Tesis de Licenciatura

Sobre la aplicación iterada del operador clique en grafos y su convergencia lineal.

Leopoldo Taravilse
ltaravilse@gmail.com
Departamento de Computación
Facultad de Ciencias Exactas y Naturales - UBA
Directora: Flavia Bonomo
Co-director: Martín Eduardo Frías Armenta

Enero de 2016

A todos los que hicieron algo para que pueda llegar a defender esta tesis, en especial, a mi familia.

Agradecimientos

A mis padres, que tanto me apoyaron para que pueda llegar al punto de poder escrirbir esta tesis. Al resto de mi familia, porque cada uno de ellos son los mejores en el rol que les toca cumplir, mi hermana Lucrecia, mis abuelos Lely, Quitín, Delicia y José, todos mis tíos y todos mis primos (todos, porque todos ellos son importantes, pero sobre todo Augusto y Lucía que son los mejores primos que alguien puede tener en el mundo.)

A Nico y a Seba, que hoy no puede estar acá pero desde donde esté se que está orgulloso y parte de esta tesis es para él, por haber estado desde siempre.

A Flavia y Martín, mis directores, por bancarse que tarde dos años en terminar esta tesis y apoyarme siempre que los necesité para ir avanzando de a pequeños pasos.

A los chicos del Bosque, Tomás, Juan Manuel, Juani, Juan Carlos, Agus, Manu, Juan Pablo (y también a Agos, que no es del Bosque pero es parte de un gran grupo de amigos) y todo el resto de mis compañeros de la secundaria, por ser un apoyo fundamental durante toda mi carrera.

A Santi Pezzoni y a Yaqui por ser tan buenos amigos.

A todos mis amigos de olimpíadas por haberme inspirado el amor por la matemática y por la programación, pero sobre todo a mi primer maestro, Ramiro, que me llevó por el camino de las matemáticas, y a Agustín, que me orientó para ser el programador que soy hoy en día.

A Patricia y Flora por ser artifices de momentos que me llenaron de alegrías y de amigos.

A mis amigos de la facu, que no los nombro a todos porque son demasiados, pero si quiero nombrar a algunos, Juli Sackmann, Melanie Sclar, Caro Lang, Franco Assenza, Juan Doldán, Gastón Bengolea, Emilio Almansi, Fede Pousa y Jons Levy entre muchos muchos otros.

A los montones de amigos con los que compartí grandes momentos o conocí gracias a la ACM ICPC (aunque algunos los conocía de antes), sobre todo a mi equipo, Quimey, Pablo y Germán, que sin ellos no hubiese llegado a ningún lado, pero también a Nico Álvarez, Fidel, Eric, Nico Bordenabe, Gastón y Nico Ponieman entre muchos otros.

A los Caloventores, Pablo, Mariano y Martín, que me dieron una oportunidad única, y a todos los que confiaron en mi (y en mis amigos) y se hicieron presentes en cada Training Camp.

A los excelentes docentes que tuve a lo largo de toda mi carrera, que como son muchos no los quiero nombrar a todos, pero no quiero dejar de nombrar a Irene por lo mucho que confió en

mí en infinidad de ocasiones.

A Aida, Mariana, Diana, Lara y Mateo porque sin su aporte no hubiese tenido una carrera para estudiar.

A todos los hinchas de Gimnasia, por compartir conmigo esta hermosa pasión.

Muchos de los arriba mencionados entran en más de una de las categorías, me restringí a ponerlos a cada uno en una sóla para no ser repetitivo.

Resumen

Uno de los primeros conceptos que se introdujo con la teoría de grafos fue el de *clique*. Una clique es un subgrafo maximal completo, es decir, un subgrafo donde todos los pares de vértices están conectados entre sí, y que además es maximal respecto de esa propiedad. En 1972, luego de que Cook [8] introdujera el concepto de problema NP-completo, Karp [21] demostró que el problema de hallar una clique máxima en un grafo general es NP-hard, y el de determinar si una clique es máxima es NP-completo. Esto hizo que muchos especialistas de la teoría de grafos estudiaran el comportamiento de las cliques en grafos generales y en familias particulares de grafos.

También en 1971, Roberts y Spencer [28] caracterizaron los grafos clique (aquellos grafos donde los vértices son las cliques maximales de un grafo y las aristas ocurren entre cliques con intersección no vacía), y en 1972, Hedetniemi y Slater [19] introdujeron el concepto de grafos iterados de cliques, que consiste en partir en un primer paso de un grafo G, y en cada paso calcular el grafo clique del grafo del paso anterior.

Los mismos Hedetniemi y Slater demostraron en [19] que existen grafos tales que al aplicarles el operador clique iteradas veces convergen al grafo trivial, así como también grafos que convergen a un ciclo de grafos no triviales. Los primeros de llaman grafos k-nulos y los segundos k-periódicos. Los grafos k-nulos y los k-periódicos se llaman también grafos k-convergentes. En dicho trabajo los autores presentan ejemplos de grafos k-nulos y de grafos k-convergentes de períodos 1 y 2.

Un año después, Victor Neumann-Lara descubre que existen familias de grafos k-divergentes dando como ejemplo a los octaedros, y Escalante reporta en 1973 [12] este descubrimiento de Neumann-Lara mostrando que además existen grafos k-periódicos de período p para todo $p \geq 1$.

A lo largo de los años se ha ido estudiando el comportamiento de los grafos iterados de cliques, como por ejemplo cuando Larrión y Neumann-Lara [22] demostraron que existen familias infinitas de grafos que divergen polinomialmente para cualquier grado (es decir, existen familias que divergen linealmente, cuadráticamente, etc).

En esta tesis presentaremos dos familias de grafos que convergen linealmente, una pregunta abierta hasta el momento. La primera de ellas decrece su número de vértices de a 4 o más, mientras que la segunda decrece su número de vértices de a 3 o más, quedando abierta la pregunta de si existen familias de grafos que convergen linealmente de a 1 o 2 vértices.

Para estas familias de grafos, daremos algoritmos polinomiales de reconocimiento, es decir, que en tiempo polinomial podremos saber si un grafo pertenece o no a esta familia.

A su vez, presentaremos familias de grafos que son auto-cliques y que son homeomorfas a distintos espacios topológicos. Los primeros grafos que mostraremos son homeomorfos a bandas de Moebious, a botellas de Klein y a planos proyectivos, y a su vez mostraremos una forma de combinar estos grafos para generar grafos homeomorfos a otros espacios topológicos.

Indice

1 Introducción **1**
1.1 Definiciones básicas y notación 1
1.2 Clases de grafos 2
1.3 Complejidad algorítmica 3

2 K-comportamiento **6**
2.1 Grafos K-divergentes 6
2.1.1 Existencia de grafos K divergentes y grafos que divergen linealmente . . . 6
2.2 Grafos K-convergentes 8
2.2.1 Grafos K-nulos 8
2.2.2 Grafos K-periódicos 9

3 Grafos de convergencia lineal **11**
3.1 Familias de grafos G que convergen de a 4 vértices o más por paso 11
3.1.1 Definición de la familia 11
3.1.2 $G(N, M)$ con $N \geq 3$ 12
3.1.3 $G(2, M)$ 14
3.1.4 $G(1, M)$ 15
3.1.5 $G(0, M)$ 16
3.2 Reconocimiento de la familia de grafos G que convergen de a 4 vértices o más por paso 17
3.2.1 $G(N, M)$ con $N \geq 4$ 17
3.2.2 $G(3, M)$ 19
3.2.3 $G(2, M)$ 20
3.2.4 $G(1, M)$ 20
3.2.5 $G(0, M)$ 21
3.3 Familias de grafos H que convergen de a 3 vértices o más por paso 22
3.3.1 Definición de la familia 22
3.4 Reconocimiento de la familia de grafos H que convergen de a 3 vértices o más por paso 24
3.4.1 $H(N, M)$ con $N \geq 4$ 25
3.4.2 $H(3, M)$ 26
3.4.3 $H(2, M)$ 27
3.4.4 $H(1, M)$ 28
3.4.5 $H(0, M)$ 29
3.4.6 Observaciones generales 29

4 Superficies **30**
4.1 Superficies 30

5 Conclusiones y trabajo futuro **34**

Capítulo 1

Introducción

Dado un grafo G, una *clique* en G es un subgrafo completo maximal. Se define el operador clique como un operador que se le aplica a un grafo G y genera un nuevo grafo $K(G)$, cuyos vértices se corresponden con las cliques del grafo original, y que contiene una arista entre dos vértices de $K(G)$ si y sólo si las cliques en G que representan esos vértices tienen intersección no vacía.
En 1971 Cook [8] introduce el concepto de problema NP-completo y problema NP-hard y un año más tarde Karp [21] demuestra que el problema de encontrar una clique máxima en un grafo es NP-hard. Esto motivó a muchos especialistas en teoría de grafos a estudiar el comportamiento de las cliques de un grafo, y es así como Roberts y Spencer [28] definen el operador clique y caracterizan aquellos grafos que son el grafo clique de otro grafo, mostrando un ejemplo de un grafo que no es grafo clique de ningún otro grafo. Por muchos años estuvo abierto el problema de saber si la clase de grafos que son K^2 de un grafo es la misma que la de los grafos clique, y recientemente Pablo de Caria [9] demostró que esto no es así, dando un ejemplo de un grafo que es K de un grafo pero no es K^2 de ningún grafo.
Si bien se conoce una caracterización para la clase de grafos clique $K(G)$, nignún algoritmo eficiente ha podido ser formulado en función de esa caracterización, y recientemente Alcón, de Figueiredo, Faria y Gutiérrez [1] demostraron que el problema de reconocimiento de esa clase de grafos es NP-completo.
En [30] se puede encontrar un compendio de los resultados sobre grafos cliques encontrados hasta el momento de dicha publicación (2001).

1.1 Definiciones básicas y notación

Denotaremos un grafo G por un par $(V(G), E(G))$, donde $V(G)$ es un conjunto finito, el conjunto de vértices de G, y $E(G)$ es un conjunto de pares no ordenados de vértices de G, llamados aristas. Sean $n = |V(G)|$ y $m = |E(G)|$. Un grafo se dice *trivial* si tiene un solo vértice.

Un vértice v es adyacente a otro vértice w en G si $(v, w) \in E(G)$. Decimos que v y w son los extremos de la arista.

Un grafo H es un subgrafo de un grafo G si $V(H) \subseteq V(G)$ y $E(H) \subseteq E(G)$. Dado un conjunto de vértices $X \subseteq V(G)$, el subgrafo de G inducido por X es el subgrafo H de G tal que $V(H) = X$ y $E(H)$ es el conjunto de aristas de G que tiene ambos extremos en X.

Dos grafos G y H son isomorfos si existe una biyección entre $V(G)$ y $V(H)$ que conserva las adyacencias. En este caso, notamos $G = H$.

Un camino en un grafo G es una secuencia de vértices distintos $P = v_1, v_2, ..., v_k$, donde $(v_i, v_{i+1}) \in E(G)$, $i = 1, ..., k-1$. Una cuerda en P es una arista que une dos vértices no consecutivos de P. Un camino inducido es un camino sin cuerdas. Denotamos por P_k al camino inducido por k vértices.

Un conjunto S es maximal (minimal) en relación a una determinada propiedad P si S satisface P, y todo conjunto S' que contiene propiamente a S (que está contenido propiamente en S) no satisface P.

Un grafo G es completo si cualquier par de vértices distintos de G son adyacentes. Llamamos K_n al grafo completo con n vértices.
Un conjunto de vértices M de un grafo G es un subgrafo completo si el subgrafo inducido por M es completo. Una clique es un subgrafo completo maximal de G.

Un concepto utilizado a lo largo de este trabajo es el de la propiedad de Helly. Una familia de subconjuntos S satisface la propiedad de Helly cuando toda subfamilia de S consistente en subconjuntos que se intersecan de a pares tiene intersección no vacía.

Definiciones no dadas aquí pueden encontrarse en [18].

1.2 Clases de grafos

Consideremos una familia finita de conjuntos no vacíos. El grafo intersección de esta familia se obtiene representando cada conjunto por un vértice y conectando dos vértices por un arista si y sólo si los correspondientes conjuntos se intersecan. Es sencillo probar que todo grafo es un grafo intersección de alguna familia.

Los grafos intersección han recibido mucha atención en el estudio de la teoría algorítmica de grafos y sus aplicaciones [24]. Algunas clases especiales muy estudiadas de grafos de intersección son los grafos de intervalos, los cordales, los arco-circulares, los circulares, los grafos de línea y los grafos clique.

El grafo clique de G, $K(G)$, es el grafo intersección de las cliques de G. Por lo tanto K es un operador que transforma un grafo en otro grafo. Podemos definir $K^j(G)$ como la j-ésima iteración del operador clique sobre el grafo G, es decir, $K^0(G) = G$ y $K^j(G) = K(K^{j-1}(G))$ para $j \geq 1$.

Dada una clase de grafos $\mathcal{H}$, definimos $K(\mathcal{H})$ como la clase de los grafos clique de grafos en $\mathcal{H}$ y $K^{-1}(\mathcal{H})$ como la clase de los grafos cuyos grafos clique están en $\mathcal{H}$ (estos últimos son llamados grafos clique-inversos de $\mathcal{H}$). Decimos que una clase $\mathcal{H}$ es fija bajo el operador K si $K(\mathcal{H}) = \mathcal{H}$, y que dos clases $\mathcal{H}$ y $\mathcal{L}$ son clique-duales si $K(\mathcal{H}) = \mathcal{L}$ y $K(\mathcal{L}) = \mathcal{H}$. En la Tabla 1.1 mostramos algunas de las clases de grafos clique caracterizadas hasta el momento.

Un grafo es clique-Helly si sus cliques satisfacen la propiedad de Helly. Un grafo G es clique-Helly hereditario si todo subgrafo inducido de G es clique-Helly.

La forma habitual de dibujar un grafo en el plano es representando los vértices por puntos y las aristas por curvas entre sus extremos, que no pasan por ningún otro vértice. Un grafo G es planar si existe una representación de G en el plano de modo que las aristas no se crucen, excepto en los vértices.
Una completa recopilación sobre clases de grafos aparece en [5].

Clase $\mathcal{A}$	$K(\mathcal{A})$	Ref.
Arco-circular Helly	Cliques circulares	[11]
Bloques	Bloques	[19]
Circular Helly	Dualmente Circular Helly	[23]
Clique-Helly	Clique-Helly	[12]
Cordales	Dualmente Cordales	[3, 6, 14]
Clockwork	Clockwork	[22]
DE	Dualmente DE	[15]
Disco-Helly	Disco-Helly	[2]
Desmontables	Desmontables	[2]
Dualmente Cordales	Cordales ∩ Clique-Helly	[6, 14]
Dualmente Circular Helly	Circular Helly	[23]
Dualmente DE	DE	[15]
Dualmente DV	DV	[17, 27]
Dualmente RDV	RDV	[4, 27]
DV	Dualmente DV	[17, 27]
Clique-Helly hereditarios	Clique-Helly hereditarios	[26]
Intervalos	Intervalos Propios	[20]
Intervalos Propios	Intervalos Propios	[20]
Intervalos Propios Minimales	Intervalos Propios	[16]
Ptolomeicos	Ptolomeicos	[2]
RDV	Dualmente RDV	[4, 27]
Sin Diamantes	Sin Diamantes	[7]
Fuertemente Cordales	Fuertemente Cordales	[2, 6]
Arboles	Bloques	[19]
UV	Dualmente UV	[3]

Tabla 1.1: Clases de grafos clique

1.3 Complejidad algorítmica

Un problema algorítmico $\pi(I, Q)$ consta de un conjunto I de todas las posibles entradas para el problema, llamado el conjunto de instancias, y de una pregunta Q sobre esas instancias. Resolver uno de estos problemas consiste en desarrollar un algoritmo cuya entrada es una instancia del problema y cuya salida es una respuesta a la pregunta del problema.
Decimos que un problema es de decisión cuando las posibles respuestas a la pregunta son SI ó NO. Por ejemplo, π podría ser el siguiente problema: "dado un grafo G, ¿es su grafo clique $K(G)$ un grafo planar?". El conjunto de instancias es el conjunto de todos los grafos y la pregunta es saber si el grafo clique de G es o no es planar. El problema dado no sólo es de decisión sino que, en particular, es un problema de reconocimiento. Es de sumo interés tanto desde el punto de vista teórico como de las aplicaciones estudiar problemas de reconocimiento para las diferentes clases de grafos.
Un problema es de optimización cuando lo que se busca a través de la pregunta es la solución óptima para el problema formulado. Por ejemplo, "dado un grafo G, ¿ cuánto vale $\chi(G)$?"
Usualmente, los problemas de optimización tienen su variante de decisión. En el caso del coloreo será: "dado un grafo G y un entero k positivo, ¿ existe un coloreo de G con a lo sumo k colores ?"
Un problema es de enumeración cuando lo que se busca no es la existencia de una solución o la solución óptima sino la cantidad de soluciones para el problema formulado. Por ejemplo, "dado

un grafo G y un entero k positivo, ¿ de cuántas formas distintas es posible colorear G con k colores ?"

Diremos que un algoritmo es polinomial cuando el número de operaciones que efectúa está acotado por una función polinomial en el tamaño de su entrada. Si el tamaño de la entrada es n y la función polinomial es $f(n)$, decimos que el algoritmo tiene complejidad $O(f(n))$. Los problemas de decisión para los que existen algoritmos polinomiales constituyen la clase P y son llamados polinomiales.

Un problema de decisión es no-determinístico polinomial cuando cualquier instancia que produce respuesta SI posee una comprobación de correctitud (también llamada certificado) verificable en tiempo polinomial en el tamaño de la instancia. Estos problemas de decisión pertenecen a la clase NP.
Claramente, P $\subseteq$ NP. Sin embargo, no se sabe si esta inclusión es estricta: uno de los principales problemas abiertos en informática teórica es saber si P $\neq$ NP.
Un problema de decisión pertenece a la clase co-NP cuando cualquier instancia que produce respuesta NO posee un certificado polinomial en el tamaño de la instancia.
Sean $\pi_1(I_1, Q_1)$ y $\pi_2(I_2, Q_2)$ dos problemas de decisión. Una transformación polinomial de π_1 en π_2 es una función $f : I_1 \to I_2$ que satisface las siguientes dos condiciones:

1. f puede computarse en tiempo polinomial.
2. Para toda instancia $D \in I_1$, D produce respuesta SI para π_1 si y sólo si $f(D)$ produce respuesta SI para π_2.

Una definición esencial en la teoría de complejidad es la definición de problema NP-completo. Un problema de decisión π pertenece a la clase NP-completo cuando se satisfacen las siguientes condiciones:

- $\pi \in$ NP.
- Para todo problema $\pi' \in$ NP, existe una transformación polinomial de π' en π.

Un problema de decisión π pertenece a la clase NP-hard cuando se satisface la siguiente condición:

- Para todo problema $\pi' \in$ NP, existe una transformación polinomial de π' en π.

La teoría de NP-completitud fue iniciada por Cook en 1971 [8]. Allí probó que el problema de satisfactibilidad de la lógica matemática es NP-completo, en un resultado que se conoce como el Teorema de Cook. Primero Karp [21], y tiempo después Garey y Johnson [13], presentaron largas listas de problemas NP-completos en el campo de la combinatoria, la lógica, la teoría de conjuntos, la teoría de grafos y otras áreas de la matemática discreta.
La técnica standard para probar que un problema π es NP-completo es la siguiente: elegir en forma apropiada un problema π' que ya sabemos que es NP-completo y luego probar que $\pi \in$ NP y que π' es transformable polinomialmente en π. Si sólo probáramos esta segunda parte, habríamos probado que el problema π es NP-hard.
No se conoce ningún algoritmo polinomial para resolver un problema NP-completo. Surge de la definición de NP-completitud que si se encontrara un algoritmo polinomial para un problema en esta clase, todo problema en NP sería polinomial (y estaría probado, en consecuencia, que P

= NP).

Similarmente, se define una clase de complejidad para los problemas cuya respuesta es un número natural, en la cual entran la mayoría de los problemas de enumeración relacionados con problemas NP-completos y algunos problemas de aritmética como por ejemplo calcular la permanente de una matriz [31].
Sean $\pi_1(I_1, Q_1)$ y $\pi_2(I_2, Q_2)$ dos problemas cuyas respuestas son números naturales. Una transformación polinomial de π_1 en π_2 es una función $f : I_1 \to I_2$ que satisface las siguientes dos condiciones:

1. f puede computarse en tiempo polinomial.
2. Para toda instancia $D \in I_1$, D produce la misma respuesta para π_1 que $f(D)$ para π_2.

En forma análoga a los problemas de decisión, un problema π pertenece a la clase #P-completo cuando se satisfacen las siguientes condiciones:

- $\pi \in$ #P.
- Para todo problema $\pi' \in$ #P, existe una transformación polinomial de π' en π.

Un problema π pertenece a la clase #P-hard cuando se satisface la siguiente condición:

- Para todo problema $\pi' \in$ #P, existe una transformación polinomial de π' en π.

La teoría de #P-completitud fue iniciada por Valiant en 1977 [31, 32]. Allí probó, entre otros resultados, que el problema de calcular la cantidad de cliques de un grafo es #P-completo. En base a un resultado de Simon [29], quien observó que la transformación genérica utilizada en la demostración del Teorema de Cook se podría hacer respetando la cantidad de soluciones, se puede probar también que el problema de contar la cantidad de valuaciones distintas que hacen verdadera a una instancia del problema de satisfactibilidad de la lógica matemática es #P-completo [25].

Resultados sobre complejidad de algoritmos pueden encontrarse en [13] y [25].
La comlejidad algorítmica es relevante para este trabajo de tesis ya que el problema de hallar clique máxima es NP-Hard al igual que el problema de contar la cantidad de cliques de un grafo y determinar cuáles son estas cliques.

Capítulo 2

K-comportamiento

En este capítulo estudiaremos los posibles K-comportamientos que puede tener un grafo.
Existen tres K-comportamientos posibles para un grafo:

- K-divergencia: Esto es, cuando al aplicar el operador clique el número de vértices diverge.
- K-peridiocidad: Un grafo es K-periódico cuando al aplicar el operador clique iteradas veces converge a un ciclo de longitud $L \geq 1$ de grafos no triviales.
- K-nulidad: Este caso es el caso de los grafos que al aplicarles el operador clique iteradas veces llegan en una cantidad finita de pasos al grafo trivial.

Los grafos K-periódicos y los K-nulos se denominan grafos K-convergentes.
En este capítulo presentaremos ejemplos de los tres comportamientos así como también daremos familias de grafos que cumplen con algunas propiedades que son de importancia en la teoría de cliques.

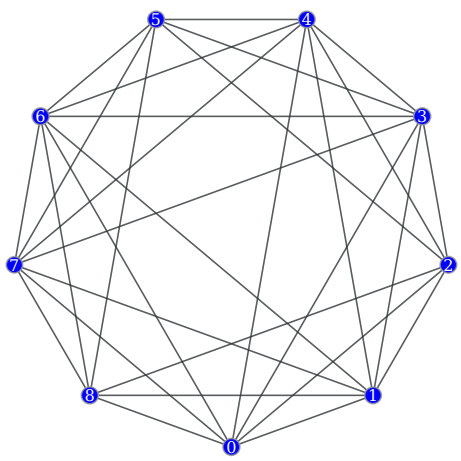
2.1 Grafos K-divergentes

2.1.1 Existencia de grafos K divergentes y grafos que divergen linealmente

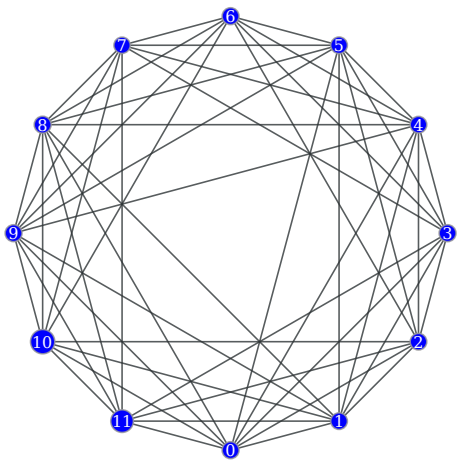
En 1972, Hedetniemi y Slater [19] demostraron que existen grafos K-divergentes.
Muchos años después, Larrión y Neumann-Lara [22] encontraron una familia de grafos que divergen linealmente.
Dicha familia consiste de los grafos $G(R, S)$ que describimos a continuación:

- C_n es un ciclo de n vértices en el que las aristas son (v_i, v_{i+1})
- C_n^s se define como un grafo con n vértices enumerados de 0 a $n-1$ donde v_i es vecino de $v_{i+1}, v_{i+2}, \ldots, v_{i+s}$. A su vez, v_i es por la misma definición vecino de $v_{i-1}, v_{i-2}, \ldots, v_{i-s}$. En este caso, tanto las sumas como s se toman módulo n.
- $G(R, S)$ se define para $R \geq 3$ y $S \geq 3$ como C_{RS}^{S-1} agregandole las aristas de la forma v_i, v_{i+s+1} con $i \in \{0, s, 2s, \ldots, (r-1)s\}$.

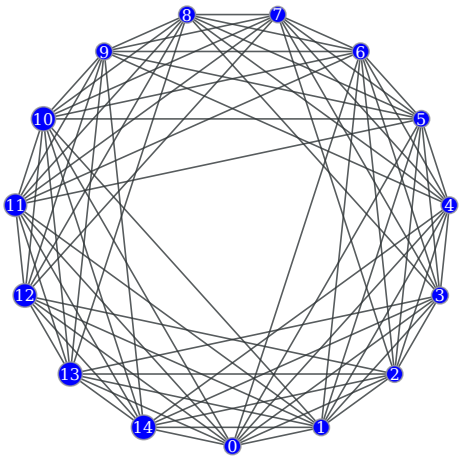
$G(R, S)$ tiene rs vértices y $rs(s-1)+r$ aristas. Lo que prueban Larrión y Neumann-Lara es que $K(G(R, S)) = G(R, S+1)$. Veamos a continuación algunos ejemplos de estos grafos.



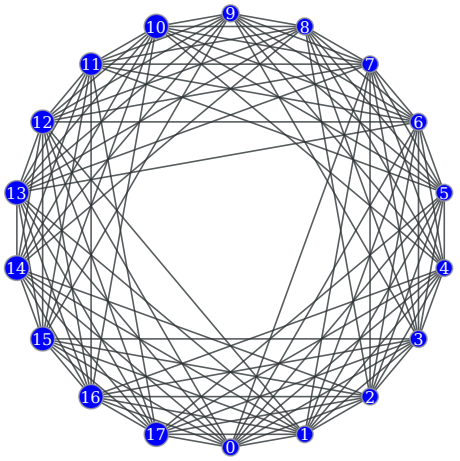
$G(3,3)$



$G(3,4)$



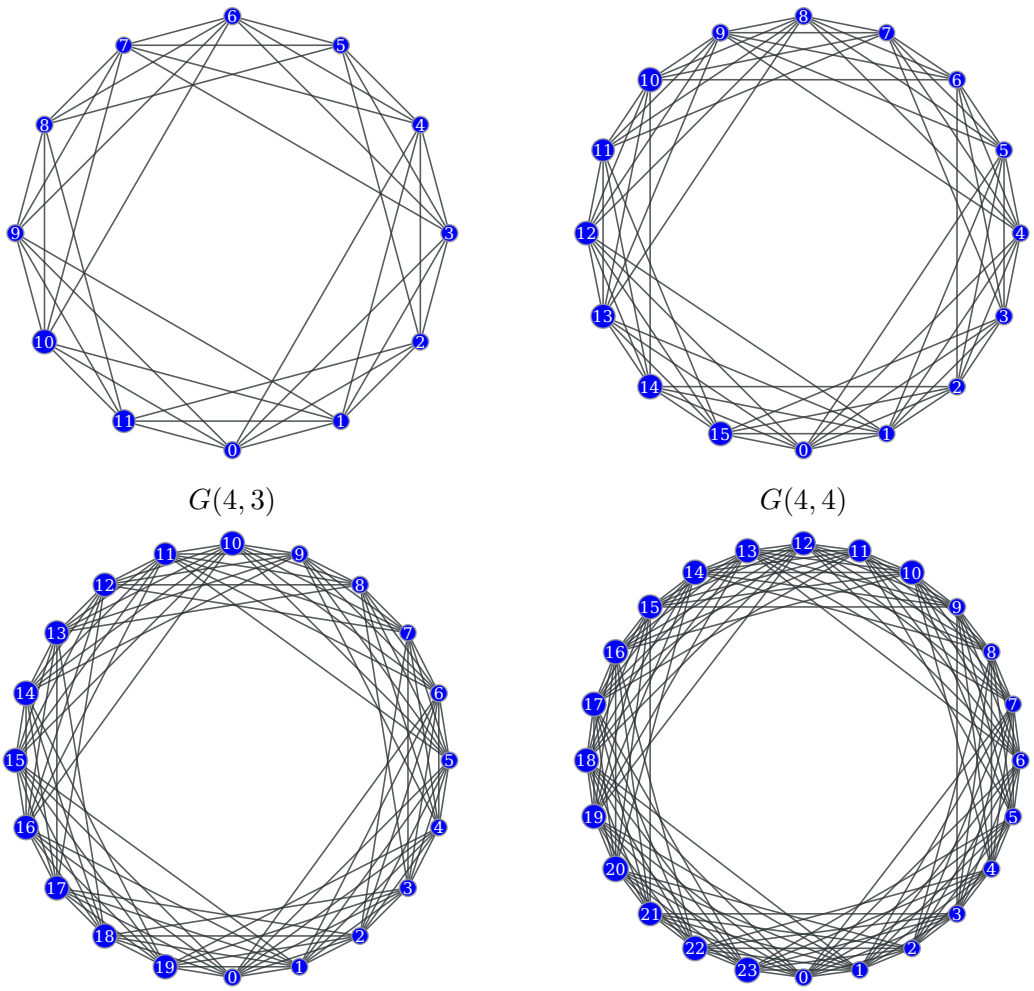
$G(3,5)$



$G(3,6)$

Figura 2.1: $G(3,S)$ **con** $S = \{3,4,5,6\}$

En la Figura 2.1 podemos ver los grafos $G(3,3)$,$G(3,4)$, $G(3,5)$ y $G(3,6)$, siendo cada uno el resultado de aplicar el operador clique al grafo anterior, es decir $G(3,6) = K(G(3,5)) = K^2(G(3,4)) = K^3(G(3,3))$.
Esto no ocurre sólo para $R = 3$, veamos también grafos en los que $R = 4$.



$G(4,3)$ $G(4,4)$

$G(4,5)$ $G(4,6)$

Figura 2.2: $G(4,S)$ **con** $S = \{3,4,5,6\}$

Si bien para grafos no tan grandes ya no se llegan a ver en la figura todas las aristas, las mismas son ilustrativas y sirven para ver cómo existen grafos que divergen linealmente en la cantidad de vértices de R en R para $R \geq 3$.
En el mismo trabajo los autores demuestran que existen grafos que divergen de R en R para $R = 1$ y para $R = 2$ y también demuestran que existen familias de grafos que divergen polinomialmente para cualquier grado $n \geq 1$.

2.2 Grafos K-convergentes

2.2.1 Grafos K-nulos

Es fácil ver que existen grafos K-nulos, y que existen grafos que se anulan convergiendo linealmente hasta llegar al grafo trivial. El ejemplo con el que es más fácil de ver esto es P_n, los caminos de n vértices, que cumplen con $P_{n-1} = K(P_n)$.
Escalante [12] demostró que si un grafo G es Clique-Helly, entonces $K^2(G)$ es G menos sus

vértices dominados. Dado que los árboles son Clique-Helly y los vértices dominados de los árboles son las hojas, podemos afirmar que si T es un árbol con más de una hoja y $H(T)$ son las hojas del árbol T entonces $K^2(T) = T \setminus H(T)$.
Para el caso particular de los árboles podemos demostrar este hecho de la siguiente manera: Dado un árbol T, las cliques de T son las aristas de T, es decir, al ser libres de K_3, todas sus cliques son K_2. Por cada vértice v que no es una hoja, todas las cliques que inciden en dicho vértice son una arista que representan un vértice en $K(T)$ y todas ellas forman una clique ya que tienen a v en común. Luego en $K^2(T)$ esta clique se convierte en un nuevo vértice v' que es análogo a v. Las hojas, en cambio, como sólo tienen una clique incidente, no generan nuevas aristas en $K(T)$ y luego los vértices que representen a la arista que incide en la hoja van a formar parte solamente de la clique que forma el otro vértice donde inciden, y como T tiene más de una arista, ese vértice no es una hoja. Solo nos falta ver que v' y w' son vecinos en $K^2(T)$ si y solo si v y w son vecinos en T.
La ida de esta demostración es fácil, ya que al haber una arista que los conecta, esta arista se convierte en un vértice que va a ser la intersección de las cliques que vayan a parar a v' y w'. La vuelta tampoco es muy difícil de probar, ya que si las cliques que van a parar a v' y w' tienen intersección no vacía, es porque había una arista que conectaba v y w.

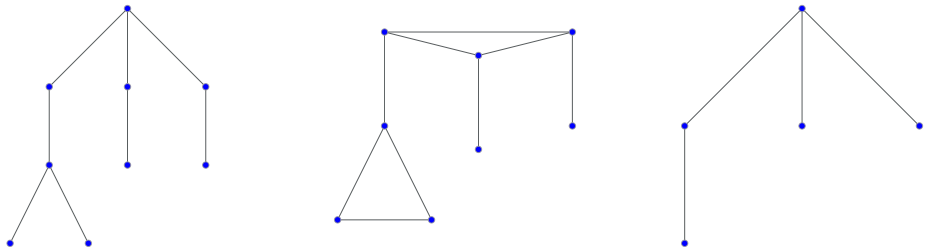


Figura 2.3: Un árbol T con sus primeros dos grafos cliques

En la Figura 2.3 podemos ver un árbol T, el grafo $K(T)$ y el grafo $K^2(T)$ como ejemplo para visualizar el comportamiento del operador clique en árboles.
En el primer paso, las tres aristas que salen de la raíz conforman un K_3, luego los tres hijos de la raíz, que tienen un único hijo cada uno, representan las aristas que unen ese K_3 con los vértices del siguiente nivel, por último, el nieto de la raíz de más a la izquierda, que tiene dos hijos, forma otro K_3 con sus dos hijos y su padre.
Por último vemos como estas cliques mapean a vértices en el grafo $K^2(T)$ y generan un grafo que es T menos sus hojas.
Si tomamos un camino simple, en cada paso eliminamos un vértice del camino, es decir $K(P_n) = P_{n-1}$. Al pegar n caminos simples por su extremo, cada dos pasos del operador clique eliminamos n vértices, es decir, que dicho grafo se anula de a n vértices por cada 2 pasos.

2.2.2 Grafos K-periódicos

En [12] Escalante demuestra que existen grafos periódicos para cualquier período. Esto es, que para todo n existe un grafo G tal que $K^n(G) = G$ y para todo $1 \leq i < n$ se tiene que $K^i(G) \neq G$. Este resultado es importante, pero la teoría no estaba completa hasta el momento ya que no se había podido encontrar una familia infinita de grafos K-periódicos que puedan converger en

una cantidad lineal de vértices.
Dicho resultado será de lo que trata el próximo capítulo de esta tesis.

Capítulo 3

Familias de grafos que convergen linealmente

En este capítulo presentaremos familias de grafos que convergen linealmente al aplicarles el operador clique. Estas familias son cerradas con respecto a dicho operador. Hasta el momento no se conocían familias de grafos cerradas por el operador clique que convergieran linealmente al aplicarles dicho operador.

3.1 Familias de grafos G que convergen de a 4 vértices o más por paso

3.1.1 Definición de la familia

A continuación presentaremos una familia de grafos tales que al aplicarles el operador clique, convergen linealmente, de a $M \geq 4$ vértices por paso. Esta familia está compuesta por los grafos $G(N, M)$ que definiremos a continuación, siendo $K(G(N, M)) = G(N-1, M)$ cuando $N \geq 1$ y $G(0, M)$ autoclique, es decir, que $K(G(0, M)) = G(0, M)$. Definiremos estos grafos para $N \in \mathbb{Z}_{\geq 0}$ y $M \geq 4$, por lo que la convergencia será de a M vértices para cualquier valor de M que sea mayor o igual a 4.

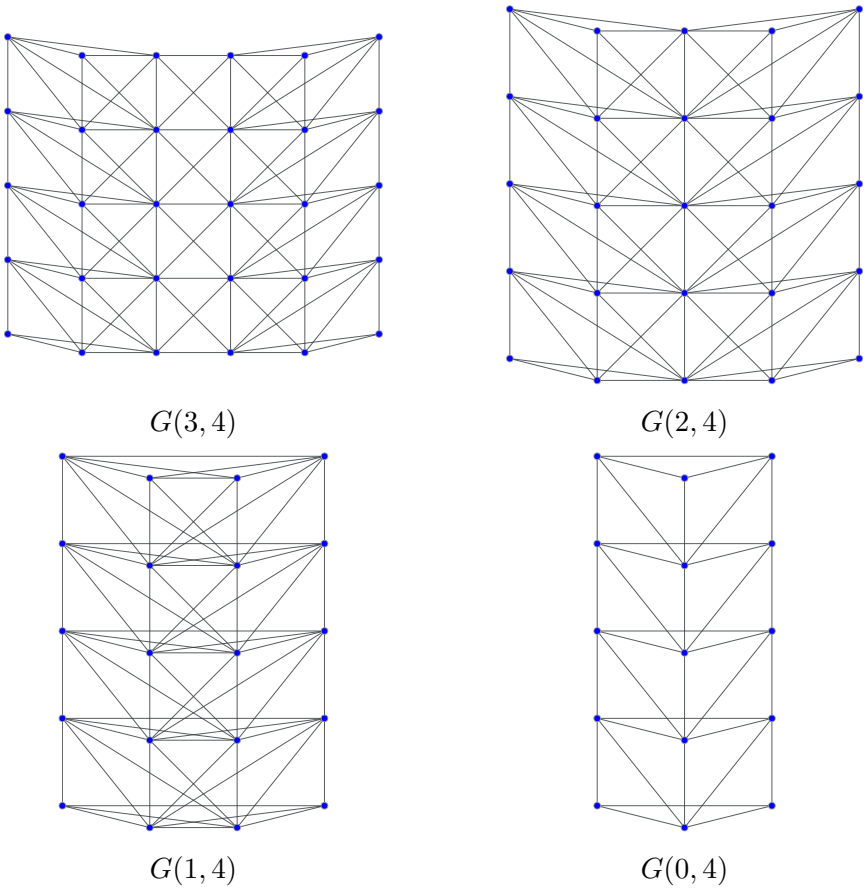
Definición 3.1 *Los vértices de $G(N, M)$ son $v_{i,j}$ con $i \in \{-1, 0, 1, \ldots, N, N+1\}$ y $j \in \mathbb{Z}_M$. Cuando $N \geq 2$ las aristas de $G(N, M)$ son las siguientes:*

- *v_{i_0,j_0} es adyacente a v_{i_1,j_1} si y sólo si $|i_0 - i_1| \leq 1$, $|j_0 - j_1| \leq 1$, $i_0, i_1 \in \{0, 1, \ldots, N\}$ y $j_0, j_1 \in \mathbb{Z}_M$.*
- *$v_{-1,i}$ es adyacente a $v_{-1,i+1}, v_{0,i}, v_{0,i+1}, v_{1,i}$ y $v_{1,i+1}$ con $i \in \mathbb{Z}_M$.*
- *$v_{n+1,i}$ es adyacente a $v_{n+1,i+1}, v_{n,i}, v_{n-1,i}, v_{n-1,i+1}$ y $v_{n,i+1}$ con $i \in \mathbb{Z}_M$.*

Cuando $N = 1$ las aristas son las mismas más las aristas $\{(v_{-1,i}, v_{2,i})\}$ con $i \in \mathbb{Z}_M$.
Cuando $N = 0$ las aristas son $\{\{v_{0,i}, v_{0,i+1}\}, \{v_{1,i}, v_{0,i}\}, \{v_{1,i}, v_{0,i+1}\}, \{v_{-1,i}, v_{0,i}\}, \{v_{-1,i}, v_{0,i+1}\}, \{v_{1,i}, v_{-1,i}\}, \{v_{-1,i}, v_{-1,i+1}\}, \{v_{1,i}, v_{1,i+1}\}\}$.

A continuación presentamos algunos de los grafos de esta familia. Todos los grafos que veremos en las figuras son $G(N, M)$ con $M = 4$ y el primer y último nivel de las figuras (es decir, el

de más arriba y el de más abajo) representan los mismos vértices, ya que la topología de estos grafos es cilíndrica.



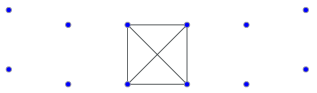
$G(3,4)$ $G(2,4)$

$G(1,4)$ $G(0,4)$

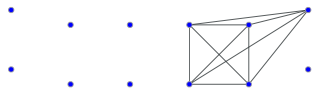
Figura 3.1: $G(N,4)$ **con** $N = \{3,2,1,0\}$

3.1.2 $G(N,M)$ con $N \geq 3$

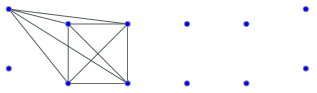
Para $G(N,M)$ con $M \geq 3$ existen 5 tipos de cliques, las mismas se repiten en todos sus niveles y son las que vemos a continuación en la Figura 3.2
Estas cliques las veremos únicamente representadas en $G(3,4)$ ya que para $N > 3$ las cliques son las mismas. Las únicas cliques que se agregan son las de tipo 1 que aparecen en las columnas del medio que se van agregando, y también veremos qué pasa con este tipo de cliques cuando son más de una columna y cómo se intersecan.



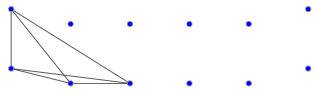
Cliques de tipo 1



Cliques de tipo 2



Cliques de tipo 3



Cliques de tipo 4

Cliques de tipo 5

Figura 3.2: Cliques de $G(3, M)$

Las cliques de tipo 1 se intersecan entre sí formando nuevas cliques de tipo 1 (si imaginamos por ejemplo un $G(9, M)$ vamos a tener 9 columnas de cliques de tipo 1, y se van a intersecar todas formando 8 columnas de cliques de tipo 1).
Analizemos por separado ahora el caso de $G(3, M)$ y el de $G(N, M)$ con $N > 3$.
Si $N > 3$ entonces tenemos dos columnas de cliques de tipo 1 en los costados, que se intersecan, la de la izquierda, con las cliques de tipo 3 y 4, y la de la derecha, con las cliques de tipo 2 y 5.
Las cliques de tipo 1 del borde izquierdo las vamos a denominar $C_{1,i}$ y van a parar a los vértices $v_{1,i}$ en el grafo $K(G)$
Las cliques $C_{1,i}$ de tipo 1 del borde izquierdo están conformadas por los vértices $(v_{1,i}, v_{1,i+1}, v_{2,i}, v_{2,i+1})$ y se intersecan con las cliques de tipo 3: la que está conformada por los vértices $(v_{-1,i}, v_{0,i}, v_{0,i+1}, v_{1,i}, v_{1,i+1})$ (a la que llamaremos $C_{0,i}$ y que va a parar a $v_{0,i}$), la que está conformada por los vértices $(v_{-1,i+1}, v_{0,i+1}, v_{0,i+2}, v_{1,i+1}, v_{1,i+2})$ y la que está conformada por los vértices $(v_{-1,i-1}, v_{0,i-1}, v_{0,i}, v_{1,i-1}, v_{1,i})$. Estas últimas dos cliques naturalmente serán $C_{0,i+1}$ y $C_{0,i-1}$.

Las cliques $C_{1,i}$ también intersecarán a las cliques de tipo 4 conformadas por los vértices $(v_{-1,i}, v_{-1,i+1}, v_{0,i+1}, v_{1,i+1})$ a las que llamaremos $C_{-1,i}$ y con las cliques $C_{-1,i-1}$ definidas de la misma manera.
Por el lado derecho, las cliques $C_{N-2,i}$ intersecarán también a las cliques $C_{N-1,i}, C_{N-1,i+1}, C_{N-1,i-1}, C_{N,i}$ y $C_{N,i-1}$ análogamente.
Por último, las cliques $C_{0,i}$ intersecarán a las cliques $C_{0,i-1}, C_{0,i+1}, C_{-1,i}$ y $C_{-1,i-1}$ y las cliques $C_{-1,i}$ intersecarán a las cliques $C_{-1,i-1}$ y $C_{-1,i+1}$
De esta forma un $G(N, M)$ con $N > 3$ tendrá como grafo clique a $K(G(N, M)) = G(N-1, M)$ ya que las cliques que se intersecan se corresponden con las aristas de $G(N-1, M)$.
Si $N = 3$ para las cliques de tipo 2, 3, 4 y 5 sus intersecciones no cambian, y el único cambio es que las cliques $C_{1,i}$ y $G_{N-2,i}$ son la misma clique y podemos observar que esto va a producir un $G(2, M)$ como $K(G(3, M))$.

3.1.3 $G(2, M)$

A continuación vemos en la Figura 3.3 los cuatro tipos de cliques del grafo $G(2, M)$.

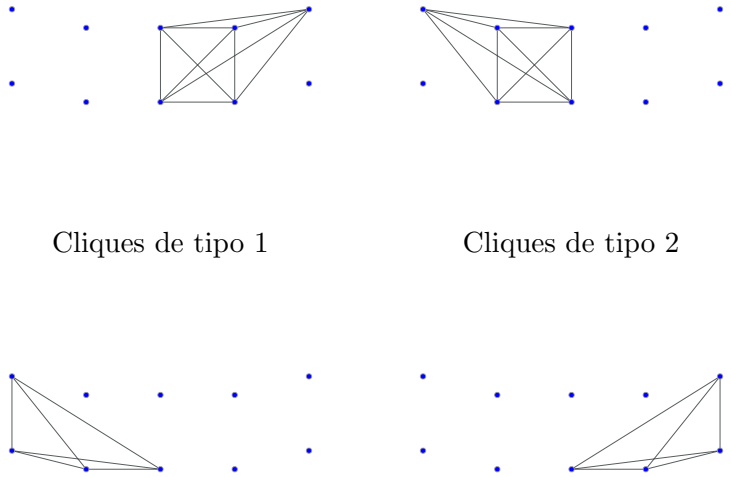


Figura 3.3: Cliques de $G(2, M)$

En este caso se intersecan cliques de todos los pares de tipos. Primero le daremos nombre a las cliques y luego mostraremos sus intersecciones.
A las cliques de tipo 1 formadas por los vértices $(v_{1,i}, v_{2,i}, v_{3,i}, v_{1,i+1}, v_{2,i+1})$ las denominaremos $C_{1,i}$.
A las cliques de tipo 2 formadas por los vértices $(v_{-1,i}, v_{0,i}, v_{1,i}, v_{0,i+1}, v_{1,i+1})$ las denominaremos $C_{0,i}$.
A las cliques de tipo 3 formadas por los vértices $(v_{-1,i-1}, v_{-1,i}, v_{0,i}, v_{1,i})$ las denominaremos $C_{-1,i}$

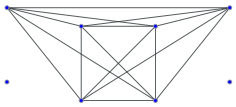
Por último, a las cliques de tipo 4 formadas por los vértices $(v_{1,i}, v_{2,i}, v_{3,i}, v_{3,i-1})$ las denominaremos $C_{2,i}$.
Las intersecciones que se forman son las siguientes:

- Tipo 1: $C_{1,i}$ se interseca con $C_{1,i-1}$ y $C_{1,i+1}$.
- Tipo 2: $C_{0,i}$ se interseca con $C_{0,i-1}$ y $C_{0,i+1}$.
- Tipo 3: $C_{-1,i}$ se interseca con $C_{-1,i-1}$ y $C_{-1,i+1}$.
- Tipo 4: $C_{2,i}$ se interseca con $C_{2,i-1}$ y $C_{2,i+1}$.
- Tipo 1 y tipo 2: $C_{1,i}$ se interseca con $C_{0,i}$, $C_{0,i-1}$ y $C_{0,i+1}$.
- Tipo 1 y tipo 3: $C_{1,i}$ se interseca con $C_{-1,i}$ y $C_{-1,i-1}$.
- Tipo 1 y tipo 4: $C_{1,i}$ se interseca con C_{2,i_1}, $C_{2,i}$ y $C2, i+1$.
- Tipo 2 y tipo 3: $C_{0,i}$ se interseca con $C_{-1,i-1}$, $C_{-1,i}$ y $C_{-1,i+1}$.
- Tipo 2 y tipo 4: $C_{0,i}$ se interseca con $C_{2,i}$ y $C_{2,i-1}$.
- Tipo 3 y tipo 4: $C_{-1,i}$ se interseca con $C_{2,i}$.

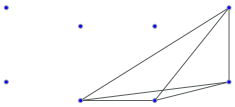
Podemos observar que estas intersecciones dan lugar a que $K(G(2, M)) = G(1, M)$.

3.1.4 $G(1, M)$

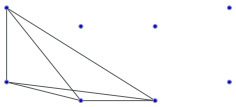
Veamos ahora en la Figura 3.4 las cliques de $G(1, M)$.
Las cliques de tipo 1 que contienen a los vértices $(v_{-1,i}, v_{0,i}, v_{1,i}, v_{2,i}, v_{0,i+1}, v_{1,i+1})$ serán identificadas como $C_{0,i}$.
Las cliques de tipo 2 que contienen a los vértices $(v_{0,i+1}, v_{1,i+1}, v_{2,i+1}, v_{2,i})$ serán denominadas $C_{1,i}$.
Por último denominaremos $C_{-1,i}$ a las cliques de tipo 3 que contienen a los vértices $(v_{-1,i}, v_{-1,i+1}, v_{0,i+1}, v_{1,i+1})$.
Las intersecciones de las cliques darán lugar al grafo $G(0, M)$ y no las daremos ya que es bastante fácil de observar que esto es así.



Cliques de tipo 1



Cliques de tipo 2

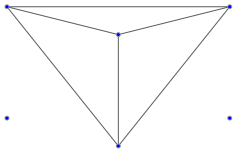


Cliques de tipo 3

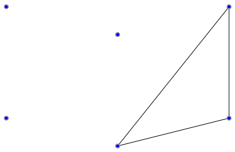
Figura 3.4: Cliques de $G(1, M)$

3.1.5 $G(0, M)$

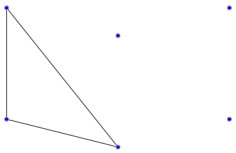
Por último observamos las cliques de $G(0, M)$ en la Figura 3.5 y no es difícil observar que dichos grafos son autocliques, es decir, que $K(G(0, M)) = G(0, M)$.



Cliques de tipo 1



Cliques de tipo 2



Cliques de tipo 3

Figura 3.5: Cliques de $G(0, M)$

3.2 Reconocimiento de la familia de grafos G que convergen de a 4 vértices o más por paso

Para reconocer estos grafos es necesario descomponer el reconocimiento de los mismos en cinco categorías, una más de las que usamos para analizar los grafos en la sección anterior.

3.2.1 $G(N, M)$ con $N \geq 4$

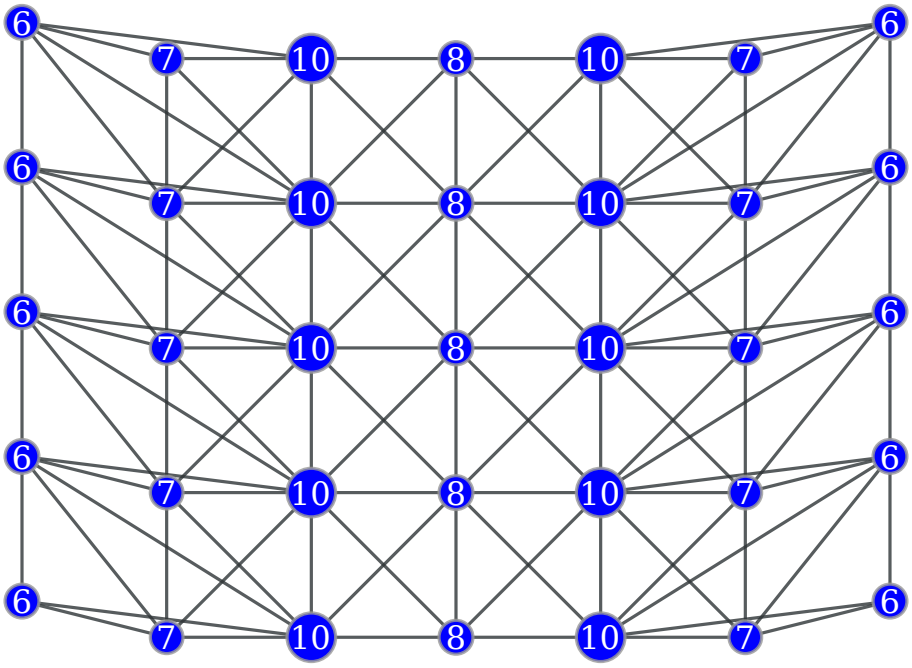
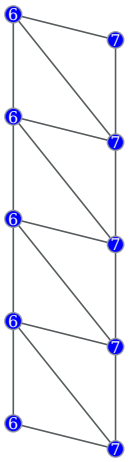


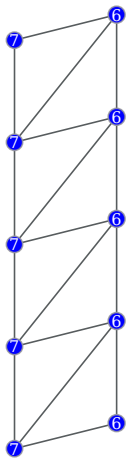
Figura 3.6: Grados de los vértices de $G(4, 4)$

La forma en la que vamos a detectar los $G(N, M)$ con $N \geq 4$ será formando ciclos según los grados de los vértices. Primero contaremos que haya $2M$ vértices de grado 6, $2M$ vértices de grado 7, $2M$ vértices de grado 10 y $(N-3)M$ vértices de grado 8. Para identificar N y M simplemente contamos la cantidad de vértices de cada grado, y si los vértices de grado 6, 7 y 10 son la misma cantidad, entonces ese número es $2M$ y hacemos la cuenta para calcular N, y en caso de que nos de que N es entero seguimos con el algoritmo, si N no es entero o la cantidad de vértices de grado 6, 7 y 10 es impar sabemos que el grafo no es $G(N, M)$.
Primero consideramos los subgrafos inducidos por vértices de grado 6, de grado 7 y de grado 10 (cada uno por separado, es decir, tres subgrafos de $G(N, M)$). Para cada uno de estos subgrafos

contamos la cantidad de componentes conexas. Si los tres subgrafos tienen dos componentes conexas cada uno, entonces seguimos con el algoritmo (caso contrario determinamos que el grafo no es $G(N, M)$).
Luego cada una de las dos componentes conexas de los subgrafos inducidos por los vértices de grado 6, los de grado 7 y los de grado 10 deben ser un ciclo simple. Si esto pasa, seguimos con el algoritmo.
Ahora el primer paso será pegar los ciclos de grado 6 y los de grado 7 para obtener los siguientes grafos:



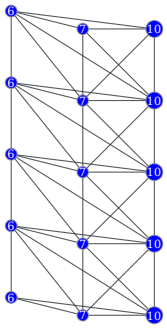
Ciclos de grado 6 y 7 (izquierda)



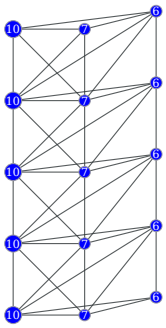
Ciclo de grado 6 y 7 (derecha)

Figura 3.7: Ciclos de grado 6 y 7 pegados en $G(4, M)$

Hasta ahora todo lo que hicimos se puede lograr en tiempo lineal en $|V(G)|+|E(G)|$. Detectar si los ciclos se pueden pegar de esta manera también ya que para cada vértice de grado 6 tiene que haber sólo dos aristas a vértices de grado 7 y viceversa, y luego sólo hay 4 formas de intentar pegar los ciclos (hay dos sentidos en los que se pueden pegar, y para cada sentido, hay dos vértices para identificar como el vecino más cercano en la figura al vértice que elijamos como inicial en el ciclo).
Si logramos identificar estos ciclos luego nos queda pegar también los de grado 10 de la siguiente manera:



Ciclos de grado 6, 7 y 10 (izquierda)



Ciclo de grado 6, 7 y 10 (derecha)

Figura 3.8: Ciclos de grado 6, 7 y 10 pegados en $G(4, M)$

Para esto, pegamos los de grado 10 a los de grado 6 de la misma manera que hicimos con los de grado 6 y 7, y una vez que tenemos identificado cada vértice vemos si los de grado 7 y 10 tienen las aristas que corresponden.
Luego vamos a tener que ir pegando ciclos de grado 8 a los de grado 10, y luego los de grado 8 entre sí. La manera de hacer esto es identificar K_4 entre los de grado 10 u 8 que ya tenemos pegados al grafo original e ir pegandolos como corresponde, y chequear que se generen los ciclos correspondientes. Una vez que hicimos esto, ya tenemos indexados todos los vértices solo nos queda comprobar que existe un isomorfismo de grafos, pero conociendo el morfismo biyectivo entre vértices, por lo que sólo nos queda verificar que el morfismo biyectivo de vértices es un isomorfismo de grafos, es decir, que conserva las aristas.
Todo esto podemos hacerlo en tiempo lineal.

3.2.2 $G(3, M)$

El próximo paso será identificar los grafos $G(3, M)$.

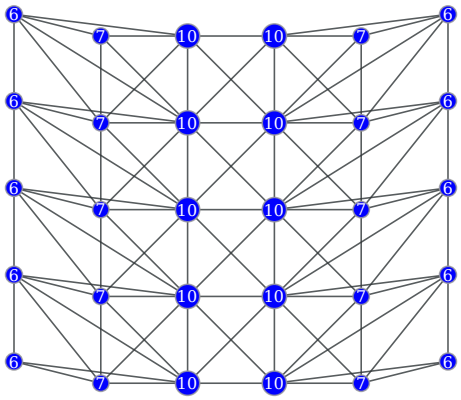


Figura 3.9: Grados de los vértices de $G(3, 4)$

Para estos grafos el procedimiento es muy similar al paso anterior, con la única diferencia de

que no hay vértices de grado 8, luego los vértices de grado 10 se pegarán entre sí y debemos hacerlo de manera análoga a como pegábamos los vértices de grado 10 con los de grado 8.
Estos grafos, al igual que los anteriores, son reconocibles en tiempo lineal.

3.2.3 $G(2, M)$

Para $G(2, M)$ la estrategia es similar con la diferencia de que en este caso los grados de los vértices son distintos
En este caso la primera etapa (pegar los ciclos de grado 6 con los ciclos de grado 7) es similar. Luego, si consideramos sólo los grafos inducidos por los de grado 12 junto con cada uno de los otros pares de ciclos pegados por separado, podemos pegarlos de la misma manera que hicimos anteriormente con los de grado 10.
Por último, como ya tenemos pegados los vértices de grado 6, 7 y 12 de los dos lados, es cuestión de ubicar los índices correctamente para que de los dos lados se peguen a la misma altura los vértices de grado 12, pero como estos forman un ciclo entre sí esto también se puede hacer linealmente. Luego podemos reconocer los $G(2, M)$ en tiempo lineal.

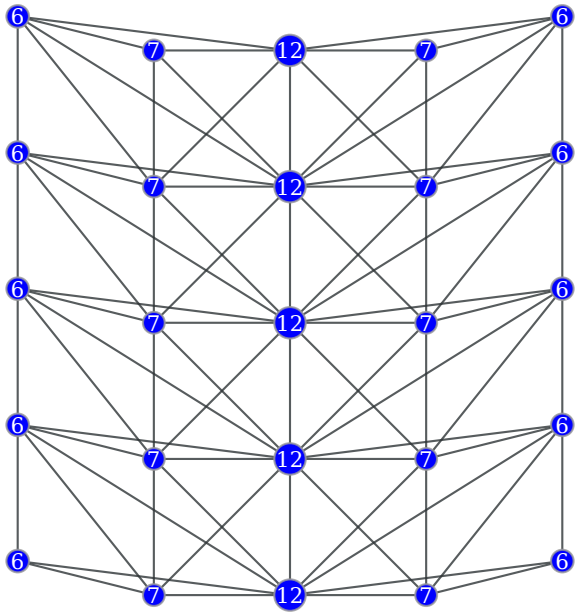


Figura 3.10: Grados de los vértices de $G(2, 4)$

3.2.4 $G(1, M)$

Los grafos $G(1, M)$ son un poco más difíciles de reconocer ya que están conformados por dos ciclos de longitud 7 y dos ciclos de longitud 9, pero todos los ciclos tienen intersección no vacía con todos los demás ciclos, por lo que no podemos en este caso separar en dos ciclos y luego pegarlos.

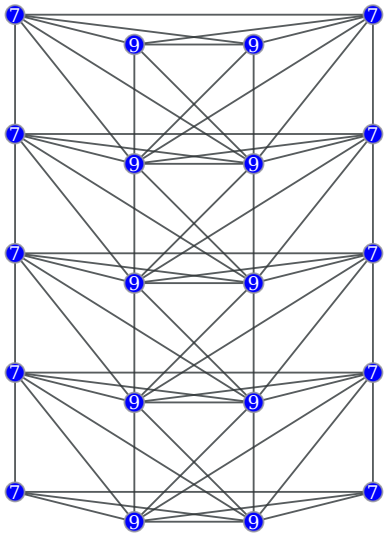


Figura 3.11: Grados de los vértices de $G(1,4)$

Estos grafos son de toda la familia, aquellos cuyo reconocimiento requiere de un algoritmo un poco más complicado, pero aún así se pueden reconocer en tiempo lineal.
Lo primero que hacemos es elegir arbitrariamente un vértice de grado 7 como $v_{-1,0}$. De sus vecinos de grado 7, que son sólo 3, hay uno sólo que va a compartir los 4 vecinos de grado 9, ese vértice será $v_{2,0}$. En todo momento cuando asumimos que hay un vértice que cumple con una condición, si ningún vértice cumple con la condición reconocemos que el grafo no es $G(1,M)$, y si asumimos que sólo uno cumple, pero cumple más de uno, nos vamos a dar cuenta cuando querramos ver si el morfismo biyectivo de vértices que dimos es efectivamente un isomorfismo de grafos.
Una vez que identificamos $v_{-1,0}$ y $v_{2,0}$ vamos a hacer uso de la propiedad de que $v_{0,i}$ y $v_{1,i}$ son indistinguibles, en el sentido de que sus vecindades coinciden, exceptuando por una arista que es la arista que los une. De esta manera, tenemos de entre los 4 vecinos de estos dos vértices, 6 posibilidades para elegir un conjunto de dos vértices que sean $v_{0,0}$ y $v_{1,0}$ por un lado, y $v_{0,1}$ y $v_{1,1}$ por el otro. Para esto vamos a probar todas las posibilidades.
Una vez que elegimos estos 6 vértices (que forman un K_6), existe sólo una forma de elegir $v_{-1,1}$, que es el único vecino de $v_{-1,0}$ de grado 7 que es también vecino de $v_{0,1}$ y de $v_{1,1}$. Análogamente hay una sóla forma de elegir $v_{2,1}$.
Ahora que elegimos $v_{-1,1}$ y $v_{2,1}$ hay una sóla forma de elegir $v_{0,2}$ y $v_{1,2}$ que es tomar los dos vértices de grado 9 que son vecinos de $v_{i,1}$ para todo $-1 \leq i \leq 2$. De esta forma seguimos con el mismo procedimiento hasta que encontramos el isomorfismo de grafos o el algoritmo encuentra que el grafo no es un $G(1,M)$.

3.2.5 $G(0,M)$

Para estos grafos vamos a utilizar un algoritmo parecido al de los $G(1,M)$.

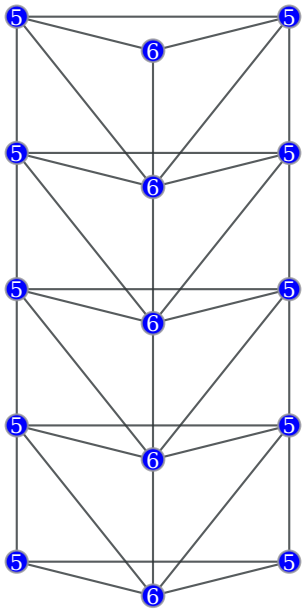


Figura 3.12: Grados de los vértices de $G(0,4)$

En este caso, vamos a tomar un vértice de grado 5 de manera aleatoria y denominarlo sin pérdida de generalidad $v_{-1,0}$. ahora de entre sus 3 vecinos de grado 5 vamos a probar con los 3 para ver cuál es $v_{1,0}$. Una vez elegido $v_{1,0}$, tenemos dos posibilidades para $v_{0,0}$ y vamos a probar con ambas. Ahora tenemos un K_3 comformado por $v_{-1,0}, v_{0,0}$ y$v_{1,0}$. Luego queda univocamente determinado $v_{0,1}$ como el único vértice de grado 6 que forma un K_4 con esos tres vértices.
A partir de este punto como los vértices de grado 6 forman un ciclo vamos completando el ciclo, y por último, cuando los tenemos todos identificados, vamos identificando $v_{-1,i}$ y $v_{1,i}$ para cada i en orden desde 1 hasta $M-1$ como los dos vértices que forman un K_4 con $v_{0,i}$ y $v_{0,i+1}$. Para distinguirlos entre ellos lo hacemos viendo que $v_{-1,i}$ es vecino de $v_{-1,i-1}$ y que $v_{1,i}$ es vecino de $v_{1,i-1}$.
De esta manera identificamos en tiempo lineal $G(0,4)$ y luego podemos identificar cualquier $G(N,M)$ en tiempo lineal, y como dividimos la identificación en una cantidad finita de casos (5) podemos correr los 5 algoritmos para identificar si el grafo que nos dan es $G(N,M)$ para cualquier valor de N en tiempo lineal corriendo 5 algoritmos lineales.

3.3 Familias de grafos H que convergen de a 3 vértices o más por paso

3.3.1 Definición de la familia

A continuación presentaremos una familia de grafos tales que al aplicarles el operador clique, convergen linealmente, de a $M \geq 3$ vértices por paso. Esta familia está compuesta por los grafos $H(N,M)$ que definiremos a continuación, siendo $K(H(N,M)) = H(N-1,M)$ cuando $N \geq 1$ y $H(0,M)$ autoclique, es decir, que $K(H(0,M)) = H(0,M)$. Definiremos estos grafos para $N \in \mathbb{Z}_{\geq 0}$ y $M \geq 4$, por lo que la convergencia será de a M vértices para cualquier valor de M que sea mayor o igual a 3.

Definición 3.2 *Los vértices de* $H(N, M)$ *son* $v_{i,j}$ *con* $i \in \{-1, 0, 1, \ldots, N, N+1\}$ *y* $j \in \{0, 1, \ldots, M-1\}$.
Cuando $N \geq 2$ *las aristas de* $H(N, M)$ *son los siguientes:*

- v_{i_0,j_0} *es adyacente a* v_{i_1,j_1} *si y sólo si* $|i_0 - i_1| \leq 1$, $|j_0 - j_1| \leq 1$, $i_0, i_1 \in \{0, 1, \ldots, n\}$ *y* $j_0, j_1 \in \mathbb{Z}_M$.
- $v_{-1,i}$ *es adyacente a* $v_{-1,i+1}, v_{0,i}, v_{0,i+1}, v_{1,i}$ *y* $v_{1,i+1}$ *con* $i \in \mathbb{Z}_M$.
- $v_{n+1,i}$ *es adyacente a* $v_{n+1,i+1}, v_{n,i}, v_{n-1,i}, v_{n-1,i+1}$ *y* $v_{n,i+1}$ *con* $i \in \mathbb{Z}_M$.

Cuando $N = 1$ *las aristas son los mismas más las aristas* $\{(v_{-1,i}, v_{2,i})\}$ *con* $i \in \mathbb{Z}_M$.
Cuando $N = 0$ *las aristas son* $\{\{v_{0,i}, v_{0,i+1}\}, \{v_{1,i}, v_{0,i}\}, \{v_{1,i}, v_{0,i+1}\}, \{v_{-1,i}, v_{0,i}\}, \{v_{-1,i}, v_{0,i+1}\}, \{v_{1,i}, v_{-1,i}\}, \{v_{-1,i}, v_{-1,i+1}\}, \{v_{1,i}, v_{1,i+1}\}\}$.
Además identificamos los vértices $(N-\alpha, 0)$ *con* (α, M) *por lo que estos grafos tienen la topología de una banda de Möebius*

En la Figura 3.13 presentamos algunos de los grafos de esta familia. Todos los grafos que veremos en las figuras son $H(N, M)$ con $M = 3$ y el primer y último nivel de las figuras (es decir, el de más arriba y el de más abajo) representan los mismos vértices, identificados de modo tal que la topología de dichos grafos sea una banda de Möebius.
Como se observa en la figura, estos grafos son localmente iguales a los grafos $G(N, M)$ y la única diferencia es cómo se identifican los vértices del primer nivel con los del último nivel, por lo que las cliques de los grafos H serán las mismas que las de los grafos G y la prueba de que $K(H(N, M)) = H(N-1, M)$ para $N \geq 1$ y que $H(0, M)$ es autoclique son análogas a las pruebas de la sección anterior.

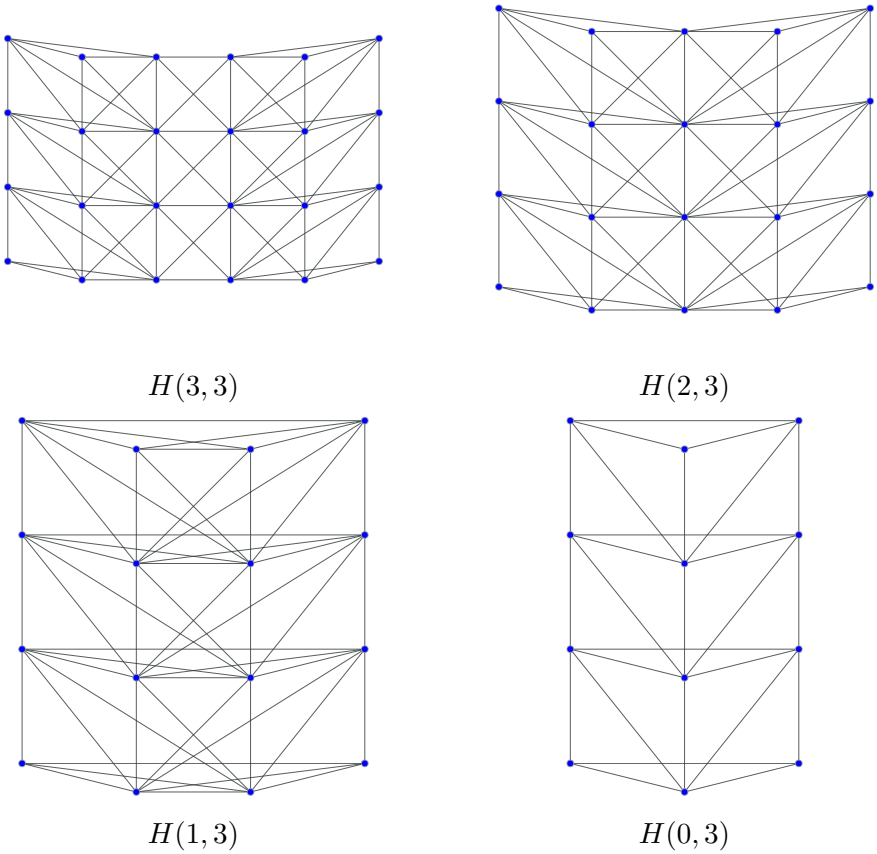


Figura 3.13: $H(N,3)$ **con** $N = \{3,2,1,0\}$

3.4 Reconocimiento de la familia de grafos H que convergen de a 3 vértices o más por paso

Para reconocer estos grafos es necesario descomponer el reconocimiento de los mismos en las mismas cinco categorías que para el caso anterior de los grafos G.

3.4.1 $H(N, M)$ con $N \geq 4$

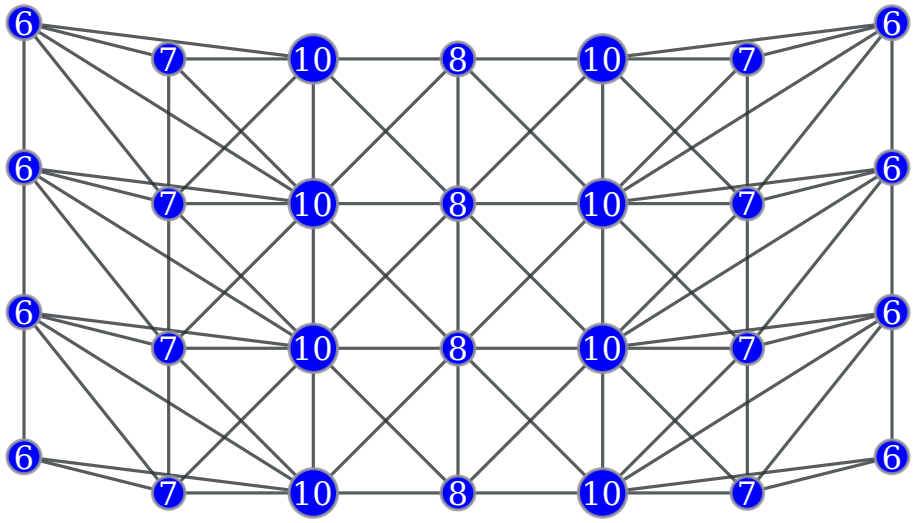


Figura 3.14: Grados de los vértices de $H(4, 3)$

En este caso, por la forma en la que se pegan los vértices, los vértices de grado 6 van a formar un sólo ciclo, y dado que el grafo que inducen estos vértices es un C_{2M} es fácil identificarlo. Lo mismo pasa con los de grado 7 y con los de grado 10, y la forma de ver si el grafo inducido por todos estos vértices es ismorfo al grafo que inducen en $H(N, M)$ los vértices de grado 6, 7 y 10 es la misma forma que con los $G(N, M)$ pegándolos.
Luego los vértices de grado 8 se pegarán por capas hasta que existan o bien un ciclo de M vértices, o un ciclo de $2M$ vértices (según la paridad de N) y para pegar estos vértices utilizamos el mismo algoritmo que utilizabamos para los grafos G, de buscar entre las 3 opciones que tienen aristas con el vértice $v_{i,0}$ (y el $v_{i+2,0}$ si el ciclo es de M vértices) correspondientes para ver cuál es el $v_{i+1,0}$ y a partir de ahí formar el ciclo que corresponda.

3.4.2 $H(3, M)$

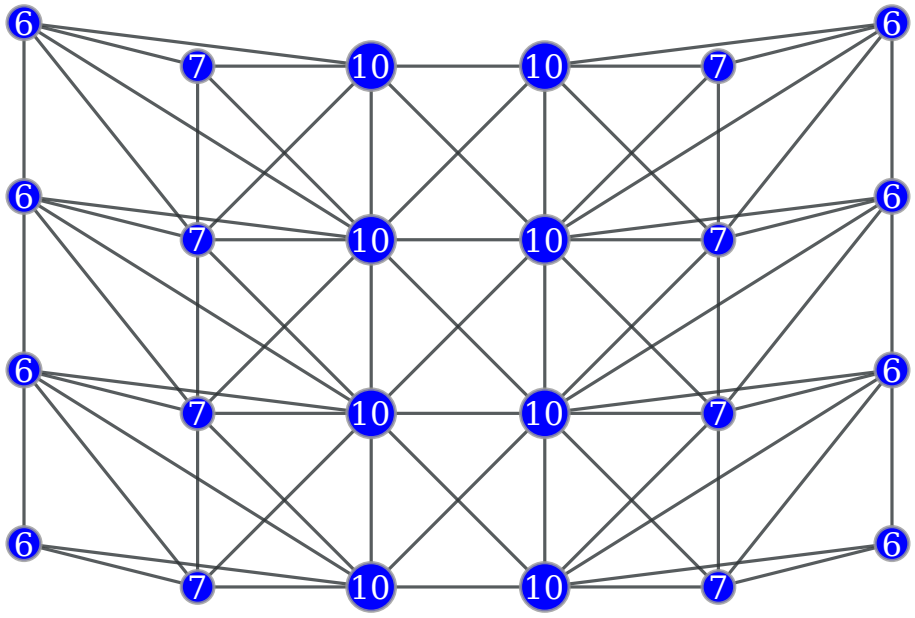


Figura 3.15: Grados de los vértices de $H(3, 3)$

En este caso la forma de identificar el grafo será análoga a la forma de identificar los $G(3, M)$ con la diferencia de que iremos identificando de a 1 ciclo en vez de identificar de a 2. La forma de ver que esto funciona es que localmente los grafos son iguales, y sólo difieren en cómo se pegan los extremos, luego el mismo algoritmo sirve para identificar estos grafos.

3.4.3 $H(2, M)$

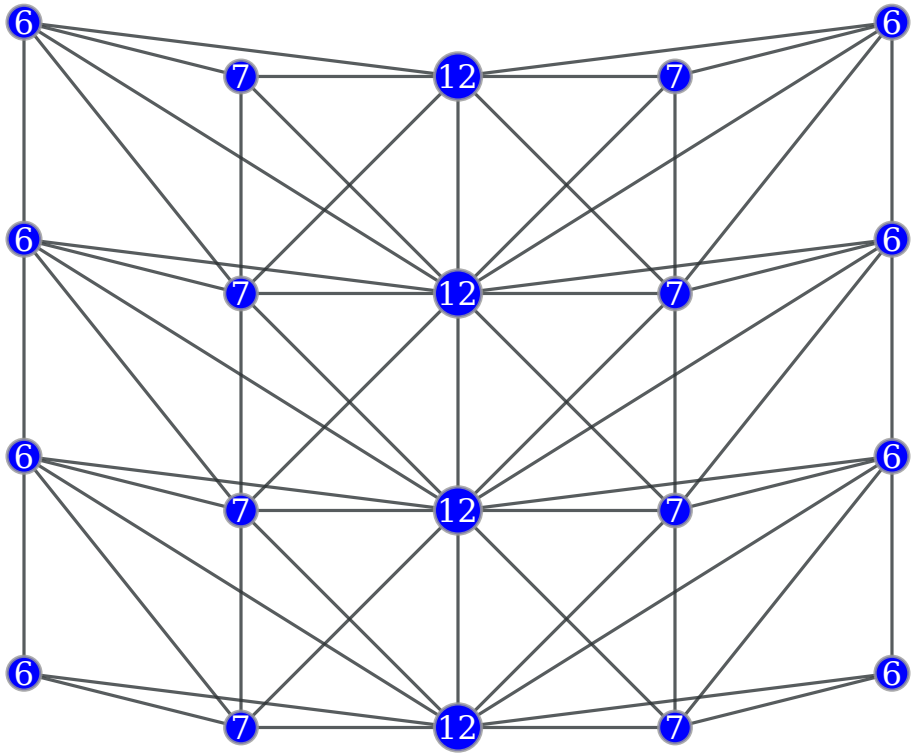


Figura 3.16: Grados de los vértices de $H(2, 3)$

En este caso, los vértices de grado 6 y 7 forman un ciclo que es identificable ya que el grafo inducido porlos vértices de grado 6 y el grafo inducido por los vértices de grado 7 no solo que contienen un ciclo sino que son un ciclo.
Los vértices de grado 12 se pueden pegar de la misma manera que se hacía en los grafos $G(2, M)$ ya que localmente los grafos son iguales y al haber una cantidad par de vértices de grado 7 se puede identificar para $v_{0,i}$ el $v_{2,i}$ a partir del ciclo de vértices de grado 7.

3.4.4 $H(1, M)$

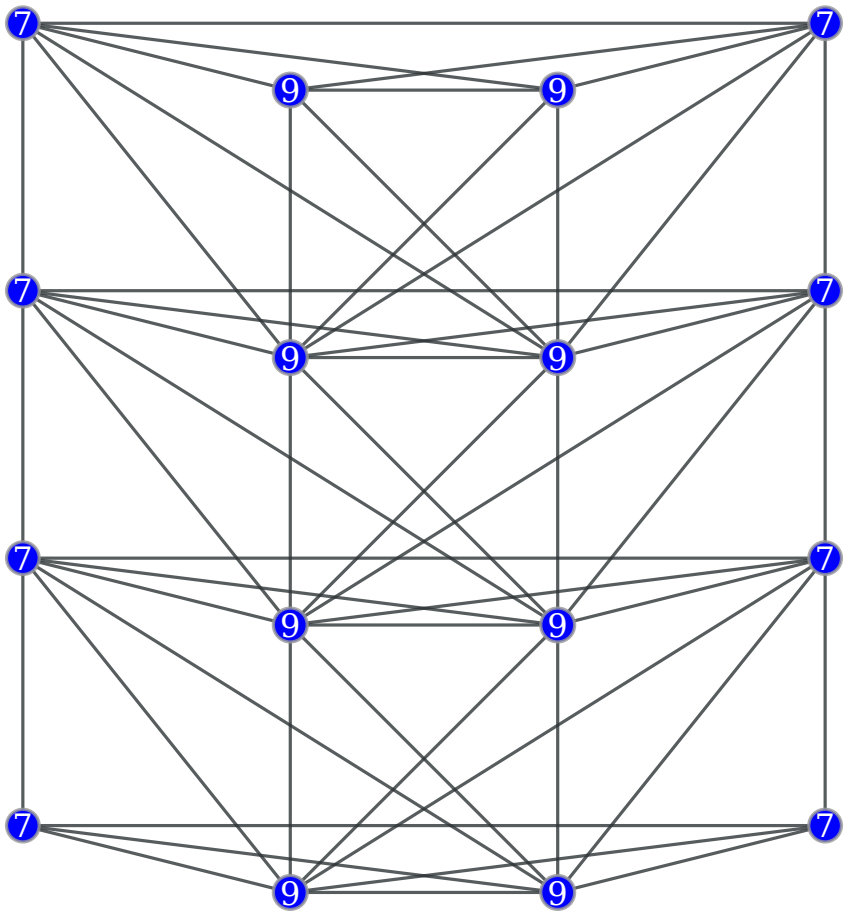


Figura 3.17: Grados de los vértices de $H(1, 3)$

Estos grafos son nuevamente los más difíciles de identificar. Primero tenemos que tener en cuenta que el grafo inducido por los vértices de grado 7 no es un ciclo, por lo que en este caso no vamos a identificar al grafo como varios ciclos que se pegan adecuadamente.
En este caso los grafos los identificamos, al igual que para los $G(1, M)$, buscando de a K_6 en K_6. Como esta búsqueda es local, y el grafo localmente es igual a $G(1, M)$, el mismo procedimiento sirve para identificar estos grafos.

3.4.5 $H(0, M)$

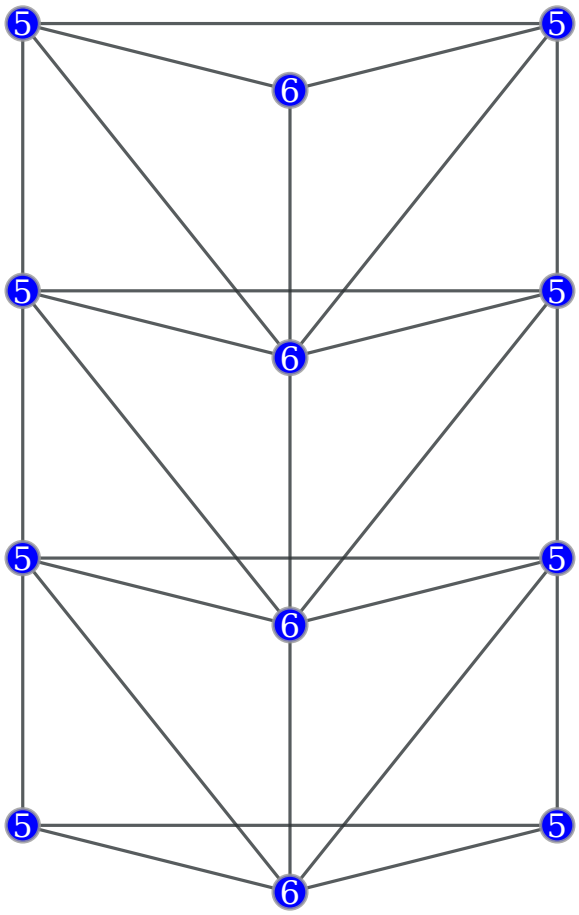


Figura 3.18: Grados de los vértices de $H(0, 3)$

Estos grafos los podemos identificar fácilmente formando el ciclo inducido por los vértices de grado 6 y luego pegandole los vértices que completan los K_4 a los costados, y viendo que estos vértices de grado 5 estén unidos acordemente.

3.4.6 Observaciones generales

En todos los grafos la forma de identificarlos es primero intentar generar el grafo que queremos generar, para poder mapear a través de un morfismo biyectivo los vértices del grafo dado con los del $H(N, M)$, y luego verificar que dicho morfismo biyectivo sea efectivamente un isomorfismo de grafos.

Capítulo 4

Superficies

A lo largo de este capítulo presentaremos superficies que son auto-cliques y daremos algunas propiedades de las mismas

4.1 Superficies

Definimos los grafos $S(N, M)$ con $N \geq 5$ y $M \geq 4$ de la siguiente manera:

Definición 4.1 *Los vértices de* $S(N, M)$ *son* $\{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, M-1\}$ *y las aristas son:*
v_{i_0,j_0} *es adyacente a* v_{i_1,j_1} *si y sólo si* $|i_0 - i_1| \leq 1$, $|j_0 - j_1| \leq 1$, $i_0, i_1 \in \{0, 1, \ldots, N\}$ *y* $j_0, j_1 \in \{0, 1, \ldots, M\}$.
Además tenemos tres formas de pegar los bordes que son:

- *Toros: Identificamos los vértices* $(0, \alpha)$ *con* (N, α) *para* $\alpha = 0, 1, \ldots, M$ *y* $(\beta, 0)$ *con* (β, M) *para* $\beta = 0, 1, \ldots, N$.
- *Botellas de Klein: Identificamos los vértices* $(0, \alpha)$ *con* (N, α) *para* $\alpha = 0, 1, \ldots, M$ *y* $(\beta, 0)$ *con* $(N - \beta, M)$ *para* $\beta = 0, 1, \ldots, N$.
- *Planos proyectivos: Identificamos los vértices* $(0, M - \alpha)$ *con* (N, α) *para* $\alpha = 0, 1, \ldots, M$ *y* $(\beta, 0)$ *con* $(N - \beta, M)$ *para* $\beta = 0, 1, \ldots, N$.

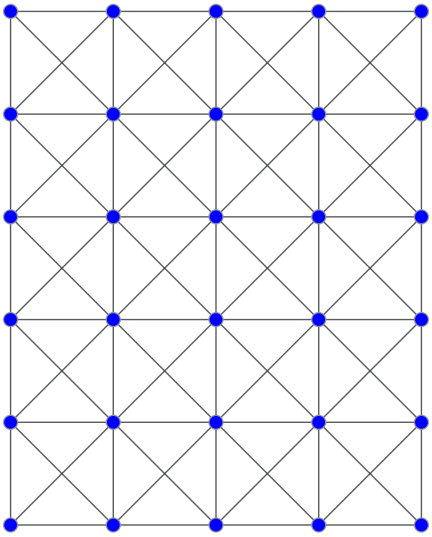


Figura 4.1: $S(4,5)$

En la Figura 4.1 podemos ver cómo sería el más chico de estos grafos, sin visualizar la forma en la que se pegan los vértices (lo cual veremos más adelante).
Para estos grafos, las únicas cliques que tendremos serán los K_4 que se unen los unos a los otros y es fácil ver que son autocliques. De hecho $K^2(G) = G$ por ser estos grafos Clique-Helly, y no tener vértices dominados, pero también podemos observar que $K(G) = G$

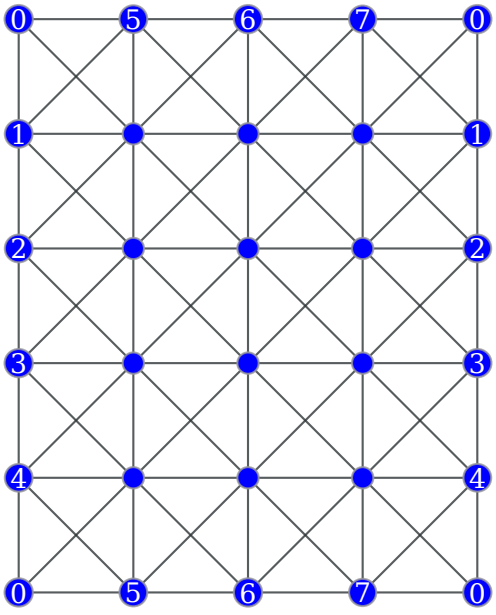


Figura 4.2: $S(4,5)$ **como Toro**

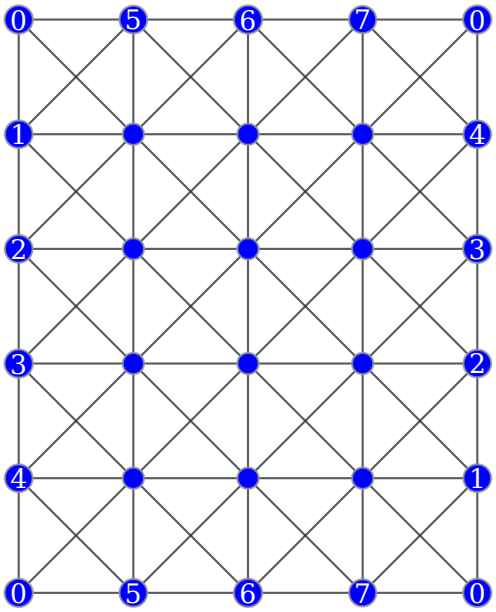


Figura 4.3: $S(4,5)$ **como Botella de Klein**

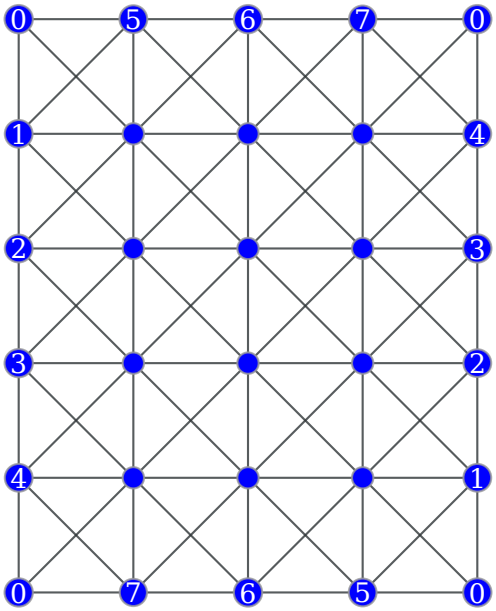


Figura 4.4: $S(4,5)$ **como Plano Proyectivo**

En la Figura 4.2 podemos ver cómo se identifican los vértices cuando el grafo se pega como toro, en la Figura 4.3 como Botella de Klein y en la Figura 4.4 como Plano Proyectivo.
También se puede hacer una suma conexa de cualquiera de estos grafos borrándoles un vértice a dos cualesquiera y pegándolos por la frontera. Esto se puede iterar, es decir, hacer la suma

conexa las veces que sea, y así podemos obtener cualquier superficie excepto la esfera. Cualquiera de estos grafos seguirá siendo Clique-Helly y sin puntos dominados por lo que para cualquiera de estos grafos G se tiene que $K^2(G) = G$.
Si hacemos estos agujeros el grafo se vería como en la Figura 4.5

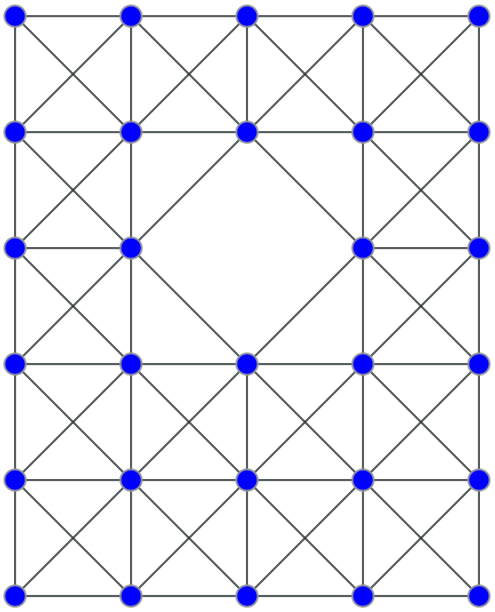


Figura 4.5: $S(4, 5)$ **con un agujero**

Cualquiera de estos grafos S al borrarle uno o más vértices se convierte en una superficie con borde. Por [12] sabemos que cada dos pasos de K se borrará un ciclo y con ellos una capa de K_4 alrededor de cada agujero.

Capítulo 5

Conclusiones y trabajo futuro

A lo largo de esta tesis trabajamos con grafos clique, estudiamos el comportamiento de aplicar iteradas veces el operador clique y dimos ejemplos de grafos que tienen la topología de cualquier superficie distinta de la esfera y que son auto-clique.

Definimos el K-comportamiento. Dimos ejemplos de grafos que son K-divergentes y que divergen linealmente, y citamos referencias donde demuestran que existen grafos K-divergentes con divergencia polinomial de cualquier orden. También dimos ejemplos de grafos que se anulan linealmente, así como demostramos que los árboles se anulan, eliminando las hojas en dos pasos, lo que demuestra que para cualquier secuencia S_i decreciente, con la diferencia de sus números decreciente y que termina en 1 existe un ejemplo de grafos tal que cada dos pasos del operador clique se obtiene el siguiente número de la secuencia como número de vértices del grafo. Estos grafos son siempre árboles a los que cada dos pasos se les van eliminando hojas. Este resultado es un corolario de un resultado más general de Escalante [12], pero en esta tesis presentamos una demostración simple para el caso de árboles.
Para grafos K-periódicos demostramos que existen grafos que convergen linealmente, un resultado que no se conocía hasta el momento, y dimos dos ejemplos de estas familias de grafos.

También dimos ejemplos de grafos que son topológicamente toros, botellas de Klein y planos proyectivos. Para estos grafos, vimos que son autoclique y vimos cómo pegar estos grafos para generar cualquier superficie topológica que no sea la esfera.

En el área de algoritmia y complejidad computacional, pudimos encontrar algoritmos lineales que reconocen los grafos del Capítulo 3 de esta tesis, y queda como trabajo futuro encontrar un algoritmo que reconozca los grafos del Capítulo 4.
Queda también como trabajo futuro intentar encontrar familias de grafos que convergan de a 2 vértices por paso o de a 1 vértice por paso a un grafo no trivial, para completar la teoría del Capítulo 3.

Bibliografía

[1] L. Alcón, L. Faria, C.M.H. de Figueiredo and M. Gutierrez, The complexity of clique graph recognition, *Theoretical Computer Science 410* (2009), 2072-2083.

[2] H. Bandelt and E. Prisner, *Clique graphs and Helly graphs*, J. Combin. Theory B 51 (1991), 34–45.

[3] C. Bornstein and J. Szwarcfiter, *Clique graphs of chordal and path graphs*, SIAM J. Discrete Math. 38 (1971), 535–545.

[4] C. Bornstein and J. Szwarcfiter, *On clique convergent graphs*, Graphs and Combinatorics 11 (1995), 213–220.

[5] A. Brandstadt, V. Bang Le and J. Spinrad, *Graph Classes: A Survey* (SIAM, 1999).

[6] A. Brandstadt, V. Chepoi, F. Dragan and V. Voloshin, *Dually chordal graphs*, SIAM J. Discrete Math. 11 (1999), 437–455.

[7] L. Chong-Keang and P. Yee-Hock, *On graphs without multicliqual edges*, J. Graph Theory 5 (1981), 443–451.

[8] S. Cook, *The complexity of theorem-proving procedures*, Proc. 3rd Ann. ACM Symp. on Theory of Computing Machinery, New York (1971), 151–158.

[9] Pablo De Caria, The images of the clique operator and its square are different, *Journal of Graph Theory 77* (2014), 39-57.

[10] C. Deng and C. Lim, *A class of clique-closed graphs*, Discrete Math. 127 (1994), 131–137.

[11] G. Durán and M. Lin, *Clique graphs of Helly circular-arc graphs*, Ars Combinatoria 60 (2001), 255–271.

[12] F. Escalante, *Uber iterierte Clique-Graphen*, Abhandlungender Mathematischen Seminar der Universität Hamburg 39 (1973), 59–68.

[13] M. Garey and D. Jonhson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman and Company, San Francisco, 1979).

[14] M. Gutiérrez, *Tree-clique graphs*, In J. Szwarcfiter, editor, Workshop International de Combinatória, Rio de Janeiro, Universidade Federal do Rio de Janeiro (1996), 7–26.

[15] M. Gutiérrez and J. Meidanis, *Recognizing clique graphs of directed edge path graphs*, manuscrito.

[16] M. Gutiérrez and L. Oubiña, *Minimun proper interval graphs*, Discrete Math. 142 (1995), 77–85.

[17] M. Gutiérrez and R. Zucchello, *Grafos ACI: Una generalización de los grafos de intervalos propios*, manuscrito.

[18] F. Harary, *Graph Theory* (Addison-Wesley, 1969).

[19] S. Hedetniemi and P. Slater, *Line graphs of triangleless graphs and iterated clique graphs*, Lecture Notes in Math., Springer-Verlag (1972) 139–147.

[20] B. Hedman, *Clique graphs of time graphs*, J. Combin. Theory B 37 (1984), 270–278.

[21] R. Karp, *Reducibility among combinatorial problems*, In R. Miller and J. Thatcher (eds.), Complexity of Computer Computations, Plenum Press, New York (1972)f, 85–103.

[22] F. Larrión and V. Neumann-Lara, *On clique divergent graphs with linear growth*, manuscrito.

[23] M. Lin, *Grafos self-clique y otras clases de grafos clique*, Tesis Doctoral, Departamento de Computación, FCEyN-UBA (2001).

[24] T. McKee and F. McMorris, *Topics in Intersection Graph Theory* (SIAM Monographs on Discrete Mathematics and Applications, 1999).

[25] C. Papadimitriou, *Computational Complexity* (Addison-Wesley, 1994).

[26] E. Prisner, *Hereditary clique-Helly graphs*, J. Combin. Math. Combin. Comput. 14 (1993), 216–220.

[27] E. Prisner and J. Szwarcfiter, *Recognizing clique graphs of directed and root path graphs*, Discrete App. Math. 94 (1999), 321–328.

[28] F. Roberts and J. Spencer, *A characterization of clique graphs*, J. Combin. Theory 10 (1971), 102–108.

[29] J. Simon, *On some central problems in computational complexity*, Ph.D. Thesis, Cornell University (1977).

[30] J. Szwarcfiter, *A survey on clique graphs*, Brazilian Summer School on Combinatorics and Algorithms, Fortaleza, Brasil (2001), 103–132.

[31] L. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science 8 (1979), 189–201.

[32] L. Valiant, *The complexity of enumeration and reliability problems*, SIAM Journal on Computing Vol. 8. N. 3 (1979), 410–421.