT(g_t)\rangle$. Queremos ver que $I = \langle g_1, g_2, \ldots, g_t\rangle$.
Es claro que $\langle g_1, g_2, \ldots, g_t\rangle \subseteq I$, ya que cada g_i está en I.
Para ver el inverso, tenemos que demostrar que $I \subseteq \langle g_1, g_2, \ldots, g_t\rangle$. Supongamos que $f \in I$ es un polinomio cualquiera. Si aplicamos el algoritmo de división de 2.1.1 para dividir f por $(g_1, g_2, \ldots, g_t)$, obtenemos una expresión de la forma

$$f = q_1 \cdot g_1 + q_2 \cdot g_2 + \cdots + q_t \cdot g_t + r$$

donde ningún término de r es divisible por ninguno de los monomios $LT(g_i)$. Supongamos que $r \neq 0$. Despejando r de la expresión anterior:

$$r = f - q_1 \cdot g_1 + q_2 \cdot g_2 + \cdots + q_t \cdot g_t,$$

y está en I porque es una resta en I. Si $r \neq 0$, entonces $LT(r) \in \langle LT(I)\rangle = \langle LT(g_1), LT(g_2), \ldots, LT(g_t)\rangle$, y del lema 2.1.2 se sigue que $LT(r)$ debe ser divisible por algún $LT(g_i)$. Esto va en contra del algoritmo de división 2.1.1, por lo que r necesariamente debe ser 0. Así se tiene que

$$f = q_1 \cdot g_1 + q_2 \cdot g_2 + \cdots + q_t \cdot g_t \in \langle g_1, g_2, \ldots, g_t\rangle,$$

por lo que $I \subseteq \langle g_1, g_2, \ldots, g_t\rangle$, que era la contención que faltaba.

□

Del teorema de la base de Hilbert se puede deducir fácilmente que todo ideal tiene una base de Gröbner:

Corolario 2.1.3. *Fijado un orden monomial $>$ en $\mathbb{F}[x_1, x_2, \ldots, x_n]$, todo ideal I de $\mathbb{F}[x_1, x_2, \ldots, x_n]$ tiene una base de Gröbner. Además, cualquier base de Gröbner de I es una base de I.*

Demostración. Dado un ideal no nulo, el conjunto $G = \{g_1, g_2, \ldots, g_t\}$ construido en la prueba del teorema anterior es una base de Gröbner por definición. Para ver que es una base de I, hemos visto en el teorema de la base de Hilbert que si $\langle LT(I)\rangle = \langle LT(g_1), LT(g_2), \ldots, LT(g_t)\rangle$, entonces $\langle I\rangle = \langle g_1, g_2, \ldots, g_t\rangle$, concluyendo así la prueba. □

Por lo explicado anteriormente, tiene sentido la siguiente definición:

Definición 2.1.9. Una base generadora $\{x^{\alpha_1}, x^{\alpha_2}, \ldots, x^{\alpha_s}\}$ de un ideal I (es decir, $I = \langle x^{\alpha_1}, x^{\alpha_2}, \ldots, x^{\alpha_s}\rangle$) es *minimal* si x^{α_i} no divide a x^{α_j} para $i \neq j$, donde $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in}) \in \mathbb{Z}^n_{>0}$.

Dicho de otra manera, una base es minimal si sus elementos no se dividen entre sí utilizando el algoritmo de división 2.1.1.

Proposición 2.1.3. *Una base minimal de un ideal es única.*

Demostración. Consideremos una $\mathbb{B} = \{x^{\alpha_1}, x^{\alpha_2}, \ldots, x^{\alpha_s}\}$ base minimal del ideal I, esto es, $I = \langle x^{\alpha_1}, x^{\alpha_2}, \ldots, x^{\alpha_s}\rangle$. Supongamos que existe otra base $\mathbb{B}' = \{x^{\beta_1}, x^{\beta_2}, \ldots, x^{\beta_t}\}$ de I. Entonces existe i tal que x^{α_i} divide a x^{β_1}. También, existe j tal que x^{β_j} divide a x^{α_i}, luego x^{β_j} divide a x^{β_1}, y por ser una base minimal, son el mismo elemento. Así obtenemos $x^{\beta_1} = x^{\alpha_i}$. Repitiendo el proceso y razonando por recurrencia, terminamos viendo que la base $\mathbb{B}'$ está contenida en la base $\mathbb{B}$ que teníamos. Haciendo el mismo proceso pero al revés, obtenemos la otra contención, y así, demostrando la unicidad de la base. □

Junto con la proposición anterior, nos aseguraremos de que todo ideal monomial tiene una base minimal única.

Proposición 2.1.4. *Todo ideal monomial tiene una base minimal.*

Demostración. Como hemos visto antes, es finitamente generado. Como el ideal I es finitamente generado, entonces tiene una base finita. Consideremos una base finita cualquiera $\langle x^{\alpha_1}, x^{\alpha_2}, \ldots, x^{\alpha_s}\rangle$ de I, esto es, $I = \langle x^{\alpha_1}, x^{\alpha_2}, \ldots, x^{\alpha_s}\rangle$. Para ver que es una base minimal, debemos ver que x^{α^i} no divide a x^{α_j} si $i \neq j$. Se realiza el proceso de división y si se da el caso x^{α_i} divide a x^{α_j} si $i \neq j$, por el lema que caracteriza los elementos de los ideales monomiales, podemos reemplazar x^{α_j} por x^{α_i} .
También sabemos por la proposición 2.1.3 que esta base es única. □

De esta última proposición podemos deducir que fijado un orden monomial, un ideal monomial tiene una base de Gröbner que es única. Esta propiedad de unicidad es importante porque garantiza que hay una forma canónica de representar el ideal monomial, y podemos usar esta base de Gröbner única para realizar diversos cálculos y resolver problemas relacionados con el ideal. Además, esta unicidad simplifica el estudio y la manipulación de los ideales monomiales en diversas aplicaciones matemáticas y computacionales. También, teniendo en cuenta lo anterior, tenemos la unicidad del resto de la división por una base de Gröbner, que es lo que realmente necesitaremos. Veremos la aplicación de esto cuando se demuestre la cota de Schwartz-Zippel.

Observación. Existe un algoritmo para calcular bases de Gröbner de un ideal, denominado algoritmo de Bucherberg. Para este trabajo solamente usaremos los resultados de existencia y las propiedades de las bases de Gröbner, no será necesario calcular una de manera explícita.

2.2. Cuerpos finitos

En esta sección se introducirán conceptos sobre los cuerpos finitos, ya que el lema de Shwartz-Zippel lo escribiremos sobre cuerpos finitos y también los necesitaremos en el siguiente capítulo, que trata sobre Network Coding.
Por definición un cuerpo es finito si tiene un número finito de elementos.
Hablaremos de las propiedades fundamentales de los cuerpos finitos y describiremos algunos métodos sobre la construcción de cuerpos finitos. Algunos de estos conceptos se han visto en asignaturas obligatorias del grado como Estructuras Algebraicas y Ecuaciones Algebraicas, y se han revisado en algunas asignaturas optativas como Criptografía, Álgebra Conmutativa y Computacional o Códigos Correctores.
Para este capítulo se ha utilizado como referencia [5].

2.2.1. Caracterización de los cuerpos finitos

Teorema 2.2.1. *El cuerpo $\mathbb{F}$ de p^r elementos existe y es único para cada primo p y cada entero positivo r.*

El cuerpo finito de q elementos se denota por $\mathbb{F}_q$. En el resto de esta sección se fijará el cardinal de un cuerpo por $q = p^r$ donde p es un número primo y r es un número natural.

Lema 2.2.1. *Si $\mathbb{F}_q$ es un cuerpo finito de q elementos entonces cada $a \in \mathbb{F}_q$ cumple $a^q = a$.*

Demostración. La tesis del lema es trivial para $a = 0$. Por el otro lado, los elementos no nulos de $\mathbb{F}_q$ forman un grupo multiplicativo de orden $q-1$. Por tanto, $a^{q-1} = 1$ para cada $a \in \mathbb{F}_q \setminus \{0\}$, y multiplicando por a obtenemos el resultado deseado. □

A partir del lema anterior se puede deducir el siguiente resultado:

Lema 2.2.2. *El grupo multiplicativo $\mathbb{F}_q^*$ es un grupo cíclico.*

La prueba de 2.2.2 se puede ver en [5] (págs. 46-47). Pueden haber varios generadores del grupo $(\mathbb{F}^*, \cdot)$, los cuales se denominan *elementos primitivos* del cuerpo $\mathbb{F}$. Esto se detalla mejor en la siguiente definición:

Definición 2.2.1. Un elemento $a \in \mathbb{F}_q$ es **primitivo** si
$\mathbb{F}_q^* = \langle a \rangle = \{a, a^2, \ldots, a^{q-2}, a^{q-1}\}$.

Observación. Para los elementos primitivos de un cuerpo finito, también se utiliza la notación $\mathbb{F}_q^* = \langle a \rangle$.

Notemos que $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0_{\mathbb{F}_q}\}$ (al tratarse de un cuerpo, todo elemento no nulo tiene inverso para el producto). También se puede construir un cuerpo finito sobre $\mathbb{F}_p$ de $q = p^r$ elementos mediante un polinomio irreducible de grado r sobre $\mathbb{F}_p$, aunque no es trivial afirmar que para cada cuerpo de p elementos existe un polinomio irreducible de grado r (pero es cierto). Se necesita un resultado auxiliar para poder demostrarlo, no obstante, se puede demostrar de otra manera: que para cada p primo y para cada natural $r > 0$ se tiene un cuerpo finito de p^r elementos.

Proposición 2.2.1. *Las raíces del polinomio $x^q - x$ son todos los elementos del cuerpo $\mathbb{F}_q$.*

Demostración. El polinomio $x^q - x$ de grado q tiene a lo sumo q raíces en $\mathbb{F}_q$. Por el lema 2.2.1, sabemos que estas q raíces son todos los elementos de $\mathbb{F}_q$. Por lo tanto, el polinomio dado es el indicado. □

Ahora veamos otra manera de representar los elementos de un cuerpo.

Proposición 2.2.2. *Si el cardinal de $\mathbb{F}_q$ es $q = p^r$, entonces el cuerpo $\mathbb{F}_q$ es isomorfo a $\mathbb{F}_p[x]/\langle f \rangle$, donde $f \in \mathbb{F}_p[x]$ es un polinomio irreducible de grado r.*

Para ver que esta representación es factible para cada cuerpo de tamaño q, es necesaria la siguiente proposición, cuya demostración está en [5](pág. 47).

Proposición 2.2.3. *Para cada cuerpo finito $\mathbb{F}_p$ y para cada entero positivo r, existe un polinomio irreducible en $\mathbb{F}_p[x]$ de grado r.*

Observación. Es común utilizar la notación $\mathbb{F}_p[x]_{<r}$ para denotar a los polinomios de grado estrictamente menor que r con coeficientes en $\mathbb{F}_p$.

A partir de los dos anteriores resultados, se puede deducir que el cuerpo $\mathbb{F}_q$ tiene estructura de espacio vectorial r-dimensional sobre el cuerpo $\mathbb{F}_p$, identificando un polinomio de $\mathbb{F}_p[x]_{<r}$ con una $r-$tupla de elementos de $\mathbb{F}_p$.

Observación. Es bien conocido de las asignaturas del grado que el cuerpo finito $\mathbb{F}_q$ es una extensión de grado r del cuerpo $\mathbb{F}_p$, y se denota de la forma: $[\mathbb{F}_q : \mathbb{F}_p] = r$.

2.2.2. Representaciones de un cuerpo finito

Para un cuerpo finito se tienen tres diferentes representaciones posibles:

- *Representación cíclica.* $\mathbb{F}_q = \{0, 1, \alpha, \ldots, \alpha^{q-2}\}$, donde α es un elemento primitivo. De esta manera para cada elemento $a \in \mathbb{F}_q$ existe $i \in \{0, 1, 2, \ldots, q-2\}$ tal que $a = \alpha^i$.
- *Representación polinomial.* Visto en la proposición 2.2.2 Así un elemento de $\mathbb{F}_q$ puede representarse como un polinomio de $\mathbb{F}_p[x]$ de grado menor o igual que r (siendo $q = p^r$) módulo f.
- *Representación como $\mathbb{F}_p$-espacio vectorial de dimensión r.* En esta situación los elementos son vectores con coordenadas en $\mathbb{F}_p$ de tamaño r.

Dependiendo de nuestros intereses, utilizaremos una representación u otra. Esta última representación será la que más interés tendrá en el Network Coding, ya que se usarán aplicaciones lineales para realizar la codificación en los nodos.

2.3. Cota de footprint y lema de Schwartz-Zippel

La idea principal en esta sección es acotar superiormente el número de ceros de un polinomio de varias variables. Esto es relevante para demostrar que existe una solución lineal al problema de Network Coding en un cuerpo de tamaño lo suficientemente grande. Antes de demostrar el teorema, demostraremos una proposición que utilizaremos para resultados posteriores.

Proposición 2.3.1. *Sea $I \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ un ideal y $\Delta(I) = \{x^\alpha \mid x^\alpha \notin LT(I)\}$. Entonces se tiene un isomorfismo (de espacios vectoriales) entre $\mathbb{F}[x_1, x_2, \ldots, x_n]/I$ y $\Delta(I)$.*

Demostración. Veamos que la aplicación $\phi : \mathbb{F}[x_1, x_2, \ldots, x_n]/I \to \Delta(I)$ definida como $\phi([f]) := [f]^G$ es un isomorfismo, donde G es una base de Gröbner de I.
Si $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, entonces denotando $r = [f]^G$, tenemos que $f = q + r$, donde $q \in I$. De ello, deducimos que $f - r \in I$, es decir, que f es congruente con r módulo I. Por el algoritmo de división, podemos afirmar que r es una combinación $\mathbb{F}$-lineal de los términos que no pertenecen a $\langle LT(I)\rangle$ y que por cada polinomio f existe un único resto.
Veamos que ϕ es un homomorfismo:

$$\phi([f] + [g]) = [f + g]^G = [f]^G + [g]^G = \sum (c_\alpha + d_\alpha)x^\alpha =$$

$$= \sum (c_\alpha)x^\alpha + \sum (d_\alpha)x^\alpha = \phi([f]) + \phi([g]).$$

También se cumple que para cualquier escalar $\lambda \in \mathbb{F}$: $\phi([\lambda f]) = [\lambda f]^G = \lambda[f]^G = \lambda\phi([f])$.
Por tanto, ϕ es lineal y es biyectiva, entonces es un isomorfismo por lo anterior. □

La proposición anterior se puede visualizar en el caso $n = 2$.

Teorema 2.3.1 (Cota de footprint)**.** *Sea I un ideal monomial de $\mathbb{F}_q[\boldsymbol{x}]$. Se cumple que*

$$|V(I)| \leq |\Delta(I)|,$$

donde $\Delta(I) = \{x^\alpha \mid x^\alpha \notin LT(I)\}$

Demostración. Definimos la aplicación $\phi : \mathbb{F}_q[\mathbf{x}] \to \mathbb{F}_q^{q^n}$ como $\phi(G) := (G(\mathbf{a}))_{\mathbf{a}\in\mathbb{F}_q^n}$.
Se trata de una aplicación de evaluación lineal donde para cada polinomio del anillo de polinomios de varias variables sobre $\mathbb{F}_q$, se hace una evaluación en los q^n elementos en $\mathbb{F}_q^n$. Que la aplicación ϕ está bien definida resulta claro, y es lineal porque

$$\phi(G+\lambda H) = (G+\lambda H)(\mathbf{a})_{\mathbf{a}\in\mathbb{F}_q^n} = (G(\mathbf{a}))_{\mathbf{a}\in\mathbb{F}_q^n} + (\lambda H(\mathbf{a}))_{\mathbf{a}\in\mathbb{F}_q^n} = \phi(G) + \lambda\phi(H).$$

En segundo lugar, consideraremos un ideal $I \subset \mathbb{F}_q[\mathbf{x}]$ y su conjunto de ceros $V(I) = \{\mathbf{a} \in \mathbb{F}_q^n \mid F(\mathbf{a}) = 0, \qquad \forall F \in I\}$.
Definimos una aplicación $\overline{\phi} : \mathbb{F}_q[\mathbf{x}] \to \mathbb{F}_q^{|V(I)|}$, donde $\phi : G \mapsto (G(\mathbf{a}))_{\mathbf{a}\in V(I)}$.
Si $f \in I$, por definición de $V(I)$, $f(\mathbf{a}) = 0, \forall \mathbf{a} \in V(I)$. Lo que significa que

$I \subset Ker(\overline{\phi})$, por lo que podría considerarse el cociente por el ideal I.
Definimos $\psi : \mathbb{F}_q[\mathbf{x}]/I \to \mathbb{F}_q^{|V(I)|}$ como

$$\psi([G]) := (G(\mathbf{a}))_{\mathbf{a}\in V(I)},$$

donde se considera la clase del polinomio $G \in \mathbb{F}_q[\mathbf{x}]$ en el cociente por el ideal I ($[G]$) donde $[G] = G + I$. Esta función está bien definida (si $[F] = [G]$ entones $F(\mathbf{a}) = G(\mathbf{a})$, $\forall \mathbf{a} \in V(I)$).
La aplicación ϕ es sobreyectiva: si tenemos q^n puntos, existe un polinomio en $\mathbb{F}_q[\mathbf{x}]$ de grado a lo sumo $q^n - 1$ que interpola los puntos $(G(\mathbf{a}))_{\mathbf{a}\in\mathbb{F}_q^n}$(por el teorema de interpolación de Lagrange en varias variables).
Así, $|V(I)| \leq |\mathbb{F}_q^n|$ porque $V(I) \subset \mathbb{F}_q^n$.
Como ϕ es sobreyectiva, entonces $\overline{\phi}$ también lo es, ya que el conjunto de llegada de ϕ, que es $V(I)$, está contenido en el conjunto de llegada de $\overline{\phi}$, que es $\mathbb{F}_q^n$. También, al ser ψ la restricción del cociente por I de $\overline{\phi}$, se tiene que ψ es sobreyectiva.
De esta sobreyectividad deducimos las desigualdades de cardinales

$$|V(I)| = \dim(\mathbb{F}_q^{|V(I)|}) \leq \dim(\mathbb{F}_q[\mathbf{x}]/I).$$

Recordamos la proposición 2.3.1. En ella, demostramos que la aplicación era un isomorfismo entre los espacios $\mathbb{F}_q[\mathbf{x}]/I$ y $\Delta(I)$, lo cual implica una igualdad de cardinales entre el conjunto de salida y el de llegada:

$$\dim(\mathbb{F}_q[\mathbf{x}]/I) = |\Delta(I)|.$$

Esta desigualdad se conoce como la "footprint bound"(en castellano, "cota de la huella"). Por lo tanto, llegamos a que $|V(I)| \leq |\Delta(I)|$. □

El conjunto $\Delta(I)$ puede verse gráficamente para $n = 2$. Supongamos que, por ejemplo, $q = 5$ y que el ideal I está generado por los monomios x^4, y^2, x^3y^2, x^3y y x^4y. Entonces $\Delta(I)$ correspondería a la zona sombreada de la figura 2.2. Esto está relacionado con el lema de Dickson.
Es bastante ilustrativo obtenerlo: la zona gris en principio es infinita, pero siempre podemos añadir los polinomios $x^5 - x$, $y^5 - y$ al ideal I y $V(I)$ no cambiará, ya que como se vio en la parte de cuerpos finitos, los elementos de $\mathbb{F}_5$ son raíces de ambos polinomios. De esta forma, se añaden los puntos $(0, 5)$ y $(5, 0)$ al ideal, obteniendo una zona gris finita.
Sobre el resto de puntos que están en el ideal, se quitaría la parte de arriba a la derecha, quedando el resto de partes intactas.
Con esta forma de verlo, podemos ver que los puntos $(3, 2)$ y $(4, 1)$ son superfluos, y que si prescindimos de los monomios asociados a esos puntos, el

ideal generado sería el mismo, ya que seguiría habiendo la misma zona gris en la figura.
De esta manera, los puntos que quedan dentro de la zona sombreada se corresponden con los monomios que están en el ideal.
Esto muy útil para obtener su cardinal.

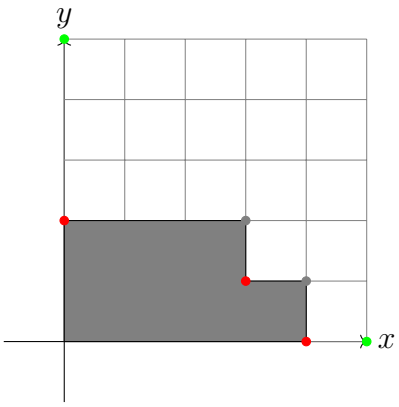


Figura 2.2: Conjunto $\Delta(I)$ para $n = 2$

La cota de Schwartz-Zippel es una consecuencia del teorema 2.3.1.

Corolario 2.3.1 (Cota de Schwartz-Zippel)**.** *Sea $q \in \mathbb{Z}_+$ tal que $q = p^r$ con p un número primo y $r > 0$. Sea $P \in \mathbb{F}_q[\boldsymbol{x}]$ un polinomio de grado d. Entonces,*

$$|V(P)| \leq q^{n-1}d,$$

donde $V(P)$ es el conjunto de ceros del polinomio P sobre el cuerpo $\mathbb{F}_q$.

Demostración. Consideremos ahora el caso $I = \langle P \rangle$, donde $P \in \mathbb{F}_q[\mathbf{x}]$ es el polinomio del enunciado.
Por la teoría de cuerpos finitos, sabemos que las raíces del polinomio $x^q - x$ son todos los elementos del cuerpo $\mathbb{F}_q$ donde $q = p^r$. Entonces, si estamos calculando los ceros de un polinomio P en $\mathbb{F}_q$, al estar estas raíces en el cuerpo, también serán raíces de $x^q - x$. Teniendo en cuenta lo anterior, se verifica la igualdad $|V(P)| = |V(P, x_1^q - x_1, x_2^q - x_2, \ldots, x_n^q - x_n)|$.
Usando $|V(I)| \leq |\Delta(I)|$, llegamos a que

$$|V(I)| \leq |\Delta(P, x_1^q - x_1, x_2^q - x_2, \ldots, x_n^q - x_n)|.$$

Si vemos que este último cardinal es menor o igual que $deg(P) \cdot q^{n-1}$, obtendremos la tesis que queríamos demostrar y se verificaría la cota $|V(P)| \leq q^{n-1}d$.

Demostrar esto es equivalente a demostrar que

$$q^n - \prod_{i=1}^{n}(q - b_i) \leq q^{n-1} \cdot deg(P) = q^{n-1} \cdot \sum_{i=1}^{n} b_i,$$

donde b_i son los grados máximos correspondientes a las variables x_i del polinomio P del enunciado. Se puede ver por inducción sobre $n \in \mathbb{N}$. Para $n = 1$ se cumple trivialmente, y para $n = 2$ se tiene que $q^2 - (q - b_1) \cdot (q - b_2) = q \cdot (b_1 + b_2) - b_1 \cdot b_2 \leq q \cdot (b_1 + b_2)$, ya que $b_i \geq 0$ para cada $i \in \{1, 2, \ldots, n\}$. Ahora lo suponemos cierto para $n - 1$ y lo probamos para n. Se cumple entonces que

$$q^{n-1} - \prod_{i=1}^{n-1}(q - b_i) \leq q^{n-2} \cdot \sum_{i=1}^{n-1} b_i.$$

Ahora multiplicando por $q - b_n$ a cada lado de la ecuación y desarrollando los productos, obtenemos (aquí asumimos que $0 < b_n < q$, que si no ocurre, se puede obtener reordenando los términos):

$$q^n - \prod_{i=1}^{n}(q - b_i) - q^{n-1} \cdot b_n \leq q^{n-1} \cdot \sum_{i=1}^{n-1} b_i - q^{n-2} \cdot b_n \cdot \sum_{i=1}^{n-1} b_i. \tag{2.1}$$

Como $b_i \geq 0$ para cada $i \in \{1, 2, \ldots, n\}$, entonces $q^{n-2} \cdot b_n \cdot \sum_{i=1}^{n-1} b_i \geq 0$, por lo que

$$q^{n-1} \cdot \sum_{i=1}^{n-1} b_i - q^{n-2} \cdot b_n \cdot \sum_{i=1}^{n-1} b_i \leq q^{n-1} \cdot \sum_{i=1}^{n-1} b_i.$$

Sumando el término $q^{n-1} \cdot b_n$ a cada lado de la desigualdad 2.1 obtenemos finalmente

$$q^n - \prod_{i=1}^{n}(q - b_i) \leq q^{n-1} \cdot \sum_{i=1}^{n} b_i,$$

concluyendo así la demostración. □

En el siguiente capítulo se verá cómo este resultado es de ayuda para demostrar que existe una solución al problema lineal del Network Coding con una fuente y varios sumideros.

Capítulo 3

Network coding

Como se mencionó al final del primer capítulo, en este capítulo se hablará de manera más detallada sobre el problema de una red de flujo con un mensaje (en $\mathbb{F}_q^n$, lo que son n mensajes en $\mathbb{F}_q$) y múltiples llegadas con una fuente.
Mientras que en el capítulo 1 tratábamos la comunicación en una red con una salida y una llegada, ahora tratamos el caso multicast, es decir, de comunicación en una red de flujo con una salida y varias llegadas, donde todas las llegadas deben obtener simultáneamente todos los mensajes enviados por la salida.
Esta parte puede ser formulada de varias formas en diferentes niveles de generalidad. Al igual que en el capítulo 1, se fijará una red de flujo $(G, s, \mathcal{T}, c)$ con $G = (V, E)$ un grafo dirigido, pero esta vez denotaremos por $T = \{t_1, t_2, \ldots, t_n\}$ como el conjunto de las n llegadas. También fijamos un cuerpo finito genérico $\mathbb{F}$.
Una referencia muy útil en este campo es [6]. También ha sido de utilidad el artículo [4].

3.1. Redes sin ciclos

Trabajaremos con redes sin ciclos (en inglés, acyclic networks). Veamos a qué nos referimos con esto.

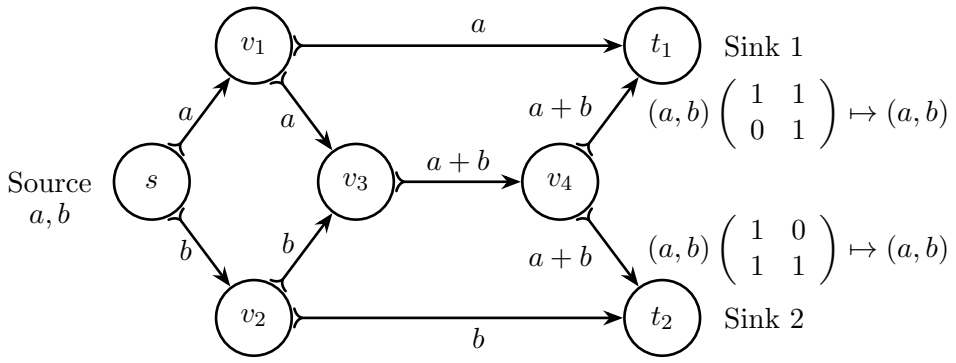
Definición 3.1.1 (ciclo)**.** Un camino $p = (v_0, v_1, \ldots, v_k)$ en G forma un **ciclo** si $v_0 = v_k$ y $k \geq 1$.

También pueden haber ciclos en un mismo vértice.

Definición 3.1.2 (bucle)**.** Un camino $p = (v_0, v_1, \ldots, v_k)$ en G es un **bucle** si es un ciclo con $k = 1$.

Definición 3.1.3. Decimos que $(G, s, \mathcal{T}, c)$ es una red de flujo **acíclica** si G es un grafo sin ciclos, es decir, que E no contiene ciclos.

Partíamos del problema del grafo mariposa expuesto en la sección 1.3. Ya se explicó en detalle en qué consistía el problema y la solución. Este ejemplo



se puede generalizar a múltiples mensajes con múltiples llegadas desde una salida (este tema es una de las partes fundamentales de la codificación en redes).
En este capítulo se tratará el caso de varios mensajes en $\mathbb{F}_q$, una fuente y múltiples llegadas.
También existe un retraso o delay en el envío de la información. El procesamiento en los nodos es la mayor causa de delay, por lo que se suele buscar un grafo simple y rápido como circuito.
No obstante, en las redes acíclicas puede presuponerse que no existe delay. ¿Por qué podemos asumirlo? Porque al no haber ciclos, cada mensaje se puede procesar de manera independiente (tanto la codificación como el envío del mensaje). El proceso de cada mensaje es independiente de los mensajes secuenciales, por lo que el problema del Network Coding es independiente de la propagación del delay, lo que incluye el proceso del delay en los nodos.
En las redes con ciclos no ocurre lo mismo, porque la propagación y la codificación de mensajes secuenciales (es decir, que se encadenan en los envíos) podrían entrelazarse entre ellos, por lo que conviene considerar el delay.
En resumen, en las redes acíclicas solo tenemos que preocuparnos del proceso y no del retraso.

Observación. En el Network Coding es común denominar a las *aristas* del grafo como canales de la red, aunque en general, se refiere a un enlace de comunicación o camino entre dos nodos de la red. En este escrito se considerará el primer caso excepto que se diga lo contrario.

Supongamos que la fuente s tiene un conjunto de canales imaginarios, de los cuales recibe el mensaje que se mandará. Ese conjunto se denotará por $In(s)$ y su cardinal por w.
La **unidad de datos** o **símbolo** depende del cuerpo base sobre el que se trabaje, por ejemplo, si el cuerpo es $\mathbb{F}_q = \mathbb{F}_2$ la unidad de datos en la red es el bit.
Un **mensaje** m está compuesto por w unidades de datos, luego $m \in \mathbb{F}^w$ como vector.
La **fuente** s recibe un mensaje m de los w canales imaginarios (un símbolo por canal), y lo manda transmitiendo un símbolo $\bar{f}_e(m) \in \mathbb{F}$ a cada canal e en la red.

Observación. La unidad de datos en la red podría ser cualquiera, por ejemplo, si quisiéramos usar bytes en lugar de bits tenemos que considerar $\mathbb{F}_{256}$, ya que $256 = 2^8$ (aquí hemos usado que 1 byte = 8 bits).

3.2. Codificación en redes con múltiples paquetes

En esta sección se verá la existencia de la solución al problema del Network Coding y se detallará la construcción de la misma. También se explicará el problema extendido del Network Coding, y por qué funciona.
A partir de ahora fijaremos un mensaje enviado, que denotaremos por $\mathbf{m} = (m_1, m_2, \ldots, m_n) \in \mathbb{F}_q^n$.
Empecemos a definir el entorno en el que demostraremos la existencia de esta solución de una manera constructiva.

Definición 3.2.1 (Descripción local de una red código en una red acíclica)**.** Sea $\mathbb{F}$ un cuerpo y w un entero positivo.
Un código de red $w-$dimensional y $\mathbb{F}-$evaluado sobre una red de comunicación acíclica consiste en una **aplicación de codificación local**

$$\bar{k_e} : \mathbb{F}^{|in(u)|} \longrightarrow \mathbb{F}$$

para cada nodo u en la red y cada canal $e \in out(u)$.

La anterior definición nos da el proceso de codificación en cada nodo del grafo, pero no nos brinda los valores concretos de $\bar{f}_e(m)$. Por ello, se da una definición equivalente pero más generalizada, que describe tanto el código de la red como los mecanismos de codificación locales, obteniendo los valores $\bar{f}_e(m)$ de manera recursiva.

Definición 3.2.2 (Descripción global de una red código en una red acíclica)**.** Sea $\mathbb{F}$ un cuerpo y w un entero positivo.
Un código de red $w-$dimensional y $\mathbb{F}-$evaluado sobre una red de comunicación acíclica consiste en una aplicación de codificación local $\bar{k}_e : \mathbb{F}^{|in(u)|} \longrightarrow \mathbb{F}$ y una **aplicación de codificación global** $\bar{f}_e : \mathbb{F}^w \longrightarrow \mathbb{F}$ para cada canal e en la red tal que:

- Para cada nodo u y cada canal $e \in out(u)$, $\bar{f}_e(m)$ está unívocamente determinado por $(\bar{f}_d(m), d \in in(u))$ y $\bar{k}_e$ es la aplicación:

$$(\bar{f}_d(m), d \in in(u)) \longmapsto \bar{f}_e(m).$$

- Para los w canales imaginarios e, las aplicaciones $\bar{f}_e$ son las proyecciones del espacio $\mathbb{F}^w$ a las w diferentes coordenadas, respectivamente.

La definición dada anteriormente es suficiente para obtener los $f_e(m)$ siempre y cuando conozcamos los valores iniciales, es decir, los que se envían desde la fuente s.
Estas definiciones son muy abstractas pero no es mucho más complejo que dar a los canales un valor determinado. La aplicación f_e asocia a cada arista un valor determinado, y k_e nos da las reglas de cómo los valores de llegada a un nodo generan el valor que sale por la arista e (es decir, cómo los valores de llegada a un nodo se codifican para dar lugar al valor de salida). Veámoslo con un ejemplo aplicado a la red mariposa:

Ejemplo 3.2.1. Sea $m = (a, b)$ el mensaje que se envía. En la red mariposa, las asignaciones locales de codificación son:

$$\tilde{k}_{sv_1}(a, b) = a$$

$$\tilde{k}_{sv_2}(a, b) = b$$

$$\tilde{k}_{v_1v_3}(a) = \tilde{k}_{v_1t_1}(a) = a$$

$$\tilde{k}_{v_2v_3}(b) = \tilde{k}_{v_2t_2}(b) = b$$

$$\tilde{k}_{v_3v_4}(a, b) = \tilde{k}_{v_4t_1}(a, b) = \tilde{k}_{v_4t_2}(a, b) = a + b$$

Las asignaciones globales de codificación serán :

$$\tilde{f}_e(m) = a \quad \text{para} \quad e = os, sv_1, v_1v_3 \text{ y } v_1t_1$$

$$\tilde{f}_e(m) = b \quad \text{para} \quad e = o's, sv_2, v_2v_3 \text{ y } v_2t_2$$

$$\tilde{f}_e(m) = a + b \quad \text{para} \quad e = v_3v_4, v_4t_1 \text{ y } v_4t_2$$

donde os y $o's$ denotan los dos canales imaginarios que llegan a la fuente s.

Es de interés que las aplicaciones en cada nodo sean lineales, ya que este tipo de funciones suelen hacer más rápido el circuito, sobre todo en el proceso en los nodos. También sabemos que las aplicaciones lineales se pueden representar mediante matrices, por lo que interesaría caracterizarlas mediante estas.

Definición 3.2.3. Un par de canales (d, e) son **adyacentes** cuando existe un nodo u con $d \in in(u)$ y $e \in out(u)$.

La anterior definición nos es de utilidad en la siguiente.

Definición 3.2.4 (Descripción local de un código de red lineal en una red acíclica)**.** Sea $\mathbb{F}$ un cuerpo y w un entero positivo.
Un código de red lineal $w-$dimensional y $\mathbb{F}-$evaluado sobre una red de comunicación acíclica consiste en un escalar $k_{d,e}$, denominado el **núcleo de codificación local**, para cada par adyacente (d, e). El núcleo de codificación local en el nodo u es la matriz de tamaño $|in(u)| \times |out(u)|$ dada por $K_u = [k_{d,e}]$.

La estructura de la matriz K_u implica de forma implícita un orden entre los canales.

Definición 3.2.5 (Descripción global de un código de red lineal en una red acíclica)**.** Sea $\mathbb{F}$ un cuerpo y w un entero positivo.
Un código de red lineal $w-$dimensional y $\mathbb{F}-$evaluado sobre una red de comunicación acíclica consiste en un escalar $k_{d,e}$ y en un vector $w-$dimensional f_e para cada canal e, denominado **núcleo de codificación global**, tal que

- $f_e = \sum_{d \in in(u)} k_{d,e} f_d$, donde $e \in out(u)$
- Los vectores f_e para los w canales imaginarios $e \in in(s)$ forman una base del espacio vectorial $\mathbb{F}^w$.

Como se comentó sobre las definiciones 3.2.1 y 3.2.2, se puede ver que conociendo $k_{d,e}$ se puede obtener f_e.
La última condición de la definición anterior implica que la matriz formada por los vectores f_e es de rango máximo, siendo así posible encontrar su inversa (por la derecha). Esto es clave para hallar el mensaje.
Sabiendo esto, si en una fuente se genera un mensaje m de la forma de un vector $w - dimensional$, entonces un nodo u recibe los símbolos $m \cdot f_d$, $d \in in(u)$, de donde se calcula el símbolo $m \cdot f_e$ para mandarlo a cada canal $e \in out(u)$ mediante la siguiente combinación lineal:

$$m \cdot f_e = m \cdot \sum_{d \in in(u)} k_{d,e} f_d = \sum_{d \in in(u)} k_{d,e}(m \cdot f_d).$$

Si todas las aplicaciones locales de codificación son lineales entonces la aplicación de codificación global será lineal, y viceversa. Dados los núcleos de codificación locales para todos los canales en una red acíclica, los núcleos de codificación globales se pueden calcular recursivamente en cualquier orden, aunque con ciertas condiciones limitantes. Con el siguiente ejemplo sobre la red mariposa se verá más claro.

Ejemplo 3.2.2. Las matrices de núcleos de codificación local en los nodos son las siguientes:

$$K_s = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad K_{v_1} = K_{v_2} = K_{v_4} = \begin{pmatrix} 1 & 1 \end{pmatrix}, \quad K_{v_3} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Los núcleos de codificación global correspondientes son:

$$f_e = \begin{cases} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \text{para} \quad e = os, sv_1, v_1v_3 \text{ y } v_1t_1, \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \text{para} \quad e = o's, sv_2, v_2v_3 \text{ y } v_2t_2, \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \text{para} \quad e = v_3v_4, v_4t_1 \text{ y } v_4t_2. \end{cases}$$

Los núcleos de codificación local/global describen un código de red bidimensional independientemente de la elección del cuerpo base $\mathbb{F}$.

Se puede encontrar la descripción general para un código de red lineal cualquiera sobre la red mariposa en [6] (pág. 18).

Si la fuente s envía combinaciones de $m_1, m_2, \ldots, m_n$ y cada vértice envía combinaciones de los símbolos que recibe (todas $\mathbb{F}_q$-lineales), entonces cada sumidero t_k, para cada $k \in \{1, 2, \ldots, N\}$ recibe n_k combinaciones lineales de la forma

$$\sum_{i=1}^{n} a_{ij}^{(k)} m_i \quad \forall j = 1, 2, \ldots, n_k, \tag{3.1}$$

donde $n_k = |in(t_k)|$. Esta combinación lineal se puede representar mediante el producto de un vector por una matriz, de la siguiente manera:

$$(m_1, m_2, \ldots, m_n) \cdot \begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1n_k}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)} & \cdots & a_{2n_k}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}^{(k)} & a_{n2}^{(k)} & \cdots & a_{nn_k}^{(k)} \end{pmatrix} = \mathbf{m} \cdot A^{(k)} \tag{3.2}$$

La matriz $A^{(k)}$ se llama **matriz de transferencia** a t_k, y existen N de estas matrices (que es el número de llegadas). Para que t_k recupere el mensaje $\mathbf{m} = (m_1, m_2, \ldots, m_n)$, tiene que resolver el sistema $\mathbf{m} \cdot A^{(k)} = \mathbf{r}_k$, donde $\mathbf{r}_k$ son los símbolos recibidos en t_k. Si tomamos por cierto que la matriz $A^{(k)}$ es de rango máximo n para cada k (necesitaremos que $n_k \geq n$), podremos considerar la inversa por la derecha de la matriz $A^{(k)}$ (esto es, una matriz $B^{(k)} \in \mathbb{F}_q^{n_k \times n}$ tal que $A^{(k)}B^{(k)} = I_n$, donde I_n es la matriz identidad de tamaño n, para cada $k = 1, 2, \ldots N$). Así tendríamos que el sumidero t_k puede obtener el mensaje como sigue:

$$\mathbf{m} = \mathbf{r}_k \cdot B^{(k)}. \tag{3.3}$$

Una forma de simplificarlo es considerar solo problemas donde $n_k = n$ para cada $k = 1, 2, \ldots, N$, es decir, asumir que la matriz $A^{(k)}$ sea cuadrada. También debe ser regular para poder hallar su matriz inversa $B^{(k)}$ (esto es, que $\det(A^{(k)}) \neq 0$).

Ejemplo 3.2.3. En la red mariposa, las matrices de transferencia para cada llegada son: $A^{(1)} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ asociada a t_1, y $A^{(2)} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ asociada a t_2.

3.3. Existencia de un código de red lineal

Demostraremos que existe solución al problema del Network Coding de una forma existencial, y después daremos algoritmos para obtener el código. Consideramos una red de flujo acíclica sin delays, tal y como hemos explicado al comienzo del capítulo.
Asumimos que todos los canales tienen capacidad para transportar un único mensaje en $\mathbb{F}_q$ y que el mensaje es de tamaño $\mathbb{F}_q^n$ (así, $n_k = n$ para cada $k = 1, 2, \ldots, N$). También supondremos que el corte mínimo de s a t_k es n, y que

$$|\text{out}(s)| = |\text{in}(t_k)| = n \quad \text{para cada } k = 1, 2, \ldots, N.$$

Para recuperar el mensaje necesitamos recuperar los n paquetes de información, y para ello cada llegada t_k debe invertir la *matriz de transferencia*

$$A^{(k)} = \begin{pmatrix} P_{11}^{(k)}(\mathbf{b}) & P_{12}^{(k)}(\mathbf{b}) & \cdots & P_{1n}^{(k)}(\mathbf{b}) \\ P_{21}^{(k)}(\mathbf{b}) & P_{22}^{(k)}(\mathbf{b}) & \cdots & P_{2n}^{(k)}(\mathbf{b}) \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1}^{(k)}(\mathbf{b}) & P_{n2}^{(k)}(\mathbf{b}) & \cdots & P_{nn}^{(k)}(\mathbf{b}) \end{pmatrix} \in \mathbb{F}_q^{n \times n} \tag{3.4}$$

donde $P_{ij}^{(k)} \in \mathbb{F}_q[x_1, x_2, \ldots, x_r]$, r es el número de total de coeficientes en la red, y $\mathbf{b}$ son los coeficientes de las combinaciones lineales que llegan a la red. Sea $P^{(k)}(\mathbf{b}) = \det(A^{(k)})$, siendo $P^{(k)}$ un polinomio de n variables sobre $\mathbb{F}_q$, para cada $k = 1, 2, \ldots, N$.
Por el teorema de max-flow min-cut (teorema 1.1.2), existe una solución para cada sumidero t_k, luego la matriz $A^{(k)}$ debe ser regular, es decir, $P^{(k)} \neq 0$ para cada $k = 1, 2, \ldots, N$.
Podemos considerar el polinomio $P = P^{(1)} \cdot P^{(2)} \cdots P^{(N)} \in \mathbb{F}_q[x_1, x_2, \ldots, x_r]$, que es el producto de todos los anteriores.
Aplicando el lema de Schwartz-Zippel (teorema 2.3.1) al polinomio P, tenemos que

$$|V(P)| \leq q^{n-1} \cdot \deg(P).$$

Notemos que si $q > \deg(P)$, entonces $|V(P)| < q^n = |\mathbb{F}_q|$. Por tanto, existe una solución $\mathbf{b}$ al problema del Network Coding lineal, ya que

$$P(\mathbf{b}) = \det(A^{(1)}) \cdot \det(A^{(2)}) \cdots \det(A^{(N)}) \neq 0.$$

Con esto se prueba la existencia de una solución, aunque no se ha especificado la forma en la que se obtiene.

Observación. Notemos que si se escogen los coeficientes $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ uniformemente al azar, la probabilidad de encontrar una solución aumenta conforme más grande sea el tamaño del cuerpo sobre el que se trabaja

$$\Pr(P(\mathbf{b}) \neq 0) \geq 1 - \frac{\deg(P)}{q} \to 1, \quad \text{si } q \to \infty.$$

De esto podemos deducir que cuanto más grande sea el tamaño del cuerpo en un código de red lineal obtenido uniformemente al azar, habrá más probabilidad de éxito en la obtención de una solución.

3.4. Construcción de un código de red lineal

Hemos visto que existe una solución al problema lineal del Nework Coding. En esta sección daremos una demostración alternativa, que será constructiva, y también veremos un algoritmo con el que se podrá calcular el código lineal sobre la red. En verdad, para obtener el mensaje solo nos interesa la codificación en los sumideros.
Recordamos que en el capítulo 1 se demostró el teorema de max-flow min-cut, donde suponíamos la conservación de la información en cada nodo. Si consideramos el máximo flujo de s a un vértice $u \in V$, teniendo en cuenta el teorema, la tasa de información recibida por el nodo u no puede exceder

el máximo flujo de s a u (ese valor se denota por maxflow(u)). De manera similar, podemos hacer lo mismo para un conjunto de vértices:

Definición 3.4.1. Sea $\mathcal{I} = \{u_1, u_2, \ldots, u_l\}$ un conjunto de vértices de G donde ninguno de ellos es una fuente. Se define el máximo flujo de los vértices $\{u_1, u_2, \ldots, u_l\}$ como

$$\text{maxflow}(\mathcal{I}) = \text{máx}\{\text{maxflow}(u_i) \mid i = 1, 2, \ldots, l\}$$

La anterior definición es una cota superior y en general, alcanzarla depende de la topología de la red, la dimensión w y el esquema de codificación. En nuestro caso, va a poder alcanzarse siempre para cada u, por definición. Se definen tres clases de códigos de red lineales según si la cota antes mencionada es alcanzada o no.

Definición 3.4.2. Sean f_e los vectores de los núcleos de codificación globales en un código de red lineal $w-$dimensional y $\mathbb{F}-$evaluada en una red acíclica. Sea $V_u = \langle\{f_e : e \in in(u)\}\rangle$ para cada nodo u. Entonces, el código de red lineal puede ser:

- **multicast lineal** si $\dim(V_u) = w$ para cada nodo $u \in V \setminus \{s\}$ con $\text{maxflow}(u) \geq w$.
- **broadcast lineal** si $\dim(V_u) = \text{mín}\{w, \text{maxflow}(u)\}$ para cada $u \in V \setminus \{s\}$.
- **dispersión lineal** si $\dim(\langle \cup_{u \in \mathcal{I}} V_u \rangle) = \text{mín}\{w, \text{maxflow}(\mathcal{I})\}$ para cada colección $\mathcal{I}$ que no contiene una fuente.

Se pueden ver unos ejemplos sencillos e ilustrativos de este tipo de códigos de red lineales en [6] (pág. 21).
Toda dispersión lineal es un broadcast lineal y todo broadcast lineal es un multicast lineal, pero no ocurre al revés.
Estos sistemas tienen una gran variedad de aplicaciones, desde los sistemas LAN hasta la conversión de imágenes de color a blanco y negro. Cada uno se utiliza con diferentes propósitos, generalmente el multicast lineal y el broadcast lineal son los más extendidos, y el de dispersión lineal se utiliza cuando el problema requiere de una mayor especificidad, ya que este último es más fuerte que el multicast y que el broadcast.
En nuestro caso será suficiente la condición $\dim(V_{t_k}) = w$ para $1 \leq k \leq N$, porque para hallar el mensaje es suficiente con que se dé el rango máximo en la llegada, sin importar lo que ocurra en los nodos intermedios. Observamos que esta condición es claramente más débil que en el multicast lineal, ya que

esta también nos pide la condición para los nodos intermedios.
En esta sección demostraremos que existe una solución para la dispersión lineal, quedando así probado automáticamente para el resto de casos, en particular, para el nuestro, que es más débil incluso que el multicast lineal.
Una manera de construir un problema de multicast/broadcast/dispersión lineal es considerando un código de red lineal donde cada colección de núcleos de codificación global que puedan ser linealmente independientes, lo sean. Esto motiva la definición de código de red lineal genérico:

Definición 3.4.3 (código de red genérico)**.** Sea $\mathbb{F}$ un cuerpo y w un entero porsitivo.
Un código de red lineal $w-$dimensional y $\mathbb{F}-$evaluado sobre una red de comunicación acíclica es genérico si siendo $\{e_1, e_2, \ldots, e_m\}$ un conjunto de canales arbitrario con $u_j \in e_j$, se cumple que los vectores $f_{e_1}, f_{e_2}, \ldots, f_{e_m}$ son linealmente independientes siempre que

$$\langle \{f_d : d \in in(u_j)\} \rangle \not\subset \langle \{f_{e_k} : k \neq j\} \rangle \quad \text{para } 1 \leq j \leq m$$

La condición realmente quiere decir que cualquier conjunto de núcleos de codificación global que puedan ser linealmente independientes lo serán.
A continuación se verá la existencia de un código de red lineal genérico, y que además cualquier código de red lineal genérico es también un sistema de dispersión lineal. No obstante, no cualquier sistema de dispersión es un código de red lineal genérico.
Cabe destacar que la anterior definición está escrita en términos del álgebra lineal y que no se ha necesitado la noción de flujo máximo, al contrario que en las definiciones anteriores.
La existencia de un código de red lineal genérica $w-$dimensional y $\mathbb{F}-$evaluada depende del valor de w, del cuerpo base $\mathbb{F}$ y de la topología de la red.

Observación. ¿Es lo mismo una red de código que un código de red? No.
Dada una red, se realiza la codificación de la misma mediante un código de red, donde al final del proceso se llega a una red codificada o red de código. Estos casos pueden ser no lineales, aunque nosotros solo consideraremos los lineales.

Para poder considerar un sistema de este tipo, se requiere que algunas colecciones de núcleos de codificación global generen el mayor número posible de dimensiones. Esto es equivalente a que ciertas funciones tomen valores no nulos, donde las variables de los polinomios sean los núcleos indeterminados (se puede ver con que el rango de la matriz sea máximo).
El siguiente lema es consecuencia directa de la cota de Schwartz-Zippel 2.3.1, vista en el capítulo 2.

Lema 3.4.1. *Sea $g(x_1, x_2, \ldots, x_n)$ un polinomio no nulo sobre un cuerpo finito $\mathbb{F}$ ($g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$). Si $|\mathbb{F}| > deg(g)$, entonces existen $a_1, a_2, \ldots, a_n \in \mathbb{F}$ tal que $g(a_1, a_2, \ldots, a_n) \neq 0$.*

Demostración. Denotemos $|\mathbb{F}| = q$ y $deg(g) = d$. La tesis del lema es equivalente a la condición $V(g) \subsetneq \mathbb{F}_q^n$, siendo $V(g)$ el conjunto de ceros del polinomio g. Por el resultado 2.3.1, sabemos que se cumple $|V(g)| \leq d \cdot q^{n-1}$. Por hipótesis del lema, $q > d$, y utilizando la condición anterior, se tiene

$$|V(g)| \leq d \cdot q^{n-1} < q \cdot q^{n-1} = q^n = |\mathbb{F}_q^n|.$$

Lo cual implica que efectivamente, $V(g) \subsetneq \mathbb{F}_q^n$, que es lo mismo que la expresión $\exists \quad \mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{F}_q^n$ tal que $g(a_1, a_2, \ldots, a_n) \neq 0$. □

Recordemos que $V_u = \langle \{f_d : d \in In(u)\} \rangle$. Veamos ahora un algoritmo simple de construcción de un código lineal.

3.4.1. Algoritmo de construcción de un código de red lineal genérico

Sea w un entero positivo y una red acíclica de N canales. Este algoritmo construye un código de red lineal $w-$dimensional y $\mathbb{F}-$evaluado cuando el cuerpo tiene más de $\binom{N+w-1}{w-1}$ elementos. El siguiente procedimiento engloba núcleos de codificación globales que forman una red de código lineal global:

Algoritmo 2: Algoritmodemulticastlineal

Input : $w > 0$, $(G, s, \mathcal{T}, c)$ red acíclica con N llegadas
Output: El núcleo de codificación global f_e para cada $e \in E$ del código de red lineal

1 **para** $e \in E$ *cada canal en la red que no sea un canal imaginario* **hacer**
  $f_e := \mathbf{0}$ //Inicialización **para** $u \in V$ *(ordenados de la fuente al sumidero)* **hacer**
    para *cada canal* $e \in out(u)$ **hacer**
      Escoger un vector $\bar{v}$ en el espacio V_u tal que $\bar{v} \notin \langle \{f_d : d \in \psi\} \rangle$ donde ψ es cualquier conjunto de $w-1$ canales, incluyendo los posibles canales imaginarios en $in(s) \setminus \{e\}$, con $V_u \not\subset \langle \{f_d : d \in \psi\} \rangle$;
    $f_e = \bar{v}$;
2 **devolver** f_e

Observación. El algoritmo es de tipo *greedy* o *voraz,* esto quiere decir que en cada iteración se escoge el paso más simple o directo a nivel local, que no será necesariamente el óptimo a nivel global. Este tipo de algoritmos no son óptimos por lo general, pero tienen sus ventajas, como por ejemplo, garantizar la existencia de una solución de forma constructiva. No suelen implementarse en la práctica, y se buscan algoritmos más elaborados aunque también más eficientes a nivel global.

La demostración completa sobre el procedimiento del algoritmo 2 se encuentra en la página 36 de [6].
Dado que se verifica el algoritmo, podemos afirmar el siguiente teorema:

Teorema 3.4.1. *Dado un entero positivo y una red acíclica, existe una red de código lineal genérica $w-$dimensional y $\mathbb{F}-$evaluada para un cuerpo base $\mathbb{F}$ suficientemente grande.*

Demostración. El algoritmo expuesto anteriormente justifica el resultado. □

Se derivan bastantes resultados del teorema anterior:

Corolario 3.4.1. *Dado un entero positivo w y una red acíclica, existe una dispersión lineal $w-$dimensional y $\mathbb{F}-$evaluada para un cuerpo suficientemente grande.*

Demostración. Todo código de red lineal genérico es una dispersión lineal (esto se verá en un resultado posterior, en particular, en el teorema 3.4.2). □

Corolario 3.4.2. *Dado un entero positivo w y una red acíclica, existe un broadcast lineal $w-$dimensional y $\mathbb{F}-$evaluada para un cuerpo suficientemente grande.*

Corolario 3.4.3. *Dado un entero positivo w y una red acíclica, existe un multicast lineal $w-$dimensional y $\mathbb{F}-$evaluada para un cuerpo suficientemente grande.*

Los dos corolarios 3.4.3 y 3.4.2 se siguen inmediatamente del anterior porque las condiciones de la tesis son menos fuertes pero las hipótesis son las mismas. Introducimos la siguiente definición para simplificar la notación:

Definición 3.4.4. Sea $\mathcal{I} = \{u_1, u_2, \ldots, u_n\}$ un conjunto de nodos en una red. El conjunto $\text{cut}(\mathcal{I})$ son las aristas con llegada en uno de los u_i y salida de un nodo que no está en $\mathcal{I}$.

Observación. Si $s \in \mathcal{I}$, entonces $\text{cut}(\mathcal{I})$ contiene a todos los w canales imaginarios.

Ejemplo 3.4.1. Para la red mariposa, si cogemos $\mathcal{I} = \{v_2, v_4\}$, se tiene que $cut(\mathcal{I}) = \{sv_1, v_3v_4\}$. En cambio, para el conjunto de nodos $\mathcal{J} = \{s, v_2, v_4, t_1, t_2\}$ tenemos que su corte es $cut(\mathcal{J}) = \{os, o's, v_3v_4, v_1t_1\}$, donde os y $o's$ representan los dos canales imaginarios.

Antes de seguir con un nuevo resultado, introduciremos un nuevo término.

Definición 3.4.5. Supongamos un grafo $G = (V, E)$ con una fuente s y un sumidero t.
Se dice que un nodo v está en el **upstream** de u si para todos los cortes mínimos de u a t (S, T), v está en S. Decimos que un nodo v está en el **downstream** de u si para todos los cortes mínimos de u a t (A, B), v está en B.
Se dice que un nodo v es **central** cuando no es ni upstream ni downstream, esto es, que existe un corte mínimo (S, T) de u a t tal que $v \in S$ (upstream), y otro corte mínimo (S', T') tal que $v \in T'$ (downstream).

Estos conceptos son bastante utilizados en redes de flujo y teoría de la comunicación, y se pueden generalizar para grafos con varias llegadas, e incluso varias salidas. Intuitivamente, se utiliza para referirse para saber si un nodo v se encuentra "antes" o "después" de u, dando un orden a los vértices del grafo. Por ejemplo, para un nodo u que no es ni fuente ni sumidero, s está en el upstream de u y t está en el downstream de u.

Lema 3.4.2. *Sea f_e el núcleo de codificación lineal para un canal e en una red codificada linealmente sobre una red acíclica. Entonces,*

$$\langle \{f_e : e \in cut(\mathcal{I})\} \rangle = \langle \cup_{u \in \mathcal{I}} V_u \rangle$$

Demostración. Primero, notemos que

$$\langle \cup_{u \in \mathcal{I}} V_u \rangle = \langle \{f_e : e \text{ acaba en un nodo de } \mathcal{I}\} \rangle.$$

Necesitamos mostrar que el siguiente conjunto es vacío: $\Psi = \{c : f_c \notin \langle \{f_e : e \in \text{cut}(\mathcal{I})\} \rangle$y c se dirige a un nodo de $\mathcal{I}$.
Razonemos por reducción al absurdo y supongamos que existe $c \in \Psi$ tal que $c \in out(u)$. De la definición de código de red lineal, f_c es es una combinación lineal de vectores f_d donde $d \in in(u)$. Como $f_c \notin \langle \{f_e : e \in \text{cut}(\mathcal{I})\}$, existe un canal $d \in in(u)$ con $f_d \notin \langle \{f_e : e \in \text{cut}(\mathcal{I})\}$. Como d está en el upstream de c, debe ser $d \notin \Psi$. Así el final de d es el origen de c. Esto hace que c sea un canal de $\text{cut}(\mathcal{I})$, contradiciendo que $f_c \notin \langle \{f_e : e \in \text{cut}(\mathcal{I})\} \rangle$. □

Lema 3.4.3. *Sea $\mathcal{I}$ un conjunto de nodos que no son fuentes en una red acíclica con w canales imaginarios. Entonces,*

$$\min\{w, \mathit{maxflow}(\mathcal{I})\} = \min_{\mathcal{I}\subset\mathcal{J}} |\mathit{cut}(\mathcal{J})|,$$

donde $|\mathit{cut}(\mathcal{J})|$ es el cardinal del conjunto $\mathcal{J}$.

Demostración. En la prueba nos apoyaremos en el resultado de max-flow min-cut que aplica a una red con una fuente y un sumidero.
Si colapsamos todo el conjunto $\mathcal{I}$ en un sumidero, y consideramos una fuente imaginaria en el upstream de S, entonces el máximo flujo entre este par de fuente y sumidero es precisamente el mínimo entre el máximo flujo de $\mathcal{I}$ y w; y el corte mínimo entre este par es precisamente el mínimo de la capacidad del corte $\mathcal{J}$ con $\mathcal{J} \supset \mathcal{I}$, ya que consideramos que solo se puede enviar un mensaje por cada arista. □

El anterior lema es parecido al max-flow min-cut en un problema de red de flujo.

Teorema 3.4.2. *Todo código de red lineal genérico es un código de dispersión lineal.*

Demostración. Sea f_e el núcleo de codificación global para cada canal e en un código de red lineal genérico $w-$dimensional en una red acíclica. Consideremos la notación

$$\mathrm{span}(\mathcal{I}) := \langle\{f_e : e \in \mathrm{cut}(\mathcal{I})\}\rangle = \langle\cup_{u\in\mathcal{I}} V_u\rangle$$

para un conjunto de nodos $\mathcal{I}$ que no contiene la fuente. Claramente, para cada conjunto $\mathcal{J} \supset \mathcal{I}$ ($\mathcal{J}$ puede contener s), tenemos que $\mathrm{span}(\mathcal{J}) \supset \mathrm{span}(\mathcal{I})$, y por lo tanto,

$$\dim(\mathrm{span}(\mathcal{I})) \leq \dim(\mathrm{span}(\mathcal{J})) \leq |\mathrm{cut}(\mathcal{J})|,$$

es decir, llegamos a que $\dim(\mathrm{span}(\mathcal{I})) \leq \min_{\mathcal{I}\subset\mathcal{J}} |\mathrm{cut}(\mathcal{J})|$. Por el lema anterior,

$$\dim(\mathrm{span}(\mathcal{I})) \leq \min_{\mathcal{I}\subset\mathcal{J}} |\mathrm{cut}(\mathcal{J})| = \min\{w, \mathrm{maxflow}(\mathcal{I})\} \leq w. \tag{3.5}$$

Para que el código de red lineal dado sea una dispersión lineal, necesitamos que se cumpla la definición, es decir, que $\dim(\mathrm{span}(\mathcal{I})) = \min\{w, \mathrm{maxflow}(\mathcal{I})\}$ para cada conjunto $\mathcal{I}$ de nodos (de los que ninguno es una fuente). Por 3.5 esto se cumple si: o bien $\dim(\mathrm{span}(\mathcal{I})) = w$ o bien existe un conjunto $\mathcal{J}$ que

contiene a $\mathcal{I}$ tal que $\dim(\text{span}(\mathcal{I})) = |\text{cut}(\mathcal{J})|$ (donde $\mathcal{J}$ puede contener s).
Si se cumple $\dim(\text{span}(\mathcal{I})) = w$ ya se tendría.
Supongamos entonces que $\dim(\text{span}(\mathcal{I})) < w$ (no puede ser mayor) y veamos que se da

$$\dim(\text{span}(\mathcal{I})) = |\text{cut}(\mathcal{J})|. \tag{3.6}$$

Como el código de red lineal es genérico, por definición se tiene que $\dim(\text{span}(\mathcal{I}))$ es igual a o bien $|\text{cut}(\mathcal{J})|$ o bien $|\text{cut}(\mathcal{I}\cup\{s\})|$, dependiendo de si $|\text{cut}(\mathcal{I})| \leq w$ o no. Con esto se llega a la afirmación que se quería considerando $\mathcal{J}$ como $\mathcal{I}$ o $\mathcal{I} \cup \{s\}$.
Después, consideremos que hay más nodos que no son fuente fuera de $\mathcal{I}$. Sea u un nodo cualquiera de ellos, y consideremos el conjunto $\mathcal{I}' = \mathcal{I} \cup \{u\}$.
Entonces, si $\dim(\text{span}(\mathcal{I}')) = w$, consideraremos $\mathcal{J}'$ el conjunto de todos los nodos, si no, la existencia de ese conjunto se sigue de la hipótesis de inducción.
Ahora si $\dim(\text{span}(\mathcal{I}')) = \dim(\text{span}(\mathcal{I})$, entonces se verifica la ecuación 3.6 tomando $\mathcal{J}$ por $\mathcal{J}'$. Por otro lado, si asumimos $\dim(\text{span}(\mathcal{I}')) > \dim(\text{span}(\mathcal{I})$, entonces existe un canal $d \in in(u)$ tal que

$$f_d \notin \text{span}(\mathcal{I}). \tag{3.7}$$

La anterior afirmación se aplica a cada nodo u fuera de $\mathcal{I}$ que no sea una fuente. Por haber asumido $\dim(\text{span}(\mathcal{I})) < w$, esto se aplica también para el caso donde el nodo u es una fuente. Con esto, debemos mostrar que

$$\dim(\text{span}(\mathcal{I})) = |\text{span}(\mathcal{I})|. \tag{3.8}$$

Esto implicaría 3.6. En realidad la ecuación 3.8 equivale a que los mensajes codificados a través de las aristas incidentes en los nodos de $\mathcal{I}$ sean linealmente independientes.
Consideremos $\text{cut}(\mathcal{I}) = \{e_1, e_2, \ldots, e_m\}$ donde $e_j \in out(u_j)$, y apliquémoslo a 3.7 para $u = u_j$ para cada j. Entonces, existe un canal $d \in in(u)$ tal que $f_d \notin \text{span}(\mathcal{I})$. Luego

$$\langle f_d : d \in in/u_j)\rangle \not\subset \langle f_{e_k} \mid 1 \leq k \leq m\rangle = \text{span}(\mathcal{I}), \qquad j = 1, 2, \ldots, m.$$

De ello se sigue, usando la notación dada para $\text{span}(\mathcal{I})$, lo siguiente:

$$\langle f_d : d \in in(u_j)\rangle \not\subset \langle f_{e_k} : k \neq j\rangle,$$

porque claramente $\{e_k : k \neq j\}$ es un subconjunto de $\{e_1, e_2, \ldots, e_m\}$.
Por lo tanto, existe un conjunto de nodos $\mathcal{J}'$ que contiene a $\mathcal{I}'$ tal que

$$\dim(\text{span}(\mathcal{J}')) = |\text{cut}(\mathcal{J})'|.$$

Por la definición de código de red genérico, la cual muestra la condición necesaria para un código de red lineal sea genérico, los vectores $f_{e_1}, f_{e_2}, \ldots, f_{e_m}$ son linealmente independientes. Esto último verifica 3.6, que era lo que queríamos para concluir la prueba. □

Con este teorema se garantiza que los resultados obtenidos para las redes genéricas son aplicables para el resto de situaciones, y en particular, para el multicast lineal.
Realmente el multicast lineal es mucho más fuerte que la condición que exigíamos. En el multicast lineal se puede hallar la inversa a la derecha en cada nodo, mientras que para funcionar basta tenerlo para las fuentes solamente.
Aquí se ha brindado una demostración constructiva, mostrando también los algoritmos que se llevan a cabo para implementarlo, lo que demuestra su existencia. Esto último era el objetivo de la sección.

Como introducción a la optimización de algoritmos, la siguiente sección tratará de proporcionar un algoritmo más refinado que el que acabamos de ver. Para nuestro caso, es suficiente ver la existencia del multicast lineal, no obstante, en la referencia [6] (págs. 44-50) se puede encontrar una construcción para una red lineal genérica estática, que es mucho más fuerte que el multicast lineal.

3.4.2. Algoritmo más óptimo para el multicast lineal

Hemos visto anteriormente el algoritmo de construcción del problema multicast lineal. Se puede modificar el algoritmo anterior para construir de manera más eficiente un multicast lineal, en concreto, uno que sea $w-$ *dimensional* y $\mathbb{F}-$evaluado sobre una red acíclica con $|\mathbb{F}| > \eta$, donde η es el número de nodos intermedios $u_1, u_2, \ldots, u_\eta$ con $\text{maxflow}(u_i) \geq w$. También es cierto que cuanta más baja sea la cota $\binom{N+w-1}{w-1}$, más fuertes son las declaraciones de existencia. El siguiente procedimiento describe un núcleo de codificación global f_e para cada canal e en la red de manera que $\dim(V_{u_q}) = w$ para $1 \leq q \leq \eta$:

Algoritmo 3: Construcción de un código lineal genérico en una red de comunicación

Input : $w > 0$, $(G, s, \mathcal{T}, c)$ red acíclica con N llegadas
Output: El núcleo de codificación global f_e para cada $e \in E$ del código de red lineal

para *cada canal e en una red* **hacer**
  $f_e := \mathbf{0}$ //como inicialización
para $q = 1; q \leq \eta; q++$ **hacer**
  para $i = 1; i \leq w; i++$ **hacer**
    $e_{q,i} :=$ {el canal imaginario que inicia el camino $P_{q,i}$}
    // esto es una inicialización
    // después $e_{q,i}$ será actualizado dinámicamente moviéndose
    // a lo largo del camino $P_{q,i}$ hasta que $e_{q,i}$ se convierta en un
    // canal en $In(u_q)$.
para *cada nodo u, en cualquier orden de upstream a downstream* **hacer**
  para *cada canal* $e \in out(u)$ **hacer**
    Escoger un vector $\bar{v} \in V_u$ tal que $\bar{v} \notin \langle \{f_{e_{q,j}} : j \neq i\} \rangle$ para cada par (q, i); //Verificación de independencia lineal
    $f_e = \bar{v}$; **para** *cada par* (q, i) **hacer**
      $e_{q,i} = e$;
devolver f_e

Conclusión

En resumen, la codificación en redes se ha convertido en una herramienta esencial para abordar la transmisión de la información de manera eficiente en sistemas con un emisor y varios receptores. A lo largo de este trabajo, hemos explorado algunos de los retos asociados con la implementación del Linear Network Coding, incluyendo consideraciones sobre la gestión del flujo.

Hemos planteado la pregunta de cómo se extiende este concepto cuando hay múltiples emisores y receptores, lo que generaría nuevas soluciones potenciales. Puede estudiarse la noción de redes de flujo equivalentes para simplificar estos problemas más complejos a formas más manejables.
Si bien hemos utilizado el algoritmo de Ford Fulkerson como una herramienta efectiva en nuestra investigación, es importante destacar que existen algoritmos más eficientes que podrían ser objeto de investigaciones futuras. La búsqueda de algoritmos de tiempo polinomial más eficientes es un área de investigación que sigue desarrollándose.

Por último, no hemos abordado la cuestión del tamaño de los símbolos en la red, ni hemos explorado completamente las posibilidades de optimización de la complejidad memorística. Estas áreas ofrecen oportunidades adicionales para mejorar la eficiencia y el rendimiento de la codificación en redes lineales.

Bibliografía

[1] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[2] David Cox, John Little, Donal O'Shea, and Moss Sweedler. Ideals, varieties, and algorithms. *American Mathematical Monthly*, 101(6):582–586, 1994.

[3] David A Cox, John Little, and Donal O'shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.

[4] Ralf Koetter and Muriel Médard. An algebraic approach to network coding. *IEEE/ACM transactions on networking*, 11(5):782–795, 2003.

[5] Rudolf Lidl and Harald Niederreiter. *Introduction to finite fields and their applications*. Cambridge university press, 1994.

[6] Raymond W Yeung. *Network coding theory*. Now Publishers Inc, 2006.