



---

---

# UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

Facultad de Ciencias

## LAS GRAFICAS DE STALLINGS Y ALGORITMOS PARA SUBGRUPOS DE GRUPOS LIBRES

TESIS

que para obtener el título de

**Maestro en Ciencias Aplicadas**

PRESENTA:

**Marco Antonio García Morales.**

Directores de tesis:

**Dr. Lev Glebsky.**

**Dra. Marcela Mejía Carlos.**

Julio de 2017.

San Luis Potosí, S.L.P., México

Título:  
LAS GRAFICAS DE STALLINGS Y ALGORITMOS  
PARA SUBGRUPOS DE GRUPOS LIBRES

Nombre del estudiante:  
L.M. Marco Antonio García Morales.

Comité que acepta la tesis:

Dr. Lev Glebsky. (Asesor) \_\_\_\_\_

Dra Marcela Mejía Carlos. (Asesor) \_\_\_\_\_

Dr. Gelasio Salazar Anaya. (Sinodal) \_\_\_\_\_

Dr. Edgardo Ugalde Saldaña. (Sinodal) \_\_\_\_\_

Dr. César Israel Hernández Vélez. (Sinodal) \_\_\_\_\_

# Agradecimientos

Quiero agradecer a mis asesores de tesis, el Dr. Lev Glesky, por sus conocimientos invaluable que me ofreció para llevar a cabo este trabajo, por todo el tiempo que me brindo. A la Dra. Marcela Mejía Carlos por sus ideas y recomendaciones respecto al presente trabajo, y por el tiempo que dedico a la revisión del mismo.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado durante mis estudios de posgrado.

Agradezco a los miembros del jurado, el Dr. Gelasio Salazar Anaya, Dr. Edgardo Ugalde Saldaña y al Dr. César Israel Hernández Vélez, por las contribuciones que hicieron al trabajo final y por el tiempo que dedicaron para revisarlo.

Quiero agradecer a mi familia, por todo el apoyo que me han brindado a lo largo de mi vida, especialmente a mis padres quienes siempre se han preocupado por mí, quienes me han formado y educado. A quienes la ilusión de su existencia ha sido verme convertido en una persona de provecho.

Finalmente agradezco a mi novia Koymbra Jannylth, quien me ha impulsado a seguir siempre adelante, gracias por todo tu apoyo y cariño que me has brindado a lo largo de estos años.



# Índice general

<b>1. PRELIMINARES</b>	<b>9</b>
1.1. Grupo Libre . . . . .	9
1.2. Gráficas . . . . .	12
1.3. Algoritmos . . . . .	12
1.3.1. Órdenes de complejidad más comunes . . . . .	14
1.3.2. Representaciones de Gráficas . . . . .	15
<b>2. LAS GRÁFICAS DE STALLINGS</b>	<b>17</b>
2.1. Gráficas de Stallings . . . . .	17
2.2. Proceso de empalmes de Stallings . . . . .	24
2.3. Algunas ideas y el algoritmo . . . . .	28
2.3.1. Construcción de $\Gamma_0$ . . . . .	29
2.3.2. Algoritmo . . . . .	30
<b>3. SUBGRUPO MALNORMAL Y CICLONORMAL</b>	<b>33</b>
3.1. Subgrupo Malnormal . . . . .	34
3.1.1. Programa Subgrupo Malnormal . . . . .	36
3.2. Subgrupo Ciclonormal . . . . .	37
3.2.1. Programa Subgrupo Ciclonormal . . . . .	38
<b>4. S-SUBGRUPO</b>	<b>39</b>
4.1. S-Subgrupo . . . . .	39
4.1.1. Programa S-subgrupo . . . . .	43
<b>Conclusiones</b>	<b>47</b>
<b>APENDICE</b>	<b>49</b>



# Introducción

Las gráficas de Stallings tienen muchas aplicaciones interesantes, y se han convertido en una interpretación de los subgrupos (finitamente generados) de los grupos libres, son muy útiles cuando se consideran problemas algorítmicos. Sea  $H$  un subgrupo finitamente generado de un grupo libre finitamente generado  $F(X)$ , a cada subgrupo  $H$  de un grupo libre  $F(X)$  se le puede asociar una gráfica dirigida  $\Gamma(H)$ , donde sus aristas están etiquetadas con elementos de  $X$ , estas gráficas tienen toda la información esencial sobre el subgrupo  $H$ .

Existen algoritmos que pueden ayudar a resolver problemas como el decidir cuándo un elemento  $f \in F(X)$  pertenece o no a  $H$ , calcular el índice de un subgrupo  $H$ , así como también calcular cuándo un par de subgrupos  $H$  y  $K$  son conjugados; además existe un algoritmo que calcula la intersección de 2 subgrupos; así como también un algoritmo que decide cuándo un subgrupo  $H$  es normal. Los algoritmos antes mencionados se pueden ver en [1].

En el presente trabajo se expone un algoritmo cuyo objetivo es encontrar la gráfica empalmada de Stallings  $\Gamma(H)$ , correspondiente a un subgrupo finitamente generado  $H$ , el cual es llamado “Proceso de empalmes de Stallings”. Se elaboró un programa que realiza el proceso de empalmes de Stallings, el cual tendrá como datos de entrada los generadores del subgrupo  $H$  y como dato de salida será la gráfica  $\Gamma(H)$ .

Se expondrán algoritmos que nos ayudan a decidir cuándo un subgrupo  $H$  es malnormal, ciclonormal y S-subgrupo [2], estos algoritmos utilizan las gráficas empalmadas de Stallings. Todos estos algoritmos se implementaron en SageMath, (que es un sistema algebraico computacional que usa un lenguaje de programación basado en Python). Cuando un subgrupo no es ciclonormal, malnormal o S-subgrupo esto nos ayuda a verificar si tienen “The Almost Congruence Extension Property” [2].

En la mayoría de los programas se calculará la complejidad de los algoritmos utilizados, los cuales no son óptimos, pero tienen un orden de complejidad razonable.

El presente trabajo se encuentra dividido en cuatro capítulos. En el capítulo 1 se presentan algunas definiciones preliminares sobre teoría de grupos libres, teoría de gráficas y complejidad algorítmica. En el capítulo 2 se dan algunas definiciones, se describe el proceso de empalmes de Stallings, así como también se presentan algunas ideas y el algoritmo, para realizar el proceso de empalmes de Stallings. En el capítulo 3 se habla de los subgrupos malnormales y ciclonormales, se muestran 2 algoritmos, el primer algoritmo nos ayuda a decidir cuándo un subgrupo finitamente generado es malnormal; el segundo algoritmo nos ayuda a decidir cuándo un subgrupo finitamente generado es ciclonormal. También se muestra en este capítulo el pseudocódigo utilizado en SageMath. Y finalmente en el capítulo 4 se habla de los S-subgrupos [2], se exhibe un algoritmo que nos ayuda a decidir cuándo un subgrupo es S-subgrupo, también se muestra el pseudocódigo que se utilizó en SageMath.

# Capítulo 1

## PRELIMINARES

En este capítulo se introducen las definiciones y resultados preliminares. En la primera parte se presentan conceptos básicos de teoría de grupos libres, necesarios para este trabajo. En la segunda parte se presentan conceptos básicos de teoría de gráficas. Y por último, en la tercera parte se presentan conceptos sobre algoritmos. También se muestra una tabla sobre los tiempos de complejidad más comunes que existen en los algoritmos.

### 1.1. Grupo Libre

Sea  $X$  un conjunto arbitrario de un grupo, se define el conjunto  $X^{-1} = \{x^{-1} : x \in X\}$ , donde el símbolo  $x^{-1}$  es el inverso formal de  $x$  y viceversa, se asume que  $X \cap X^{-1} = \emptyset$ .

**Definición 1.1.1. (Grupo Libre).** [4] Sea  $F$  un grupo y sea  $X$  un subconjunto linealmente ordenado de  $F$ , tal que  $X \cap X^{-1} = \emptyset$  donde  $X^{-1} = \{x^{-1} : x \in X\}$ . El grupo  $F$  es llamado grupo libre con la base  $X$ , si cualquier elemento no trivial  $f$  se puede representar de forma única como  $f = x_1 x_2 \dots x_n$ , donde  $x_i \in X \cup X^{-1}$  y  $x_i x_{i+1} \neq 1$  para toda  $i$ .

Esta definición implica que  $X$  genera al grupo  $F$ . En particular, el grupo cíclico  $\mathbb{Z}$  es un grupo libre, generado por un único elemento.

**Teorema 1.1.2.** [4] Para cualquier conjunto  $X$  existe un grupo libre con base  $X$  y lo denotamos por  $F(X)$ .

Antes de hacer la demostración de este teorema, se necesitan algunas ideas y se demuestra una proposición auxiliar.

Sea  $X$  un conjunto cualquiera, se define  $X^{-1} = \{x^{-1} : x \in X\}$ , donde  $x^{-1}$  denota un nuevo símbolo correspondiente a  $x$ . Se asume que  $X \cap X^{-1} = \emptyset$  y que la expresión  $(x^{-1})^{-1}$  denota al elemento  $x$ .

El conjunto  $\Sigma = X \cup X^{-1}$  es llamado alfabeto y los elementos de este conjunto son llamados letras. Una palabra  $w$  sobre  $\Sigma$  se escribe de la forma  $x_1x_2x_3 \dots x_n$  donde  $x_i \in \Sigma$  para  $n \geq 0$ , si  $n = 0$  se obtiene la palabra vacía. Una subpalabra de la palabra  $w$  es cualquier subsecuencia de letras consecutivas.

Sean  $u$  y  $v$  dos palabras sobre  $\Sigma$  donde  $u = u_1u_2 \dots u_n$  y  $v = v_1v_2v_3 \dots v_m$ , entonces la concatenación de  $u$  y  $v$  denotada por  $u \circ v$  o simplemente  $uv$  es la palabra  $uv = u_1u_2 \dots u_nv_1v_2 \dots v_m$ . Una palabra es libremente reducida si no contiene subpalabras de la forma  $xx^{-1}$ ,  $x^{-1}x$  con  $x \in X$ .

Sea  $w$  una palabra sobre  $\Sigma$ , si se eliminan todas las subpalabras de la forma  $xx^{-1}$  y  $x^{-1}x$  con  $x \in X$ , se obtiene una palabra reducida, se denota por  $\bar{w}$  la palabra reducida correspondiente a  $w$ , esta palabra es única. La longitud de una palabra  $w$  en el alfabeto  $X \cup X^{-1}$ , es la cantidad de letras que contiene y se denota por  $|w|$ . Se expresa por  $|w|_X$  la longitud de la palabra reducida correspondiente a  $w$ .

Sea  $W$  el conjunto de todas las palabras en el alfabeto  $\Sigma$ , sean  $f$  y  $g$  dos palabras sobre  $\Sigma$ , se introduce una relación de equivalencia en  $W$ . Dos palabras  $f$  y  $g$  son equivalentes, si existe una secuencia finita de palabras  $f = f_1, f_2, f_3, \dots, f_k = g$ , tal que cada  $f_{i+1}$  se obtiene de  $f_i$  por inserción o eliminación de subpalabras de la forma  $xx^{-1}$  con  $x \in \Sigma$ . Sea  $[F]$  el conjunto de clases de equivalencia de palabras en  $W$ . La clase que contiene la palabra  $f$  se denota por  $[f]$ .

**Proposición 1.1.3.** *[4] Cualquier clase  $[f]$  contiene una única palabra reducida.*

*Demostración.* La existencia de una palabra reducida en la clase  $[f]$  es evidente. Supongamos que existen 2 palabras reducidas  $u$  y  $v$  en la clase  $[f]$ , se puede elegir una secuencia  $u = f_1, f_2, f_3, \dots, f_k = v$  con la suma mínima de  $\sum_{i=1}^k |f_i|$ . Ya que  $u$  y  $v$  son palabras reducidas y diferentes, tenemos que  $|f_1| < |f_2|$  y  $|f_{k-1}| > |f_k|$ . Por lo tanto existe  $i \in \{2, 3, \dots, k-1\}$  tal que  $|f_{i-1}| < |f_i|$  y  $|f_i| > |f_{i+1}|$ . Por consiguiente  $f_{i+1}$  se puede obtener de  $f_{i-1}$  en dos pasos.

Primero se inserta una subpalabra de la forma  $xx^{-1}$  y se obtiene a  $f_i$ , se consigue de esta manera la palabra  $f_i$ , y después borrar la palabra  $yy^{-1}$ , si estas subpalabras son disjuntas en  $f_i$  se puede ir en otro camino y obtener lo mismo, es decir, primero borrando la subpalabra de la forma  $yy^{-1}$ , y se consigue una nueva palabra  $f'_i$ , después insertar la subpalabra de la forma  $xx^{-1}$ .

Se puede remplazar a  $f_{i-1}, f_i, f_{i+1}$  en la secuencia de palabras  $u = f_1, f_2, \dots, f_k = v$  por  $f_{i-1}, f'_i, f_{i+1}$  y se obtiene una nueva secuencia de palabras  $u = f_1, \dots, f_{i-1}, f'_i, f_{i+1}, \dots, f_k = v$  con la suma de sus longitudes más pequeña que la otra, contradiciendo nuestra afirmación. Si las subpalabras  $xx^{-1}$  y  $yy^{-1}$  su intersección es no vacía, entonces  $f_{i-1} = f_{i-1}$  y se puede

borrar las palabras  $f_i, f_{i+1}$  de la secuencia de palabras  $u = f_1, f_2, f_3, \dots, f_k = v$ , de nuevo se tiene una contradicción con la suma mínima de las longitudes. Por lo tanto  $u = v$ .  $\square$

A continuación se muestra la demostración del teorema 1.1.2.

*Demostración del teorema 1.1.2.* Se define el producto en el conjunto  $[F]$  por  $[f][g] = [fg]$ , y se demuestra que  $[F]$  es un grupo libre con base  $[X] = \{[x] : x \in X\}$ , la asociatividad del producto es evidente, el elemento identidad es la clase  $[\emptyset]$  y el inverso de la clase  $[f] = [x_1x_2 \dots x_n]$ , donde  $x_i \in X \cup X^{-1}$ , es la clase  $[x_n^{-1} \dots x_2^{-1}x_1^{-1}]$ . Además  $[f] = [x_1][x_2] \dots [x_n]$  es reducida con respecto a  $[X]$  si, y solo si, la palabra  $x_1x_2 \dots x_n$  es reducida en  $X$ . La unicidad de la forma reducida de los elementos de  $[F]$  con respecto a  $[X]$ , se sigue del hecho de que cada clase contiene exactamente una palabra reducida. Notemos que la cardinalidad de  $[X]$  es igual a la cardinalidad de  $X$ .  $\square$

**Teorema 1.1.4.** *Todas las bases para un grupo libre  $F$  tienen la misma cardinalidad.*

La demostración del teorema anterior se puede ver en [4].

**Definición 1.1.5.** *El rango de un grupo libre  $F$ , se denota por  $\text{rank}(F)$ , es la cardinalidad de cualquiera de sus bases.*

## 1.2. Gráficas

**Definición 1.2.1.** Una gráfica es un par  $G = (V(G), E(G))$  de conjuntos finitos, los elementos del conjunto  $V(G)$  son llamados vértices o nodos, y los elementos del conjunto  $E(G) \subseteq V \times V$  son llamados aristas o arcos.

A una gráfica  $G$  en ocasiones también se le llama grafo.<sup>1</sup>

**Definición 1.2.2. (X-dígrafo).** [1] Sea  $X = \{x_1, x_2, \dots, x_n\}$  un alfabeto finito y  $\Gamma$  una gráfica dirigida.  $\Gamma$  es un X-dígrafo donde toda arista  $e \in E(\Gamma)$ , tiene dirección y esta etiquetada por una letra de  $X$ , denotémosla por  $\mu(e)$ .

Sea  $\Gamma$  un X-dígrafo, cada arista  $e \in E(\Gamma)$  tiene un origen lo vamos a expresar por  $\circ(e)$  y un fin el cual se denota por  $t(e)$ . Si a  $\Gamma$  se le agregan todas sus aristas inversas, vamos a tener un X-dígrafo con etiquetas en el alfabeto  $\Sigma = X \cup X^{-1}$ , donde  $X^{-1} = \{x_1^{-1}, x_2^{-1}, \dots, x_n^{-1}\}$ , es decir, si  $e \in E(\Gamma)$  se agrega una nueva arista llamada  $e^{-1}$ , donde  $\mu(e^{-1}) = (\mu(e))^{-1}$ ,  $\circ(e^{-1}) = t(e)$  y  $t(e^{-1}) = \circ(e)$ ; a este nuevo X-dígrafo se denota por  $\widehat{\Gamma}$ . Sin embargo, en la mayoría de los casos, vamos a abusar de esta notación y simplemente referirnos a ella como  $\Gamma$ .

Un camino en  $\Gamma$  es una secuencia de aristas  $p = e_1 e_2 e_3 \dots e_k$  donde  $e_i \in E(\widehat{\Gamma})$ , además  $\circ(e_i) = t(e_{i-1})$  para todo  $i \in \{2, 3, 4, \dots, k\}$ , se define el origen del camino  $p$  como  $\circ(p) = \circ(e_1)$ , y el final del camino  $p$  como  $t(p) = t(e_k)$ . La longitud de un camino  $p$  será el número de aristas que lo forman ( $k$  aristas) y se denota por  $|p| = k$ , la etiqueta del camino esta definida por  $\mu(p) = \mu(e_1)\mu(e_2) \dots \mu(e_{k-1})\mu(e_k)$  así,  $\mu(p)$  es una palabra en el alfabeto  $\Sigma$ , donde  $\mu(p)$  puede contener subpalabras de la forma  $xx^{-1}$  o  $x^{-1}x$  para algún  $x \in X$ .

Si  $v \in V(\Gamma)$  se considera la secuencia  $p = v$  como un camino, tal que  $\circ(p) = t(p) = v$  con  $|p| = 0$  y  $\mu(p) = 1$  (la palabra vacía)

**Definición 1.2.3. (Camino reducido).** Se dice que un camino  $p$  en el X-dígrafo  $\Gamma$ , es reducido si no contiene subcaminos de la forma  $ee^{-1}$  con  $e \in E(\widehat{\Gamma})$ .

## 1.3. Algoritmos

Un algoritmo se puede definir como una serie de pasos organizados que describen el proceso que se debe seguir, para dar solución a un problema específico. En informática la complejidad de tiempo de un algoritmo cuantifica la cantidad de tiempo empleado para terminar dicho algoritmo, es decir, en cuanto tiempo el algoritmo arroja el valor de salida, se debe tener en cuenta algunas consideraciones; por ejemplo, si se piensa en un típico algoritmo

<sup>1</sup>Se sabe que en México se les llama gráficas y en España grafos. En ingles se les llama graphs, así que no ayuda, pues son dos palabras que se traducen de graphs.

para ordenar los elementos de un vector, ¿tardará lo mismo un algoritmo de ordenación, en ordenar un vector con 100 valores que uno con 100000 valores?, pues aquí es donde se tiene que empezar a hablar del tamaño del problema, por lo tanto el tamaño del problema incide directamente en el tiempo que tarda el algoritmo en resolverse. Otra consideración a tener en cuenta a la hora de tratar con la complejidad, es que si se está contando el tiempo que tarda un algoritmo en resolver un problema, ¿En qué ordenador se ejecuta el algoritmo?, ¿Cuál debería ser entonces la unidad de medida de la complejidad?; ninguna unidad de tiempo sirve, porque el resultado variaría de un ordenador a otro, se puede adoptar una simplificación que permita no tener en cuenta en qué ordenador se ejecutará el algoritmo, en lugar de medir tiempos, se cuentan el número de instrucciones que debe realizar el algoritmo, se supone que cada instrucción se ejecuta en un tiempo constante. Se puede permitir esa simplificación porque lo que realmente se quiere saber, es cómo crece el número de instrucciones necesarias para resolver el problema, con respecto a su tamaño. Esto es realmente la complejidad.

En la mayor parte de los algoritmos, también influye el propio contenido de los datos. Es posible que para un problema determinado de tamaño  $n$ , algunas veces el algoritmo tarde más y otras tarde menos, dependiendo de los propios datos de entrada del problema. En éste caso, nos conviene distinguir dos opciones, que es lo peor y que es lo mejor que puede pasar para un problema de tamaño  $n$ .

**Ejemplo 1.3.1.** *Supongamos que se tienen  $n$  valores  $a_1, a_2, \dots, a_n$ , y se quiere ver si un determinado valor  $x$  se encuentra entre estos valores, lo que se hace es comparar uno a uno, si se tiene mucha suerte se encuentra a la primera, y por otro lado puede que este valor  $x$  no se encuentre entre los  $n$  valores, se llama a esto el peor de los casos.*

En este trabajo solo nos interesa el peor de los casos de ejecución del algoritmo. Sea  $T(n)$  el tiempo en que tarda en ejecutarse un algoritmo de tamaño  $n$ ; en el ejemplo anterior se usa un algoritmo que tarda en ejecutarse en  $T(n) = n$ , es decir,  $T(n) \in O(n)$ .

**Definición 1.3.2.** Sean  $f$  y  $g$  funciones. Se denota por  $O(g(n))$  al siguiente conjunto de funciones.

$$O(g(n)) = \{f(n) : \exists c, n_0 \in \mathbb{R} \text{ tal que } 0 \leq f(n) \leq cg(n) \text{ para todo } n \geq n_0 \}.$$

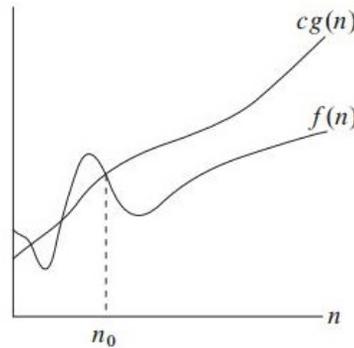


Figura 1.1: Se presenta una visión intuitiva de lo que esto significa. Figura obtenida de [6]

### 1.3.1. Órdenes de complejidad más comunes

A continuación se muestra una tabla de los órdenes de complejidad más comunes.

Orden	Nombre	Comentario
$O(1)$	Constante	Todos aquellos algoritmos que responden en un tiempo constante, sea cual sea el tamaño del problema.
$O(\text{Log}(n))$	Logaritmico	Los que el tiempo crece con un criterio logarítmico independientemente de cuál sea la base, mientras ésta sea mayor que 1.
$O(n)$	Lineal	El tiempo crece linealmente con respecto al tamaño.
$O(n^c)$	Polinomial	Aquí están muchos de los algoritmos más comunes. Intuitivamente se podrá decir que éste orden es el último de los aceptables.
$O(c^n)$	Exponencial	Aunque pudiera no parecerlo, es mucho peor que el anterior.
$O(n!)$	Factorial	Es el típico de aquellos algoritmos que para un problema complejo prueban todas las combinaciones posibles.

Tabla 1: Órdenes de complejidad más comunes.

Hay otros órdenes intermedios, e incluso superiores, pero usualmente se suelen utilizar los de la tabla de arriba, que son órdenes bastante representativos.

### 1.3.2. Representaciones de Gráficas

Cuando los datos de entrada de un algoritmo son una gráfica  $G = (V(G), E(G))$ , usualmente se mide el tamaño del conjunto de vértices  $|V(G)|$  y el tamaño del conjunto de aristas  $|E(G)|$  de la gráfica, es decir, se describe el tamaño con dos parámetros, no solo uno.

Se puede elegir entre dos formas estándares de representar una gráfica  $G = (V(G), E(G))$ , como una colección de listas de adyacencia o como una matriz de adyacencia. Estas dos representaciones se aplican a las gráficas dirigidas y a las no dirigidas. En la figura 1.2 se muestran estas dos representaciones.

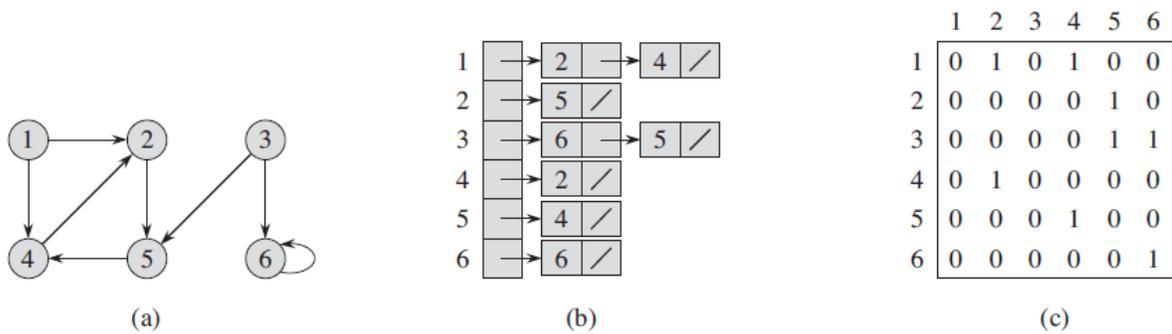


Figura 1.2: Dos representaciones de una gráfica dirigida. (a) Una gráfica dirigida  $G$  con 6 vértices y 8 aristas. (b) Una representación de lista de adyacencia de  $G$ . (c) La matriz de adyacencia correspondiente a  $G$ .

El programa donde se implementan los algoritmos *SageMath*, guarda las gráficas como una lista de adyacencia.



# Capítulo 2

## LAS GRÁFICAS DE STALLINGS

En este capítulo se hablara de las gráficas empalmadas de Stallings que son gráficas finitas, así como se hablara y describirá el proceso de empalmes de Stallings.

### 2.1. Gráficas de Stallings

La siguiente definición de  $X$ -dígrafo empalmado es importante para el proceso de empalme de Stallings.

**Definición 2.1.1. ( $X$ -dígrafo empalmado).** Sea  $\Gamma$  un  $X$ -dígrafo. Se dice que  $\Gamma$  es empalmado si para cada vértice  $v \in V(\Gamma)$  y cada letra  $a \in X$ , hay a lo más una arista en  $\Gamma$  con origen en  $v$  y etiqueta  $a$ , y hay a lo más un arista con términos en  $v$  y etiqueta  $a$ .

**Ejemplo 2.1.2.** En la figura 2.1 (izquierda) se muestra un ejemplo de un  $X$ -dígrafo no empalmado, del vértice 2 entran dos aristas con la misma etiqueta. Y en la misma figura (derecha) se muestra un  $X$ -dígrafo empalmado, no existe vértice del cual entre dos aristas con la misma etiqueta, y no existe vértice donde salgan dos aristas con la misma etiqueta.

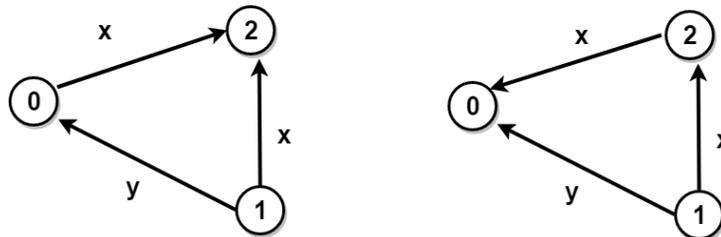


Figura 2.1: Gráfica no empalmada y una empalmada respectivamente.

Se puede observar que  $\Gamma$  es empalmado si, y solo si, para cada vértice  $v$  en  $\Gamma$  y cada  $x \in \Sigma$ , existe a lo más una arista en  $\widehat{\Gamma}$  con origen en  $v$  y etiqueta  $x$ . En un X-dígrafo empalmado el grado de cada vértice es a lo más  $2 | X |$ .

**Definición 2.1.3.** [1] Sean  $\Gamma$  y  $\Delta$  dos X-dígrafos. Entonces el mapeo  $\pi : \Gamma \rightarrow \Delta$  es llamado mapeo de X-dígrafos, donde  $\pi$  manda vértices a vértices, aristas dirigidas a aristas dirigidas, preserva las etiquetas de las aristas dirigidas y tiene las siguientes propiedades:

1.  $\circ(\pi(e)) = \pi(\circ(e))$ .
2.  $t(\pi(e)) = \pi(t(e))$ .

para cualquier  $e \in \Gamma$ .

Sea  $\Gamma$  un X-dígrafo que no es empalmado, se puede hacer un empalme, sean  $e_1$  y  $e_2$  dos aristas de  $\Gamma$  que tienen el mismo origen, es decir,  $\circ(e_1) = \circ(e_2)$  se puede “empalmar” las dos aristas formando una nueva arista con el mismo origen y la misma etiqueta, dando como resultado un nuevo X-dígrafo con una estructura natural.

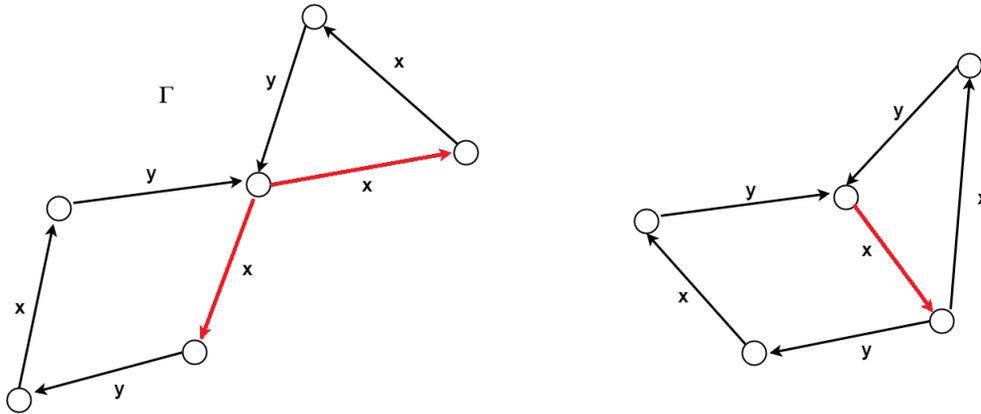


Figura 2.2: En la imagen superior se puede ver la representación de un X-dígrafo  $\Gamma$  y un empalme de  $\Gamma$

A continuación se mostrara la definición de lo que es un empalme, en un X-dígrafo.

**Definición 2.1.4.** [1] (**Folding of Graphs**). Sea  $\Gamma$  un  $X$ -dígrafo. supóngase que  $v_0 \in V(\Gamma)$  y  $f_1, f_2 \in E(\widehat{\Gamma})$ ,  $f_1 \neq f_2$  ambas aristas con origen  $v_0$ , además  $\mu(f_1) = \mu(f_2) = x \in \Sigma = X \cup X^{-1}$ . Sea  $h_i$  las aristas positivas de  $\Gamma$  correspondientes a  $f_i$  (esto es  $h_i = f_i$  si  $f_i$  es positiva, y  $h_i = f_i^{-1}$  si  $f_i$  es negativa). Dependiendo de si  $x \in X$  o  $x \in X^{-1}$  las aristas  $f_1$  y  $f_2$  son positivas o negativas en  $\widehat{\Gamma}$ .

Sea  $\Delta$  un  $X$ -dígrafo definido como sigue:

Los vértices de  $\Delta$  son los vértices de  $\Gamma$  removiendo los vértices  $t(f_1)$  y  $t(f_2)$  y agregando un nuevo vértice  $t_f$ , es decir,

$$V(\Delta) = \{\{t_f\} \cup V(\Gamma) \setminus \{t(f_1), t(f_2)\}\}$$

El conjunto de aristas de  $\Delta$  son el conjunto de aristas de  $\Gamma$  con las aristas  $h_1$  y  $h_2$  removidas y se agrega una nueva arista  $h$ , es decir,  $E(\Delta) = \{\{h\} \cup E(\Gamma) \setminus \{h_1, h_2\}\}$

Las direcciones de las aristas de  $\Delta$  serán definidas como sigue:

Si  $e \in E(\Delta)$  y  $e \neq h$  (esto es  $e \in E(\Gamma)$ ,  $e \neq h_i$ )

1.  $\circ_{\Delta}(e) = \circ_{\Gamma}(e)$  si  $\circ_{\Gamma}(e) \neq t(f_i)$       y       $\circ_{\Delta}(e) = t_f$  si  $\circ_{\Gamma}(e) = t(f_i)$  para algún  $i$ .
2.  $t_{\Delta}(e) = t_{\Gamma}(e)$  si  $t_{\Gamma}(e) \neq t(f_i)$       y       $t_{\Delta}(e) = t_f$  si  $t_{\Gamma}(e) = t(f_i)$  para algún  $i$ .

Para la arista  $h$ :

1.  $\circ_{\Delta}(h) = \circ_{\Gamma}(h_i)$  si  $\circ_{\Gamma}(h_i) \neq t(f_1), t(f_2)$       y       $\circ_{\Delta}(h) = t_f$  si  $\circ_{\Gamma}(h_i) = t(f_i)$  para algún  $i$ .
2.  $t_{\Delta}(h) = t_{\Gamma}(h_i)$  si  $t_{\Gamma}(h_i) \neq t(f_1), t(f_2)$       y       $t_{\Delta}(h) = t_f$  si  $t_{\Gamma}(h_i) = t(f_i)$  para algún  $i$ .

Las etiquetas de las aristas de  $\Delta$  se definen como sigue:  $\mu_{\Delta}(e) = \mu_{\Gamma}(e)$  si  $e \neq h$  y  $\mu_{\Delta}(h) = \mu_{\Gamma}(h_1) = \mu_{\Gamma}(h_2)$ .

Con esto  $\Delta$  es un  $X$ -dígrafo, y se dice que  $\Delta$  se obtiene mediante un solo empalme de  $\Gamma$ .

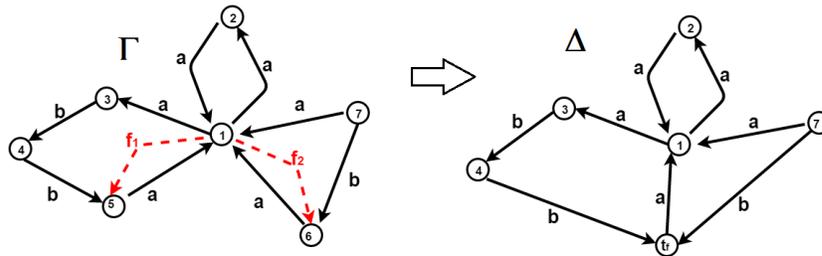


Figura 2.3: Un  $X$ -dígrafo  $\Gamma$ , y un empalme de  $\Gamma$

**Teorema 2.1.5.** [1] Sea  $\Gamma_1$  un  $X$ -dígrafo obtenido mediante un solo empalme del  $X$ -dígrafo  $\Gamma$ . Sea  $v$  un vértice de  $\Gamma$  y sea  $v_1$  el vértice correspondiente en  $\Gamma_1$ . Entonces:

1. Si  $\Gamma$  es un  $X$ -dígrafo conexo, entonces  $\Gamma_1$  es conexo.
2. Sea  $p$  un camino de  $v$  a  $v$  en  $\Gamma$  con etiqueta  $w$ . Entonces la imagen de  $p$  en  $\Gamma_1$  es un camino de  $v_1$  a  $v_1$  con etiqueta  $w$ .
3. Si  $\Gamma$  es un  $X$ -dígrafo finito, entonces  $\Gamma$  decrece en el número de aristas por 1, es decir,  $\Gamma_1$  tiene una arista menos que  $\Gamma$ .

*Demostración.* Sean  $e'_1, e'_2 \in E(\Gamma)$  las aristas que se empalman, es decir,  $e'_1$  y  $e'_2$  son las aristas que se sustituyen por una nueva arista  $e$ .

Sean  $v_1$  y  $v_2$  dos vértices en  $\Gamma$ , sean  $v'_1$  y  $v'_2$  sus vértices correspondientes en  $\Gamma_1$ . Por ser  $\Gamma$  un  $X$ -dígrafo conexo, existe un camino  $p$  en  $\Gamma$ , tal que  $\circ(p) = v_1$  y  $t(p) = v_2$ . Sea  $p = e_1 e_2 e_3 \dots e_k$  con  $e_i \in E(\Gamma)$ , si algún arista  $e'_i$  esta en  $p$ , se pueden sustituir estas aristas por  $e$ , creando un nuevo camino  $p'$  en  $\Gamma_1$ , con  $\circ(p') = v'_1$  y  $t(p') = v'_2$ . Por lo tanto  $\Gamma_1$  es un  $X$ -dígrafo conexo. Claramente se puede ver que  $\mu(p) = \mu(p')$ , con esto queda demostrado (2). (3) es claro.  $\square$

**Definición 2.1.6. (Lenguaje.)** Sea  $\Gamma$  un  $X$ -dígrafo y sea  $v \in V(\Gamma)$ , se define el lenguaje de  $\Gamma$  con respecto a  $v$  como sigue:

$$L(\Gamma, v) = \{\mu(p) : p \text{ es un camino reducido de } v \text{ a } v\}.$$

Se puede observar que no todas las palabras  $w \in L(\Gamma, v)$  son reducidas, sin embargo, si  $\Gamma$  es una gráfica empalmada, las palabras  $w \in L(\Gamma, v)$  serán libremente reducida, lo enunciaremos como sigue:

**Lema 2.1.7.** [1] Sea  $\Gamma$  un  $X$ -dígrafo empalmado. Sea  $v \in V(\Gamma)$ . Entonces todas las palabras  $w$  en  $L(\Gamma, v)$  son libremente reducidas.

*Demostración.* Supongamos que existe un palabra  $w \in L(\Gamma, v)$  que no es libremente reducida, es decir,  $w$  contiene una subpalabra de la forma  $xx^{-1}$  o  $x^{-1}x$ , se realizara el caso cuando existe una subpalabra de la forma  $xx^{-1}$  (el otro caso es similar), esto implica que existe dos aristas  $e_1$  y  $e_2$  en  $\widehat{\Gamma}$  que tienen un vértice en común  $v_0$ , con etiquetas  $\mu(e_1) = x$  y  $\mu(e_2) = x^{-1}$ . Sea  $e'_2 = e_2^{-1}$ , se tiene que  $e_1$  y  $e'_2$  son aristas de  $\widehat{\Gamma}$ , además  $\mu(e'_2) = \mu(e_1) = x$  y  $t(e_1) = t(e'_2)$  ó  $\circ(e_1) = \circ(e'_2)$ , con esto tenemos que  $\Gamma$  no es un  $X$ -dígrafo empalmado, contradiciendo nuestra hipótesis, por lo tanto todas las palabras  $w \in L(\Gamma, v)$  son libremente reducidas.  $\square$

Ahora se define el siguiente conjunto de palabras  $\overline{L(\Gamma, v)} = \{\overline{w} | w \in L(\Gamma, v)\}$ , que llamaremos el lenguaje reducido de  $\Gamma$  con respecto a  $v$ . Se puede observar que  $\overline{L(\Gamma, v)}$  contiene solo palabras reducidas, es decir, es un subconjunto de  $F(X)$ . Además  $\overline{L(\Gamma, v)}$  es un subgrupo de  $F(X)$ .

**Teorema 2.1.8.** [1] Sea  $\Gamma$  un  $X$ -dígrafo y sea  $v \in V(\Gamma)$ . Entonces el conjunto

$$\overline{L(\Gamma, v)} = \{\overline{w} \mid w \in L(\Gamma, v)\}$$

es un subgrupo de  $F(X)$

*Demostración.* Sean  $v_1, v_2 \in \overline{L(\Gamma, v)}$ , entonces existen caminos reducidos  $p_1$  y  $p_2$  que van de  $v$  a  $v$  en  $\Gamma$ , tales que  $\mu(p_1) = w_1$  y  $\mu(p_2) = w_2$  además  $\overline{w_i} = v_i$ . La concatenación de los caminos  $p_1$  y  $p_2$  nos da como resultado un camino  $p$  que va de  $v$  a  $v$ .

Sea  $p = p_1 p_2$ , si de  $p$  se eliminan todos los subcaminos de la forma  $ee^{-1}$  con  $e \in \widehat{\Gamma}$ , se obtiene un camino reducido  $p'$  que va de  $v$  a  $v$ , sea  $\mu(p') = w$  con esto se tiene que  $w \in L(\Gamma, v)$ . Si se hace la concatenación de las palabras  $w_1$  y  $w_2$ , es decir,  $w_1 w_2$  se obtiene lo siguiente  $w_1 w_2 = \mu(p_1) \mu(p_2) = \mu(p_1 p_2) = \mu(p)$ , se observa además que  $\overline{\mu(p)} = \overline{\mu(p')}$  con esto se obtiene lo siguiente  $v_1 \cdot v_2 = \overline{w_1 w_2} = \overline{\mu(p)} = \overline{\mu(p')}$  por lo tanto  $v_1 \cdot v_2 \in L(\Gamma, v)$ .

Ahora sea  $v_1 \in \overline{L(\Gamma, v)}$ , entonces existe un camino reducido  $p$  en  $\Gamma$  que va de  $v$  a  $v$ , tal que  $\mu(p) = w$  y  $\overline{w} = v_1$  como  $p$  es un camino reducido también lo es  $p^{-1}$ , además  $\mu(p^{-1}) = w^{-1}$ . Se tiene entonces que  $w^{-1} \in L(\Gamma, v)$  y  $\overline{w^{-1}} = \overline{w}^{-1} = v_1^{-1}$ , en consecuencia  $v_1^{-1} \in \overline{L(\Gamma, v)}$ . Por lo tanto  $\overline{L(\Gamma, v)}$  es un subgrupo de  $F(X)$ . □

**Lema 2.1.9.** [1] Sea  $\Gamma$  un  $X$ -dígrafo empalmado. Entonces

$$L(\Gamma, v) = \overline{L(\Gamma, v)}$$

es un subgrupo de  $F(X)$

*Demostración.* Sea  $w \in L(\Gamma, v)$ , como  $\Gamma$  es un  $X$ -dígrafo empalmado, por el lema 2.1.7 se tiene que  $w = \overline{w}$ , entonces  $w \in \overline{L(\Gamma, v)}$  esto implica  $L(\Gamma, v) \subseteq \overline{L(\Gamma, v)}$ .

Sea  $w_1 \in \overline{L(\Gamma, v)}$ , entonces existe un camino reducido  $p$  en  $\Gamma$  que va de  $v$  a  $v$ , tal que  $\mu(p) = w$  y  $w_1 = \overline{w}$  con  $w \in L(\Gamma, v)$ . Como  $p$  es un camino en  $\Gamma$  y por el lema 2.1.7 se tiene que  $w$  es libremente reducida, entonces  $w_1 = w$  de manera que  $w_1 \in L(\Gamma, v)$  que nos da la otra contención  $\overline{L(\Gamma, v)} \subseteq L(\Gamma, v)$ . Por lo tanto  $L(\Gamma, v) = \overline{L(\Gamma, v)}$ . □

Ahora enunciaremos un teorema de gran importancia en el proceso de empalmes de Stallings.

**Teorema 2.1.10.** [1] Sea  $\Gamma$  un  $X$ -dígrafo y sea  $\Gamma'$  un  $X$ -dígrafo obtenido de  $\Gamma$  mediante un solo empalme,  $v \in V(\Gamma)$  y  $v' \in V(\Gamma')$  su vértice correspondiente. Entonces

$$\overline{L(\Gamma, v)} = \overline{L(\Gamma', v')}$$

*Demostración.* Sean  $e_1$  y  $e_2$  las aristas en  $\Gamma$  que se empalman para obtener el X-dígrafo  $\Gamma'$  y  $e$  la nueva arista.

Primero demostraremos que  $\overline{L(\Gamma, v)} \subseteq \overline{L(\Gamma', v')}$ . Sea  $w_1 \in \overline{L(\Gamma, v)}$ , entonces existe un camino reducido  $p$  en  $\Gamma$  que va de  $v$  a  $v$  tal que  $\mu(p) = w$  además  $\bar{w} = w_1$ . La imagen de  $p$  en  $\Gamma'$  es un camino  $p'$  que va de  $v'$  a  $v'$  con la misma etiqueta, es decir,  $\mu(p') = w$ . Si en  $p'$  se quitan todos los subcaminos de la forma  $ee^{-1}$  con  $e \in \widehat{\Gamma}$  se obtiene un camino reducido  $p''$  en  $\Gamma'$  que va de  $v'$  a  $v'$  tal que  $\mu(p'') = \mu(p') = \mu(p) = w$  con esto tenemos que  $w_1 \in \overline{L(\Gamma', v')}$  entonces  $\overline{L(\Gamma, v)} \subseteq \overline{L(\Gamma', v')}$ .

Ahora sea  $w_1 \in \overline{L(\Gamma', v')}$ , entonces existe un camino reducido  $p'$  en  $\Gamma'$  que va de  $v'$  a  $v'$  tal que  $\mu(p') = w$  además  $\bar{w} = w_1$ . Vamos a dividir el camino  $p'$  en subcaminos  $p' = p_0 f_0 p_1 f_1 \dots f_k p_{k+1}$  donde  $f_i = e^\pm$  y los caminos  $p_i$  no contienen a la arista  $e^\pm$ , estos caminos también son caminos en  $\Gamma$ . Supongamos que para algún  $i$  se tiene que  $f_i = e$ , esto implica que  $t(p_i) = \circ(e)$  y  $\circ(p_{i+1}) = t(e)$  por definición de folgings of graph  $p_i$  y  $p_{i+1}$  están unidos por  $d_i \in \{e_1, e_2\}$ , entonces  $p_i d_i p_{i+1}$  es una camino reducido en  $\Gamma$  con la misma etiqueta que  $p_i f_i p_{i+1}$  en  $\Gamma'$ . Ahora sea  $f_i = e^{-1}$  para algún  $i$  es una caso similar al anterior, se puede encontrar  $d_i \in \{e_1^{-1}, e_2^{-1}\}$  tal que  $p_i d_i p_{i+1}$  es una camino reducido en  $\Gamma$  con la misma etiqueta que  $p_i f_i p_{i+1}$  en  $\Gamma'$ . Entonces  $p = p_0 d_0 p_1 d_1 \dots d_k p_{k+1}$  es un camino reducido que va de  $v$  a  $v$  con  $\mu(p) = w$ , por lo tanto  $\bar{w} = w_1 \in \overline{L(\Gamma, v)}$ , con esto se tiene la otra contención y se puede concluir que  $\overline{L(\Gamma, v)} = \overline{L(\Gamma', v')}$   $\square$

Sea  $H = \langle h_1, h_2, \dots, h_k \rangle < F(X)$  un subgrupo de  $F(X)$ , vamos a construir un X-dígrafo, utilizando los generadores de  $H$ , este X-dígrafo se denota por  $\Gamma_0$ , primero dibujamos un vértice base el cual se expresa por  $1_H$ , por cada palabra  $h_i$  dibujamos por separado un camino  $p_i$  con  $t(p_i) = \circ(p_i) = 1_H$  y  $\mu(p_i) = h_i$ , se puede observar que  $\Gamma_0$  tiene  $k$  círculos.

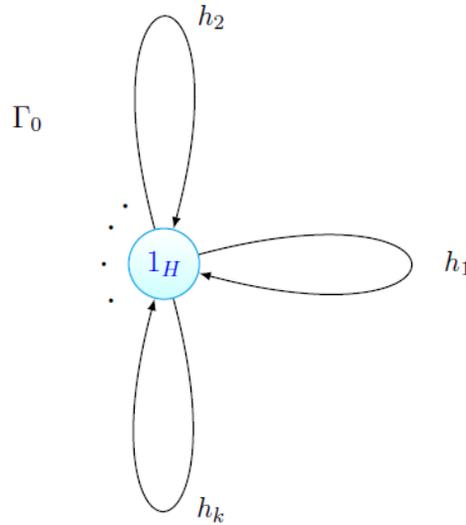


Figura 2.4: X-dígrafo  $\Gamma_0$  construido a partir de un subgrupo  $H$  finitamente generado.

La construcción de  $\Gamma_0$  la haremos más adelante por medio de un ejemplo, ahora seguiremos con el siguiente teorema.

**Teorema 2.1.11.** *Sea  $H < F(X)$  finitamente generado, con  $H = \langle h_1, h_2, h_3, \dots, h_k \rangle$ , entonces*

$$\overline{L(\Gamma_0, 1_H)} = H$$

*Demostración.* Sabemos que  $\overline{L(\Gamma_0, 1_H)}$  es un subgrupo de  $F(X)$  además  $h_i \in \overline{L(\Gamma_0, 1_H)}$ , por lo tanto  $H \leq \overline{L(\Gamma_0, 1_H)}$ .

Ahora demostraremos la otra contención, sea  $w \in \overline{L(\Gamma_0, 1_H)}$  entonces existe un camino reducido  $p$  en  $\Gamma_0$  que va de  $1_H$  a  $1_H$  con  $\mu(p) = w$  además  $\bar{v} = w$ , se considera el camino  $p = v_1 v_2 v_3 \dots v_m$ , donde  $v_i \in V(\Gamma_0)$  con  $v_i$  adyacente a  $v_{i+1}$  además  $v_1 = v_m = 1_H$ , entre estos vértices existen vértices que son igual a  $1_H$ , por lo que se puede dividir el camino  $p$  en subcaminos  $p_i$  que solo contienen al vértice  $1_H$  en los extremos, el camino  $p_i$  van de  $1_H$  a  $1_H$  por lo tanto  $p = p_1 p_2 \dots p_n$ , entonces  $\mu(p) = \mu(p_1) \mu(p_2) \mu(p_3) \dots \mu(p_n) = \overline{h_{i_1} h_{i_2} h_{i_3} \dots h_{i_n}}$ . Por lo tanto  $\overline{h_{i_1} h_{i_2} h_{i_3} \dots h_{i_n}} = \mu(p) = \bar{v} = w \in H$  con esto tenemos que  $\overline{L(\Gamma_0, 1_H)} < H$  y en consecuencia  $\overline{L(\Gamma_0, 1_H)} = H$ .  $\square$

## 2.2. Proceso de empalmes de Stallings

Describiremos el proceso de empalmes de Stallings. Sea  $H = \langle h_1, h_2, \dots, h_k \rangle$  un subgrupo finitamente generado de  $F(X)$ , vamos a construir a  $\Gamma_0$ , primero se dibuja un punto base  $1_H$ , por cada palabra  $h_i$  dibujamos por separado un camino  $p_i$  con  $t(p_i) = o(p_i) = 1_H$  y  $\mu(p_i) = h_i$  se observa que  $\Gamma_0$  tiene  $k$  círculos. Si  $\Gamma_0$  es un grafo empalmado se termina el proceso, en otro caso si  $\Gamma_0$  no es empalmado, hacemos un empalme, para obtener un nuevo grafo  $\Gamma_1$ . Continuando con este mismo proceso se obtiene una serie de empalmes, hasta llegar a un grafo empalmado  $\Gamma$ .

$$\Gamma_0 \rightarrow \Gamma_1 \rightarrow \dots \rightarrow \Gamma_{M-1} \rightarrow \Gamma_M = \Gamma$$

Este proceso se llama "Proceso de empalmes de Stallings".

Sea  $H < F(X)$  libremente generado, es decir,  $H = \langle h_1, h_2, \dots, h_k \rangle$  donde  $h_i$  son palabras libremente reducidas. Haciendo el proceso de empalmes de Stallings se obtiene una gráfica empalmada y conexa la cual se denota por  $\Gamma(H)$ , además esta gráfica es única e independiente de la secuencia de empalmes, con  $H = L(\Gamma(H), 1_H)$ , para más detalles ver [1].

**Ejemplo 2.2.1.** Sea  $H = \langle xyxy, x^2y \rangle$ . Donde  $\Gamma_3 = \Gamma(H)$  a esta gráfica se le llama la gráfica empalmada de Stallings, correspondiente a  $H$ . Se tiene que  $L(\Gamma(H), 1_H) = H$ .

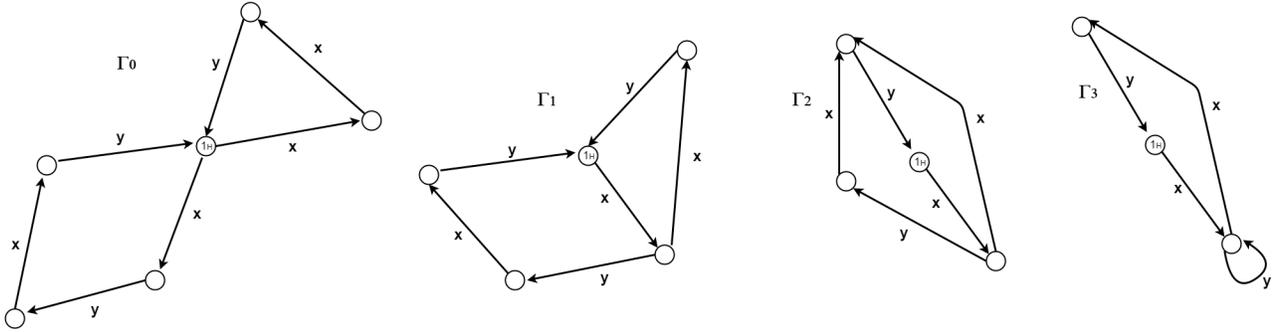


Figura 2.5: Proceso de empalmes de Stallings.

Se puede hacer la siguiente pregunta, ¿dado un  $X$ -dígrafo empalmado, este es una gráfica empalmada de Stallings para algún subgrupo finitamente generado  $H$ ? Las siguientes proposiciones nos ayudan a resolver esta pregunta.

**Proposición 2.2.2.** [1] Sea  $\Gamma$  un  $X$ -dígrafo conexo y  $v \in V(\Gamma)$ . Para cada vértice  $u \neq v$  de  $\Gamma$  elegimos un camino reducido  $p_u$  en  $\Gamma$  de  $v$  a  $u$ . Denotamos por  $p_v = v$  el camino de longitud cero, que consiste solo del vértice  $v$ . Para cada arista  $e \in E(\widehat{\Gamma})$ , se denota  $p_e = \overline{p_{o(e)}e(p_{t(e)})^{-1}}$  de modo que  $p_e$  es un camino en  $\Gamma$  que va de  $v$  a  $v$ . Se denota por  $[e] = \overline{\mu(p_e)}$ . Entonces el subgrupo  $H = L(\Gamma, v)$  de  $F(X)$  es generado por el conjunto.

$$S = \{[e] : \text{donde } e \text{ es una arista positiva en } \Gamma \}$$

En particular, si  $\Gamma$  es finita, el subgrupo  $H$  es finitamente generado.

*Demostración.* Demostraremos primero que  $\langle S \rangle \leq H$ . Sea  $[e] \in S$  entonces existe un camino  $p_e$  que va de  $v$  a  $v$  tal que  $\overline{\mu(p_e)} = [e]$ . Sea  $p'_e$  un camino reducido obtenido de  $p_e$ , entonces  $\overline{\mu(p'_e)} = \overline{\mu(p_e)}$ , con esto tenemos que  $[e] = \overline{\mu(p'_e)} \in \overline{L(\Gamma, v)} = H$  por lo tanto  $S \subseteq H$  entonces  $\langle S \rangle \leq H$ .

Queda por demostrar que cualquier elemento de  $H$  puede expresarse como el producto de elementos de  $S$  y sus inversos. Notemos que por definición  $p_{e^{-1}} = (p_e)^{-1}$  entonces  $[e^{-1}] = [e]^{-1}$ . Con esto es suficiente demostrar que cualquier elemento de  $H$  puede expresarse como el producto de elementos  $[e]$  donde  $e \in E(\widehat{\Gamma})$ .

Sea  $h \in H$  con  $h \neq 1$ . Entonces existe un camino  $p$  no trivial que va de  $v$  a  $v$  con  $\mu(p) = w$  y  $\overline{w} = h$ , sea  $p = e_1 e_2 \dots e_k$  con  $e_i \in E(\widehat{\Gamma})$  sea  $v_1 = v_{k+1} = v$  y  $\circ(e_i) = v_i$ , elegimos un camino  $p'$  de  $v$  a  $v$  de la siguiente forma  $p' = p_{e_1} p_{e_2} \dots p_{e_k} = (p_{v_1} e_1 p_{v_2}^{-1})(p_{v_2} e_2 p_{v_3}^{-1}) \dots (p_{v_k} e_k p_{v_{k+1}}^{-1})$  se puede ver que si se hacen reducciones de subcaminos de  $p'$  se puede llevar  $p'$  a  $p$ , entonces  $\overline{\mu(p')} = \overline{\mu(p)} = h$  por otro lado tenemos que  $\overline{\mu(p')} = \overline{p_{e_1} \cdot p_{e_2} \dots p_{e_k}} = [e_1][e_2] \dots [e_k] \in \langle S \rangle$  por lo tanto se tiene que  $\langle S \rangle = H$   $\square$

Si una gráfica  $T$  es un árbol, entonces para cualquiera dos vértices  $u, v$  de  $T$  existe un único camino reducido en  $T$  de  $u$  a  $v$  el cual se denota por  $[u, v]$ .

**Proposición 2.2.3.** [1] Sea  $\Gamma$  un  $X$ -dígrafo conexo empalmado y  $v \in V(\Gamma)$ . Sea  $T$  el árbol generador de  $\Gamma$  y sea  $T^+$  el conjunto de las aristas positivas de  $\Gamma$  que se encuentran fuera de  $T$ . Para cada  $e \in T^+$  se denota  $p_e = [v, \circ(e)]e[t(e), v]$  también para cada  $e \in T^+$  se denota  $[e] = \mu(p_e) = \overline{\mu(p_e)}$ . Denotemos por

$$Y_T = \{[e] \mid e \in T^+\}$$

Entonces  $Y_T$  es una base libre para el subgrupo  $H = L(\Gamma, v)$ .

*Demostración.* Vamos a extender la definición de  $p_e$  y  $[e]$  para cada arista  $e \in E(\widehat{\Gamma})$  (sea positiva o negativa, o que este o no este en  $T$ ) ponemos  $p_e = [v, \circ(e)]e[t(e), v]$  y  $[e] = \overline{\mu(p_e)}$ . Se observa que en este caso  $p_{e^{-1}} = (p_e)^{-1}$  y  $[e^{-1}] = [e]^{-1}$ . Es fácil ver que si  $e$  es una arista positiva en  $T$ , entonces  $p_e$  puede transformarse por reducción de subcaminos al camino trivial, y por lo tanto  $[e] = 1$ . Y por la proposición 2.2.2, se sabe que  $H$  es generado por el conjunto  $S = \{[e] : \text{donde } e \text{ es una arista positiva en } \Gamma \}$ , por lo tanto  $H$  es generado por el conjunto  $Y_T = \{[e] \mid e \in T^+\}$ , queda por demostrar que  $Y_T$  es una base libre de  $H$ .

Para ver que  $Y_T$  es una base libre de  $H$ , es suficiente demostrar que cualquier palabra libremente reducida no trivial en  $Y_T^{\pm 1}$  define un elemento no trivial en  $F(X)$ . Sea  $h = [e_1] \dots [e_k]$

donde  $k \geq 1$ ,  $e_i \in Y_T^+ \cup Y_T^-$  y  $e_i \neq e_{i+1}^{-1}$ . Se necesita demostrar que  $h \neq 1$ . Ya que  $T$  es un árbol para cualquiera 2 vértices  $u, u'$  de  $\Gamma$  se tiene que  $[u, v][v, u'] = [u, u']$ . Por definición de  $h$  y de  $[e_i]$  se tiene que  $h = \bar{p}$ , donde  $p$  es el siguiente camino de  $v$  a  $v$  en  $\Gamma$ .

$$p = [v, \circ(e_1)]e_1[t(e_1), v][v, \circ(e_2)]e_2 \dots [v, \circ(e_k)]e_k[t(e_k), v]$$

Se puede transformar a  $p$  en el siguiente camino mediante la afirmación de que  $[u, v][v, u'] = [u, u']$ .

$$p' = [v, \circ(e_1)]e_1[t(e_1), \circ(e_2)]e_2 \dots e_i[t(e_i), \circ(e_{i+1})]e_{i+1} \dots e_k[t(e_k), v].$$

Se observa que los caminos  $[v, \circ(e_1)]$ ,  $[t(e_k), v]$  y  $[t(e_i), \circ(e_{i+1})]$  son caminos en  $T$  y  $e_i \notin E(T)$ , entonces si  $p'$  no es un camino reducido esto implica que  $t(e_i) = \circ(e_{i+1})$  y  $e_i = e_{i+1}^{-1}$ , pero esto es imposible por nuestra suposición de  $h$ . Por lo tanto  $p'$  es un camino no reducido en  $\Gamma$ , como  $\Gamma$  es empalmado esto implica que  $\mu(p')$  es una palabra reducida en el alfabeto  $X^\pm$ , por lo que se tiene que es una palabra no trivial en  $F(X)$ , además

$$1 \neq \overline{\mu(p)} = \overline{\mu(p')} = \overline{\mu(p)} = h.$$

Con esto  $h \neq 1$ , que es lo que se quiere demostrar. □

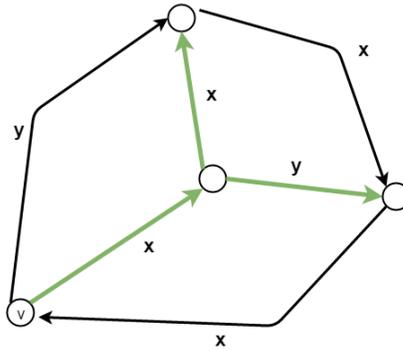


Figura 2.6: Sea  $H = L(\Gamma, v)$ , el árbol generador de  $\Gamma$  lo resaltado, entonces  $\{x^2y^{-1}, x^3y^{-1}x^{-1}, xyx\}$  es una base libre de  $H$ , es decir,  $L(\Gamma, v) = \langle x^2y^{-1}, x^3y^{-1}x^{-1}, xyx \rangle$

Si se hace el proceso de empalmes de Stallings con el subgrupo  $H = \langle x^2y^{-1}, x^3y^{-1}x^{-1}, xyx \rangle$  obtenemos como  $\Gamma(H)$  la figura 2.6

### 2.3. Algunas ideas y el algoritmo

En esta sección vamos a dar algunas ideas y observaciones para hacer más eficiente el proceso de empalmes de Stallings. Si se supone que conocemos el alfabeto con el cual se está trabajando se puede hacer más eficiente el algoritmo, por ejemplo si se tiene el grupo libre  $F(a, b)$ . A continuación se dará una definición y observaciones que nos ayudara a entender mejor el teorema 2.3.2.

**Definición 2.3.1.** *La función  $\log^* : \mathbb{N} \rightarrow \mathbb{N}$  asigna a cada número natural  $n$  el menor número natural  $k$  tal que:*

$$\underbrace{\log \circ \log \circ \dots \circ \log(n)}_{k \text{ veces}} \leq 1$$

Donde estamos usando el logaritmo base 2. Se tiene la siguiente igualdad en esta función  $\log^*(2^n) = \log^*(n) + 1$ . Se observa que  $\log^*(2^{2^{2^2}}) = \log^*(2, 10^{19728}) = 5$ . Se puede ver que  $\log^*$  crece muy lentamente, por consiguiente se puede considerar como una constante.

**Teorema 2.3.2.** *[5] Sea  $F(X)$  un grupo libre y  $H < F(X)$  un subgrupo finitamente generado. Sea  $H = \langle h_1, h_2, \dots, h_k \rangle$  donde  $n = \sum_{i=1}^k |h_i|$ . Entonces existe un algoritmo para el proceso de empalme de Stallings, donde los datos de entrada sean los generadores de  $H$  y nos dé como dato de salida  $\Gamma(H)$  en un tiempo  $O(n \cdot \log^*(n))$ .*

La demostración del teorema anterior se puede ver en [5]. Entonces se tiene un algoritmo para el proceso de empalme de Stallings que es “casi lineal”.

**Definición 2.3.3.** *Un vértice  $v \in V(\Gamma)$  lo llamamos empalmado si no existen aristas  $e_1, e_2$  en  $\Gamma$  con la misma etiqueta, y tal que  $\circ(e_1) = \circ(e_2) = v$  o  $t(e_1) = t(e_2) = v$ . En caso contrario el vértice  $v$  lo llamamos no empalmado.*

Sea  $\Gamma$  un  $X$ -dígrafo, y sea  $L$  la lista de vértices que contiene exactamente a todos los vértices no empalmados de  $\Gamma$ , esta lista la llamamos “UNFOLDED”. Se observa que en el proceso de empalmes de Stallings, cuándo se construye  $\Gamma_0$  la lista “UNFOLDED” de  $\Gamma_0$ , solo puede contener a lo más un vértice que es al vértice  $1_H$ . Se puede ver que si  $\Gamma$  es un  $X$ -dígrafo empalmado si, y solo si, todos sus vértices son vértices empalmados.

Se darán algunas observaciones que nos ayudaran a decidir cuándo un vértice es no empalmado.

**Observación 1:** Supongamos que  $\Gamma$  es un  $X$ -dígrafo y que existe un vértice  $v \in V(\Gamma)$ , tal que  $d(v) > 2 \mid X \mid$ ; entonces  $v$  es un vértice no empalmado. Para verificar si un vértice es empalmado o no, nos tomara un número limitado de operaciones. De hecho solo se necesita la lista de aristas que tienen como vértice de incidencia a  $v$ . Resultando un algoritmo de

tiempo  $O(1)$  para verificar si un vértice es empalmado o no.

**Observación 2:** Un empalme elemental es una operación esencialmente local, es decir, cada que identificamos dos aristas solo se necesita trabajar con tres vértices y sus aristas adyacentes. Cualquier vértice que no sea el vértice inicial o terminal, de alguna de las aristas en este empalme elemental, tendrán el mismo número de aristas entrantes y salientes después del empalme elemental.

**Observación 3:** Al principio solo puede haber a los más un vértice que es no empalmado, aquí se inicia nuestro bucle y el algoritmo termina cuando no hay vértices no empalmados.

### 2.3.1. Construcción de $\Gamma_0$

El algoritmo para construir a  $\Gamma_0$  es el siguiente:

Sea  $H = \langle h_1, h_2, \dots, h_k \rangle < F(X)$  un subgrupo de  $F(X)$ , primero se dibuja un vértice base el cual se denota por  $1_H$ , por cada palabra  $h_i$  se dibuja por separado un camino  $p_i$  con  $t(p_i) = o(p_i) = 1_H$  y  $\mu(p_i) = h_i$ , se observa que  $\Gamma_0$  tiene  $k$  círculos.

Sea  $H = \langle h_1, h_2, \dots, h_k \rangle$  un subgrupo finitamente generado de  $F(X)$ . El siguiente programa toma como datos de entrada los generadores de un subgrupo  $H$  y el dato de salida es el X-dígrafo  $\Gamma_0$ , ya se había hablado del tamaño de un problema, en este caso el tamaño del problema va a depender de los generadores de  $H$ , sea  $n = \sum_{i=1}^k |h_i|$ , se observa que  $n = |E(\Gamma_0)|$ .

```
def c_grafo(subgru):
    g1=DiGraph(loops=true, multiedges=true)
    g1.add_vertex(0)
    for word in subgru:
        k=g1.num_verts()
        l=len(word)
        v=range(k, k+l-1)
        v.insert(0,0)
        k1=0
        k2=mod(1, 1)
        for ltr in word:
            if ltr < 0:
                g1.add_edge(v[k2], v[k1], str(-ltr))
            else:
                g1.add_edge(v[k1], v[k2], str(ltr))
            k1=k2
            k2=mod(k2+1, 1)
    return g1
```

Figura 2.7: Código que se utilizó en SageMath para crear a  $\Gamma_0$ .

Sea subgru= $\langle h_1, h_2, \dots, h_k \rangle$ ,  $n = \sum_{i=1}^k |h_i|$ ,  $T$  el número de pasos para que el programa finalice, se tiene que en la línea 5,6,7,8,9,10 se hace un paso en cada línea, entonces se tiene que  $T = 2 + \sum_{i=1}^k (6 + 3|h_i|) = 2 + 6k + 3 \sum_{i=1}^k |h_i| = 2 + 6k + 3n \leq 9n$ , con esto se obtiene que  $T(n) \in O(n)$ , es decir, corre en tiempo lineal.

### 2.3.2. Algoritmo

Mostraremos un algoritmo <sup>1</sup> para el proceso de empalme de Stallings.

**Iniciación:** Se nos da de entrada  $h_1, h_2, h_3, \dots, h_k$  palabras reducidas en  $F(X)$  se construye a  $\Gamma_0$  en un tiempo de orden  $O(n)$ .

#### Empalmes:

Mientras la lista “UNFOLDED” sea no vacía, se hace lo siguiente, de lo contrario  $\Gamma_0$  es la gráfica empalmada de Stallings y termina el algoritmo. :

1. Sea  $v$  un vértice de la lista “UNFOLDED”. Entonces existen por lo menos 2 aristas  $e_1, e_2 \in \hat{\Gamma}$  con la misma etiqueta y tales que  $\circ(e_1) = \circ(e_2)$
2. Encontrar las aristas  $e_1, e_2 \in \hat{\Gamma}$  mencionadas en el paso 1.  
Entonces se tiene 4 posibles casos que son:

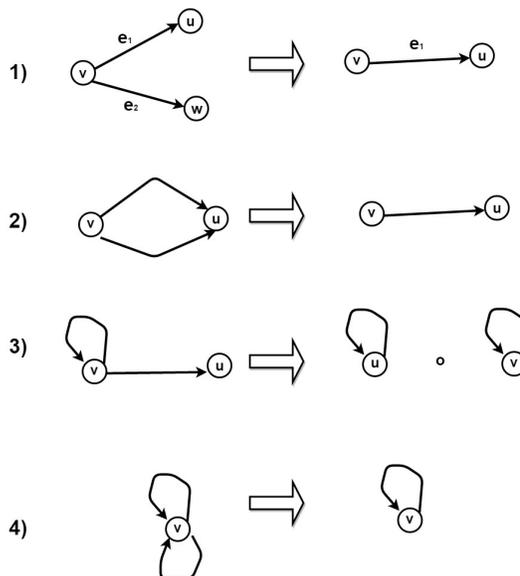


Figura 2.8: Los cuatro posibles casos para empalmar 2 aristas.

<sup>1</sup>Este algoritmo fue presentado por Nicholas W.M. Touikan en su artículo llamado “A fast algorithm for Stallings Folding Process”

Solamente se hablara del caso 1), los otros 3 casos son similares.

**CASO 1) :**

- Unimos los vértices  $u$  y  $w$ .
- Se verifica si  $w$  se encuentra en la lista UNFOLDED, y si se encuentra lo borramos de esta lista.
- Concatenamos la lista de aristas que entran y salen del vértice  $u$ , con la lista de aristas que entran y salen del vértice  $w$ .
- Se elimina la arista  $e_2$ .
- Se verifica si el vértice  $v$  es empalmado o no; si no lo es eliminarlo de la lista UNFOLDED, si sigue siendo empalmado lo dejamos en esta lista, se hace lo mismo con el vértice  $u$ .

Los otros casos son similares. Como se tiene un ciclo "WHILE", el ciclo termina cuando la lista UNFOLDED sea vacía. Cuando esto pasa ya no tenemos vértices empalmados, por lo tanto tampoco tenemos aristas que se puedan empalmar y tenemos como resultado a  $\Gamma(H)$ . En el apéndice se muestran el Pseudocódigo en SageMath que hace el proceso de empalme de Stallings.

En el siguiente capítulo se hablara de los subgrupos malnormales y ciclonormales de grupos libres, y de cómo usando las gráficas empalmadas de Stallings se puede ver si un subgrupo finitamente generado de un grupo libre es malnormal o ciclonormal.



# Capítulo 3

## SUBGRUPO MALNORMAL Y CICLONORMAL

En este capítulo se tratarán los subgrupos malnormales y ciclonormales, así como se muestran algoritmos para decidir cuándo un subgrupo de un grupo libre (el subgrupo finitamente generado) es malnormal o no, o si es ciclonormal o no, para esto se utilizan las gráficas empalmadas de Stallings.

**Definición 3.0.4.** [1] (*producto de X-dígrafos*) Sean  $\Gamma$  y  $\Delta$  X-dígrafos. Se define el producto de X-dígrafos  $\Gamma \times \Delta$  como sigue:

Los vértices de  $\Gamma \times \Delta$  es el conjunto  $V(\Gamma) \times V(\Delta)$ , es decir,  $V(\Gamma \times \Delta) = V(\Gamma) \times V(\Delta)$ , donde  $V(\Gamma) \times V(\Delta)$  es el producto cartesiano de los conjuntos de vértices de  $\Gamma$  y  $\Delta$  respectivamente. Para cada par de vértices  $(v, u), (v', u') \in V(\Gamma \times \Delta)$  ( $v, v' \in V(\Gamma)$  y  $u, u' \in V(\Delta)$  además  $x \in X$ ), se introduce una arista con etiqueta  $x$ , con origen en  $(v, u)$  y final  $(v', u')$  si existen aristas que cumplen lo siguiente:

1.  $e_1 \in E(\Gamma)$ , tal que  $\circ(e_1) = v$ ,  $t(e_1) = v'$ ,  $\mu(e_1) = x$ .
2.  $e_2 \in E(\Delta)$ , tal que  $\circ(e_2) = u$ ,  $t(e_2) = u'$ ,  $\mu(e_2) = x$ .

**Teorema 3.0.5.** [1] Sean  $\Gamma$  y  $\Delta$  dos X-dígrafos empalmados. Entonces  $\Gamma \times \Delta$  también es un X-dígrafo empalmado.

*Demostración.* Supongamos que  $\Gamma \times \Delta$  no es empalmado, entonces existe un vértice  $(v, u) \in \Gamma \times \Delta$  del cual salen o entran 2 aristas con la misma etiqueta, sin pérdida de generalidad se puede decir que existen dos aristas  $e_1^*, e_2^* \in E(\Gamma \times \Delta)$ , tal que  $\circ(e_1^*) = \circ(e_2^*) = (v, u)$  y  $\mu(e_1^*) = \mu(e_2^*) = x$  con  $x \in X$ , definamos además  $t(e_1^*) = (v_1, u_1)$  y  $t(e_2^*) = (v_2, u_2)$ . Por definición de  $\Gamma \times \Delta$  existen aristas  $e_1, e_2 \in E(\Gamma)$  tales que  $\circ(e_1) = v$ ,  $\circ(e_2) = v$  con  $\mu(e_1) = \mu(e_2) = x$ , esto implica que  $\Gamma$  no es un X-dígrafo empalmado contradiciendo nuestra hipótesis, por lo tanto  $\Gamma \times \Delta$  es empalmado.  $\square$

El siguiente programa fue realizado en SageMath y hace el producto de 2 X-dígrafos.

```
def prodT(g1, g2)
    g3=DiGraph(loops=true, multiedges=true)
    for i in g1.edges():
        for j in g2.edges():
            if i[2]=j[2]:
                g3.add_edge((i[0], j[0]), (i[1], j[1]), i[2])
    return g3
```

Se ejecuta este programa con  $\text{prodT}(\Gamma(H), \Gamma(H))$ , sea  $n = |E(\Gamma(H))|$ . En el segundo renglón se hace un paso, dentro de los ciclos se hacen 2 operaciones, por lo tanto  $T(n) = 1 + 2n^2$ , con esto se tiene que  $T(n) \in O(n^2)$ , es decir, su tiempo es de orden cuadrático.

### 3.1. Subgrupo Malnormal

En esta sección se hablara de los subgrupos malnormales de los grupos libres, se trataran algunas proposiciones y después un teorema que ayuda a realizar un algoritmo, con el cual se puede decidir cuándo un subgrupo  $H < F(X)$  es malnormal o no.

**Definición 3.1.1.** *Sea  $H$  un subgrupo de un grupo  $G$ . Se dice que  $H$  es un subgrupo Malnormal de  $G$  si para cualquier  $g \in G \setminus H$  ocurre lo siguiente:*

$$H^g \cap H = 1$$

donde  $H^g = \{ghg^{-1} | h \in H\}$

**Proposición 3.1.2.** *Sea  $H, K < F(X)$  dos subgrupos de  $F(X)$ . Sea  $g \in F(X)$  tal que las clases laterales  $KgH$  y  $KH$  son distintas. Supongamos que  $gHg^{-1} \cap K \neq 1$ .*

*Entonces existe un vértice  $v \times u$  en  $\Gamma(H) \times \Gamma(K)$  que no pertenece al componente conexo que contiene el vértice  $1_H \times 1_K$ , tal que el subgrupo  $L(\Gamma(H) \times \Gamma(K), v \times u)$  es conjugado con  $gHg^{-1} \cap K$  en  $F(X)$ .*

*Demostración.* Ver [1] □

**Proposición 3.1.3.** *Sea  $H, K < F(X)$  dos subgrupos de  $F(X)$  y  $g \in F(X)$ . Entonces para algún vértice  $v \times u \in V(\Gamma(H) \times \Gamma(K))$  el subgrupo  $L(\Gamma(H) \times \Gamma(K), v \times u)$  es conjugado a un subgrupo de la forma  $gHg^{-1} \cap K$  para algún  $g \in F(X)$ . Además si  $v \times u$  no pertenece al componente conexo que contiene el vértice  $1_H \times 1_K$ , entonces el elemento  $g$  puede ser elegido de modo que  $KgH \neq KH$ .*

*Demostración.* Ver [1] □

A continuación se enuncia un teorema que nos ayuda a decidir cuándo un subgrupo de un grupo libre es malnormal, utilizando las gráficas empalmadas de Stallings.

**Teorema 3.1.4.** [1] *Sea  $H \leq F(X)$  un subgrupo. Entonces  $H$  es malnormal en  $F(X)$  si, y solo si, todos los componentes en  $\Gamma(H) \times \Gamma(H)$  que no contenga al vértice  $1_H \times 1_H$  son árboles.*

*Demostración.* Sea  $H$  un subgrupo malnormal en  $F(X)$ . Supongamos que existe un componente  $C$  en  $\Gamma(H) \times \Gamma(H)$  el cual no es un árbol y no contiene al vértice  $1_H \times 1_H$ , sea  $v \times u$  un vértice en  $C$ , entonces como  $C$  no es un árbol existe un camino  $p$  no trivial en  $C$  que es reducido que va de  $v \times u$  a  $v \times u$ . Definamos  $A = L(C, v \times u) \neq 1$ , entonces por la proposición 3.1.3 existe un  $g \in F(X)$  tal que  $HgH \neq HH = H$  (tal que  $g \in F \setminus H$ ), además  $H^g \cap H$  es conjugado con el subgrupo  $A$ , esto implica que  $H^g \cap H \neq 1$  contradiciendo que  $H$  es malnormal.

Sean todos los componentes conexos de  $\Gamma(H) \times \Gamma(H)$  árboles, entonces para cualquier componente  $C$  y cualquier vértice de  $C$  tenemos  $L(C, v) = 1$ , la implicación opuesta del teorema es obvia y se sigue de la proposición 3.1.2.  $\square$

**Teorema 3.1.5.** [1] *Existe un algoritmo que, si se tiene un conjunto de palabras reducidas  $h_1, h_2, \dots, h_k$  en  $F(X)$  decide cuándo o no el subgrupo  $H = \langle h_1, h_2, \dots, h_k \rangle$  es malnormal en  $F(X)$  o si  $H$  no es malnormal.*

*Demostración.* Primero construimos la gráfica empalmada de Stallings  $\Gamma(H)$ , esto lo se hace mediante el proceso de empalmes de Stallings, como  $H$  es finitamente generada entonces  $\Gamma(H)$  es finita y  $\Gamma(H) \times \Gamma(H)$  también es finita, se construye el X-dígrafo  $\Gamma(H) \times \Gamma(H)$ , después checamos que cada uno de los componentes que no contienen al vértice  $1_H \times 1_H$  sean árboles, si todos los componentes son árboles entonces  $H$  es malnormal. Si al menos existe un componente que no contiene al vértice  $1_H \times 1_H$  y además que no sea un árbol, entonces  $H$  no es malnormal.  $\square$

Con el teorema anterior se puede decidir cuándo un subgrupo  $H$  finitamente generado de un grupo libre  $F(X)$  es malnormal o no.

**Ejemplo 3.1.6.** *Sea  $H = \langle xyxy, x^2y \rangle$  un subgrupo del grupo libre  $F$ ,  $H$  no es malnormal. Construimos  $\Gamma(H)$  y  $\Gamma(H) \times \Gamma(H)$ .*

*En la figura 3.1 se puede ver que en  $\Gamma(H) \times \Gamma(H)$  existe un componente que no es un árbol, por lo tanto el subgrupo  $H = \langle xyxy, x^2y \rangle$  no es malnormal.*

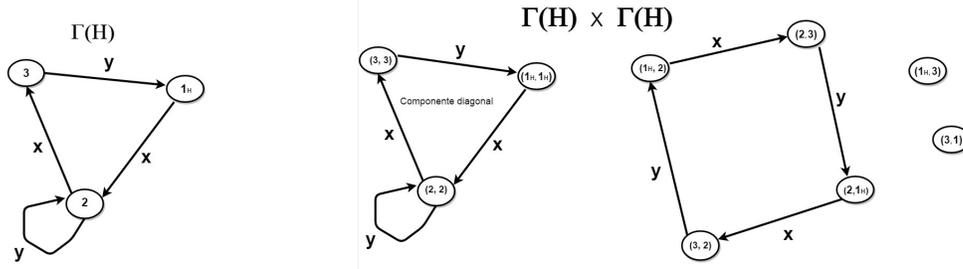


Figura 3.1:  $\Gamma(H)$  y  $\Gamma(H) \times \Gamma(H)$  correspondiente al subgrupo  $H = \langle xyxy, x^2y \rangle$ .

### 3.1.1. Programa Subgrupo Malnormal

El siguiente programa decide cuando un subgrupo finitamente generado de un grupo libre es malnormal o no. Se analizara solo el programa de “Malnormal”, donde dentro de este se encuentran unos subprogramas de los que ya se han hablado; donde Subgru es un conjunto de las palabras libres generadoras de  $H$ , es decir,  $\text{subgru} = \{h_1, h_2, \dots, h_k\}$ .

```
# PROGRAMA PARA VER SI UN SUBGRUPO ES MALNORMAL#

def malnormal(subgru):
    H=c_grafo(subgru)
    HH=empalmar(H)
    g2=prodT(HH,HH)
    g3=g2.to_undirected()
    for component in g3.connected_components():
        g4=g3.subgraph(component)
        if not g4.has_vertex((0,0)):
            if not g4.is_tree():
                print 'El subgrupo no es Malnormal'
                return
    print 'El subgrupo es Malnormal'
    return
```

Figura 3.2: Código que se utilizó en SageMath para la implementación del algoritmo mencionado en el teorema 3.1.5.

En la segunda línea se crea el X-dígrafo  $\Gamma_0$ , en la tercer línea se hace el proceso de empalmes de Stallings y se encuentra  $\Gamma(H)$ , en la línea 4 se crea el X-dígrafo  $\Gamma(H) \times \Gamma(H)$ , en la línea 5 lo que se hace es convertir a  $\Gamma(H) \times \Gamma(H)$  en un grafo no dirigido, esto con un comando de SageMath. En el programa se tiene un ciclo, que se repite a lo más  $n(n-1)$  veces donde  $n = \sum_{i=0}^k |h_i|$ , dentro del ciclo se tiene los comandos de SageMath “ $g4.has\_vertex((0,0))$ ”

y “g4.is tree” que tienen un tiempo lineal y  $O(|E| + |V|)$  respectivamente. Por lo tanto el programa para ver si un subgrupo  $H$  es malnormal o no, tiene un tiempo de complejidad de orden polinomial.

## 3.2. Subgrupo Ciclonormal

Ahora se trataran los subgrupos ciclonormales de un grupo libre.

**Definición 3.2.1.** *Sea  $H$  un subgrupo de un grupo  $G$ . Se dice que  $H$  es un subgrupo Ciclonormal de  $G$  si para cualquier  $g \in G \setminus H$  ocurre lo siguiente:*

$$H^g \cap H \text{ es cíclico}$$

donde  $H^g = \{ghg^{-1} | h \in H\}$ .

**Teorema 3.2.2.** [1] *Existe un algoritmo que dado un conjunto de palabras reducidas  $h_1, h_2, \dots, h_k$  en  $F(X)$ , decide cuándo o no el subgrupo  $H = \langle h_1, h_2, \dots, h_k \rangle$  es ciclonormal en  $F(X)$ .*

*Demostración.* Esto se sigue de la proposición 3.1.2 que si  $g \in F \setminus H$  entonces  $gHg^{-1} \cap H$  es trivial o isomorfo a  $L(C, v \times u)$ , donde  $C$  es un componente conexo de  $\Gamma(H) \times \Gamma(H)$  que no contiene al vértice  $1_H \times 1_H$ , además  $v \times u$  es un vértice de  $C$ . Se observa que el tipo de isomorfismo de  $L(C, v \times u)$  solo depende de  $C$  y no de la elección del vértice  $v \times u$  en  $C$ . Para comprobar si  $H$  es ciclonormal se construye primero  $\Gamma(H)$ , después se construye  $\Gamma(H) \times \Gamma(H)$ , por cada componente conexo  $C$  de  $\Gamma(H) \times \Gamma(H)$  que no contenga al vértice  $1_H \times 1_H$  se construye su árbol generador  $T_C$  en  $C$ . Sea  $n_C$  el número de arista positivas en  $C - T_C$ . Este número es igual al rango de  $L(C, v \times u)$ . Si para cada  $n_C$  se tiene que  $n_C \leq 1$  el subgrupo  $H$  es ciclonormal. Si para algún  $C$  se tiene que  $n_C \geq 2$  el subgrupo  $H$  no es ciclonormal.  $\square$

### 3.2.1. Programa Subgrupo Ciclonormal

El siguiente programa, decide si un subgrupo finitamente generado de un grupo libre es ciclonormal o no.

```
##PROGRAMA PARA VER SI UN SUBGRUPO ES CYCLONORMAL##

def cyclonormal(subgru):
    H=c_grafo(subgru)
    HH=empalmar(H)
    g2=prodT(HH,HH)
    g3=g2.to_undirected()
    for component in g3.connected_components():
        g4=g3.subgraph(component)
        S=g4.min_spanning_tree()
        a=len(S)
        b=g4.size()
        if not g4.has_vertex((0,0)):
            if b-a>=2:
                print 'H_no_es_ciclonormal'
                return 1
    print 'H_es_Ciclonormal'
    return 0
```

Figura 3.3: Código que se utilizó en SageMath para la implementación del algoritmo mencionado en el teorema 3.2.2.

Este programa tiene un tiempo de complejidad de orden polinomial.

# Capítulo 4

## S-SUBGRUPO

En este capítulo se tratarán un tipo especial de subgrupos de grupos libres. Se mostrará un algoritmo, así como un programa que nos ayuda a decidir cuándo un subgrupo finitamente generado es un S-subgrupo en  $F$ . Los S-subgrupos no tienen ACEP (Almost Congruence Extension Property), para más detalles ver [2].

### 4.1. S-Subgrupo

**Definición 4.1.1.** Sea  $H$  un subgrupo de un grupo libre  $F$ . Se dice que  $H$  es un S-subgrupo de  $F$  (notación:  $H <_S F$ ), si existe un  $a \notin H$  y  $w \in H$  tal que  $awa^{-1} \in H$ , y para todo  $b \in H$  se tiene que  $awa^{-1} \neq bwb^{-1}$ .

Sea  $H$  un subgrupo y  $a, w \in H$ . Se usará la siguiente notación  $w^a = \{a^{-1}wa\}$  y  $w^H = \{h^{-1}wh \mid h \in H\}$ , es decir,  $w^H$  son las clases de conjugación de  $w$  en  $H$ .

La siguiente definición que es equivalente a 4.1.1.

**Definición 4.1.2.** Sea  $H$  un subgrupo de un grupo libre  $F$ . Se dice que  $H$  es un S-subgrupo de  $F$ . Si existe un  $w \in H$ , tal que  $w^H = w^F \cap H$ .

Se exponen algunos teoremas que ayudan a decidir, cuando un subgrupo finitamente generado de un grupo libre es S-subgrupo o no lo es.

**Teorema 4.1.3.** [2] Sea  $H < F$ , con  $a \in F \setminus H$ . Si existe  $u = awa^{-1} \in H \cap H^a$  tal que  $w$  no es una potencia propia en  $F$ . Entonces  $H$  es un S-subgrupo.

*Demostración.* Supongamos que  $H$  no es S-subgrupo, entonces existe un  $b \in H$  tal que:

$$\begin{aligned} bwb^{-1} &= awa^{-1} \\ w &= b^{-1}awa^{-1}b \\ w &= (b^{-1}a)w(b^{-1}a)^{-1} \\ w(b^{-1}a) &= (b^{-1}a)w \end{aligned}$$

Es decir,  $w$  y  $b^{-1}a$  conmutan, pero  $F$  es un grupo libre, esto solo pasa si  $w$  y  $b^{-1}a$  son potencias propias del mismo elemento, es decir,  $w = r^{n_1}$  y  $b^{-1}a = r^{n_2}$ . Pero  $w$  no es una potencia propia en  $F$  esto implica que  $w = r$ , por lo tanto

$$\begin{aligned} b^{-1}a &= w^{n_2} \\ a &= bw^{n_2} \in H \end{aligned}$$

Esto implica que  $a \in H$ , contradiciendo que  $a \in F \setminus H$ . Por lo que se tiene que  $H$  es un S-subgrupo.  $\square$

**Teorema 4.1.4.** [2] Sea  $F$  un grupo libre, y  $H$  un subgrupo de  $F$  tal que  $\text{rank}(H \cap H^a) \geq 2$  para algún  $a \in F \setminus H$ . Entonces  $H$  es un S-subgrupo.

*Demostración.* Sea  $\{u, v\}$  un subconjunto de los generadores libres de  $H \cap H^a$ , se demostrara que un elemento en  $\{u, v, uv\}$ , no es una potencia propia en  $F$

Supongamos que todos son potencias propias de algún elemento en  $F$ . Sean  $u = x^n, v = y^m$  y  $uv = z^p$ . Se tiene la siguiente ecuación sobre un grupo libre  $F$   $x^n y^m = z^p$ , las soluciones de este tipo de ecuaciones son potencias propias del mismo elemento, ver [3]. Esto implica que  $\langle u, v \rangle$  es cíclico. Por lo tanto un elemento en  $\{u, v, uv\}$  no es una potencia propia en  $F$ .

Sea  $h \in \{u, v, uv\}$  tal que  $h$  no es potencia propia en  $F$ , además  $h \in H \cap H^a$ , esto implica que  $h = awa^{-1}$  donde también  $w$  no es una potencia propia en  $F$ . Por el teorema anterior se tiene que  $H$  es un S-subgrupo.  $\square$

**Teorema 4.1.5.** [2] Sea  $F$  un grupo libre y  $H$  un subgrupo de  $F$  tal que  $H$  no es ciclonormal. Entonces  $H$  es un S-subgrupo.

*Demostración.*  $H$  no es ciclonormal, esto implica que existe  $a \in F \setminus H$  tal que  $H \cap H^a$  no es cíclico, con esto se tiene que  $\text{rank}(H \cap H^a) \geq 2$ . Por el teorema 4.1.4 se tiene que  $H$  es un S-subgrupo.  $\square$

El teorema anterior nos ayuda a decir cuando un subgrupo es un S-subgrupo. Ya se tiene un algoritmo que nos dice cuando un subgrupo  $H$  de  $F$  es ciclonormal o no. Entonces solamente nos quedamos con subgrupos  $H$  que son ciclonormales. Por lo tanto la pregunta es, ¿ todos los subgrupos ciclonormales, no son S-subgrupos? la respuesta es NO, existen subgrupos ciclonormales que son S-subgrupos, y existen otros que no son S-subgrupos.

Sea  $\Gamma$  un X-dígrafo. Se denota por  $P(u, v, w)$  el camino que va del vértice  $u$  al vértice  $v$  con etiqueta  $w$ .

**Proposición 4.1.6.** *Sea  $F$  un grupo libre  $H < F$ , y  $\Gamma(H)$  la gráfica empalmada de Stallings correspondiente a  $H$ . Entonces  $H$  es un S-subgrupo si, y solo si, existen caminos cíclicamente reducidos  $P(v, v, w)$  y  $P(v', v', w)$  en  $\Gamma(H)$ , que son ciclos reducidos que van de  $v$  a  $v$  y de  $v'$  a  $v'$  respectivamente, estos ciclos con la misma etiqueta  $w$ , tal que  $P(v, v, w)$  no es permutación cíclica de  $P(v', v', w)$*

*Demostración.* Ver [2] □

Sea  $H$  un subgrupo finitamente generado de un grupo libre. La proposición 4.1.6 <sup>1</sup> nos ayuda a decidir cuándo un subgrupo ciclonormal, es un S-subgrupo o no es S-subgrupo.

**Ejemplo 4.1.7.** *Sea  $H = \langle xyxy, x^2y \rangle$  un subgrupo del grupo libre  $F$ ,  $H$  es ciclonormal pero no es un S-subgrupo.*

*Se puede ver en la figura 4.1, que en  $\Gamma(H)$  existen caminos cíclicamente reducidos diferentes con la misma etiqueta, pero uno es permutación cíclica del otro  $P(1_H, 1_H, xyxy)$  y  $P(v, v, xyxy)$ .*

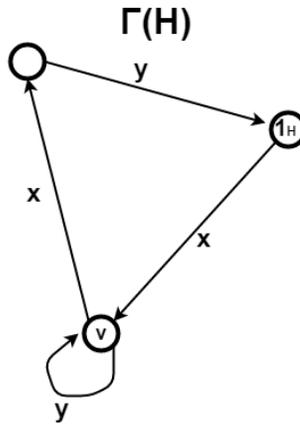


Figura 4.1: Gráfica empalmada de Stallings correspondiente a  $H = \langle xyxy, x^2y \rangle$ .

Sea  $H$  un subgrupo del grupo libre  $F$ ,  $\Gamma(H)$  su gráfica empalmada de Stallings correspondiente a  $H$ , si en  $\Gamma(H)$  existen 2 caminos cíclicamente reducidos con la misma etiqueta  $w$ , entonces existe un ciclo en un componente  $C$  en  $\Gamma(H) \times \Gamma(H)$ , tal componente no contiene el vértice  $1_H \times 1_H$ , esto por la definición del producto de X-dígrafos. Entonces si se quiere ver la existencia de estos caminos cíclicamente reducidos, basta hacer el producto de  $\Gamma(H) \times \Gamma(H)$ .

<sup>1</sup>Esta proposición es un resultado de la tesis doctoral de Nevarez Nieto Saúl y el Dr. Glesky Lev.

El objetivo principal es hacer un algoritmo y un programa en SageMath, que ayude a decidir cuándo un subgrupo finitamente generado de un grupo libre es un S-subgrupo, esto utilizando las gráficas empalmadas de Stallings.

**Teorema 4.1.8.** *Existe un algoritmo, que dado un conjunto de palabras reducidas  $h_1, h_2, \dots, h_k$  en  $F(X)$ , decide cuándo o no el subgrupo  $H = \langle h_1, h_2, \dots, h_k \rangle$  es S-subgrupo en  $F(X)$ .*

*Demostración.* Esto se sigue de la proposición 4.1.6.

Verificar que  $H$  sea ciclonormal, si no es ciclonormal entonces  $H$  es un S-subgrupo. Si  $H$  es ciclonormal se construye la gráfica empalmada de Stallings  $\Gamma(H)$ , esto se hace mediante el proceso de empalmes de Stallings, como  $H$  es finitamente generada entonces  $\Gamma(H)$  es finita, y  $\Gamma(H) \times \Gamma(H)$  también es finita; se construye el X-dígrafo  $\Gamma(H) \times \Gamma(H)$ .

Cada componente  $C$  de  $\Gamma(H) \times \Gamma(H)$  que no contenga al vértice  $1_H \times 1_H$  tienen a lo mas un ciclo reducido  $C'$ , cada ciclo reducido  $C'$  proviene de dos ciclos  $C_1$  y  $C_2$  de  $\Gamma(H)$ ,  $C_1$  y  $C_2$  tienen la misma etiqueta. Si por cada ciclo reducido  $C'$  se tiene que  $C_1 = C_2$ , entonces  $H$  no es S-subgrupo. Si por lo menos para algún ciclo reducido  $C'$  se tiene que  $C_1 \neq C_2$ , entonces  $H$  es un S-subgrupo.  $\square$

### 4.1.1. Programa S-subgrupo

Se hicieron dos programas, uno de ellos deja solo los ciclos reducidos que existen en los componentes de  $\Gamma(H) \times \Gamma(H)$ , el otro programa encuentra los ciclos  $C_1$  y  $C_2$  en  $\Gamma(H)$ , los compara y arroja como resultado si son iguales o no.

Sea  $C$  un componente de  $\Gamma(H) \times \Gamma(H)$  que no contiene al vértice  $1_H \times 1_H$ . El componente  $C$  tiene a lo más un ciclo reducido. Se define  $|V(C)| = n_1$ , el siguiente programa de “Quitar”, tiene un tiempo de orden  $O(n_1)$ .

```

def Quitar(C):
    i=0
    while i==0:
        i=1
        for j in C.vertices():
            l=C.degree(j)
            if l<=1:
                C.delete_vertex(j)
                i=0
    return C

```

Figura 4.2: Programa que deja solo los ciclos reducidos que existen en los componentes de  $\Gamma(H) \times \Gamma(H)$ .

Sea  $CC$  un ciclo reducido en  $\Gamma(H) \times \Gamma(H)$ , Se sabe que  $CC$  proviene de dos ciclos  $C_1$  y  $C_2$  en  $\Gamma(H)$  con la misma etiqueta, el siguiente programa (figura 4.3) nos dice si  $C_1$  y  $C_2$  son los mismos o no. Donde  $|V(CC)| = |E(CC)| = n_2$ ; este programa tiene un tiempo de  $O(n_2)$ .

```

def preimagen(CC):
    if CC.order() <= 2:
        LL=CC.vertices()
        z_list=LL
        x_tup, y_tup=zip(*z_list)
        x_list=list(x_tup); y_list=list(y_tup)
        C1=DiGraph(loops=true, multiedges=true)
        C2=DiGraph(loops=true, multiedges=true)
        i=0
        a=len(x_list)
        while i < a:
            C1.add_edge(x_list[i-1], x_list[i])
            C2.add_edge(y_list[i-1], y_list[i])
            i=i+1

    if CC.order() > 2:
        LL=CC.cycle_basis()
        z_list=LL[0]
        x_tup, y_tup=zip(*z_list)
        x_list=list(x_tup); y_list=list(y_tup)
        C1=DiGraph(loops=true, multiedges=true)
        C2=DiGraph(loops=true, multiedges=true)
        i=0
        a=len(x_list)
        while i < a:
            C1.add_edge(x_list[i-1], x_list[i])
            C2.add_edge(y_list[i-1], y_list[i])
            i=i+1

    return C1==C2

```

Figura 4.3: Programa que encuentra los ciclos  $C_1$  y  $C_2$  en  $\Gamma(H)$ , los compara y arroja como resultado si son iguales o no.

Estos dos programas anteriores los vamos a unir en uno solo. El nuevo programa (figura 4.4) nos ayuda a decidir cuándo un subgrupo  $H = \langle h_1, h_2, h_3, \dots, h_k \rangle$  de  $F(X)$ , es S-subgrupo o no es S-subgrupo. Este programa tiene un tiempo de complejidad de orden polinomial.

```

def Sgrupo(subgru):
    if cyclonormal(subgru)==0:
        H=c_grafo(subgru)
        HH=empalmar(H)
        g2=prodT(HH,HH)
        g3=g2.to_undirected()
        g4=Quitar(g3)
        for component in g4.connected_components():
            g5=g4.subgraph(component)
            if not g5.has_vertex((0,0)):
                if not preimagen(g5):
                    print 'ciclos diferentes, no es S-subgrupo'
                    return
            print 'ciclos iguales, no es S-subgrupo'
        return
    print 'es S-subgrupo'
    return

```

Figura 4.4: Programa que decide cuándo un subgrupo  $H$  es un S-subgrupo o no.



# Conclusiones

Se puede ver que las gráficas de Stallings se han convertido en una representación de los subgrupos de los grupos libres, con las gráficas de Stallings se pueden resolver problemas por medio de algoritmos, tal como hemos visto en este trabajo. Se ha estudiado el tiempo de complejidad del proceso de empalme de Stallings como lo ha hecho Nicholas W. M. Touikan en [5], así como también Carmel Levy [7]; Levy hace una conjetura sobre el orden de complejidad del proceso de empalmes de Stallings. Un camino para continuar el trabajo es bajar el orden de complejidad, esto se podría implementar usando el algoritmo que utiliza Nicholas W. M. Touikan y conociendo el rango del grupo libre  $F(X)$  el cual tendrá como subgrupo a  $H$ .

En el programa Sgrupo ayuda a decidir si un subgrupo  $H$  es S-subgrupo o no, pero se puede ver que dentro de dicho programa se encuentra otro que es el de cyclonormal, con esto se tiene que el proceso de empalmes de Stallings se realiza dos veces en el programa S-subgrupo. Una tarea podría ser hacer un algoritmo para verificar cuando un subgrupo  $H$  de  $F(X)$ , es un S-subgrupo utilizando las gráficas de Stallings y la proposición 4.1.6, sin la necesidad de utilizar que  $H$  sea cyclonormal o no.



# APENDICE

A continuación se muestran los códigos que se utilizaron en **SageMath** para la implementación de los algoritmos mencionados en este trabajo.

El código para construir a  $\Gamma_0$  es el siguiente:

```
def c_grafo(subgru):
    g1=DiGraph(loops=true, multiedges=true)
    g1.add_vertex(0)
    for word in subgru:
        k=g1.num_verts()
        l=len(word)
        v=range(k, k+l-1)
        v.insert(0,0)
        k1=0
        k2=mod(1, l)
        for ltr in word:
            if ltr < 0:
                g1.add_edge(v[k2], v[k1], str(-ltr))
            else:
                g1.add_edge(v[k1], v[k2], str(ltr))
            k1=k2
            k2=mod(k2+1, l)
    return g1
```

Se tiene un algoritmo que nos ayuda a encontrar la gráfica empalmada de Stallings, correspondiente a un subgrupo  $H$  finitamente generado de un grupo libre  $F(X)$ , este algoritmo es el proceso de empalmes de Stallings.

Se realizó un programa en SageMath, el cual hace todos los posibles empalmes de un  $X$ -dígrafo. Se puede ver parte del código en la siguiente página.

Código en SageMath que realiza todos los posibles empalmes de un X-dígrafo  $\Gamma$ .

```

#Empalme de aristas salientes

def empalme_saliente(G):
    for i in G.vertices():
        l=G.outgoing_edges(i)
        for ii in l:
            l.remove(ii)
            for iii in l:
                a=ii; b=iii
                if a[2]==b[2]:
                    ll=G.incoming_edges(b[1])
                    lll=G.outgoing_edges(b[1])
                    for s in ll:
                        if not((s[0], a[1], s[2]) in G.edges()):
                            G.add_edge(s[0], a[1], s[2])
                    for s in lll:
                        if not((a[1], s[1], s[2]) in G.edges()):
                            G.add_edge(a[1], s[1], s[2])
                    G.delete_vertex(b[1])
                return G
    return G
*****
#Empalme de aristas entrantes

def empalme_entrante(G):
    for i in G.vertices():
        l=G.incoming_edges(i)
        for ii in l:
            l.remove(ii)
            for iii in l:
                a=ii; b=iii
                if a[2]==b[2]:
                    ll=G.incoming_edges(b[0])
                    lll=G.outgoing_edges(b[0])
                    for s in ll:
                        if not((s[0], a[0], s[2]) in G.edges()):
                            G.add_edge(s[0], a[0], s[2])
                    for s in lll:
                        if not((a[0], s[1], s[2]) in G.edges()):
                            G.add_edge(a[0], s[1], s[2])
                    G.delete_vertex(b[0])
                return G
    return G
*****
# Empalmar un grafo

def empalmar(G):
    f=0
    while f<2:
        g_n=copy(G)
        G=empalme_saliente(G)
        G=empalme_entrante(G)
        if g_n==G:
            break;
    return G

```

Si unimos estos dos programas, se tiene como resultado el proceso de empalmes de Stallings.

# Bibliografía

- [1] ILYA KAPOVICH AND ALEXEI MYASNIKOV, “Stalling Foldings and Subgroups of Free Groups”. Journal of Algebra Volume 248, Issue 2, 15 February 2002, Pages 608-668.
- [2] LEV GLEBSKY; NEVAREZ NIETO SAUL, “Almost Congruence Extension Property for subgroups of free groups”. Journal of group theory. Por publicar.
- [3] R.C. LYNDON AND M.P. SCHUTZENBERGER , “The equation  $a^m = b^n c^p$  in a free group”. Michigan Math. J., vol. 9, pp. 289-298, 1962.
- [4] OLEG BOGOPOLSKI, “Introduction to Group Theory”. European Mathematical Society, 2002.
- [5] NICHOLAS W. M. TOUIKAN., “A Fast Algorithm for Stallings Folding Process”. International Journal of algebra and computation 16 (6 2006).
- [6] THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD L. RIVEST, CLIFFORD STEIN., “Introduction to algorithms”. The MIT press, Cambridge, Massachusetts England, 2009.
- [7] CARMEL LEVY., “Stallings Folding Process in (almost) Linear Time”. 16 (2006).