



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO
DEPARTAMENTO DE COMPUTACIÓN

Particiones de cografías en gráficas multipartitas completas

TESIS QUE PRESENTA

Sergio Eduardo Juárez Martínez

PARA OBTENER EL GRADO DE

Maestro en Ciencias en Computación

DIRECTORES DE LA TESIS

Dr. César Hernández Cruz
Dra. Dolores Lara Cuevas

México, Ciudad de México

Octubre 2020

Abstract

Graph Theory is the branch of Discrete Mathematics that studies the mathematical structures known as graphs. A graph G is an ordered pair of disjoint sets (V, E) such that E is a set of unordered pairs of elements of V . We call V the set of vertices of G and E the set of edges of G . If an element of E contains two vertices, we say that those vertices are adjacent. In this thesis, we work with a particular kind of graphs known as cographs, that can be characterized in several ways. One of this ways is that cographs are the graphs that does not contain the path on 4 vertices, P_4 , as an induced subgraph.

A classic problem in Graph Theory is graph coloring, which consists in deciding if it is possible to label the vertices of a graph with a given number of different labels, also known as colors, in a way such that if two vertices are adjacent, then those two vertices have different labels. A generalization of the graph coloring problem is the matrix partition problem, which consists in deciding if the vertices of a graph can be labeled with a given number of labels in a way such that vertices with the same label satisfy a homogeneity property. In this thesis we address a problem of matrix partitions. We study the classes of cographs defined in the following way. Given an integer i greater than or equal to one, the class M_i is the class of the cographs whose set of vertices accepts a partition in i parts such that each part induces a multipartite complete graph. The class M_1 , which is the class of multipartite complete cographs, has been widely studied. However, the classes M_i for values of i greater than one have not been previously studied. The base of our research is the study of the class M_2 . For this study, we use the research done about polar cographs as guide. We characterize the class M_2 through its set of minimal obstructions, we present an algorithm to recognize its elements and a certifying algorithm. We also study a set of subclasses of the class M_2 that we call the classes (α, β) - M_2 . Similarly, we characterize the class M_3

4

through its set of minimal obstructions and present two families of minimal obstructions for any class M_i .

Resumen

La Teoría de Gráficas es la rama de las Matemáticas Discretas encargada del estudio de los objetos matemáticos conocidos como gráficas. Una gráfica G es una pareja ordenada de conjuntos ajenos (V, E) tal que E es un conjunto de parejas no ordenadas de elementos de V . Llamamos a V el conjunto de vértices de G y a E el conjunto de aristas de G . Si dos elementos de V forman una pareja que está en el conjunto E , decimos que estos son adyacentes. Con frecuencia, las gráficas son representadas con un dibujo en el que los vértices aparecen como puntos o pequeños círculos y las aristas como líneas que unen a los vértices adyacentes. En esta tesis trabajamos con un tipo particular de gráficas conocidas como cográficas, que pueden ser caracterizadas de múltiples maneras. Una de ellas es que son las gráficas que no tienen a la trayectoria de 4 vértices, P_4 , como subgráfica inducida.

Un problema clásico en la Teoría de Gráficas es la coloración de gráficas que consiste en determinar si los vértices de una gráfica se pueden etiquetar con un número determinado de etiquetas diferentes, también llamadas colores, de forma tal que, si dos vértices son adyacentes, estos tienen etiquetas diferentes. Una generalización de las coloraciones de gráficas son las particiones matriciales, que consisten en determinar si los vértices de una gráfica se pueden etiquetar con un número determinado de etiquetas diferentes de manera tal que los vértices con la misma etiqueta cumplan con una propiedad de homogeneidad. En esta tesis abordamos un problema de particiones matriciales. Estudiamos a las clases de cográficas que se definen de la siguiente manera. Dado i , un entero mayor o igual a uno, la clase M_i es la clase de las cográficas cuyo conjunto de vértices acepta una partición en i partes tal que cada parte induce una gráfica multipartita completa. La clase M_1 , que es la clase de las cográficas multipartitas completas, ha sido ampliamente estudiada. Sin embargo, las clases M_i para valores de i mayores a uno no han sido estudiadas con anterioridad. Nuestra investigación tiene como base el estudio

de la clase M_2 , para el cuál tomamos como guía la investigación realizada sobre las cográficas polares. Caracterizamos a la clase M_2 a través de su conjunto de obstrucciones mínimas, presentamos un algoritmo para reconocer a sus elementos y un algoritmo certificador y estudiamos a un conjunto de subclases de esta clase a las que llamamos clases (α, β) - M_2 . De igual manera, caracterizamos a la clase M_3 a través de su conjunto de obstrucciones mínimas y proporcionamos dos familias de obstrucciones mínimas para cualquier clase M_i .

Índice general

1. Introducción	15
2. Antecedentes	17
2.1. Introducción a la Teoría de Gráficas	17
2.1.1. Conceptos básicos y notación	17
2.1.2. Operaciones en gráficas	19
2.1.3. Algunas gráficas distinguidas	20
2.1.4. Caminos y conexidad	21
2.1.5. Árboles	21
2.1.6. Clases de gráficas hereditarias	23
2.2. Cográficas	24
2.2.1. Caracterización de las cográficas	24
2.2.2. Coárboles	24
3. Estado del arte	27
3.1. Cografías polares	27
3.2. Obstrucciones mínimas para cográficas $(s, 1)$ -polares	30
3.3. Obstrucciones mínimas para cográficas 2-polares	32
3.4. Un algoritmo lineal para el reconocimiento de cográficas	35
4. Resultados	39
4.1. Términos y algoritmos generales	40
4.1.1. Coárbol binario	40
4.1.2. Algoritmo para generar un coárbol binario	40
4.1.3. Algoritmo para generar todos los coárboles binarios de una gráfica	42
4.1.4. Subcoárbol	45
4.1.5. Algoritmo para encontrar obstrucciones mínimas	47

4.2.	La clase M_2	52
4.2.1.	Obstrucciones mínimas	54
4.2.2.	Reconocimiento de la clase M_2	59
4.2.3.	Algoritmo certificador	59
4.3.	Las clases (α, β) - M_2	70
4.3.1.	Definición y clases conocidas	70
4.3.2.	Conjunto de parejas mínimas	72
4.3.3.	Cálculo del conjunto de parejas mínimas de una gráfica	73
4.3.4.	Algoritmo para generar obstrucciones mínimas	78
4.3.5.	Resultados del algoritmo	82
4.4.	Particiones en más de dos partes	88
4.4.1.	Obstrucciones mínimas de la clase M_3	88
4.4.2.	Familia O de obstrucciones	98
4.4.3.	Familia P de obstrucciones	105
5.	Conclusiones y trabajo futuro	107
5.1.	Trabajo a futuro	108
A.	Obstrucciones mínimas de algunas clases (α, β)-M_2	111
A.1.	Algunas obstrucciones mínimas de la clase $(3, 3)$ - M_2	111
A.2.	Algunas obstrucciones mínimas de la clase $(3, 4)$ - M_2	111
A.3.	Algunas obstrucciones mínimas de la clase $(3, 5)$ - M_2	112
A.4.	Algunas obstrucciones mínimas de la clase $(3, 6)$ - M_2	113
A.5.	Algunas obstrucciones mínimas de la clase $(3, 7)$ - M_2	114
A.6.	Algunas obstrucciones mínimas de la clase $(4, 4)$ - M_2	115
A.7.	Algunas obstrucciones mínimas de la clase $(4, 5)$ - M_2	115
A.8.	Algunas obstrucciones mínimas de la clase $(4, 6)$ - M_2	116
B.		119
C.	Familias O_n de obstrucciones	121

Índice de figuras

2.1.	Ejemplo de la representación de una gráfica.	18
2.2.	Ejemplo de una subgráfica y de una subgráfica inducida de la gráfica de la Figura 2.1.	19
2.3.	Las gráficas K_4 , P_4 y C_4	20
2.4.	Ejemplo de un árbol arraigado. El vértice distinguido como raíz aparece como tal.	22
2.5.	El nodo a es ancestro de los nodos b, c y d . Los nodos b, c y d son descendientes de a . El ancestro común más profundo de los nodos c y d es el nodo a . El nodo b es hijo del nodo a y éste a su vez es padre del nodo b . Los nodos c y d son hojas del árbol. Los nodos a y b son nodos internos.	23
2.6.	Una cográfica y su coárbol.	25
3.1.	Obstrucciones mínimas para las gráficas polares.	28
3.2.	Obstrucciones mínimas para las gráficas monopolares.	29
3.3.	Obstrucciones mínimas para las gráficas polares.	30
3.4.	Obstrucciones mínimas para las cográficas 2-polares obtenidas con el Lema 3.3.1.	32
3.5.	Obstrucciones mínimas de las cográficas 2-polares con 7 vértices.	33
3.6.	Familia A de obstrucciones mínimas de las cográficas 2-polares con 8 vértices.	33
3.7.	Obstrucciones mínimas de las cográficas 2-polares con 9 vértices.	34
3.8.	Familia B de obstrucciones mínimas de las cográficas 2-polares con 8 vértices.	34
4.1.	(b) y (c) son dos coárboles binarios diferentes que representan a la cográfica (a).	41

4.2.	Ejemplo de la ejecución del Algoritmo 3. A la izquierda se muestra el cóarbol original, mientras se marca con gris el nodo que se está procesando. A la derecha aparece el cóarbol binario que se va construyendo.	43
4.3.	El cóarbol (b) es subcóarbol binario del cóarbol (a). Las etiquetas en los nodos de ambos cóarboles binarios indican la asignación de los nodos de (b) a los nodos de (a).	45
4.4.	Obstrucciones mínimas para la clase M_2	55
4.5.	H_1, H_2 y H_3 son los cóarboles binarios correspondientes a la obstrucción H . El cóarbol binario I_1 es el único que corresponde a la obstrucción I . El cóarbol binario J_1 es el único que corresponde a la obstrucción J	60
4.6.	Ejemplos del resultado de la ejecución del Algoritmo 9.	61
4.7.	Ejemplo de la ejecución del Algoritmo 10. Se muestran en color gris los nodos del árbol que están siendo procesados. Los colores de las hojas corresponden a los colores que asigna el algoritmo.	64
4.8.	Ejemplo de la ejecución del Algoritmo 10. Se muestran en color gris los nodos del árbol que están siendo procesados. Los colores de las hojas corresponden a los colores que asigna el algoritmo.	65
4.9.	Ejemplos del resultado de la ejecución del Algoritmo 10 en los que se encuentra una obstrucción.	65
4.10.	Ejemplo de la ejecución del Algoritmo 11. Se muestran en color gris los nodos del árbol que están siendo procesados. El procesamiento de las hojas hermanas se realiza en una sola imagen. Los colores de las hojas corresponden a los colores que asigna el algoritmo.	68
4.11.	Ejemplo del resultado de la ejecución del Algoritmo 11 para cóarboles que incluyen una obstrucción.	70
4.12.	Obstrucciones mínimas para la clase $(1, \infty)-M_2$	71
4.13.	Obstrucción mínima de la clase $(1, 1)-M_2$	83
4.14.	Algunas obstrucciones mínimas de la clase $(1, 2)-M_2$	83
4.15.	Algunas obstrucciones mínimas de la clase $(1, 3)-M_2$	83
4.16.	Algunas obstrucciones mínimas de la clase $(1, 4)-M_2$	83
4.17.	Algunas obstrucciones mínimas de la clase $(1, 5)-M_2$	84
4.18.	Algunas obstrucciones mínimas de la clase $(2, 2)-M_2$	84
4.19.	Algunas obstrucciones mínimas de la clase $(2, 3)-M_2$	85

4.20. Algunas obstrucciones mínimas de la clase $(2, 4)-M_2$	86
4.21. Familia O_3 de obstrucciones mínimas de la clase M_3	90
4.22. Familia Q_3 de obstrucciones mínimas de la clase M_3	91
4.23. Ejemplos de O -componentes $(G_1, G_2, G_3$ y $G_4)$ y de gráficas que no son O -componentes $(G_5, G_6, G_7$ y $G_8)$	100
4.24. La gráfica G es una O -componente que contiene a $K_1, K_2, K_3,$ K_4 y K_6 . En rojo se marcan los vértices que se agregan a A_1 . Se puede apreciar que el resto de los vértices inducen la gráfica $K_1 + K_2 + K_3 + K_4$	102
4.25. La gráfica G es una O -componente que contiene a $K_1, K_2, K_3,$ K_4 y a K_6 . En rojo se marcan los vértices que se agregan a A_1 . A la derecha se muestra $G - A_1$. En azul se muestran los vértices de una O -obstrucción que contiene a K_1, K_2, K_3 y a K_4	104
4.26. La gráfica G es una O -componente que contiene a $K_1, K_2, K_3,$ K_4 y a K_6 . En rojo se marcan los vértices que se agregan a A_1 . A la derecha se muestra $G - A_1$. En azul se muestran los vértices de una O -obstrucción que contiene a K_1, K_2, K_3 y a K_4	104
C.1. Familia O_4 de obstrucciones. Parte 1.	121
C.2. Familia O_4 de obstrucciones. Parte 2.	122
C.3. Familia O_4 de obstrucciones. Parte 3.	123
C.4. Familia O_4 de obstrucciones. Parte 4.	124
C.5. Familia O_4 de obstrucciones. Parte 5.	125

Índice de tablas

B.1. Resumen de las obstrucciones mínimas encontradas con el Algoritmo 16	119
---	-----

Capítulo 1

Introducción

Una gráfica G es una pareja ordenada de conjuntos ajenos (V, E) tal que E es un conjunto de parejas no ordenadas de elementos de V . Llamamos a V el conjunto de vértices de G y a E el conjunto de aristas de G . Decimos que dos vértices de G son adyacentes si la pareja formada por estos está en E . Sea V' un subconjunto de V , la subgráfica de G inducida por V' es la gráfica que tiene como conjunto de vértices a V' y como conjunto de aristas al subconjunto de E de las parejas no ordenadas de elementos de V' . Una gráfica multipartita completa es una gráfica cuyo conjunto de vértices acepta una partición (A_1, A_2, \dots, A_n) tal que, para cualesquiera enteros diferentes $1 \leq i, j \leq n$, los vértices en A_i no son adyacentes entre sí pero sí son adyacentes a cada uno de los vértices en A_j . Un clan es un conjunto de vértices de G tal que todos sus elementos son adyacentes entre sí.

Una clase hereditaria de gráficas C es un conjunto de gráficas tal que, si C contiene a $G = (V, E)$, entonces, para cualquier subconjunto V' de V , C contiene a la subgráfica de G inducida por V' . Toda clase hereditaria de gráficas C puede ser caracterizada a través de un conjunto de gráficas S tal que toda gráfica en C no tiene a ninguna gráfica de S como subgráfica inducida. Llamamos a S el conjunto de obstrucciones mínimas de C . Las cografías son la clase hereditaria de gráficas cuyo conjunto de obstrucciones mínimas contiene únicamente a la trayectoria de 4 vértices, P_4 .

En 1990, Damascke, P. [1] mostró que cualquier clase hereditaria de cografías puede ser caracterizada por un conjunto finito de obstrucciones mínimas.

En esta tesis estudiamos a las clases de cografías que se definen de la siguiente manera. Sea i un entero mayor o igual a uno, la clase M_i es la clase de las cografías cuyo conjunto de vértices acepta una partición en

i partes tal que cada parte induce una gráfica multipartita completa. Nos referimos a estas clases en conjunto como las clases M_i . Este problema no ha sido estudiado con anterioridad. Sin embargo, utilizamos la investigación realizada acerca de las cografías polares como guía para el estudio de la clase M_2 , que sirve como base de nuestra investigación.

Los resultados principales de nuestra tesis son los siguientes. Presentamos un algoritmo que, dada una clase de cografías fija, representada a través de su conjunto de obstrucciones mínimas, es capaz de determinar si una cografía pertenece a dicha clase en tiempo lineal con respecto al orden de la gráfica. Caracterizamos a la clase M_2 a través de su conjunto de obstrucciones mínimas. Presentamos un algoritmo de tiempo lineal para el reconocimiento de los elementos de la clase M_2 . Presentamos un algoritmo certificador de tiempo lineal para la clase M_2 . Tomando como base el estudio de las cografías (s, k) -polares, estudiamos un conjunto de subclases de la clase M_2 a las que llamamos clases (α, β) - M_2 . Como resultado principal de este estudio, presentamos un algoritmo que, dados tres enteros α , β y n mayores o iguales a uno, genera las obstrucciones mínimas de la clase (α, β) - M_2 con hasta n vértices. Presentamos el conjunto de obstrucciones mínimas de la clase M_3 . Finalmente, presentamos dos familias de obstrucciones mínimas para cualquier clase M_i . Los algoritmos presentados en la sección de resultados fueron implementados en lenguaje C++. El código se encuentra disponible en <https://github.com/serj1495/ParticionesDeCografiasEnGraficasMultipartitasCompletas>.

La tesis está organizada de la siguiente manera. En el Capítulo 2 se presentan de manera detallada y formal las definiciones necesarias para entender el problema que resolvemos con nuestro trabajo de tesis. Estas definiciones proporcionan una introducción a la Teoría de Gráficas y, dentro de esta área, a las cografías, que son el tipo de gráficas con las que trabajamos en todo el documento. En el Capítulo 3 presentamos los resultados en la investigación de las cografías polares que dan forma a nuestro estudio de la clase M_2 . En el Capítulo 4, presentamos los resultados de nuestra investigación que comprenden un conjunto de lemas, teoremas, algoritmos y listas de obstrucciones mínimas para varias clases hereditarias de cografías. La correctitud de cada uno de nuestros resultados se demuestra de manera formal. Finalmente, en el Capítulo 5, presentamos las conclusiones de nuestra investigación a la vez que proponemos varias formas de continuar con la investigación que iniciamos con este trabajo de tesis.

Capítulo 2

Antecedentes

En este capítulo proporcionamos las definiciones necesarias para presentar el problema principal de nuestro trabajo de tesis. Primero establecemos definiciones y notación de Teoría de Gráficas, área a la cual pertenece nuestra investigación. Posteriormente presentamos a las cográficas, la clase de gráficas con las que trabajamos a lo largo de todo el documento.

2.1. Introducción a la Teoría de Gráficas

A continuación proporcionamos conceptos básicos del área de Teoría de Gráficas. Dado que nuestra investigación pertenece a dicha área, estos conceptos serán utilizados a lo largo de todo el documento. De manera general, las definiciones que se presentan fueron tomadas de Bollobás [2]. Algunas de éstas pueden diferir de las proporcionadas en Bollobás [2] de forma no sustancial.

2.1.1. Conceptos básicos y notación

Una *gráfica* G es una pareja ordenada de conjuntos ajenos (V, E) tal que E es subconjunto del conjunto de parejas no ordenadas de elementos de V . Decimos que V es el conjunto de *vértices* de G y que E es el conjunto de *aristas* de G .

Si G es una gráfica, denotamos a su conjunto de vértices como $V(G)$ y a su conjunto de aristas como $E(G)$. Sin embargo, cuando G es la única gráfica de la que estamos hablando, podemos referirnos a su conjunto de vértices

simplemente como V y a su conjunto de aristas como E .

Dada una arista $\{x, y\}$ de una gráfica, decimos que $\{x, y\}$ **une** a los vértices x y y y la denotamos como xy . Los vértices x y y son los **extremos** de dicha arista. Si $xy \in E(G)$, entonces x y y son **adyacentes** en G . Si el contexto es claro, simplemente decimos que x y y son adyacentes. Notemos que la arista xy es exactamente la misma arista que yx . Un vértice que no es extremo de arista alguna es llamado un **vértice aislado**. Un conjunto de vértices es un **conjunto independiente** si no contiene elementos que sean adyacentes.

Decimos que una gráfica G es una **gráfica finita** si tanto V como E son finitos. A lo largo de este documento, trabajaremos exclusivamente con gráficas finitas. Si G es una gráfica finita, decimos que el **orden** de G es el número de vértices de G , i.e., $|V(G)|$. El **tamaño** de G es el número de aristas de G , i.e., $|E(G)|$.

Las gráficas suelen ser representadas por medio de dibujos. En esta representación los vértices aparecen como pequeños círculos y las aristas como líneas que conectan a los vértices que son sus extremos. Podemos ver un ejemplo de esto en la Figura 2.1. Las etiquetas que aparecen a lado de cada vértice indican el nombre con el que se denota dicho vértice. Usualmente estas etiquetas no son incluidas.

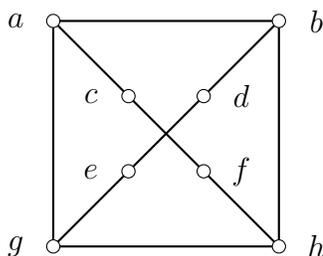


Figura 2.1: Ejemplo de la representación de una gráfica.

Dadas dos gráficas $G = (V, E)$ y $G' = (V', E')$, decimos que G' es **subgráfica** de G , denotado $G' \subseteq G$, si $V' \subseteq V$ y $E' \subseteq E$. La subgráfica de G **inducida** por $V' \subseteq V$, a la que denotamos como $G[V']$, es la gráfica cuyo conjunto de vértices es V' y cuyas aristas unen a dos vértices si y sólo si estos son adyacentes en G .

En la Figura 2.2 se muestran dos gráficas. Ambas gráficas son subgráficas de la gráfica de la Figura 2.1, a la que llamaremos G para los fines de este

ejemplo. Dado que la gráfica (a) no incluye las aristas ab y ac , ésta no es una subgráfica inducida de G . Por otra parte, la gráfica (b) es la subgráfica de G inducida por el conjunto de vértices $\{a, b, c, d, g, h\}$.

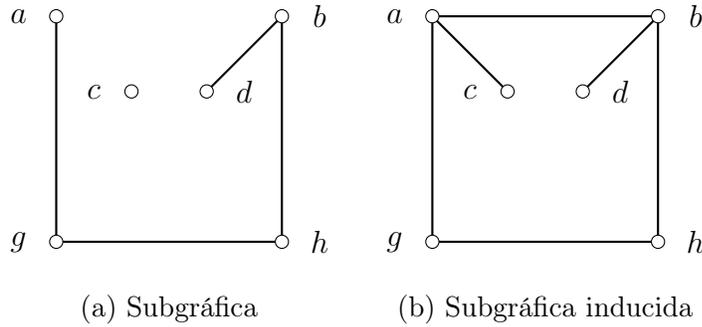


Figura 2.2: Ejemplo de una subgráfica y de una subgráfica inducida de la gráfica de la Figura 2.1.

Decimos que dos gráficas $G = (V, E)$ y $G' = (V', E')$ son **isomorfas** si existe una biyección $\phi : V \rightarrow V'$ tal que $xy \in E$ si y sólo si $\phi(x)\phi(y) \in E'$. A lo largo de este documento no hacemos distinción entre gráficas isomorfas. Es decir que, si dos gráficas G y G' son isomorfas, las consideramos como la misma gráfica.

Una gráfica G es **bipartita** con clases V_1 y V_2 si $V(G) = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ y cada arista en $E(G)$ une a un vértice de V_1 con un vértice de V_2 . Decimos que G tiene una **bipartición** (V_1, V_2) . De manera similar, G es **r -partita** (**multipartita**) con clases V_1, V_2, \dots, V_r si $V(G) = \bigcup_{i=1}^r V_i$, $V_i \cap V_j = \emptyset$ para cualesquiera $1 \leq i < j \leq r$ y ninguna arista une vértices de la misma clase.

2.1.2. Operaciones en gráficas

A continuación presentamos algunas operaciones que se pueden aplicar a las gráficas. Éstas son útiles para construir gráficas a partir de otras gráficas. También nos sirven para describir algunas gráficas y estructuras que se utilizan en el documento.

Dada una gráfica G , el **complemento** \overline{G} de G , es la gráfica tal que $V(\overline{G}) = V(G)$ y $E(\overline{G})$ es el conjunto de todas las parejas no ordenadas de elementos de V menos las aristas de $E(G)$; así, dos vértices son adyacentes en \overline{G} si y sólo si no son adyacentes en G .

Dada una gráfica G y uno de sus vértices x , denotamos como $G - x$ a la gráfica $G[V - \{x\}]$. Es decir, la subgráfica de G inducida por el conjunto de vértices $V - \{x\}$.

Dadas dos gráficas G y H tales que $V(G) \cap V(H) = \emptyset$, decimos que la gráfica $G + H = (V(G) \cup V(H), E(G) \cup E(H))$ es la **unión ajena** de G y H . Por otra parte, obtenemos la **unión completa** de G y H , denotada como $G \oplus H$, a partir de $G + H$ al agregar todas las aristas entre los vértices de G y los vértices de H . Al usar notación aditiva para la unión ajena, es coherente utilizar kG para denotar la unión ajena de k copias de G .

2.1.3. Algunas gráficas distinguidas

A continuación se presentan algunas familias de gráficas a cuyos elementos nos referimos por medio de un nombre específico, debido a que éstos son únicos salvo isomorfismo. A lo largo del documento usaremos dicho nombre para referirnos a cualquier copia isomorfa de estas gráficas.

El tamaño de una gráfica de orden n es al menos 0 y a lo más $\binom{n}{2}$. La gráfica (única salvo isomorfismo) de orden n y tamaño $\binom{n}{2}$ es la **gráfica completa** de orden n y se denota como K_n . En K_n cualesquiera dos vértices son adyacentes. Decimos que la gráfica K_1 es una gráfica **trivial**.

Una **trayectoria** es una gráfica P tal que $V(P) = x_0, \dots, x_\ell$ y $E(P) = x_0x_1, \dots, x_{\ell-1}x_\ell$. Utilizamos la notación $P = x_0 \dots x_\ell$, y decimos que P es una trayectoria de x_0 a x_ℓ . Usamos la notación P_ℓ para referirnos a la trayectoria de longitud ℓ . Análogamente, un **ciclo** es una gráfica C , con al menos tres vértices, y tal que $V(C) = x_0, \dots, x_\ell$ y $E(C) = x_0x_1, \dots, x_{\ell-1}x_\ell, x_\ell x_0$. Utilizamos la notación $C = x_0 \dots x_\ell x_0$. Para referirnos al ciclo de longitud ℓ usamos C_ℓ .

En la Figura 2.3 podemos ver ejemplos de una gráfica completa, un camino y un ciclo.

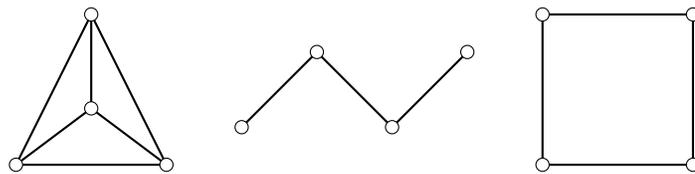


Figura 2.3: Las gráficas K_4 , P_4 y C_4 .

Una gráfica es **r -partita completa** (o bien, **multipartita completa**)

si es r -partita (o multipartita) y todo par de vértices de clases distintas son adyacentes. Nos referimos a las gráficas 2-partitas completas como **bipartitas completas**. Utilizamos $K_{n,m}$ para denotar a la gráfica bipartita completa que tiene n vértices en una de sus clases y m vértices en la otra. Notemos que una gráfica es r -partita completa si y sólo si es la unión completa de a lo más r conjuntos independientes ajenos.

2.1.4. Caminos y conexidad

Un **camino** W en una gráfica G es una sucesión alternante de vértices y aristas de G , digamos $x_0, e_1, x_1, e_2, \dots, e_\ell, x_\ell$ en donde $e_i = x_{i-1}x_i$, $0 < i \leq \ell$. De acuerdo con la terminología anterior, W es un (x_0x_ℓ) -camino y es denotado como $x_0x_1 \dots x_\ell$; la **longitud** de W es ℓ . Este camino es llamado un **paseo** si todas sus aristas son distintas. Notemos que una trayectoria es un camino en el que todos sus vértices son distintos (lo que coincide con la definición de trayectoria considerada anteriormente). Un paseo cuyos vértices de inicio y fin coinciden es un **circuito**. Para ser más precisos, un circuito es un paseo cerrado sin vértices de inicio y fin distinguidos. Si un camino $W = x_0 \dots x_\ell$ es tal que $\ell \geq 3$, $x_0 = x_\ell$ y los vértices x_i , $0 < i < \ell$ son distintos los unos de los otros y de x_0 , decimos que W es un **ciclo** en G .

Una gráfica es **conexa** si para cada par de vértices distintos existe un camino de entre ellos. Una gráfica que no es conexa es **inconexa**. Dada una gráfica G , una subgráfica de G conexa y máxima por contención (es decir que se vuelve inconexa si se le agrega cualquier otro vértice) es llamada una **componente conexa** de G . Así, una gráfica es conexa si y sólo si tiene una única componente conexa. Además, toda gráfica puede expresarse como la unión ajena de sus componentes conexas.

2.1.5. Árboles

A continuación se exponen los conceptos de árbol y de árbol arraigado (también conocido como árbol enraizado), dos tipos de gráficas que serán ampliamente utilizadas en el documento, en particular, en los algoritmos que presentamos.

Una gráfica que no contiene ciclo alguno es un **bosque**; un **árbol** es un bosque conexo. Notemos que un bosque es un conjunto de árboles sin aristas entre ellos.

Si bien los textos sobre algoritmos mantienen una noción uniforme de lo que es un árbol arraigado, es difícil encontrar una definición formal. Por lo que a continuación se presenta una definición y notación propias.

Un **árbol arraigado** es un árbol con un vértice distinguido al que llamamos **raíz**. Un árbol arraigado se puede denotar como una pareja ordenada (T, r) , en donde T es un árbol y $r \in V(T)$. Nos referiremos al árbol arraigado (T, r) simplemente como T , indicando que r es su raíz únicamente cuando sea necesario.

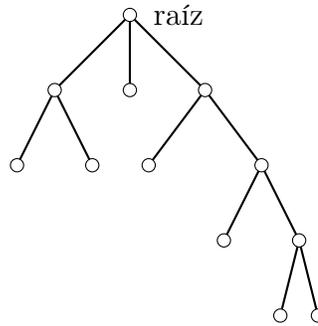


Figura 2.4: Ejemplo de un árbol arraigado. El vértice distinguido como raíz aparece como tal.

En la Figura 2.4 podemos observar la representación de un árbol arraigado. A partir de ahora supondremos que el nodo que se dibuja con mayor altura en la representación es la raíz, sin la necesidad de incluir una etiqueta para identificarlo.

Sea T un árbol arraigado con raíz r , y sean u y v nodos de T . Denotamos a la única trayectoria desde v hasta r como $P_T(v)$, y establecemos un orden parcial en $V(T)$ con la relación \geq tal que $u \geq v$ si y sólo si u es un vértice en $P_T(v)$. Si u y v son vértices de T distintos tales que $u \geq v$, decimos que u es **ancestro** de v , y que v es **descendiente** de u . El **ancestro común más profundo** de dos vértices de T es el vértice que está tanto en $P_T(u)$ como en $P_T(v)$ y que se encuentra a mayor distancia de r . Si v es un vértice de T , decimos que v es una **hoja** si no tiene descendientes, y decimos que es un **nodo interno** en el caso contrario. Si x y y son nodos de un árbol tales que x es ancestro de y y estos son adyacentes, decimos que x es el padre de y y que y es hijo de x . Podemos ver estos conceptos ilustrados en la Figura 2.5.

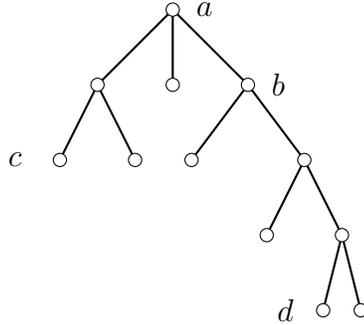


Figura 2.5: El nodo a es ancestro de los nodos b , c y d . Los nodos b , c y d son descendientes de a . El ancestro común más profundo de los nodos c y d es el nodo a . El nodo b es hijo del nodo a y éste a su vez es padre del nodo b . Los nodos c y d son hojas del árbol. Los nodos a y b son nodos internos.

2.1.6. Clases de gráficas hereditarias

A continuación se presenta el concepto de clase hereditaria de gráficas y otras definiciones relacionadas. Todas las clases de gráficas que estudiamos en este trabajo son clases hereditarias. Las definiciones fueron tomadas de Kitaev [3].

Una clase X de gráficas que contiene una gráfica G si y sólo si contiene también a todas las subgráficas inducidas de G es llamada una **clase hereditaria** de gráficas.

Dado un conjunto de gráficas M (finito o infinito), denotamos como $Free(M)$ a la clase de gráficas que contiene todas las gráficas que no tienen a ninguna gráfica de M como subgráfica inducida. Decimos que las gráficas en M son **subgráficas inducidas prohibidas** para la clase $Free(M)$, y que las gráficas en $Free(M)$ son libres de M . Diremos también que las gráficas en $Free(M)$ son libres de G para cualquier gráfica G en el conjunto M . Por convención, llamamos a los elementos de M **obstrucciones** de $Free(M)$.

Una clase X de gráficas es hereditaria si y sólo si existe un conjunto M tal que $X = Free(M)$.

Una gráfica es una **subgráfica inducida prohibida mínima**, también llamada una **obstrucción mínima** de una clase hereditaria X si G no pertenece a X pero toda subgráfica inducida de G (con excepción de G misma) pertenece a X . Denotamos como $Forb(X)$ al conjunto de todas las obstrucciones mínimas de la clase hereditaria X .

Para cualquier clase hereditaria X , tenemos que $X = Free(Forb(X))$. Además, $Forb(X)$ es el único conjunto mínimo con esta propiedad.

2.2. Cográficas

A continuación presentamos el concepto de *cográfica*. Las cográficas son la clase de gráficas que estudiamos en este trabajo de tesis. Éstas fueron introducidas independientemente por varios investigadores a principios de la década de 1970 y finalmente unificadas por Corneil, Lerchs y Stewart Burlingham en 1981, [4]; de este artículo serán tomadas las definiciones básicas.

2.2.1. Caracterización de las cográficas

Una *cográfica* se define recursivamente de la siguiente manera:

- (I) Una gráfica con un solo vértice es una cográfica.
- (II) Si G_1 y G_2 son cográficas, también lo es $G_1 + G_2$.
- (III) Si G es una cográfica, también lo es su complemento \overline{G} .

De igual forma, en [4], se presentan las siguientes equivalencias para las cográficas (entre otras):

Teorema 2.2.1 ([4]). *Sea G una gráfica. Las siguientes afirmaciones son equivalentes.*

- (1) G es una cográfica.
- (2) G no contiene a P_4 como subgráfica inducida.
- (3) El complemento de cualquier subgráfica inducida no trivial conexa de G es inconexa.

2.2.2. Coárboles

De la definición de las cográficas, podemos observar que éstas son todas las gráficas que se pueden obtener a partir de un solo nodo aplicando un número finito de operaciones de unión ajena y complemento. Esta serie de

operaciones puede ser representada a través de un árbol único salvo isomorfismo conocido como *coárbol*. En este documento utilizamos la definición de coárbol proporcionada por Corneil [5].

Sean G una gráfica y T un árbol arraigado con raíz r cuyos nodos están etiquetados y cuyas hojas representan cada una un vértice de G , decimos que T es el *coárbol* de G si cumple con las siguientes condiciones:

1. Cada uno de sus nodos internos tiene la etiqueta 0 o la etiqueta 1.
2. Los nodos etiquetados con 0 y los etiquetados con 1 son alternantes en cualquier camino desde r .
3. Si G es conexa, entonces r tiene etiqueta 0. En el caso contrario, r tiene etiqueta 1.
4. dos vértices $x, y \in V(G)$ son adyacentes si y sólo si el camino desde la hoja que representa a x hasta r se encuentra con el camino desde la hoja que representa a y hasta r en un nodo con etiqueta 1.

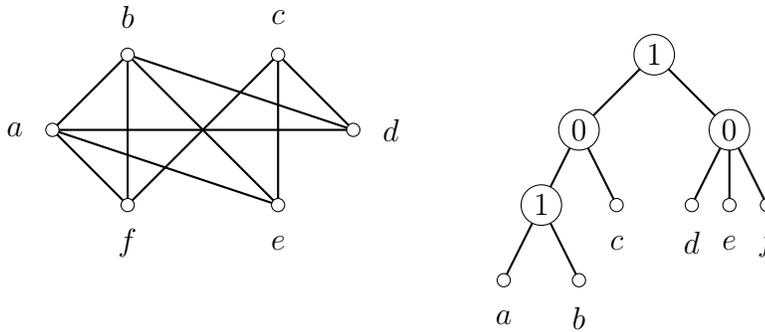


Figura 2.6: Una cográfica y su coárbol.

Notemos que todos los nodos internos de un coárbol tienen al menos dos hijos. Además, si el árbol T con raíz r es el coárbol de una gráfica inconexa G , cada uno de los hijos de r son los coárboles de las componentes conexas de G . Por otra parte, si G es conexa, los hijos de r serán los coárboles de las componentes conexas de \overline{G} . De esta forma, todo nodo x del coárbol T es la raíz de un coárbol al que denotamos como T_x . La cográfica representada por T_x es la subgráfica de G inducida por los vértices representados por las hojas de T_x . A esta subgráfica la denotamos por $G[x]$.

Capítulo 3

Estado del arte

En el presente capítulo hacemos una revisión de los artículos que dan forma a la investigación que realizamos. El problema de determinar si una cográfica acepta una partición tal que cada una de sus partes es una gráfica multipartita completa no ha sido estudiado con anterioridad. Sin embargo, la investigación realizada sobre las cográficas polares nos sirve como referencia acerca de cómo encontrar las obstrucciones mínimas de una clase hereditaria de cográficas en donde cada uno de sus elementos acepta una partición tal que cada parte cumple con una condición. Esta investigación también nos muestra que se pueden estudiar subclases de las clases anteriormente descritas al agregar restricciones a cada una de las partes. Los primeros tres artículos de los que hablamos abordan el tema de las cográficas polares.

3.1. Cográficas polares

En [6], Ekim, de Werra y Mahadeev introducen la clase de las cográficas polares y presentan una caracterización de la misma mediante un conjunto de obstrucciones mínimas (que existe dado que las cográficas polares son una clase hereditaria de gráficas). La metodología que los autores utilizan para demostrar su resultado principal, i.e., que el conjunto de gráficas que se propone (Figura 3.1) es, en efecto, el conjunto de obstrucciones mínimas para las cográficas polares, fue una guía importante para realizar la demostración de que nuestra propuesta de conjunto de obstrucciones mínimas de la clase

M_2^1 es correcta. En este artículo también se consideran las cográficas monopolares, una subclase de las cográficas polares, cuyo estudio sirve como inspiración para el estudio de algunas subclases de M_2 .

Dada una cográfica G , decimos que G es una **cográfica polar** si su conjunto de vértices V acepta una partición (A, B) tal que A induce una gráfica multipartita completa y B induce una unión ajena de clanes. Decimos que G es (s, k) -polar si existe una partición (A, B) de los vértices de G en donde A induce en G una unión completa de a lo más s conjuntos independientes (es decir una gráfica s -partita) y B induce en G una unión ajena de a lo más k clanes. Notemos que las cográficas polares son las gráficas (∞, ∞) -polares.

Teorema 3.1.1. *Si G es una cográfica, entonces G es polar si y sólo si no contiene como subgráfica inducida a ninguna de las gráficas de la Figura 3.1.*

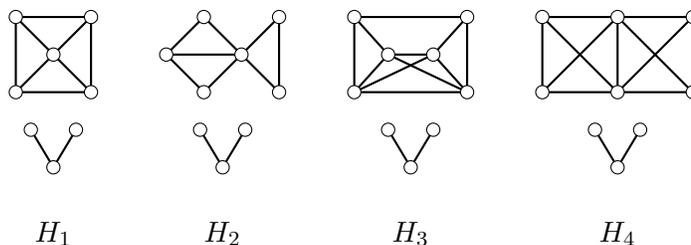


Figura 3.1: Obstrucciones mínimas para las gráficas polares.

La demostración de que las gráficas de la Figura 3.1 conforman el conjunto de obstrucciones mínimas de las cográficas polares se realiza describiendo a cada una de éstas en términos de la unión ajena y la unión completa de gráficas más pequeñas.

$$(1) H_1 = P_3 + (\overline{K_2} \oplus P_3) = P_3 + (K_1 \oplus P_4)$$

$$(2) H_2 = P_3 + (K_1 \oplus (P_3 + K_2))$$

$$(3) H_3 = P_3 + (\overline{P_3} \oplus \overline{P_3})$$

$$(4) H_4 = P_3 + (K_2 \oplus 2K_2)$$

¹Una cográfica está en la clase M_2 si acepta una partición en dos gráficas multipartitas completas.

Primero se demuestra que ninguna de estas gráficas es una cografía polar y que, por lo tanto, cualquier gráfica que tenga a alguna de ellas como subgráfica inducida tampoco es una cografía polar. Por último se muestra que si una cografía G no es polar, entonces debe de tener a alguna de estas gráficas como subgráfica inducida.

En este artículo también se presenta y se caracteriza, a través de su conjunto de obstrucciones mínimas, a la clase de las cografías monopolares. El conjunto de obstrucciones mínimas se muestra en la Figura 3.2

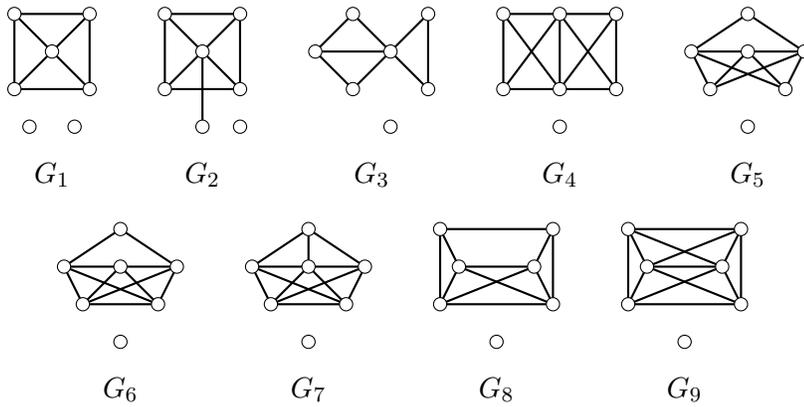


Figura 3.2: Obstrucciones mínimas para las gráficas monopolares.

Como podemos observar, varias de las obstrucciones mínimas de las gráficas monopolares se asemejan a obstrucciones mínimas de las gráficas polares. Por ejemplo, las gráficas G_1 , G_3 , G_4 y G_8 son subgráficas de H_1 , H_2 , H_3 y H_4 respectivamente. Notemos que en los cuatro casos se obtiene una obstrucción mínima G para las gráficas monopolares, a partir de una obstrucción mínima H para las gráficas polares al eliminar vértices de la componente conexa de H que forma un P_3 . En los últimos tres casos, el P_3 es reemplazado por un vértice aislado. Esto nos lleva a pensar en que se puede encontrar una relación entre el conjunto de obstrucciones mínimas de una clase y los conjuntos de obstrucciones mínimas de sus subclases.

3.2. Obstrucciones mínimas para cográficas $(s, 1)$ -polares

En [7], Contreras-Mendoza y Hernández-Cruz exhiben el conjunto de obstrucciones mínimas de las cográficas $(\infty, 1)$ -polares. Este conjunto es utilizado para describir cómo se puede obtener el conjunto de obstrucciones mínimas de cualquiera de las clases de cográficas $(s, 1)$ -polares dado un entero $s \geq 2$. En nuestra investigación encontramos un resultado similar, el Lema 4.3.1, donde describimos a las obstrucciones mínimas para las gráficas que aceptan una partición en un conjunto independiente y una gráfica multipartita completa.

A continuación reproducimos el resultado antes mencionado.

Teorema 3.2.1 ([7]). *Sea G una cográfica. Entonces G es $(\infty, 1)$ -polar si y sólo si no contiene alguna de las gráficas de la Figura 3.3 como subgráfica inducida. Este conjunto es llamado el conjunto de obstrucciones esenciales.*

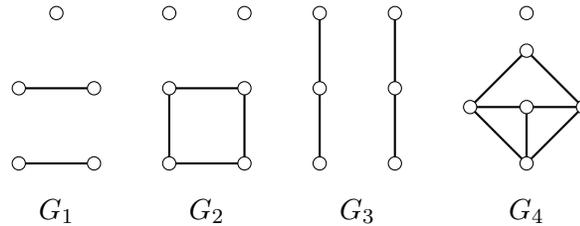


Figura 3.3: Obstrucciones mínimas para las gráficas polares.

Haciendo uso de estas obstrucciones esenciales, los autores caracterizan, para cualquier entero s , con $s \geq 2$, a las obstrucciones mínimas para la clase de cográficas $(s, 1)$ -polares. Notemos que, además de las obstrucciones esenciales, existen cuatro familias esporádicas de obstrucciones mínimas, y se plantea una regla recursiva para generar obstrucciones mínimas para las cográficas $(s, 1)$ -polares utilizando obstrucciones mínimas no esenciales para las cográficas $(t, 1)$ -polares, con $t < s$.

Teorema 3.2.2. *Sea G una cográfica y $s \geq 2$ un entero. Entonces G es una obstrucción mínima de las cográficas $(s, 1)$ -polares si y sólo si es una de las siguientes gráficas:*

3.2. OBSTRUCCIONES MÍNIMAS PARA COGRÁFICAS $(S, 1)$ -POLARES 31

- Una de las cuatro obstrucciones esenciales.
- $2K_{s+1}$.
- $K_2 + (\overline{K_2} \oplus K_s)$.
- $K_1 + (C_4 \oplus K_{s-1})$.
- $\overline{(s+1)K_2}$.
- El complemento de G es inconexo con componentes G_1, \dots, G_t tales que $t \leq s$, y cada G_i es el complemento de una obstrucción mínima no esencial de la clase de cográficas $(s_i, 1)$ -polares con $\sum_{i=1}^t s_i = s - t + 1$.

El resultado presentado en el Teorema 3.2.2 es un ejemplo de cómo, al restringir un problema a un subproblema bien definido, es posible encontrar soluciones parciales al problema general. En particular, se obtuvo una caracterización por obstrucciones mínimas para una subclase infinita de las cográficas (s, k) -polares.

Más aún, una interpretación de los Teoremas 3.2.1 y 3.2.2 es que resulta posible pensar a las gráficas $(\infty, 1)$ -polares como el límite cuando s tiende al infinito de las gráficas $(s, 1)$ -polares. Podemos observar que las gráficas que son obstrucciones mínimas de las gráficas $(s, 1)$ -polares para cualquier entero $s \geq 2$ también son obstrucciones mínimas de las gráficas $(\infty, 1)$ -polares. En nuestra investigación encontramos un resultado similar para las gráficas (α, β) - M_2^2 , ya que las obstrucciones mínimas de la clase $(1, \infty)$ - M_2 son obstrucciones mínimas de la clase $(1, m)$ - M_2 , para cualquier valor de m con $m \geq 2$. Con base en la idea anterior, a partir de las obstrucciones mínimas de las clases $(2, m)$ - M_2 , con $m \in \{3, \dots, 7\}$ que generamos computacionalmente, encontramos algunas obstrucciones mínimas de la clase $(2, \infty)$ - M_2 . Aplicando este mismo proceso, podemos encontrar obstrucciones mínimas para la clase α, ∞ - M_2 dado un entero $\alpha \geq 3$.

²La clase (α, β) - M_2 es la clase constituida por todas las gráficas que aceptan una partición en dos gráficas multipartitas completas, una formada por a lo más α conjuntos estables y la otra formada por a lo más β conjuntos estables.

3.3. Obstrucciones mínimas para cográficas 2-polares

En [8], Hell, Hernández-Cruz y Linhares-Sales exhiben el conjunto de obstrucciones mínimas de la clase de las cográficas 2-polares. Para construir este conjunto primero se presentan resultados preliminares sobre la estructura de las obstrucciones mínimas de las cográficas (k, k) -polares (llamadas simplemente cográficas k -polares) para cualquier entero positivo k . Posteriormente se presenta el complemento parcial, una operación que conserva la 2-polaridad, para construir las 24 obstrucciones mínimas inconexas de las cográficas 2-polares a partir de un conjunto de cuatro obstrucciones mínimas.

El siguiente lema describe la estructura de las obstrucciones mínimas para las cográficas k -polares con el máximo número posible de componentes conexas.

Lema 3.3.1. *Sean l y k enteros tales que $1 \leq l \leq k + 1$. Salvo isomorfismo, hay exactamente una obstrucción mínima para las cográficas k -polares con $k+2$ componentes en total y l componentes triviales. Esta obstrucción mínima es isomorfa a*

$$lk_1 + (k - l + 1)k_2 + k_{l,l}$$

Aplicando el Lema 3.3.1, podemos encontrar tres obstrucciones mínimas para las cográficas 2-polares. Éstas se muestran en la Figura 3.4.

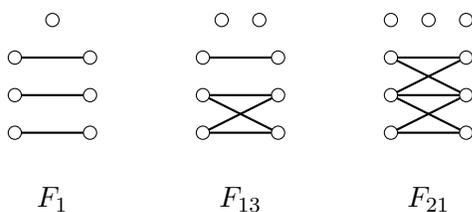


Figura 3.4: Obstrucciones mínimas para las cográficas 2-polares obtenidas con el Lema 3.3.1.

Ahora introducimos una operación que preserva la 2-polaridad y la propiedad de ser cográfica, por lo que resulta útil en el estudio de las obstrucciones mínimas para las cográficas 2-polares. Sea H una gráfica, un **complemento**

3.3. OBSTRUCCIONES MÍNIMAS PARA COGRÁFICAS 2-POLARES 33

parcial de H es una gráfica obtenida de H al dividir a sus componentes conexas en dos gráficas H' y H'' , y tomando de forma separada el complemento de cada una.

Tomando todos los posibles complementos parciales de las gráficas F_1 , F_{13} y F_{21} (Figura 3.4), encontramos tres familias de obstrucciones mínimas; éstas se muestran en las Figuras 3.5, 3.6 y 3.7 respectivamente. Podemos encontrar una cuarta familia de obstrucciones mínimas al tomar todos los posibles complementos parciales de la gráfica F_7 (Figura 3.8) que también es una obstrucción mínima de las gráficas 2-polares. La gráfica F_7 se puede construir de forma natural agregando un K_2 a una de las obstrucciones mínimas para $(2, 1)$ -polaridad en cografías.

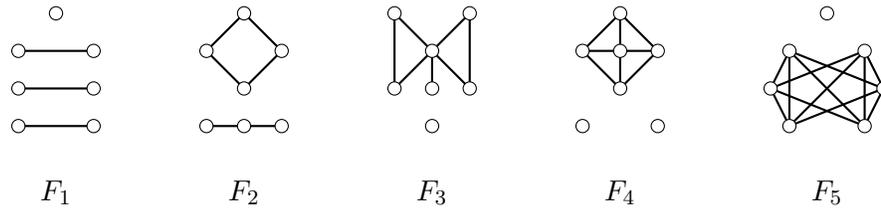


Figura 3.5: Obstrucciones mínimas de las cografías 2-polares con 7 vértices.

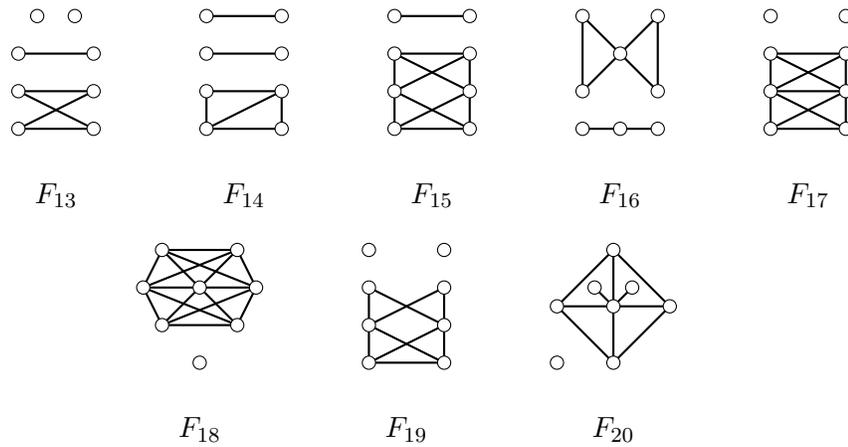


Figura 3.6: Familia A de obstrucciones mínimas de las cografías 2-polares con 8 vértices.

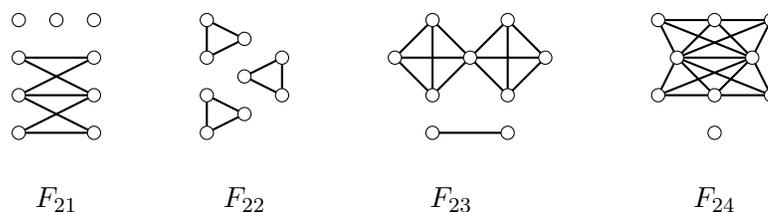


Figura 3.7: Obstrucciones mínimas de las cográficas 2-polares con 9 vértices.

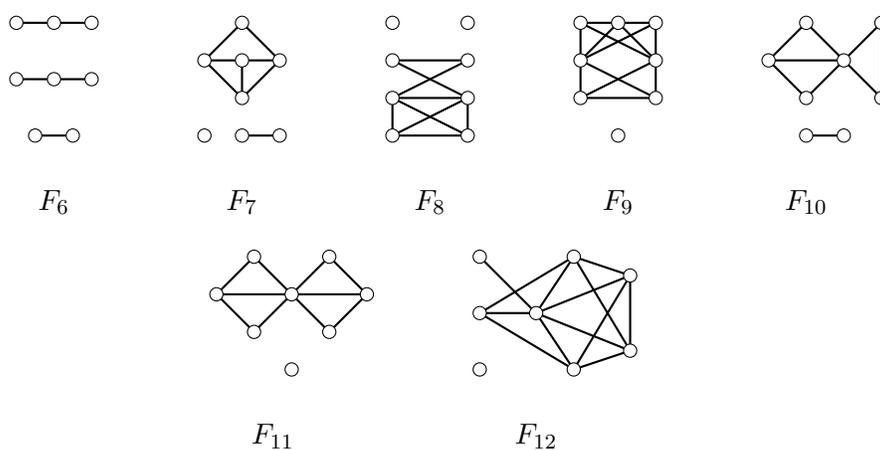


Figura 3.8: Familia B de obstrucciones mínimas de las cográficas 2-polares con 8 vértices.

En este artículo podemos observar un ejemplo en el que, dada una clase hereditaria de cográficas C y una operación o tales que C es cerrada bajo o , es posible encontrar familias de obstrucciones mínimas para C a partir de obstrucciones mínimas de C ya conocidas aplicándoles la operación o . Las clases de cográficas que estudiamos en nuestra investigación son cerradas bajo la unión completa y, de manera parecida a lo que sucede en [8], encontramos una regla con la que, dado un entero $i \geq 2$, podemos construir obstrucciones mínimas para las gráficas que aceptan una partición en i gráficas multipartitas completas aplicando la unión completa a obstrucciones mínimas de las gráficas que aceptan una partición en $i - 1$ gráficas multipartitas completas (Teorema 4.4.4).

3.4. Un algoritmo lineal para el reconocimiento de cografías

En [5], Corneil, Perl y Stewart presentan un algoritmo que, dada una gráfica, determina si ésta es una cografía y, si lo es, devuelve su cóarbol. Éste es un algoritmo incremental, se construye un cóarbol agregando los vértices de la gráfica recibida como entrada uno a uno. Para agregar un nuevo vértice x , de la gráfica de entrada G , al cóarbol T que se va construyendo, el algoritmo ejecuta primero una subrutina que se encarga de marcar los nodos de T empezando por sus hojas. Una hoja de T se marca sólo si ésta representa un vértice de G adyacente a x . La subrutina continúa marcando nodos de G subiendo hacia la raíz. En nuestra investigación presentamos un algoritmo similar, el Algoritmo 5, que ayuda a determinar si una cografía H es subcografía de otra cografía I marcando los vértices de un cóarbol binario³ de I partiendo desde las hojas y subiendo hacia la raíz.

A continuación se reproducen los resultados de [5], empezando por el algoritmo de marcado antes mencionado.

Algoritmo 1: Marcar

Entrada: T , un cóarbol con raíz R cuyas hojas son vértices de una gráfica G ; x , el vértice de G que se busca agregar a T

Salida: Se marcan y se desmarcan algunos nodos de T

- 1 $D \leftarrow \{\}$ // El conjunto de los nodos que han sido marcados y desmarcados
 - 2 Marcar todas las hojas de T que sean adyacentes a x
 - 3 **para cada** nodo marcado u de T tal que todos sus hijos están en el conjunto D **hacer**
 - 4 Desmarcar a u
 - 5 Agregar u a D
 - 6 **si** $u \neq R$ **entonces**
 - 7 | Marcar al padre de u
-

Sean T un cóarbol, x un vértice y M el conjunto de nodos de T que se encuentran marcados al terminar la ejecución del Algoritmo 1 al recibir como

³Un cóarbol binario es un cóarbol en el que cada nodo tiene necesariamente dos hijos y pueden haber nodos adyacentes con la misma etiqueta.

entrada a T y a x . El Teorema 3.4.1 puede ser utilizado para determinar si la gráfica que se obtiene al agregar x a la gráfica representada por T es una cográfica. Las siguientes definiciones son necesarias para presentar dicho teorema. Sea α un nodo de M de profundidad máxima en T y sea β un nodo en $M - \{\alpha\}$ de profundidad máxima en T . Decimos que un nodo γ de T con etiqueta 1 está correctamente marcado si y sólo si todos sus hijos, excepto uno, fueron marcados y desmarcados. Un camino legítimamente alternante en un coárbol marcado es un camino alternante de nodos correctamente marcados y nodos sin marcar, con etiqueta 0, cuyos extremos son nodos con etiqueta 1.

Teorema 3.4.1. *Si G es una cográfica con árbol T , entonces $G + x$ es una cográfica si y sólo si:*

- M es vacío o
- se cumplen las siguientes dos condiciones
 - $M - \{\alpha\}$ consiste exactamente de los nodos con etiqueta 1 de un camino legítimamente alternante que termina en R .
 - α es un nodo con etiqueta 0 cuyo padre es β o α es un nodo con etiqueta 1 cuyo abuelo, si existe, es β .

El Algoritmo 2, que constituye el resultado principal del artículo, utiliza el Algoritmo 1 y el Teorema 3.4.1 para determinar si una gráfica es una cográfica, y construir su coárbol en caso de que lo sea.

El Algoritmo 1 consolida un ejemplo de un algoritmo que funciona marcando los nodos de un coárbol empezando por las hojas y dirigiéndose hacia la raíz⁴. Por otro lado, el Algoritmo 1 es un ejemplo de un algoritmo que, al ser ejecutado, no sólo resuelve un problema de decisión, sino que proporciona una prueba (o *certificado*) de que su respuesta es correcta. En este caso, la prueba es el coárbol de la gráfica recibida como entrada. Sin embargo, cuando la gráfica que se recibe como entrada no es una cográfica, el Algoritmo 1 sólo devuelve *null*. Si desconfiáramos de una implementación de este algoritmo, y quisiéramos comprobar que la salida es correcta, podríamos utilizar el coárbol en un caso, pero en el otro, no tendríamos información adicional.

En nuestra investigación, el Algoritmo 12 es un algoritmo certificador, es decir, devuelve certificados para verificar, de manera eficiente, que la salida

⁴En inglés un *bottom-up* algorithm.

3.4. UN ALGORITMO LINEAL PARA EL RECONOCIMIENTO DE COGRÁFICAS 37

Algoritmo 2: AlgoritmoDeReconocimiento

Entrada: G , una gráfica cuyo conjunto de vértices es

$$V = \{v_1, v_2, \dots, v_n\}$$

Salida: El córbol de G si G es una cográfica, *null* en el caso contrario

```
1 Crear el córbol  $T$  cuya raíz  $R$  tiene etiqueta 1 si  $v_1$  y  $v_2$  son
  adyacentes o etiqueta 0 en el caso contrario
2 para cada  $x$  en  $V(G) - \{v_1, v_2\}$  hacer
3   Marcar( $T, x$ )
4   si Todos los nodos de  $T$  fueron marcados y desmarcados
     entonces
5     si  $R$  tiene etiqueta 1 entonces
6        $\lfloor$  Agregar a  $x$  como hijo de  $R$ 
7     en otro caso
8        $\lfloor$   $R \leftarrow$  un nuevo nodo con etiqueta 1 cuyos hijos sean  $x$  y  $R$ 
9   si no, si Ninguno de los nodos de  $T$  fue marcado entonces
10    si  $R$  tiene etiqueta 0 entonces
11       $\lfloor$  Agregar a  $x$  como hijo de  $R$ 
12    en otro caso
13       $\lfloor$   $R \leftarrow$  un nuevo nodo con etiqueta 0 cuyos hijos sean  $x$  y  $R$ 
14  si no, si La gráfica representada por  $T$  agregando  $x$  es una
     cográfica entonces
15     $\lfloor$   $u \leftarrow$  el nodo marcado más profundo de  $T$ 
16     $\lfloor$  Encontrar en qué nodo del córbol con raíz  $u$  se debe insertar
      $x$ 
17  en otro caso
18     $\lfloor$  devolver null
19  Desmarcar todos los nodos de  $T$ 
```

del algoritmo es correcta, en cualquier caso. Nuestro algoritmo no sólo es capaz de determinar si una cográfica G pertenece a la clase M_2 , sino que devuelve una coloración de las hojas de su coárbol T tal que si $G \in M_2$, las hojas de T tienen uno de dos colores y las hojas del mismo color inducen una gráfica multipartita completa en G (un *sí-certificado*). En el caso contrario, algunas de las hojas de T tendrán un color distintivo que indica que esos vértices inducen una obstrucción mínima de M_2 en G (un *no-certificado*).

Capítulo 4

Resultados

En este capítulo presentamos el producto de nuestra investigación. La primera sección del capítulo tiene como resultado principal un algoritmo capaz de determinar si una cográfica pertenece a una clase hereditaria de gráficas fija en tiempo lineal. A partir de la segunda sección abordamos el problema principal de nuestra tesis, determinar si una cográfica acepta una partición en un número dado de gráficas multipartitas completas. Comenzamos realizando un estudio detallado de la clase de cográficas que aceptan una partición en dos gráficas multipartitas completas, a la que llamamos M_2 . En la segunda sección del capítulo caracterizamos a la clase M_2 a través de su conjunto de obstrucciones mínimas, proporcionamos un algoritmo para reconocer a sus elementos (utilizando el resultado principal de la primera sección) y presentamos un algoritmo certificador que no sólo es capaz de reconocer si una cográfica G pertenece a M_2 , sino que encuentra una partición de G en dos gráficas multipartitas completas o una obstrucción mínima de la clase como subgráfica inducida de G . En la tercera sección estudiamos a las clases $(\alpha, \beta) - M_2$, subclases de M_2 cuyos elementos aceptan una partición en dos gráficas multipartitas completas de tamaños restringidos. El resultado principal de esta sección es un algoritmo para encontrar obstrucciones mínimas para cualquier clase $(\alpha, \beta) - M_2$. La cuarta y última sección del capítulo da un paso en la generalización de los resultados de las secciones anteriores. En éste se proporcionan algunas familias de obstrucciones mínimas para caracterizar a la clase de cográficas que aceptan una partición en i gráficas multipartitas completas dado un entero $i \geq 2$.

4.1. Términos y algoritmos generales

En esta sección se presenta un conjunto de conceptos y algoritmos útiles para cográficas en general. El resultado principal de la sección es un algoritmo de tiempo lineal capaz de determinar si una cográfica, representada por su coárbol, pertenece a una clase hereditaria fija de cográficas caracterizada por su conjunto de obstrucciones mínimas.

4.1.1. Coárbol binario

Tomando como base el concepto de coárbol, podemos imaginar otra estructura de tipo árbol para la representación de las cográficas en la que cada nodo tenga a lo más un número k de hijos. Esta limitante resulta útil para formular algoritmos rápidos en cográficas. El menor valor que puede tomar k es 2, lo que resulta en una representación de cualquier cográfica mediante un coárbol binario. Esta representación es la que utilizaremos principalmente en el presente capítulo.

Sean $G = (V, E)$ una cográfica, $C = \{c_1, c_2, \dots, c_n\}$ el conjunto de las componentes conexas de G , $D = \{d_1, d_2, \dots, d_m\}$ el conjunto de las componentes conexas de \overline{G} , (C_1, C_2) una partición en dos partes de C y (D_1, D_2) una partición en dos partes de D . Decimos que el árbol binario arraigado etiquetado, (T, r) , es un **coárbol binario** de G si se puede construir de la siguiente manera: Si G consta de un sólo vértice, entonces T sólo contiene a r , que es igual al único vértice de G . De lo contrario, si G es conexa, entonces r tiene la etiqueta 1, uno de los hijos de r es el coárbol binario de $G - D_1$ y el otro es el coárbol binario de $G - D_2$. Y finalmente, si G es inconexa, entonces r tiene la etiqueta 0, uno de sus hijos es el coárbol binario de $G - C_1$ y el otro el coárbol binario de $G - C_2$.

Claramente, una cográfica puede ser representada por más de un coárbol binario diferente como se muestra en la Figura 4.1. Sin embargo, la propiedad de que dos vértices son adyacentes si y sólo si su ancestro común más profundo tiene la etiqueta 1 se mantiene.

4.1.2. Algoritmo para generar un coárbol binario

Dada una cográfica G , el Algoritmo 3 construye recursivamente un coárbol binario de G a partir del coárbol de G . Las hojas del coárbol son la base de la recursión; al procesar una hoja del coárbol original, simplemente se crea un

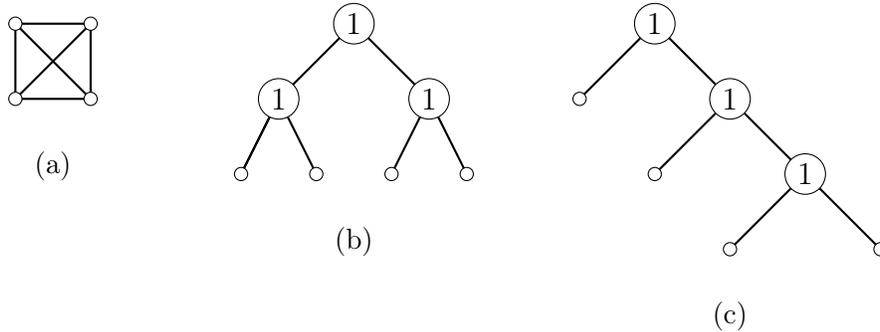


Figura 4.1: (b) y (c) son dos cóarbols binarios diferentes que representan a la cográfica (a).

nuevo nodo para el cóarbol binario, inicialmente sin etiqueta, y se devuelve. Los nodos internos de un cóarbol siempre tienen al menos dos hijos. Si r es un nodo interno del cóarbol original con dos hijos, entonces se crea un nuevo nodo en el cóarbol binario con la misma etiqueta que r , y sus subárboles izquierdo y derecho son los obtenidos al procesar recursivamente a su hijo izquierdo y derecho, respectivamente. Si r tiene al menos tres hijos, se procesa creando un nuevo cóarbol binario de la siguiente forma: La raíz del cóarbol binario tiene la misma etiqueta que r , como subárbol izquierdo de la raíz, se tiene al cóarbol binario resultante de procesar recursivamente al primer hijo de r , y como subárbol derecho, se tiene un nodo con la misma etiqueta que r , que a su vez tiene como subárbol izquierdo al árbol binario resultante de procesar recursivamente al segundo hijo de r y como subárbol derecho un nuevo nodo con la misma etiqueta que él. El árbol binario resultante es un árbol cargado a la derecha. La Figura 4.2 muestra una ejecución ilustrativa del algoritmo.

En términos de las particiones de las componentes conexas de la gráfica, el algoritmo realiza lo siguiente. Si la etiqueta de r es 0, entonces el cóarbol con raíz en r representa una cográfica inconexa y se elige la partición de sus vértices en la que la primera parte es una componente conexa y la segunda parte es el resto. Si la etiqueta de r es uno, como la cográfica representada es conexa, se elige una componente conexa del complemento de la cográfica para la primera parte y el resto de las componentes del complemento para la segunda.

Dado que el Algoritmo 3 recorre a lo más una vez cada nodo de r , su tiempo de ejecución es $O(n)$ en donde n es el número total de nodos del

Algoritmo 3: CrearArbolBinario

Entrada: r la raíz del coárbol**Salida:** r' la raíz del coárbol binario

```

1 si  $r$  es un nodo interno entonces
2    $r' \leftarrow$  nuevo nodo de árbol binario
3    $r'.etiqueta = r.etiqueta$ 
4    $s \leftarrow r'$ 
5    $i \leftarrow 0$ 
6   mientras  $i < r.children.size - 2$  hacer
7      $s.izquierda \leftarrow$  CrearArbolBinario( $r.hijos[i]$ )
8      $s.derecha \leftarrow$  nuevo nodo de árbol binario
9      $s \leftarrow s.derecha$ 
10     $s.etiqueta \leftarrow r.etiqueta$ 
11     $i \leftarrow i + 1$ 
12   $s.izquierda \leftarrow$  CrearArbolBinario( $r.hijos[i]$ )
13   $s.derecha \leftarrow$  CrearArbolBinario( $r.hijos[i + 1]$ )
14 en otro caso
15    $r' \leftarrow r$ 
16 devolver  $r'$ 

```

árbol con raíz r .

4.1.3. Algoritmo para generar todos los coárboles binarios de una gráfica

Podemos obtener todos los coárboles binarios correspondientes a un coárbol haciendo uso del Algoritmo 4. Este algoritmo recibe como entrada la raíz del coárbol, r , y devuelve un conjunto de nodos, S , cada uno de cuyos elementos es la raíz de un coárbol binario. En el algoritmo, los nodos internos son procesados creando un nuevo coárbol binario para cada posible partición del conjunto de hijos de dicho nodo. Al procesar las hojas, simplemente se crea un nuevo nodo que será una hoja en los árboles binarios.

Dado que el número de coárboles binarios que representan a una gráfica crece de forma exponencial, el tiempo en el que se pueden generar dichos coárboles binarios crece al menos de forma exponencial. El Algoritmo 4 no

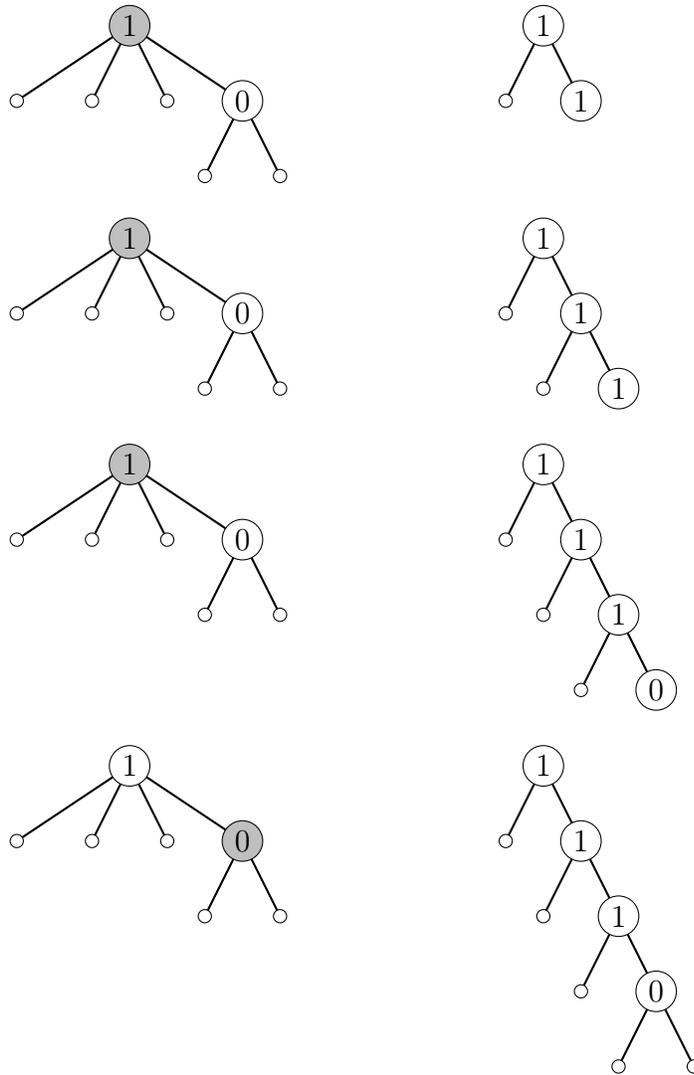


Figura 4.2: Ejemplo de la ejecución del Algoritmo 3. A la izquierda se muestra el córbol original, mientras se marca con gris el nodo que se está procesando. A la derecha aparece el córbol binario que se va construyendo.

es óptimo, ya que todos los nodos con excepción de la raíz y las hojas se procesan múltiples veces. Esto no repercute en el resultado principal de la sección, el Algoritmo 7, que sirve para identificar a los elementos de una clase hereditaria de cográficas C , ya que, aunque se requiere encontrar todos los córbol binarios de cada una de las obstrucciones mínimas de C , se

Algoritmo 4: CrearÁrbolesBinarios

Entrada: r la raíz de un coárbol T **Salida:** $S = \{r'_1, r'_2, \dots, r'_n\}$ con r_i la raíz de un coárbol binario de T
para todo $1 \leq i \leq n$

```

1  $S \leftarrow \emptyset$ 
2 si  $r$  es una hoja entonces
3    $\lfloor$  Agregar  $r$  a  $S$ 
4 en otro caso
5   para cada partición en dos partes  $(A, B)$  del conjunto de hijos
6     de  $r$  hacer
7     si  $A$  tiene sólo un elemento  $a$  entonces
8        $\lfloor L \leftarrow \text{CrearÁrbolesBinarios}(a)$ 
9     en otro caso
10       $\lfloor a' \leftarrow$  nuevo nodo de coárbol con la etiqueta de  $r$ 
11       $\lfloor$  Agregar todos los elementos de  $A$  como hijos de  $a'$ 
12       $\lfloor L \leftarrow \text{CrearÁrbolesBinarios}(a')$ 
13     si  $B$  tiene sólo un elemento  $b$  entonces
14        $\lfloor R \leftarrow \text{CrearÁrbolesBinarios}(b)$ 
15     en otro caso
16        $\lfloor b' \leftarrow$  nuevo nodo de coárbol con la etiqueta de  $r$ 
17        $\lfloor$  Agregar todos los elementos de  $B$  como hijos de  $b'$ 
18        $\lfloor R \leftarrow \text{CrearÁrbolesBinarios}(b')$ 
19     para cada  $l \in L$  y cada  $r \in R$  hacer
20        $\lfloor s \leftarrow$  nuevo nodo de coárbol binario con la etiqueta de  $r$ 
21        $\lfloor s.\text{izquierda} \leftarrow l$ 
22        $\lfloor s.\text{derecha} \leftarrow r$ 
23        $\lfloor$  Agregar  $s$  a  $S$ 

```

contempla que este cómputo se realice antes de la ejecución del algoritmo.

4.1.4. Subcoárbol

A continuación presentamos el concepto de subcoárbol binario que será utilizado para determinar si una cográfica H es subgráfica inducida de una cográfica G en el Algoritmo 6.

Sean T y U dos coárboles binarios y u_1, u_2 y u_3 nodos de U , decimos que U es un **subcoárbol binario** de T si existe una función inyectiva $f : V(U) \rightarrow V(T)$ tal que, si u_1 es una hoja, entonces $f(u_1)$ es una hoja también; si no, entonces $u_1.\text{etiqueta} = f(u_1).\text{etiqueta}$ y, si u_3 es el ancestro común más profundo de u_1 y u_2 , entonces $f(u_3)$ es el ancestro común más profundo de $f(u_1)$ y $f(u_2)$. Llamamos a f la **función de coasignación** de U a T .

El concepto de subcoárbol binario es diferente del de subárbol dado que, si T y U son coárboles con U subcoárbol binario de T , entonces tenemos que los nodos de U se pueden encontrar dispersos entre los nodos de T a diferencia de lo que se tendría si U fuera subárbol de T . Esto se puede apreciar en la Figura 4.3.

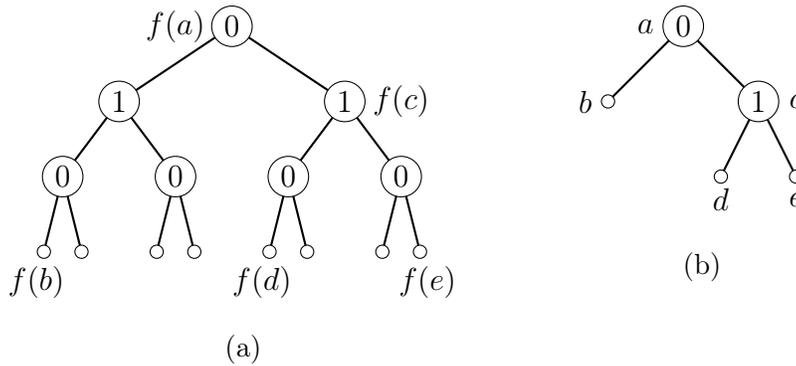


Figura 4.3: El coárbol (b) es subcoárbol binario del coárbol (a). Las etiquetas en los nodos de ambos coárboles binarios indican la asignación de los nodos de (b) a los nodos de (a).

Lema 4.1.1. Sean G y H cográficas, T_G un coárbol binario de G y T_H un coárbol binario de H . Si T_H es subcoárbol binario de T_G , entonces H es una subgráfica inducida de G .

Demostración. Sean h_1 y h_2 hojas diferentes de T_H y h_3 el ancestro común más profundo de h_1 y h_2 . Como T_H es subcoárbol binario de T_G , entonces existe una función de coasignación, f , de T_H a T_G . Luego, tenemos que $f(h_1)$

y $f(h_2)$ son hojas de T_G y que las etiquetas de h_3 y $f(h_3)$ coinciden. Así, h_1 y h_2 son adyacentes en H si y sólo si $f(h_1)$ y $f(h_2)$ son adyacentes en G . Luego, $G[f[V(H)]]$ es una subgráfica inducida de G que es isomorfa a H . Así, H es una subgráfica inducida de G . \square

Lema 4.1.2. Sean G y H cográficas y B_G un coárbol binario de G . Si H es una subgráfica inducida de G , entonces existe un coárbol binario B_H de H tal que B_H es subcoárbol binario de B_G .

Demostración. Como H es una subgráfica inducida de G , $V(H)$ es un subconjunto de las hojas de B_G . Consideremos el siguiente proceso recursivo para crear el árbol B_H .

Entrada: $V(H)$
Salida: r , la raíz de B_H

- 1 **si** $V(H)$ tiene un solo elemento **entonces**
- 2 $r \leftarrow v$, el único elemento de $V(H)$
- 3 **en otro caso**
- 4 $v \leftarrow$ el nodo más profundo de B_G que es ancestro de cada elemento de $V(H)$
- 5 $r \leftarrow$ nuevo nodo de coárbol binario con la misma etiqueta que v
- 6 Asignar al hijo izquierdo de r el resultado de procesar recursivamente los elementos de $V(H)$ que estén en la rama izquierda de v
- 7 Asignar al hijo derecho de r el resultado de procesar recursivamente los elementos de $V(H)$ que estén en la rama derecha de v
- 8 **devolver** r ,

Notemos que cada nodo de B_H se construye tomando como base a un nodo de B_G . Esto ocurre en la línea 2 si dicho nodo es una hoja o en la línea 4 si es un nodo interno. Sea x un nodo de B_H , denotamos por x' al nodo de B_G que se toma como base para construir a x .

Sea x un nodo de B_H , notemos lo siguiente. Si x es una hoja de B_H , entonces x es un vértice de H , por lo que $x = x'$, y si x es un nodo interno, entonces la etiqueta de x es la misma que la etiqueta de x' . Además, el hijo derecho de x es ancestro de los mismos vértices de H que el hijo derecho de x' . Análogamente para el hijo izquierdo. De esto se sigue que para cualesquiera dos vértices de H , su ancestro común más profundo en B_H tiene la misma

etiqueta que su ancestro común más profundo en B_G .

Veamos que B_H es un coárbol binario de H . Notemos que las hojas de B_H son todos los vértices de H . Dado que H es una subgráfica inducida de G , cualesquiera dos vértices de H son adyacentes si y sólo si su ancestro común más profundo en B_G tiene etiqueta 1. Luego, dos vértices de H son adyacentes si y sólo si su ancestro común más profundo en B_H tiene etiqueta 1. Así, B_H es un subcoárbol binario de H .

Sea f un subconjunto de $V(B_H) \times V(B_G)$ tal que $(x, x') \in f$ para cualquier nodo x de $V(B_H)$, veamos que f es una función de coasignación de B_H a B_G . Como cada pareja en f tiene como primer elemento a un nodo único de B_H y como segundo elemento un nodo único de B_G , f es una función inyectiva. Sean x_1 , x_2 y x_3 nodos de B_H . Es claro que si x es una hoja, entonces $f(x) = x'$ es una hoja. Y si x es un nodo interno, entonces $f(x) = x'$ tiene la misma etiqueta que x . Si x_3 es el ancestro común más profundo de x_1 y x_2 , notemos lo siguiente:

- el conjunto de vértices de H que son descendientes de x_1 es igual al conjunto de vértices de H que son descendientes de $f(x_1)$.
- el conjunto de vértices de H que son descendientes de x_2 es igual al conjunto de vértices de H que son descendientes de $f(x_2)$.
- el hijo derecho de x_3 es ancestro de los mismos vértices de H que el hijo derecho de $f(x_3)$.

Supongamos sin pérdida de generalidad que x_1 es descendiente del hijo izquierdo de x_3 y que x_2 es descendiente de su hijo derecho. Luego, $f(x_1)$ es descendiente del hijo izquierdo de $f(x_3)$ y $f(x_2)$ es descendiente de su hijo derecho. Así, $f(x_3)$ es el ancestro común más profundo de $f(x_1)$ y $f(x_2)$. Luego, f es una función de coasignación de B_H a B_G .

□

4.1.5. Algoritmo para encontrar obstrucciones mínimas

La presente sección aborda el problema de determinar si una cográfica G tiene a otra cográfica H como subgráfica inducida haciendo uso de los conceptos de coárbol binario y subcoárbol. Se proporciona un algoritmo (Algoritmo 6) para resolver este problema tal que, si se fija el tamaño de H , su tiempo de ejecución crece de forma lineal con respecto al tamaño de G . Este algoritmo

es útil para identificar a las gráficas pertenecientes a una clase caracterizada a través de su conjunto de obstrucciones mínimas de forma rápida.

Algoritmo para determinar si un coárbol binario es subcoárbol binario de otro

Sean T y U coárboles binarios y u un nodo de U , decimos que $f: V(U) \rightarrow \{\text{marcado}, \text{no_marcado}\}$ es una **función de verificación** de T para U si $f(u) = \text{marcado}$ si y sólo si el coárbol binario con raíz u es subcoárbol binario de T . Si $f(u) = \text{marcado}$, decimos que f **marca** a u .

El Algoritmo 5 recibe como entradas dos coárboles binarios, G y H representados por sus raíces g y h respectivamente, y devuelve una función de verificación, f_g , de G para H . Este algoritmo crea la función de verificación de cada subárbol de G para H , empezando por aquellos cuya raíz es más profunda. De esta manera, si la función de verificación de G para H evaluada en h es *marcado*, entonces H es subcoárbol de G .

Algoritmo 5: Función_de_coasignación

Entrada: g y h , las raíces de dos coárboles binarios para las gráficas G y H respectivamente

Salida: $func$, la función de verificación de G para H

- 1 $func \leftarrow$ nueva función de coasignación tal que $func(x) = \text{no_marcado}$ para todo $x \in V(H)$
 - 2 **si** g es una hoja **entonces**
 - 3 $func$ marca a todas las hojas de H
 - 4 **en otro caso**
 - 5 $v_{izq} \leftarrow$ Función_de_coasignación($g.izquierda, h$)
 - 6 $v_{der} \leftarrow$ Función_de_coasignación($g.derecha, h$)
 - 7 **para cada** $nodo$ de H **hacer**
 - 8 **si** $v_{izq}(nodo) = \text{marcado}$ o $v_{der}(nodo) = \text{marcado}$ **entonces**
 - 9 $func(nodo) \leftarrow \text{marcado}$
 - 10 **si no, si** $nodo.etiqueta = g.etiqueta$ y v_{izq} marca a uno de los hijos de $nodo$ y v_{der} al otro **entonces**
 - 11 $func(nodo) \leftarrow \text{marcado}$
 - 12 **devolver** $func$
-

Teorema 4.1.3. *La ejecución del Algoritmo 5, Función_de_coasignación(g, h) regresa una función, $func$, tal que $func$ es una función de verificación de $árbol(g)$ para $árbol(h)$.*

Demostración. Sea n un nodo de $árbol(h)$. Para probar que $func$ es una función de verificación de $árbol(g)$ para $árbol(h)$, tenemos que probar que $func(n) = marcado$ si y sólo si $árbol(n)$ es subcoárbol de $árbol(g)$.

Supongamos primero que el algoritmo ha sido ejecutado y que $func$ marca a n . Procedamos por inducción sobre la altura de g .

Caso base: Si g tiene altura 0, entonces g es una hoja, por lo que $func$ marca únicamente a las hojas de $árbol(h)$. Como $func$ marca a n , entonces n es una hoja. Luego, la función $f = \{(n, g)\}$ es una función de coasignación de $árbol(n)$ a $árbol(g)$, por lo que $árbol(n)$ es subcoárbol de $árbol(g)$.

Paso inductivo: Si g tiene altura $k > 0$. Supongamos como hipótesis inductiva que, para todo nodo de un coárbol binario, g' , de altura $k' < k$ se cumple que, si $func' = \text{Función_de_coasignación}(g', h)$ marca a un nodo n' de $árbol(h)$, entonces $árbol(n')$ es subcoárbol de $árbol(g')$.

Como g no es una hoja, el algoritmo debió de entrar al bloque de instrucciones de las líneas 5 a 11. En las líneas 5 y 6 se crean dos funciones que cumplen con la hipótesis inductiva, ya que $g.izquierda$ y $g.derecha$ tienen ambas una altura menor a k . Como $func$ marca a n , entonces n debe de cumplir la condición de la línea 8 o la condición de la línea 10. Si se cumple la condición de la línea 8, entonces $v_{izq}(n) = marcado$ o $v_{der}(n) = marcado$, por lo que $árbol(n)$ es subcoárbol de $árbol(g.izquierda)$ o de $árbol(g.derecha)$, y por lo tanto es subcoárbol de $árbol(g)$. De lo contrario, se cumple la condición de la línea 10, entonces v_{izq} marca a $n.izquierda$ o a $n.derecha$ y v_{der} marca al otro.

Supongamos sin pérdida de generalidad que v_{izq} marca a $n.izquierda$ y que v_{der} marca a $n.derecha$. Sean

$$f_i: V(\text{árbol}(n.izquierda)) \rightarrow V(\text{árbol}(g.izquierda))$$

la función de coasignación de $árbol(n.izquierda)$ a $árbol(g.izquierda)$ y

$$f_d: V(\text{árbol}(n.derecha)) \rightarrow V(\text{árbol}(g.derecha))$$

la función de coasignación de $árbol(n.derecha)$ a $árbol(g.derecha)$; mostremos que la función $f = f_i \cup f_d \cup \{(n, g)\}$ es una función de coasignación de $árbol(n)$ a $árbol(g)$.

Como los dominios de f_i y f_d son ajenos y ninguno contiene a n , entonces f es una función. Como los rangos de f_i y f_d son ajenos, ninguno contiene a g , y como tanto f_i como f_d son inyectivas, entonces f es inyectiva. Por otra parte, por la condición de la línea 10, sabemos que $n.etiqueta = g.etiqueta$. También sabemos que, sea $x \in V(\text{árbol}(n.izquierda))$, si x es una hoja, entonces $f(x) = f_i(x)$ es una hoja y si no, entonces $x.etiqueta = f_i(x).etiqueta = f(x).etiqueta$. Análogamente para un $y \in V(\text{árbol}(n.derecha))$ y f_d . Finalmente, si n es el ancestro común más profundo de dos nodos z_1 y z_2 , entonces z_1 es descendiente de $n.derecha$ y z_2 es descendiente de $n.izquierda$ o viceversa. Supongamos lo primero sin pérdida de generalidad. Luego, por la condición de la línea 10, v_{izq} marca a uno y v_{der} marca al otro. Supongamos sin pérdida de generalidad que v_{izq} marca a z_1 y v_{der} marca a z_2 . Entonces, $f(z_1) = f_i(z_1) \in V(\text{árbol}(g.izquierda))$ y $f(z_2) = f_d(z_2) \in V(\text{árbol}(g.derecha))$, por lo que el ancestro común más profundo de $f(z_1)$ y $f(z_2)$ es $g = f(n)$. Así, f es una función de coasignación de $\text{árbol}(n)$ a $\text{árbol}(g)$ y $\text{árbol}(n)$ es subcoárbol de $\text{árbol}(g)$.

Recíprocamente, supongamos $\text{árbol}(n)$ es subcoárbol de $\text{árbol}(g)$. Seguiremos la ejecución del algoritmo para mostrar que, al final de la misma, $func$ marcará a n . Sea f la función de coasignación de $\text{árbol}(n)$ a $\text{árbol}(g)$, procedamos por inducción sobre la altura de g .

Caso base: Si la altura de g es 0, entonces g es una hoja, por lo que se cumple con la condición de la línea 2 y se ejecuta la línea 3, haciendo que $func$ marque todas las hojas de H . Como $\text{árbol}(n)$ es subcoárbol de $\text{árbol}(g)$ y $\text{árbol}(g)$ sólo tiene un nodo, entonces n debe de ser una hoja. Luego, $func$ marca a n .

Paso inductivo: Si g tiene altura $k > 0$. Supongamos como hipótesis inductiva que todo coárbol, g' , con altura $k' < k$ cumple que, siendo n' un nodo de $\text{árbol}(h)$, si $\text{árbol}(n')$ es subcoárbol de $\text{árbol}(g')$, entonces $func' = \text{Función_de_coasignación}(g', h)$ marca a n' .

Como g no es una hoja, el algoritmo ejecuta las líneas 5 y 6, y posteriormente el bloque de las líneas 8 a 11 para cada nodo de H . Si n es marcada por v_{izq} o v_{der} , entonces se ejecuta la línea 9 y $func$ marca a n . En el caso contrario, probemos que se cumple la condición de la línea 10.

Mostremos primero que $f(n) = g$ procediendo por contradicción. Supongamos que $f(n) = x$ para algún $x \in V(\text{árbol}(g)) - \{g\}$. Como x es descendiente de g , tiene altura menor a k . También sabemos que f es una función de coasignación de $\text{árbol}(n)$ a $\text{árbol}(x)$, por lo que, por hipótesis inductiva, n debería de ser marcado ya sea por v_{izq} o por v_{der} , lo que es una contradicción.

Luego, $f(n) = g$, y por lo tanto n no es una hoja y $f(n).etiqueta = g.etiqueta$.

Mostremos ahora que tanto $n.izquierda$ como $n.derecha$ son marcados cada uno ya sea por v_{izq} o por v_{der} . Sabemos que $f(n.izquierda)$ y $f(n.derecha)$ son descendientes de r . Como $f|_{V(\text{árbol}(n.izquierda))}$ es una función de coasignación de $\text{árbol}(n.izquierda)$ a $\text{árbol}(f(n.izquierda))$ y $f(n.izquierda) \neq g$ ya que f es inyectiva, entonces $\text{árbol}(n.izquierda)$ es subcoárbol de algún descendiente de g , al que llamaremos y . Como y tiene altura menor a k , su función de verificación correspondiente marca a $n.izquierda$ (por hipótesis inductiva), y por la condición de la línea 8, sus ancestros también lo marcan. Luego v_{izq} o v_{der} marcan a $n.izquierda$. Análogamente para $n.derecha$. Así, tanto $n.izquierda$ como $n.derecha$ están marcados cada uno ya sea en v_{izq} o en v_{der} .

Mostremos, por último, que uno es marcado por v_{izq} y el otro es marcado por v_{der} . Como el ancestro común más profundo de $n.izquierda$ y $n.derecha$ es n , y $f(n) = g$, entonces el ancestro común más profundo de $f(n.izquierda)$ y $f(n.derecha)$ debe de ser g . Luego, $f(n.izquierda)$ está en una rama de g y $f(n.derecha)$ está en la otra. Supongamos sin pérdida de generalidad que $f(n.izquierda)$ está en la rama izquierda de g y $f(n.derecha)$ está en la rama derecha. Como $f|_{V(\text{árbol}(n.izquierda))}$ es una función de coasignación de $\text{árbol}(n.izquierda)$ a $\text{árbol}(g.izquierda)$ y por hipótesis inductiva, entonces v_{izq} marca a $n.izquierda$. De forma análoga, v_{der} marca a $n.derecha$. Concluyendo, como $n.etiqueta = r.etiqueta$ y tanto $n.izquierda$ como $n.derecha$ son marcados uno por v_{izq} y el otro por v_{der} , se cumple la condición de la línea 10 y $func$ marca a n . Así, al final de la ejecución del algoritmo, n estará marcado.

□

Dado que, para cada nodo de G se crea una función de verificación cuyo dominio es el conjunto de los nodos de H , el tiempo de ejecución del algoritmo tiene crecimiento $O(|V(G)||V(H)|)$.

Determinar si una cográfica es subcográfica de otra

Usando el Algoritmo 5 como subrutina, se puede idear un algoritmo para determinar si una cográfica H es subcográfica de otra cográfica G , al buscar todas las posibles formas del córbol binario de H en un córbol binario fijo de G .

Algoritmo 6: Es_subgráfica

Entrada: g y h , las raíces de dos coárboles, G y H respectivamente.**Salida:** *verdadero* si la cográfica representada por H es subgráfica de la cográfica representada por G . *falso* en el caso contrario.

```

1  $g\_bin \leftarrow$  CrearÁrbolBinario( $g$ )
2  $h\_bins \leftarrow$  las raíces de todos los coárboles binarios correspondientes
   a  $H$ 
3 para cada  $bin$  en  $h\_bins$  hacer
4    $f =$  Función_de_coasignación( $g\_bin, bin$ )
5   si  $f(bin) = marcado$  entonces
6      $\lfloor$  devolver verdadero
7 devolver falso

```

Como la línea 1 Algoritmo 6 se ejecuta en tiempo $O(|V(G)|)$, la complejidad temporal de éste depende del número de coárboles binarios correspondientes a H (que crece con mayor rapidez). Sin embargo, si se fija H , la complejidad temporal de éste es simplemente $O(|V(G)|)$. Fijar H resultará útil cuando se esté resolviendo un problema específico como el de encontrar una obstrucción mínima en una gráfica. La aplicación de este algoritmo en el presente trabajo de tesis es desarrollar algoritmos que nos permitan identificar a los elementos de una clase hereditaria de cográficas en tiempo lineal. Esto se muestra en el Algoritmo 7.

El Algoritmo 7 es una variación del Algoritmo 6 en el que se fija una clase C cuyos elementos se desea identificar. Al computar el conjunto de coárboles binarios de cada una de las obstrucciones mínimas de C antes de la ejecución del algoritmo 7, la línea 2 del algoritmo se puede ejecutar en tiempo constante. Gracias a esto obtenemos un algoritmo que determina si una gráfica G pertenece a la clase C en tiempo $O(|V(G)|)$.

4.2. La clase M_2

A partir de esta sección abordamos el problema principal de la tesis, determinar si una cográfica acepta una partición en un número dado de gráficas multipartitas completas. Empezamos por dar nombre a las clases de

Algoritmo 7: Pertenece_a_la_clase

Entrada: g , la raíz del coárbol de una cográfica G .**Salida:** *verdadero* si G pertenece a la clase hereditaria de cográficas C

- 1 $g_bin \leftarrow \text{CrearÁrbolBinario}(g)$
 - 2 $C_bins \leftarrow$ las raíces de todos los coárboles binarios de todas las obstrucciones mínimas de la clase C
 - 3 **para cada** bin **en** C_bins **hacer**
 - 4 $f = \text{Función_de_coasignación}(g_bin, bin)$
 - 5 **si** $f(bin) = \text{marcado}$ **entonces**
 - 6 **devolver** *verdadero*
 - 7 **devolver** *falso*
-

cográficas que estudiamos, y procedemos con el análisis de una de estas clases que nos servirá de base para estudiar el resto.

Dado un entero positivo i , la **clase** M_i es la clase de las cográficas tales que su conjunto de vértices acepta una partición (A_1, A_2, \dots, A_i) , donde A_j induce una gráfica multipartita completa para cada $1 \leq j \leq i$. Decimos que (A_1, A_2, \dots, A_i) es una M_i -partición.

Dado que las gráficas multipartitas completas conforman una clase hereditaria de gráficas, la clase M_i también es una clase hereditaria de gráficas para cada entero positivo i . Si G es una gráfica que tiene una M_i -partición (A_1, \dots, A_i) y x es un vértice de G , entonces x es elemento de A_j para algún $1 \leq j \leq i$. Como $G[A_j] - x$ es una gráfica multipartita completa, entonces $(A_1, \dots, A_j - \{x\}, \dots, A_i)$ es una M_i -partición de $G - x$, que resulta ser un elemento de M_i .

Sea i un entero positivo. Como M_i es una clase hereditaria de gráficas, entonces puede ser caracterizada mediante un conjunto de obstrucciones mínimas, y se puede identificar a sus elementos en tiempo lineal con el Algoritmo 7. El siguiente lema contiene una observación que resultará de utilidad, la clase M_i es cerrada bajo la unión completa.

Lema 4.2.1. *Sea i un entero mayor o igual a 2. La clase M_i es cerrada bajo la unión completa.*

Demostración. Sean $G, H \in M_i$, sabemos que existen M_i -particiones (A_1, \dots, A_i) y (B_1, \dots, B_i) de G y H , respectivamente. Veamos que $P = (A_1 \cup B_1, \dots, A_i \cup$

B_i) es una M_i -partición de $G \oplus H$.

Claramente, cada uno de los vértices de $G \oplus H$ se encuentra en exactamente uno de los elementos de P . Así, P es una partición de los vértices de $G \oplus H$. Luego, como la unión completa de gráficas multipartitas completas es una gráfica multipartita, tenemos que $(G \oplus H)[A_j \cup B_j] = G[A_j] \oplus G[B_j]$ es una gráfica multipartita completa para cualquier $1 \leq j \leq i$. Así, P es una M_i partición de $G \oplus H$ y $G \oplus H \in M_i$. \square

Notemos que la clase M_1 es la clase de las gráficas multipartitas completas. En la presente sección nos enfocaremos en la clase M_2 , que es la más pequeña de las clases M_i después de M_1 (que ha sido ampliamente estudiada). Caracterizamos a M_2 a través de su conjunto de obstrucciones mínimas, proporcionamos un algoritmo de tiempo lineal para identificar a sus elementos y presentamos un algoritmo certificador que no sólo determina si una gráfica G pertenece a M_2 , sino que colorea las hojas de su córbol indicando si se encontró una M_2 -partición de G o una obstrucción mínima de la clase como subgráfica inducida de G .

4.2.1. Obstrucciones mínimas

El primer paso en nuestro estudio de la clase M_2 es caracterizar a la misma a través de su conjunto de obstrucciones mínimas, lo cual se realiza en el Teorema 4.2.4, cuya demostración requiere del Teorema 4.2.2 y del Lema 4.2.3. Recordemos que la gráfica conocida como **Paw** es la gráfica obtenida de K_3 al agregar un vértice nuevo y hacerlo adyacente a exactamente un vértice de K_3 , o bien $K_1 \oplus (K_1 + K_2)$.

Teorema 4.2.2 ([9]). *Sea G una gráfica perfecta, G es libre de Paw si y sólo si cada componente de G es libre de K_3 o multipartita completa.*

Aplicando este teorema, podemos concluir lo siguiente. Dado que las cográficas son gráficas perfectas y toda cográfica libre de K_3 es bipartita, si una cográfica G es libre de Paw, entonces G es bipartita o multipartita completa.

Lema 4.2.3. *Sea G una cográfica conexa. Si G es bipartita, entonces G es bipartita completa.*

Demostración. Sea r la raíz del córbol de G . Si G es trivial, es claro que G es bipartita completa. En el caso contrario, r tiene etiqueta 1. Como G es bipartita, entonces es libre de K_3 . Luego, r tiene exactamente dos hijos,

ninguno de los cuales puede contener un K_2 . Así cada uno de los hijos de r representa a un conjunto independiente. Luego, G es la unión completa de dos conjuntos independientes. Es decir, G es una gráfica bipartita completa. \square

Teorema 4.2.4. *Para una cográfica G , las siguientes afirmaciones son equivalentes.*

- (a) $G \in M_2$.
- (b) G no contiene a ninguna de las gráficas de las Figuras 4.4 como subgráficas inducidas.

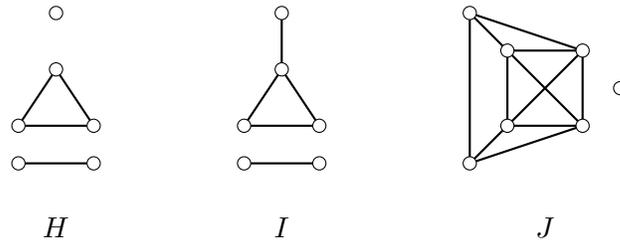


Figura 4.4: Obstrucciones mínimas para la clase M_2 .

Demostración. Notemos que las gráficas H, I y J pueden ser construidas de la siguiente manera:

- (1) $H = K_1 + K_2 + K_3$.
- (2) $I = Paw + K_2$.
- (3) $J = (\overline{P_3} \oplus \overline{P_3}) + K_1$.

Supongamos primero que $G \in M_2$ y procedamos probando la contrapositiva de la afirmación. Es decir, probemos que si G tiene a H , a I o a J como subgráficas inducidas, entonces $G \notin M_2$. Veamos que ninguna partición en dos partes de G es una M_2 -partición.

Sea (X, Y) una partición de G . Si G tiene a H como subgráfica inducida a H . Si ambos vértices de K_2 en H se encuentran en X , entonces la existencia de cualquier vértice adicional en X , implicaría que $G[X]$ contiene un $\overline{P_3}$. Como los vértices restantes inducen una gráfica que no es multipartita completa, esta partición no es una M_2 -partición. Supongamos entonces que un vértice

de K_2 está en X y el otro en Y . Como la gráfica inducida por los vértices restantes de H contiene un K_3 , podemos suponer sin pérdida de generalidad que dos de sus vértices se encuentran en X , por lo que X contiene una copia inducida de $\overline{P_3}$. Como la partición fue elegida arbitrariamente, tenemos que H no pertenece a M_2 . El argumento para probar que I no está en M_2 es análogo al anterior.

Si G tiene a J como subgráfica inducida, como J tiene un vértice aislado, para que admitiera una M_2 -partición, el resto de sus vértices (que inducen un $\overline{P_3} \oplus \overline{P_3}$) deben de poder dividirse en dos partes de manera tales que una induzca un conjunto independiente y la otra una gráfica multipartita completa. Siempre que tomamos uno de los vértices de uno de los dos $\overline{P_3}$ para formar el conjunto independiente, ninguno los vértices del otro $\overline{P_3}$ puede ser agregado al mismo, pues es adyacente al vértice que agregamos primero. Así, la subgráfica inducida $\overline{P_3} \oplus \overline{P_3}$ no acepta una partición en un conjunto independiente y una gráfica multipartita completa. Luego, J no está en M_2 .

Recíprocamente, supongamos que G es libre de H , I y J y probemos que es elemento de M_2 . Para ello consideramos cuatro casos que son exhaustivos. Los primeros tres casos cubren toda situación en la que G es inconexa, mientras que el último caso se cumple si G es conexa.

Caso 1: G tiene al menos dos componentes conexas no triviales.

Consideremos la partición de V en dos partes (A, B) tal que A contiene únicamente una componente no trivial y B el resto. Como $G[A]$ y $G[B]$ contienen ambas componentes no triviales, las dos poseen un K_2 . Luego, ni $G[B]$ ni $G[A]$ pueden contener un Paw , o G tendría a I como subgráfica inducida. Dado que $G[A]$ y $G[B]$ son cográficas, son también gráficas perfectas, y al ninguna tener un Paw como subgráfica inducida, cada una es bipartita o multipartita completa [9].

Si tanto $G[A]$ como $G[B]$ son gráficas multipartitas completas, entonces $G \in M_2$. Si ambas son bipartitas, entonces G es bipartita también y acepta una partición en dos conjuntos independientes, cada uno de los cuales es una gráfica multipartita completa, por lo que $G \in M_2$. Si $G[A]$ es bipartita y $G[B]$ es multipartita completa, como $G[A]$ es una cográfica conexa, entonces es una gráfica multipartita completa y $G \in M_2$.

Finalmente, si $G[A]$ es multipartita completa y $G[B]$ es bipartita. Si $G[B]$ tiene una sola componente, $G[B]$ es bipartita completa y $G \in M_2$. Si $G[B]$ tiene más de una componente, como al menos una es no trivial, debe tener a $\overline{P_3}$ como subgráfica inducida. Luego, $G[A]$ debe ser libre de K_3 o G tendría a H como subgráfica inducida. Así, $G[A]$ es bipartita. Como ambas son

bipartitas, $G \in M_2$.

Caso 2: G tiene exactamente una componente conexa no trivial y al menos una trivial.

Como G contiene al menos una componente trivial, la única partición que puede aceptar en dos gráficas multipartitas completas es una partición en un conjunto independiente y una gráfica multipartita completa. Luego, la componente no trivial de G , a la que llamaremos G' , debe de aceptar una partición en un conjunto independiente y una gráfica multipartita completa.

Si G' es bipartita, entonces acepta una partición en dos conjuntos independientes, y por lo tanto $G \in M_2$. Si G' es una gráfica multipartita completa, entonces $G \in M_2$. Si G no es una gráfica bipartita ni multipartita completa, dado que es una cografía, y por lo tanto una gráfica perfecta, G contiene un Paw . Sea y la raíz del córbol de G' y sea z , descendiente de y , el nodo más profundo que tiene un Paw como subgráfica inducida, probemos por inducción sobre la distancia desde y hasta z , denotada por d , que $G'[y]$ acepta una partición en un conjunto independiente y una gráfica multipartita completa.

Caso base: $d = 0$, o bien, $y = z$.

Notemos que z tiene etiqueta 1, pues Paw es una gráfica conexa. Dado que z tiene etiqueta 1, todos sus hijos inducen gráficas multipartitas completas menos uno, w , que tiene etiqueta 0. Mostremos por contradicción que todos los hijos de w inducen gráficas multipartitas completas. Supongamos que alguno de los hijos de w contiene un \overline{P}_3 . Como el nodo más profundo que contiene un \overline{P}_3 debe tener etiqueta 0, w tiene un hijo de etiqueta 1 y éste a su vez tiene al menos 2 hijos, uno de los cuales contiene a \overline{P}_3 y el otro que tiene al menos un K_1 . Luego, dicho hijo contiene un Paw , lo que es una contradicción.

Si w tiene un sólo hijo que no es un vértice, el resto de sus hijos forman un conjunto independiente, C . Si eliminamos este conjunto independiente, como el único hijo de w que queda induce una gráfica multipartita completa, entonces $G'[w] - C$ es una gráfica multipartita completa. Luego, $G'[z] - C$ es la unión completa de varias gráficas multipartitas completas y por lo tanto es una gráfica multipartita completa. De esto se sigue que $G'[z]$ acepta una partición en un conjunto independiente, C , y una gráfica multipartita completa $G'[z] - C$.

Si w tiene al menos dos hijos no triviales, notemos que ninguno de ellos puede contener a K_3 , o de lo contrario w contendría a $K_2 + K_3$ y G no sería libre de I . Luego, todos los hijos de w inducen gráficas bipartitas, es decir que w induce también una gráfica bipartita. En otras palabras, $G'[w]$

acepta una partición en dos conjuntos independientes. Si sustraemos uno de estos conjuntos independientes, denotado por D , entonces $G'[w] - D$ es un conjunto independiente. Luego $G'[z] - D$ es la unión completa de al menos una gráfica multipartita completa y un conjunto independiente. Así, $G'[z] - D$ es una gráfica multipartita completa. Luego, $G'[z]$ acepta una partición en un conjunto independiente, D y una gráfica multipartita completa, $G'[z] - D$.

Como en todos los casos z acepta una partición en un conjunto independiente y una gráfica multipartita completa y $y = z$, entonces y acepta la misma partición.

Paso inductivo: $d \geq 2$.

Notemos que d siempre será par, ya que tanto y como z son nodos con etiqueta 1. Sea k un entero tal que $k \geq 2$. Supongamos, como hipótesis inductiva, que si G'' es una cografía conexa libre de H, I y J tal que la distancia, d' , entre la raíz, y' de su cóarbol y el nodo más profundo que contiene un Paw es igual a $k - 2$, entonces G'' acepta una partición en un conjunto independiente y una gráfica multipartita completa.

Dado que G' es libre de J , todos los hijos de y , menos uno, inducen gráficas multipartitas completas. Dicho hijo, v , tiene etiqueta 0 y al menos uno de sus hijos debe de contener un Paw . Denotemos a dicho hijo como u . El resto de los hijos de v deben de ser vértices, o de lo contrario, $G'[v]$ contendría a $K_2 + K_3$ como subgráfica inducida, por lo que G contendría a I . Denotemos a este conjunto de vértices como E . Luego, $G'[u]$ es una cografía que cumple con las condiciones de la hipótesis inductiva, por lo que acepta una partición en un conjunto independiente, D y una gráfica multipartita completa. Tenemos entonces que $G'[u] - D$ es una gráfica multipartita completa. Se sigue que $(G'[v] - D) - E$ es una gráfica multipartita completa. Luego, $(G' - D) - E$ es una unión completa de gráficas multipartitas completas por lo que también es una gráfica multipartita completa. Notemos que, dado que v tiene etiqueta 0, no existen aristas entre los vértices en D y los vértices en E , es decir que $D \cup E$ es un conjunto independiente. Así, G' acepta una partición en un conjunto independiente, $D \cup E$ y una gráfica multipartita completa, $(G' - D) - E$.

Como G' acepta una partición en un conjunto independiente y una gráfica multipartita completa, entonces $G \in M_2$.

Caso 3: G es un conjunto independiente con al menos dos vértices.

Dado que G es una gráfica multipartita completa, se sigue inmediatamente que está en M_2 .

Caso 4: G es conexa.

Dado que toda cografía inconexa libre de H, I y J acepta una partición

en dos gráficas multipartitas completas, y como una cográfica conexa es o un vértice aislado o una unión completa de cográficas inconexas se sigue de los casos anteriores y del hecho de que la clase M_2 es cerrada bajo uniones completas (Lema 4.2.1), que $G \in M_2$. □

4.2.2. Reconocimiento de la clase M_2

Haciendo uso del Algoritmo 7 se puede determinar si una cográfica pertenece o no a la clase M_2 . Como se especificó en la Sección 4.1.5, el tiempo de este algoritmo crece de forma lineal de acuerdo con el tamaño de la gráfica de entrada si encontramos primero todos los coárboles binarios de las obstrucciones de la clase. Como conocemos las obstrucciones mínimas de la clase M_2 , que son finitas, se puede buscar cada una en tiempo lineal y por lo tanto se puede reconocer si una cográfica pertenece a la clase M_2 en tiempo lineal. El Algoritmo 8, que es una instancia del Algoritmo 7 corresponde a este proceso. Los árboles binarios de cada una de las obstrucciones de la clase M_2 se muestran en la Figura 4.5.

Algoritmo 8: Pertenece_a_M2

Entrada: g , la raíz del coárbol de una cográfica G .

Salida: *verdadero* si G pertenece a la clase hereditaria de cográficas C

```

1  $g\_bin \leftarrow$  CrearÁrbolBinario( $g$ )
2  $C\_bins \leftarrow \{H_1, H_2, H_3, I_1, J_1\}$ 
3 para cada  $bin$  en  $C\_bins$  hacer
4    $f =$  Función_de_coasignación( $g\_bin, bin$ )
5   si  $f(bin) = marcado$  entonces
6     devolver verdadero
7 devolver falso

```

4.2.3. Algoritmo certificador

Si bien, el Algoritmo 8 es capaz de identificar a las cográficas que pertenecen a la clase M_2 , no es posible determinar a partir de éste cuáles son las dos partes en las que se puede dividir una gráfica de la clase. La presente

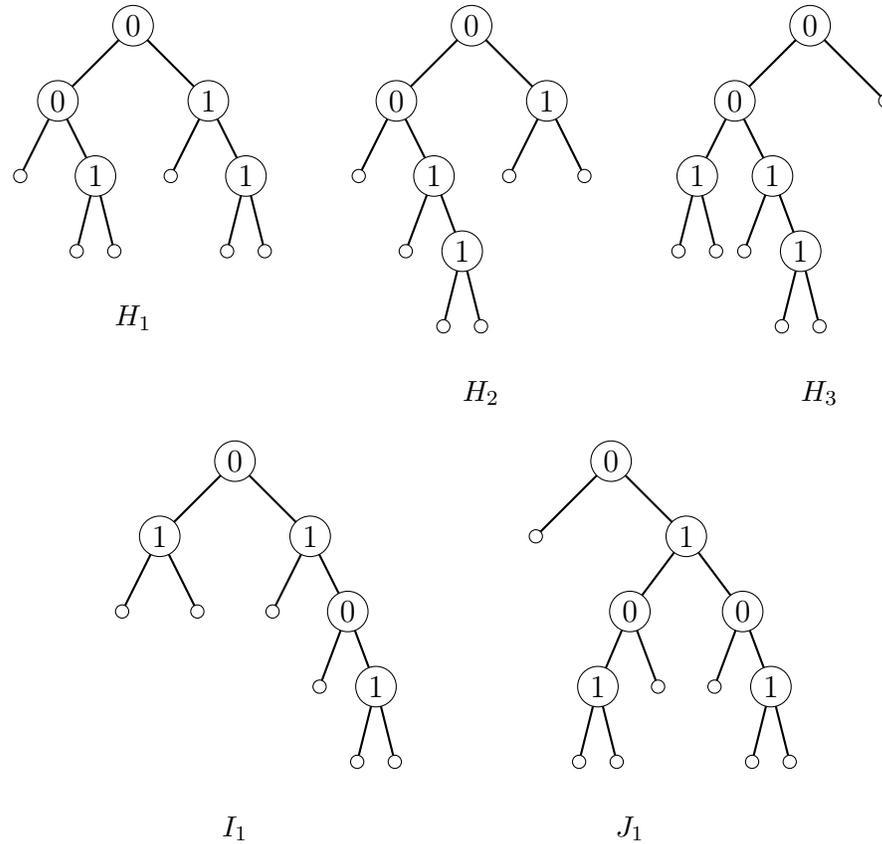


Figura 4.5: H_1 , H_2 y H_3 son los coárboles binarios correspondientes a la obstrucción H . El coárbol binario I_1 es el único que corresponde a la obstrucción I . El coárbol binario J_1 es el único que corresponde a la obstrucción J .

subsección muestra un algoritmo (Algoritmo 12) que, dada una cográfica G representada por su coárbol T , devuelve una coloración de las hojas de este último. Si G pertenece a M_2 , cada una de las hojas de T tendrá uno de dos colores, *verde* o *azul*. Cada uno de estos cuales corresponde a una parte de la M_2 -partición de G . En el caso contrario, las hojas de T correspondientes a los vértices que forman una obstrucción mínima tendrán un color que indique de qué obstrucción mínima se trata (*amarillo* para H , *anaranjado* para I y *rojo* para J). El Algoritmo 12 hace uso de los Algoritmos 10 y 11, que funcionan de la misma manera para casos específicos del problema. La correctitud de estos algoritmo se sigue de la demostración del Teorema 4.2.4

Algoritmo para reconocer gráficas bipartitas completas conexas

El Algoritmo 9 es un algoritmo que resulta útil para los algoritmos subsecuentes. Éste recibe la raíz de un coárbol, g , y devuelve *verdadero* si la gráfica representada por dicho coárbol es una gráfica bipartita completa conexas, coloreando los vértices de una parte de color *azul* y los de la otra parte de *verde*. En el caso contrario, se colorean con *amarillo* tres hojas cuyo ancestro común más profundo sea un nodo con etiqueta 1. Es decir que se colorean los vértices que inducen un K_3 en la gráfica. El bloque de la línea 9 a la 28 se ejecuta sólo si g tiene exactamente dos hijos. En las líneas 10 a 18 se busca un K_3 en el primer hijo de g y en las líneas 19 a 27 se busca en el segundo hijo. La Figura 4.6 muestra el resultado de la ejecución de este algoritmo para algunos coárboles.

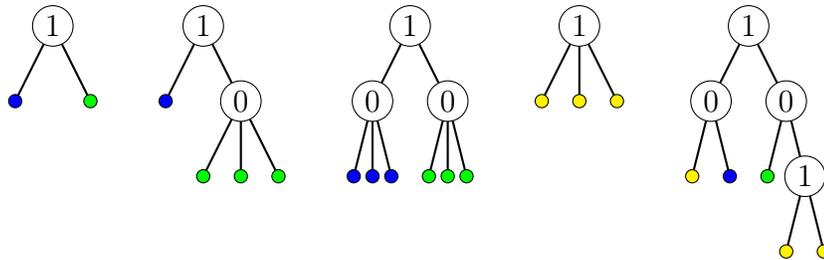


Figura 4.6: Ejemplos del resultado de la ejecución del Algoritmo 9.

Caso 1

El algoritmo 10 corresponde al *Caso 1* de la demostración del Teorema 4.2.4. Éste recibe como entrada la raíz de un coárbol que representa una cográfica inconexa que tiene al menos dos componentes conexas no triviales. En el bloque de las líneas 1 a 12 se aborda el caso en el que la gráfica tiene exactamente dos componentes conexas y se busca un *Paw* que pueda formar la obstrucción I . En el bloque de las líneas 13 a 17 se aborda el caso en el que hay al menos 3 componente conexas y se busca un K_3 en cada componente para formar la obstrucción H . Si no se encuentra ninguna de las obstrucciones mínimas, se devuelve *verdadero* y cada una de las hojas del coárbol tendrán color *azul* o *verde*. Las Figuras 4.7 y 4.8 muestran la ejecución del algoritmo para gráficas sin ninguna de las obstrucciones. La

Algoritmo 9: Es_bipartita_completa

Entrada: g , la raíz de un coárbol, G **Salida:** Verdadero si la gráfica representada por G es bipartita completa. Falso en el caso contrario. Las hojas de $\text{árbol}(g)$ se colorean.

```

1 si  $g$  es una hoja entonces
2   |  $g.color \leftarrow azul$ 
3   | devolver verdadero
4 si no, si  $g.etiqueta = 0$  entonces
5   | devolver falso
6 si no, si  $g.hijos.tamaño > 2$  entonces
7   | Marcar con amarillo: una hoja en cada uno de tres hijos
8   |   diferentes de  $g$ 
9   | devolver falso
9 en otro caso // Hay exactamente dos hijos
10  | si  $g.hijos[0]$  es una hoja entonces
11  |   |  $g.hijos[0].color \leftarrow azul$ 
12  |   en otro caso
13  |     | para cada  $gchild$  en  $g.hijos[0].hijos$  hacer
14  |       | si  $gchild$  es una hoja entonces
15  |         |   |  $gchild.color \leftarrow azul$ 
16  |         |   en otro caso
17  |           |   | Marcar con amarillo: dos hojas que tengan como
18  |             |   |   ancestro común más profundo a  $gchild$  y una hoja
19  |             |   |   descendiente de  $g.hijos[1]$ 
20  |             |   | devolver falso
21  |             | si  $g.hijos[1]$  es una hoja entonces
22  |             |   |  $g.hijos[1].color \leftarrow verde$ 
23  |             |   en otro caso
24  |               |   | para cada  $gchild$  en  $g.hijos[1].hijos$  hacer
25  |                 |   | si  $gchild$  es una hoja entonces
26  |                   |   |   |  $gchild.color \leftarrow verde$ 
27  |                   |   |   en otro caso
28  |                     |   |   | Marcar con amarillo: dos hojas que tengan como
29  |                       |   |   |   ancestro común más profundo a  $gchild$  y una hoja
30  |                       |   |   |   descendiente de  $g.hijos[0]$ 
31  |                       |   |   | devolver falso
32  |                       |   | devolver verdadero
33  |                       | devolver verdadero
34  | devolver verdadero

```

Figura 4.9 muestra el resultado de la ejecución para tres gráficas, cada una de las cuales contiene una obstrucción.

Algoritmo 10: M2_Caso_1

Entrada: g , la raíz de un coárbol con etiqueta 0 y al menos dos hijos que no son hojas

Salida: Verdadero si G pertenece a la clase M_2 . Falso en el caso contrario. Las hojas de $\text{árbol}(g)$ se colorean.

```

1 si  $g.hijos.tamaño = 2$  entonces
2   para  $gchild$  en  $g.hijos[0]$  hacer
3     si  $gchild$  es una hoja entonces
4        $gchild.color \leftarrow azul$ 
5     en otro caso
6       para cada  $ggchild$  en  $gchild.hijos$  hacer
7         si  $ggchild$  es una hoja entonces
8            $ggchild.color \leftarrow azul$ 
9         en otro caso // Se marca la obstrucción  $I$ 
10          Marcar con anaranjado: una hoja en  $ggchild.hijos[0]$ ,
              una hoja en  $ggchild.hijos[1]$ , una hoja en un hermano
              de  $ggchild$ , una hoja en un hermano de  $gchild$  y dos
              hojas cuyo ancestro común más profundo sea el
              hermano de  $g.hijos[0]$ 
11          devolver falso
12   Repetir el procedimiento de las líneas 2 a 11 para  $g.hijos[1]$ , pero
      marcando con color verde en vez de azul
13 en otro caso
14   para  $child$  en  $g.hijos$  hacer
15     si  $Es\_bipartita\_completa(child) = falso$  entonces
16       Marcar con amarillo: dos hojas cuyo ancestro común más
              profundo sea un hermano de  $child$  que no sea una hoja y una
              hoja en un hermano diferente
17       devolver falso
18 devolver verdadero

```

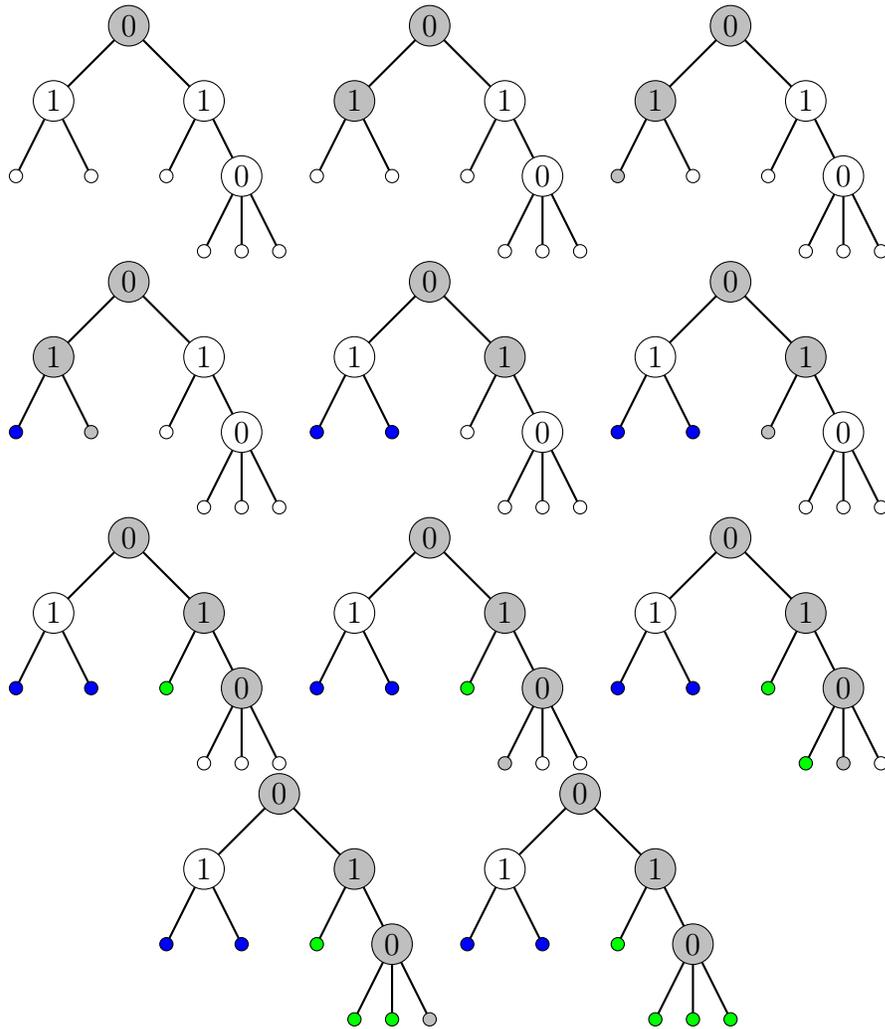


Figura 4.7: Ejemplo de la ejecución del Algoritmo 10. Se muestran en color gris los nodos del árbol que están siendo procesados. Los colores de las hojas corresponden a los colores que asigna el algoritmo.

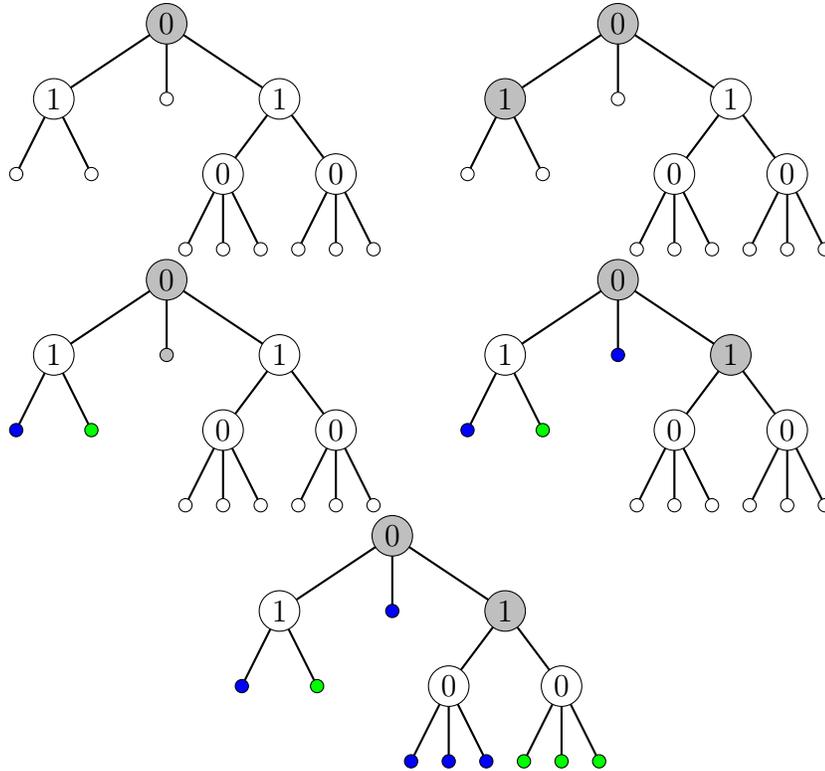


Figura 4.8: Ejemplo de la ejecución del Algoritmo 10. Se muestran en color gris los nodos del árbol que están siendo procesados. Los colores de las hojas corresponden a los colores que asigna el algoritmo.

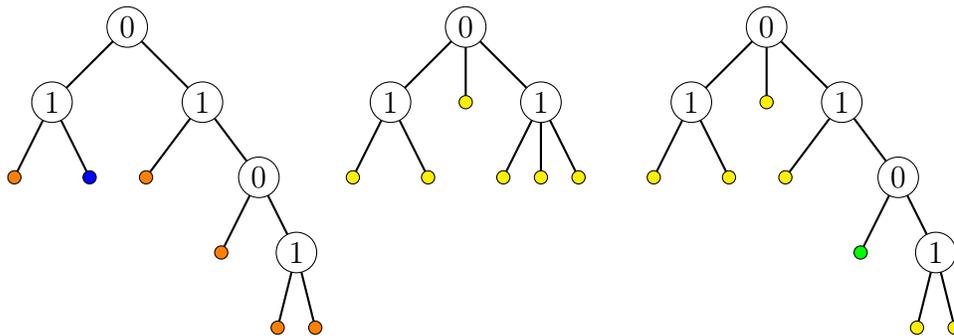


Figura 4.9: Ejemplos del resultado de la ejecución del Algoritmo 10 en los que se encuentra una obstrucción.

Caso 2

El algoritmo 11 corresponde al *Caso 2* de la demostración del Teorema 4.2.4. Éste recibe como entrada la raíz, g , de un coárbol que representa una cográfica inconexa que tiene exactamente una componente conexa no trivial y al menos una trivial. En el bloque de las líneas 6 a 28 se procesa el hijo de g que no es una hoja. En las líneas 7 a 19 se procesan los nietos de g y se registra si alguno tiene un hijo que no sea una hoja (es decir que dicho nieto de g corresponde a una gráfica no multipartita completa) en la variable *aux_gchild*. La cantidad de hijos diferentes de una hoja de éste se registra en *ggchildren_no_hojas*. Si hay más de un nieto que tenga hijos que no son hojas, se marca la obstrucción J (Línea 18). Una vez procesados los nietos de g , se decide cómo será procesado el nieto de g que no corresponde a una gráfica multipartita completa. Si tal hijo no existe, la partición ya se habrá hecho (Líneas 20 y 21), esto corresponde a una parte del caso base del *Caso_2* de la demostración del Teorema 4.2.4. Si dicho nieto tiene un solo hijo que no es una hoja, se procesa recursivamente (Líneas 22 y 23), esto corresponde al paso inductivo del *Caso_2* de la demostración ya mencionada. Y finalmente, si tiene más de un hijo que no es una hoja, se busca que todos estos hijos sean bipartitas, esta es la otra parte del caso base del *Caso_2* de la demostración. La Figura 4.10 muestra un ejemplo de la ejecución del algoritmo para un coárbol cuya cográfica no contiene a ninguna de las obstrucciones mínimas de M_2 . La Figura 4.11 muestra el resultado de la ejecución para coárboles que contienen una obstrucción.

Algoritmo certificador

Finalmente, el Algoritmo 12 utiliza los algoritmos anteriores para colorear las hojas del coárbol recibido como entrada, g . En el caso de que la gráfica sea conexa (líneas 4 a 8), simplemente se llama el algoritmo para cada una de los hijos de g . Esto no significa que sea un algoritmo recursivo, dado que, para las gráficas inconexas y las hojas, el algoritmo no vuelve a ser llamado. En el caso de que la gráfica sea inconexa, se ejecuta el bloque de las líneas 10 a 21. En las líneas 10 a 15 se cuenta el número de componentes conexas de la gráfica representada (es decir que se cuentan los hijos de g que no son hojas). Y por último se toma la decisión de qué caso debe llamarse.

Algoritmo 11: M2_Caso_2

Entrada: g , la raíz de un coárbol con etiqueta 0 que tiene exactamente un hijo que no es una hoja y al menos uno que es una hoja

Salida: Verdadero si G pertenece a la clase M_2 . Falso en el caso contrario. Las hojas de G se colorean.

```

1   $aux\_gchild \leftarrow null$ 
2   $ggchildren\_no\_hojas \leftarrow 0$ 
3  para  $child$  en  $g.hijos$  hacer
4  | si  $child$  es una hoja entonces
5  | |  $child.color \leftarrow azul$ 
6  | en otro caso // Sólo se ejecuta una vez
7  | | para  $gchild$  en  $child.hijos$  hacer
8  | | | si  $gchild$  es una hoja entonces
9  | | | |  $gchild.color \leftarrow verde$ 
10 | | | en otro caso
11 | | | | para  $ggchild$  en  $gchild.hijos$  hacer
12 | | | | | si  $ggchild$  es una hoja entonces
13 | | | | | |  $ggchild.color \leftarrow verde$ 
14 | | | | | si no, si  $aux\_gchild = null$  o  $aux\_gchild = gchild$  entonces
15 | | | | | |  $aux\_gchild \leftarrow gchild$ 
16 | | | | | |  $ggchildren\_no\_hojas \leftarrow ggchildren\_no\_hojas + 1$ 
17 | | | | | en otro caso
18 | | | | | | Marcar con rojo: Un hijo de  $g$  que sea una hoja, dos hojas cuyo
19 | | | | | | | ancestro común más profundo sea  $ggchild$ , una hoja en un
20 | | | | | | | | hermano de  $ggchild$ , dos hojas cuyo ancestro común más
21 | | | | | | | | | profundo sea un hijo de  $aux\_gchild$  que no es una hoja y una
22 | | | | | | | | | hoja en un hijo de  $aux\_gchild$  diferente del anterior
23 | | | | | | | | | devolver falso
24 | | | | | si  $ggchildren\_no\_hojas = 0$  entonces
25 | | | | | |  $devolver verdadero$ 
26 | | | | | si no, si  $ggchildren\_no\_hojas = 1$  entonces
27 | | | | | |  $devolver M2\_Caso\_2(aux\_gchild)$ 
28 | | | | | en otro caso
29 | | | | | | para  $ggchild$  en  $aux\_gchild$  hacer
30 | | | | | | | si  $Es\_bipartita\_completa(ggchild) = falso$  entonces
31 | | | | | | | | Marcar con amarillo: dos hojas cuyo ancestro común más profundo
32 | | | | | | | | | sea un hermano de  $ggchild$  que no sea una hoja y un hijo de  $g$  que
33 | | | | | | | | | sea una hoja
34 | | | | | | | | | devolver falso
35 | | | | | | devolver verdadero

```

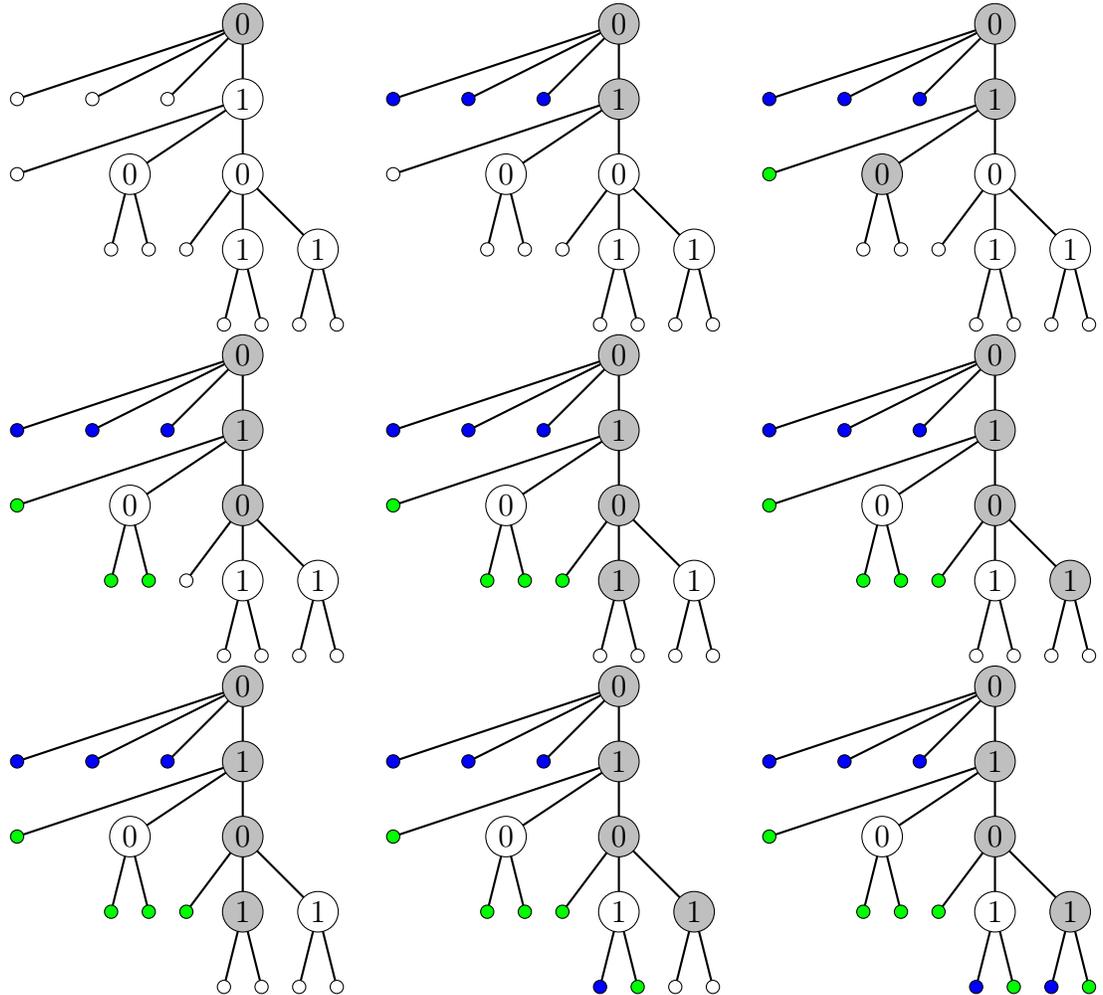


Figura 4.10: Ejemplo de la ejecución del Algoritmo 11. Se muestran en color gris los nodos del árbol que están siendo procesados. El procesamiento de las hojas hermanas se realiza en una sola imagen. Los colores de las hojas corresponden a los colores que asigna el algoritmo.

Algoritmo 12: M2_Certificador

Entrada: g , la raíz de un coárbol, G **Salida:** Verdadero si la gráfica representada por G pertenece a la clase M_2 . Falso en el caso contrario. Las hojas de G se colorean.

```

1 si  $g$  es una hoja entonces
2   |  $g.color \leftarrow azul$ 
3   | devolver verdadero
4 si no, si  $g.etiqueta = 1$  entonces
5   | para  $child$  en  $g.hijos$  hacer
6   |   | si M2_Certificador( $child$ ) = falso entonces
7   |   |   | devolver falso
8   |   |   | devolver verdadero
9 en otro caso
10  |  $componentes\_no\_triviales \leftarrow 0$ 
11  | para  $child$  en  $g.hijos$  hacer
12  |   | si  $child$  es una hoja entonces
13  |   |   |  $child.color \leftarrow azul$ 
14  |   |   | en otro caso
15  |   |   |   |  $componentes\_no\_triviales \leftarrow$ 
16  |   |   |   |   |  $componentes\_no\_triviales + 1$ 
17  |   | si  $componentes\_no\_triviales = 0$  entonces
18  |   |   | devolver verdadero
19  |   | si no, si  $componentes\_no\_triviales = 1$  entonces
20  |   |   |   | devolver M2_Caso_2( $g$ )
21  |   |   |   | en otro caso
22  |   |   |   |   | devolver M2_Caso_1( $g$ )
22 devolver verdadero

```

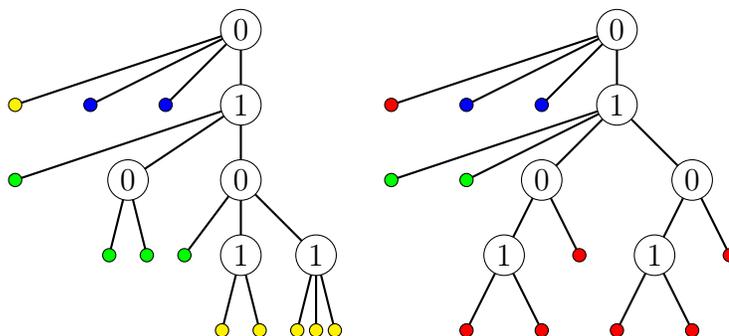


Figura 4.11: Ejemplo del resultado de la ejecución del Algoritmo 11 para cóarboles que incluyen una obstrucción.

4.3. Las clases (α, β) - M_2

Al igual que con las cográficas polares, podemos obtener subclases de M_2 al limitar el tamaño de sus partes. En la presente sección estudiamos un conjunto de subclases de M_2 a las que llamamos clases (α, β) - M_2 . El estudio de estas clases consolida un ejemplo de cómo los conjuntos de obstrucciones mínimas de un conjunto de clases C_1, C_2, \dots tales que $C_i \subset C_{i+1}$ para todo entero $i \geq 1$ se relacionan de manera que podemos encontrar fórmulas y algoritmos para generar dichos conjuntos de obstrucciones mínimas. El resultado principal de esta sección es un algoritmo de fuerza bruta para encontrar algunas obstrucciones mínimas de cualquier clase (α, β) - M_2 . Este algoritmo es utilizado para generar los conjuntos de obstrucciones mínimas de varias clases.

4.3.1. Definición y clases conocidas

Sean α y β enteros tales que $0 < \alpha \leq \beta$, una gráfica G está en la **clase** (α, β) - M_2 si y sólo si su conjunto de vértices acepta una partición (A, B) tal que $G[A]$ es una gráfica multipartita completa formada por a lo más α conjuntos independientes y $G[B]$ es una gráfica multipartita completa formada por a lo más β conjuntos independientes. Decimos que (A, B) es una M_2 -partición de G de tamaño (α, β) .

Notemos que la clase M_2 es la clase (∞, ∞) - M_2 y que la clase $(1, 1)$ - M_2 es la clase de todas las cográficas bipartitas. En la demostración del Teorema 4.2.4 hablamos de las cográficas que aceptan una partición en un conjunto

independiente y una gráfica multipartita completa. Es decir, las gráficas de la clase $(1, \infty)$ - M_2 . A partir de esta demostración, podemos encontrar las obstrucciones mínimas de la clase; éstas nos serán de ayuda para entender el comportamiento de las clases $(1, \beta)$ - M_2 .

Lema 4.3.1. *Si G es una cografía, entonces G está en la clase $(1, \infty)$ - M_2 si y sólo si G es libre de las gráficas de la Figura 4.12.*

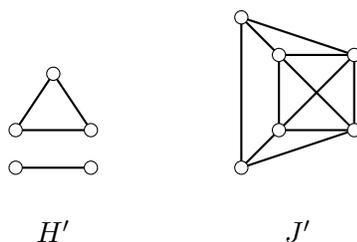


Figura 4.12: Obstrucciones mínimas para la clase $(1, \infty)$ - M_2 .

Demostración. Notemos que $H' = K_2 + K_3$ y $J' = \overline{P_3} \oplus \overline{P_3}$.

Supongamos primero que $G \in (1, \infty)$ - M_2 . Veamos que $H' \notin (1, \infty)$ - M_2 . Procedamos por contradicción. Supongamos que $H' \in (1, \infty)$ - M_2 . Luego, H (Figura 4.4) también está en $(1, \infty)$ - M_2 y por lo tanto $H \in M_2$, lo que es una contradicción. Así, $H' \notin (1, \infty)$ - M_2 . Análogamente para J' . Como ni H' ni J' están en $(1, \infty)$ - M_2 y toda subgráfica inducida de G sí lo está, G es libre de H' y J' .

Recíprocamente, si G es libre de H' y J' , mostremos que $G \in (1, \infty)$ - M_2 . Sea r la raíz del cóarbol de G . Consideremos los siguientes casos que son exhaustivos.

Consideremos el caso en el que G es inconexa. Si G es una gráfica vacía, es claro que $G \in (1, \infty)$ - M_2 . Si G tiene exactamente una componente no trivial y al menos una trivial. Entonces, del Caso 2 de la segunda parte de la demostración del Teorema 4.2.4, se sigue que $G \in (1, \infty)$ - M_2 . Si G tiene al menos dos componentes no triviales. Como G es libre de H' , entonces cada componente de G es libre de K_3 . Es decir que cada componente de G es bipartita. Luego, G es bipartita. Así, $G \in (1, 1)$ - M_2 , por lo que $G \in (1, \infty)$ - M_2 .

Por otra parte, consideremos el caso en el que G es conexa. Si G es isomorfa a K_1 , es claro que $G \in (1, \infty)$ - M_2 . En el caso contrario, como G es

libre de J' , todos los hijos de r representan gráficas multipartitas completas excepto a lo más uno. Si todos sus hijos representan gráficas multipartitas completas, entonces G es la unión completa de varias gráficas multipartitas completas, por lo que G es multipartita completa. Si r tiene un hijo s que no representa una gráfica multipartita completa, como $G[s]$ es inconexa y libre de H' y J' , entonces, procediendo inductivamente, $G[s]$ acepta una partición (A, B) en un conjunto independiente $G[A]$ y una gráfica multipartita completa $G[B]$. Notemos que $G' = G - V(G[s])$ es una gráfica multipartita completa dado que todos los hijos de r diferentes de s representan gráficas multipartitas completas. Como todos los vértices de G' son adyacentes a los vértices de $G[s]$, entonces $G' \oplus G[B] = G - A$ es una gráfica multipartita completa. Luego, $(A, V(G) - A)$ es una partición de G en un conjunto independiente y una gráfica multipartita completa. Así, $G \in (1, \infty)\text{-}M_2$. \square

4.3.2. Conjunto de parejas mínimas

En esta subsección, presentamos el concepto de conjunto de parejas mínimas, que nos ayudará a determinar si una cográfica G pertenece a una clase $(\alpha, \beta)\text{-}M_2$ sin tener que revisar todas las posibles M_2 -particiones de G .

Sean $P_1 = (\alpha_1, \beta_1)$ y $P_2 = (\alpha_2, \beta_2)$ parejas de enteros tales que $0 < \alpha_1 \leq \beta_1$ y $0 < \alpha_2 \leq \beta_2$. Decimos que P_1 **domina** a P_2 si $\alpha_1 \geq \alpha_2$ y $\beta_1 \geq \beta_2$. Es fácil observar que ésta es una relación reflexiva, transitiva y antisimétrica, por lo que es una relación de orden.

Sea G una gráfica en M_2 , denotamos por $S(G)$ al conjunto de todas las parejas ordenadas de enteros (α, β) tales que G acepta una M_2 -partición de tamaño (α, β) .

Sea G una gráfica en M_2 , el **conjunto de parejas mínimas** de G , denotado por $\mu(G)$, es el conjunto de todas las parejas $(\alpha, \beta) \in S(G)$ tales que, para cualquier otro $(\alpha', \beta') \in S(G)$, (α, β) no domina a (α', β') . Al ser $(S(G), \leq)$ un orden parcial, toda pareja en $S(G)$ domina a alguna pareja en $\mu(G)$.

Lema 4.3.2. *Si G es una gráfica en M_2 y (α, β) una pareja de enteros mayores a cero tales que $\alpha \leq \beta$, entonces G pertenece a la clase $(\alpha, \beta)\text{-}M_2$ si y sólo si existe una pareja de enteros $(\alpha', \beta') \in \mu(G)$ tal que (α, β) domina a (α', β') .*

Demostración. Supongamos primero que $G \in (\alpha, \beta)$ - M_2 . Sabemos que G acepta una M_2 -partición (A, B) tal que $G[A]$ es una gráfica multipartita completa formada por α' conjuntos independientes con $\alpha' \leq \alpha$ y $G[B]$ es una gráfica multipartita completa formada por β' conjuntos independientes con $\beta' \leq \beta$. Como (A, B) es una M_2 -partición de G , entonces $(\alpha', \beta') \in S(G)$. Supongamos sin pérdida de generalidad (dado que la dominación es una relación transitiva) que $(\alpha', \beta') \in \mu(G)$. Luego, como $\alpha' \leq \alpha$ y $\beta' \leq \beta$, entonces (α, β) domina a $(\alpha', \beta') \in \mu(G)$.

Recíprocamente. Como $(\alpha', \beta') \in \mu(G)$, entonces $(\alpha', \beta') \in S(G)$. Luego, existe una M_2 -partición (A, B) de G de tamaño (α', β') . Como (α, β) domina a (α', β') , entonces $G[A]$ es una gráfica multipartita completa formada por a lo más α conjuntos independientes y $G[B]$ es una gráfica multipartita completa formada por a lo más β conjuntos independientes. Así, $G \in (\alpha, \beta)$ - M_2 . \square

4.3.3. Cálculo del conjunto de parejas mínimas de una gráfica

A continuación presentamos una serie de lemas para encontrar el conjunto de parejas mínimas de una gráfica.

Lema 4.3.3. *Si G es una gráfica vacía, entonces G es un elemento de M_2 y $\mu(G) = \{(0, 1)\}$.*

Demostración. Sea (A, B) una M_2 -partición de G . Si ni A ni B son vacíos, entonces (A, B) es de tamaño $(1, 1)$. Si uno de los dos es vacío, entonces (A, B) es de tamaño $(0, 1)$. Luego, $S(G) = \{(0, 1), (1, 1)\}$. Como $(1, 1)$ domina a $(0, 1)$ y $(0, 1)$ no domina a ningún elemento de $S(G)$, entonces $\mu(G) = \{(0, 1)\}$. \square

Lema 4.3.4. *Si $G \in M_2$ es una gráfica inconexa con al menos una componente trivial y exactamente una componente no trivial que es una gráfica multipartita completa formada por n conjuntos independientes, entonces $\mu(G) = \{(1, n - 1)\}$.*

Demostración. Notemos que, como G contiene vértices aislados, toda M_2 -partición que G acepta es una partición en un conjunto independiente y una gráfica multipartita completa. Sea (A, B) una M_2 -partición de G y $G[A]$ un conjunto independiente, abordemos los siguientes dos casos que son exhaustivos.

Caso 1: $n = 2$.

Notemos que la componente no trivial de G es bipartita y por lo tanto G también lo es. Si tanto $G[A]$ como $G[B]$ son conjuntos independientes, entonces (A, B) es de tamaño $(1, 1)$. En el caso contrario B contiene al menos un par de vértices que son adyacentes. Luego, A contiene a todos los vértices aislados de G y B contiene sólo vértices de la componente no trivial de G . Como ésta es bipartita, entonces (A, B) es de tamaño $(1, 2)$. Así, $S(G) = \{(1, 1), (1, 2)\}$. Luego, $\mu(G) = \{(1, 1)\} = \{(1, n - 1)\}$.

Caso 2: $n > 2$.

Como G no es bipartita, entonces B debe contener al menos un par de vértices adyacentes. Luego, A debe de contener a todos los vértices aislados de G , y B sólo puede contener vértices de la componente no trivial de G . Sea (B_1, B_2, \dots, B_n) una partición de la componente no trivial de G tal que $G[B_i]$ es un conjunto independiente para todo $0 \leq i \leq n$. Si A contiene todos los vértices de B_j para algún $0 \leq j \leq n$, entonces (A, B) es de tamaño $(1, n - 1)$. En el caso contrario, (A, B) es de tamaño $(1, n)$. Así, $S(G) = \{(1, n - 1), (1, n)\}$. Luego, $\mu(G) = \{(1, n - 1)\}$. □

Lema 4.3.5. *Sea $G \in M_2$ una gráfica inconexa con al menos una componente trivial y exactamente una componente no trivial que no es una gráfica multipartita completa. Si T es el cóarbol de G y r la raíz de T , entonces $\mu(G) = \{(1, B(r))\}$, en donde $B(x)$ se define de la siguiente manera para cualquier nodo x de T tal que x tiene etiqueta 0 y al menos un hijo que no es una hoja.*

1. *Si x es el nodo más profundo con etiqueta 0 que tiene al menos un hijo que no es una hoja.*
 - a) *Si x tiene al menos dos hijos que no son hojas, entonces $B(x) = 1$.*
 - b) *Si x tiene un solo hijo, y , que no es una hoja, tal que $G[y]$ es una gráfica multipartita completa formada por n conjuntos independientes, entonces $B(x) = n - 1$.*
2. *En el caso contrario. Sea y el único hijo de x que no es una hoja, n el número de hijos de y y x' el único hijo de y tal que $G[x']$ no es una gráfica multipartita completa, entonces $B(x) = n - 1 + B(x')$.*

Demostración. Podemos verificar que estos casos son exhaustivos a través del *Caso 2* de la demostración del Teorema 4.2.4.

Sea (A, B) una M_2 -partición de G , notemos que (A, B) es de tamaño $(1, \beta)$ para algún entero $\beta \geq 2$. Luego $(1, \beta) \in \mu(G)$ si y sólo si β es mínimo. Es decir que $\mu(G)$ tendrá un solo elemento, y éste será el tamaño de una M_2 -partición de G en un conjunto independiente y una gráfica multipartita completa con el menor número posible de partes.

Sea x un nodo de T . Mostremos que si x tiene etiqueta 0 y al menos un hijo que no es una hoja, entonces una M_2 -partición de $G[x]$ en un conjunto independiente y una gráfica multipartita completa con el menor número posible de partes será de tamaño $(1, B(x))$. Sea z el nodo más profundo de T con etiqueta 0 y al menos un hijo que no es una hoja, y d la distancia desde x hasta z , probemos por inducción sobre d .

Caso base: $d = 0$, es decir que $x = z$.

Por el *Caso 2* de la demostración del Teorema 4.2.4, sabemos que x tiene al menos dos hijos que no son hojas o x tiene un solo hijo y que no es una hoja, y tal que $G[y]$ es una gráfica multipartita completa formada por n conjuntos independientes.

Si x tiene al menos dos hijos que no son hojas, cada uno de estos hijos es la raíz del coárbol de una gráfica bipartita. Luego, $G[x]$ es bipartita. Así, $G[x]$ acepta una M_2 -partición (A, B) de tamaño $(1, 1)$. Como $G[x]$ no es un conjunto independiente, entonces (A, B) es una M_2 -partición de $G[x]$ en un conjunto independiente y una gráfica multipartita completa con el menor número posible de partes. Finalmente, notemos que (A, B) es de tamaño $(1, B(x))$ dado que se cumple la primera condición 1.a) del lema.

Si x tiene un solo hijo y que no es una hoja, y tal que $G[y]$ es una gráfica multipartita completa formada por n conjuntos independientes, entonces se sigue del Lema 4.3.4 que $\mu(G[x]) = (1, n - 1)$. Es claro que no existe una M_2 -partición de $G[x]$ en un conjunto independiente y una gráfica multipartita completa con menos de $n - 1$ partes. Así, dado que se cumple la condición 1.b) del lema, una M_2 -partición de $G[x]$ en un conjunto independiente y una gráfica multipartita completa con el menor número posible de partes será de tamaño $(1, n - 1) = (1, B(x))$.

Paso inductivo: $d > 0$.

Sea x' un nodo de T con etiqueta 0 y al menos un hijo que no es una hoja. Supongamos como hipótesis inductiva que, si la distancia desde x' hasta z es menor a d , entonces una M_2 -partición de $G[x']$ en un conjunto independiente y una gráfica multipartita completa con el menor número posible de partes es de tamaño $(1, B(x'))$.

Por el *Caso 2* de la demostración del Teorema 4.2.4, sabemos que x tiene

un sólo hijo y que no es una hoja, mientras que el resto son hojas. Asimismo, sabemos que y tiene un solo hijo x' tal que $G[x']$ no es una gráfica multipartita completa. Es decir que x' , que tiene etiqueta 0, tiene al menos un hijo que no es una hoja. Por hipótesis inductiva, una M_2 -partición de $G[x']$ en un conjunto independiente y una gráfica multipartita completa con el menor número posible de partes es de tamaño $(1, B(x'))$.

Sean (A', B') una partición de $G[x']$ de tamaño $(1, B(x'))$ tal que $G[A']$ es un conjunto independiente y (A, B) una partición de $G[x]$ tal que $G[A]$ es un conjunto independiente y $G[B]$ una gráfica multipartita completa. Como G no es bipartita, entonces A debe contener a todos los vértices de G que son hijos de x (es decir, todos los hijos de x que son hojas), mientras que B sólo puede contener vértices de $G[y]$. Para que $G[B]$ sea una gráfica multipartita completa, ésta debe de ser la unión completa de un conjunto de gráficas multipartitas completas. Como todos los hijos de y representan gráficas multipartitas completas menos x' , entonces A debe de contener un subconjunto S de los vértices de $G[x']$ tal que $G[x'] - S$ sea una gráfica multipartita completa.

Sea β' un entero tal que $G[x'] - S$ es una gráfica multipartita completa formada por β' conjuntos independientes y n el número de hijos de y , calculemos el tamaño (α, β) de (A, B) en función de β' y n . Sabemos que $\alpha = 1$ y que β es el número de conjuntos independientes que forman a la gráfica multipartita completa $G[x'] - S$. Como $G[x'] - S$ es la unión completa de n gráficas multipartitas completas, entonces β es la suma del número de conjuntos independientes que conforman a cada una de estas gráficas multipartitas completas. Todos los hijos de y que no son x' son hojas o representan gráficas multipartitas completas inconexas. Es decir, conjuntos independientes. En cualquier caso, cada uno de los $n - 1$ hijos de y representa un conjunto independiente. Como $G[x'] - S$ es la unión completa de β' conjuntos independientes, entonces $\beta = n - 1 + \beta'$.

Finalmente, encontremos el menor β' posible. Si $S = A'$, entonces $\beta' = B(x')$. Como $B(x')$ es mínimo por hipótesis inductiva, entonces el mínimo valor que puede tomar β' es $B(x')$. Luego, el valor mínimo de β es $n - 1 + B(x')$. Así, una M_2 -partición de $G[x]$ en un conjunto independiente y una gráfica multipartita completa con el menor número posible de partes es de tamaño $(1, n - 1 + B(x'))$. Como x cumple la condición 2 del lema, este tamaño es igual a $(1, B(x))$.

□

Lema 4.3.6. *Sea $G \in M_2$ una gráfica inconexa no bipartita con dos componentes conexas no triviales. Si las componentes de G son una la unión ajena de α conjuntos independientes y la otra la unión ajena de β conjuntos independientes con $\alpha \leq \beta$, entonces $\mu(G) = \{(\alpha, \beta)\}$.*

Demostración. Sea (A, B) una M_2 -partición de G . Como G no es bipartita, entonces A debe de contener todos los vértices de una de las componentes de G y B debe contener todos los vértices de la otra componente. Como ésta es la única posible M_2 -partición de G y es de tamaño (α, β) , entonces $\mu(G) = \{(\alpha, \beta)\}$. \square

Lema 4.3.7. *Si $G \in M_2$ es una gráfica inconexa bipartita con al menos dos componentes conexas no triviales, entonces $\mu(G) = \{(1, 1)\}$.*

Demostración. Como G es bipartita, entonces G acepta una M_2 -partición de tamaño $(1, 1)$. Como G tiene un \overline{P}_3 , entonces G no es una gráfica multipartita completa. Luego, G no acepta M_2 -particiones de tamaño $(0, \beta)$ para cualquier entero $\beta \geq 1$. Así, cualquier M_2 -partición de G será de tamaño (α, β') para algunos $\alpha \geq 1$ y $\beta \geq 1$. Luego, $\mu(G) = \{(1, 1)\}$. \square

Lema 4.3.8. *Si G es una gráfica isomorfa a K_1 , entonces $\mu(G) = \{0, 1\}$.*

Demostración. Es claro que la única M_2 -partición posible de G es de tamaño $(0, 1)$. Luego, $\mu(G) = \{(0, 1)\}$. \square

Lema 4.3.9. *Sean $G_1, G_2 \in M_2$, $G = G_1 \oplus G_2$ y $S'(G)$ el conjunto de todas las parejas (α, β) tales que para algunos $(\alpha_1, \beta_1) \in \mu(G_1)$ y $(\alpha_2, \beta_2) \in \mu(G_2)$ alguna de las siguientes condiciones se cumple*

- $\alpha = \alpha_1 + \alpha_2$ y $\beta = \beta_1 + \beta_2$ o
- $\alpha = \min(\alpha_1 + \beta_2, \alpha_2 + \beta_1)$ y $\beta = \max(\alpha_1 + \beta_2, \alpha_2 + \beta_1)$.

Una pareja de enteros P está en $\mu(G)$ si y sólo si $P \in S'(G)$ y P no domina a ningún otro elemento de $S'(G)$.

Demostración. Notemos que (A, B) es una M_2 -partición de G si y sólo si existen M_2 -particiones (A_1, B_1) y (A_2, B_2) de G_1 y G_2 respectivamente tales que $(A, B) = (A_1 \cup A_2, B_1 \cup B_2)$.

Veamos que $S'(G) \subset S(G)$. Sea (x, y) un elemento de $S'(G)$. Sabemos que existen $(\alpha_1, \beta_1) \in \mu(G_1)$ y $(\alpha_2, \beta_2) \in \mu(G_2)$ tales que se cumple alguna de las

condiciones mencionadas en el lema. Como $(\alpha_1, \beta_1) \in \mu(G_1)$, entonces existe una M_2 -partición (A_1, B_1) de G_1 de tamaño (α_1, β_1) . Análogamente, existe una M_2 -partición (A_2, B_2) de G_2 de tamaño (α_2, β_2) . Luego, $(A_1 \cup A_2, B_1 \cup B_2)$ es una M_2 -partición de G de tamaño (x, y) o $(A_1 \cup B_2, B_1 \cup A_2)$ es una M_2 partición de G de tamaño (x, y) . En cualquiera de los casos, $(x, y) \in S(G)$. Así, $S'(G) \subset S(G)$.

Para la contención restante, sea $(x, y) \in \mu(G)$, entonces existe una M_2 -partición (A, B) de G de tamaño (x, y) . Luego, existen M_2 -particiones (A_1, B_1) de tamaño (α_1, β_1) y (A_2, B_2) de tamaño (α_2, β_2) de G_1 y G_2 respectivamente tales que $(A, B) = (A_1 \cup A_2, B_1 \cup B_2)$. Como todos los vértices de A_1 son adyacentes a todos los vértices de A_2 y todos los vértices de B_1 son adyacentes a todos los vértices de B_2 , se cumple que $\alpha = \alpha_1 + \alpha_2$ y $\beta = \beta_1 + \beta_2$, o bien, se cumple que $\alpha = \min(\alpha_1 + \beta_2, \alpha_2 + \beta_1)$ y $\beta = \max(\alpha_1 + \beta_2, \alpha_2 + \beta_1)$.

Veamos que $(\alpha_1, \beta_1) \in \mu(G_1)$. Procedamos por contradicción. Supongamos que $(\alpha_1, \beta_1) \notin \mu(G_1)$. Como (A_1, B_1) existe, entonces $(\alpha_1, \beta_1) \in S(G_1)$. Luego debe de existir otra pareja $(\alpha'_1, \beta'_1) \in S(G_1)$ tal que (α_1, β_1) domina a (α'_1, β'_1) . De esto se sigue que existe una M_2 -partición (A'_1, B'_1) de G_1 de tamaño (α'_1, β'_1) . Como $(A'_1 \cup A_2, B'_1 \cup B_2)$ es una M_2 -partición de G cuyo tamaño es dominado por (x, y) , entonces $(x, y) \notin \mu(G)$. Lo cual es una contradicción. Así, $(\alpha_1, \beta_1) \in \mu(G_1)$. Análogamente, $(\alpha_2, \beta_2) \in \mu(G_2)$. Así, $(x, y) \in S'(G)$, y por lo tanto $\mu(G) \subseteq S'(G)$. Luego, como $S'(G) \subset S(G)$ y $(x, y) \in \mu(G)$, entonces (x, y) no domina a elemento alguno de $S(G)$, y por lo tanto tampoco domina elemento alguno de $S'(G)$.

Recíprocamente, sea $(x, y) \notin \mu(G)$, mostremos que (x, y) no está en $S'(G)$ o (x, y) domina a algún elemento de $S'(G)$. Sabemos por la definición de $\mu(G)$ que (x, y) no está en $S(G)$ o (x, y) domina a algún elemento de $S(G)$. Si (x, y) no está en $S(G)$, dado que $S'(G)$ es subconjunto de $S(G)$, entonces (x, y) no está en $S'(G)$. En el caso contrario, si $(x, y) \in S(G)$, como $(x, y) \notin \mu(G)$, entonces (x, y) domina a algún elemento de $S(G)$ diferente de sí mismo. Luego, (x, y) domina a algún elemento de $\mu(G)$. Recordemos que $\mu(G) \subseteq S'(G)$. Así, (x, y) domina a algún elemento de $S'(G)$. □

4.3.4. Algoritmo para generar obstrucciones mínimas

En esta subsección presentamos cuatro algoritmos. Los primeros dos (Algoritmos 13 y 14) se utilizan para calcular el conjunto de parejas mínimas de una gráfica. El Algoritmo 15 resuelve el problema de decisión de si una

gráfica pertenece a una clase (α, β) - M_2 . El Algoritmo 16 es capaz de generar las obstrucciones mínimas de una clase (α, β) - M_2 con a lo más un número n de vértices, por lo que es utilizado en la siguiente sección para generar obstrucciones mínimas para algunas clases (α, β) - M_2 .

Algoritmos para el cálculo del conjunto de parejas mínimas

El Algoritmo 13 evalúa la función $B(x)$ descrita en el Lema 4.3.5 para un nodo x de un coárbol que representa a una gráfica G . Es decir que éste encuentra el menor entero n tal que $G[x]$ acepta una partición en un conjunto independiente y una gráfica multipartita completa formada por n conjuntos independientes. Dado que este algoritmo es una implementación directa del Lema 4.3.5, la demostración de este lema funciona también para el algoritmo.

Algoritmo 13: Función_B

Entrada: x , el nodo de un coárbol con etiqueta 0 y al menos un hijo que no es una hoja. x es la raíz del coárbol de una cográfica en M_2 .

Salida: El valor de $B(x)$, de acuerdo con la definición del Lema 4.3.5.

```

1 si  $X$  tiene más de un hijo que no es una hoja entonces
2   └─ devolver 1
3 en otro caso
4   └─  $y \leftarrow$  el hijo de  $x$  que no es una hoja.
5     └─ si  $y$  es la raíz del coárbol de una cográfica multipartita completa
6         └─ devolver  $y.hijos.tamaño - 1$ 
7     └─ en otro caso
8         └─  $x' \leftarrow$  el único hijo de  $y$  que tiene al menos un hijo que no es
9             └─ una hoja
                └─ devolver  $y.hijos.tamaño - 1 + \text{Función\_B}(x')$ 

```

El algoritmo 14 recibe como entrada una gráfica G , representada a través de la raíz de su coárbol y devuelve su conjunto de parejas mínimas $\mu(G)$. Éste funciona aplicando los Lemas 4.3.3, 4.3.5, 4.3.6, 4.3.7, 4.3.8 y 4.3.9.

Notemos que los casos considerados son exhaustivos. Las condiciones de las líneas 2 y 4 cubren todos los casos en los que la gráfica G es conexa,

Algoritmo 14: Conjunto_De_Parejas_Mínimas**Entrada:** g , la raíz del cóarbol de una cográfica G .**Salida:** $\mu(G)$

```

1  $\mu(G) \leftarrow \emptyset$ 
2 si  $g$  es una hoja entonces
3    $\lfloor$  Agregar  $(0, 1)$  a  $\mu(G)$ 
4 si no, si  $g$  tiene etiqueta 1 entonces
5    $\mu(G) \leftarrow$  Conjunto_De_Parejas_Mínimas( $g.hijos[0]$ )
6   para cada  $h$  en  $g.hijos$  con excepción de  $g.hijos[0]$  hacer
7      $T \leftarrow$  Conjunto_De_Parejas_Mínimas( $h$ )
8      $S' \leftarrow \emptyset$ 
9     para cada  $(\alpha_1, \beta_1) \in \mu(G)$  y cada  $(\alpha_2, \beta_2) \in T$  hacer
10       $\lfloor$  Agregar  $(\alpha_1 + \alpha_2, \beta_1 + \beta_2)$  a  $S'$ 
11       $\lfloor$  Agregar  $(\min(\alpha_1 + \beta_2, \alpha_2 + \beta_1), \max(\alpha_1 + \beta_2, \alpha_2 + \beta_1))$  a  $S'$ 
12       $\mu(G) \leftarrow \emptyset$ 
13      Agregar a  $\mu(G)$  todos los elementos de  $S'$  que no dominan a
14      ningún otro elemento de  $S'$ 
15 en otro caso
16    $n \leftarrow$  el número de hijos de  $g$  que no son hojas
17   si  $n = 0$  entonces
18      $\lfloor$  Agregar  $(0, 1)$  a  $\mu(G)$ 
19   si no, si  $n = 1$  entonces
20      $\lfloor$  Agregar  $(0, \text{Función\_B}(g))$  a  $\mu(G)$ 
21   si no, si  $n = 2$  y  $G$  no es bipartita entonces
22      $\lfloor$   $\alpha \leftarrow g.hijos[0].hijos.tamaño$ 
23      $\lfloor$   $\beta \leftarrow g.hijos[1].hijos.tamaño$ 
24      $\lfloor$  Agregar  $(\min(\alpha, \beta), \max(\alpha, \beta))$  a  $\mu(G)$ 
25   en otro caso
26      $\lfloor$  Agregar  $(1, 1)$  a  $\mu(G)$ 
27 devolver  $\mu(G)$ 

```

mientras que las condiciones de las líneas 16, 18, 20 y 24 cubren todos los casos en los que la gráfica G es inconexa.

En la línea 3 del algoritmo se aplica el Lema 4.3.8. En la línea 17 se aplica

el Lema 4.3.3. En la línea 19 se aplica el Lema 4.3.5. En la línea 23 se aplica el Lema 4.3.6 y en la línea 25 se aplica el Lema 4.3.7.

El único caso en el que no se aplica directamente un lema de la subsección anterior es en el caso en el que G es una gráfica conexa no trivial. Éste corresponde al bloque de líneas 5 a 13 del algoritmo. En este bloque se calcula el conjunto de parejas mínimas de G de forma incremental aplicando el Lema 4.3.9. Sea $G = G_1 \oplus G_2 \oplus \dots \oplus G_n$ para un entero $n > 1$, primero se calcula $\mu(G_1)$, posteriormente $\mu(G_1 \oplus G_2)$ y así sucesivamente hasta encontrar $\mu(G)$. En la línea 5 se inicializa $\mu(G)$ con los elementos de G_1 . En el bloque de las líneas 6 a 13 se procesan el resto de las G_i con $1 < i \leq n$. En cada iteración i se calcula $\mu(G_i)$ y se guarda en T (línea 7). Luego, en las líneas 10 y 11 se aplica el Lema 4.3.9 para calcular $S'(G_1 \oplus \dots \oplus G_i)$. Finalmente, en la línea 13 se actualiza $\mu(G)$ con los elementos de $S'(G_1 \oplus \dots \oplus G_i)$ que no dominan a ningún elemento del mismo conjunto. Así, al final de cada iteración, $\mu(G)$ tiene asignado el valor de $\mu(G_1 \oplus \dots \oplus G_i)$, y al final de la última iteración, $\mu(G) = \mu(G_1 \oplus \dots \oplus G_n)$.

Algoritmo para el problema de decisión

El Algoritmo 15 es una implementación del Lema 4.3.2. Éste recibe como entrada la raíz del córbol de una gráfica G , elemento de M_2 , y dos enteros α y β tales que $\alpha \leq \beta$. Luego, calcula $\mu(G)$ y determina si $G \in (\alpha, \beta)$ - M_2 .

Algoritmo 15: Pertenece_a_Alfa_Beta_M2

Entrada: g , la raíz del córbol de una gráfica $G \in M_2$; α , un entero mayor o igual a uno; β , un entero mayor o igual a α

Salida: *verdadero* si $G \in (\alpha, \beta)$ - M_2 . *falso* en el caso contrario

- 1 $\mu(G) \leftarrow$ Conjunto_De_Parejas_Mínimas(g)
- 2 **para cada** (α', β') **en** $\mu(G)$ **hacer**
- 3 **si** (α', β') *domina a* (α, β) **entonces**
- 4 | **devolver** *verdadero*
- 5 **devolver** *falso*

Algoritmo para generar las obstrucciones mínimas de una clase (α, β) - M_2

El Algoritmo 16 recibe como entrada tres enteros α , β y n mayores a uno tales que $\alpha \leq \beta$, y devuelve S , un conjunto con los coárboles de las obstrucciones mínimas de la clase (α, β) - M_2 con a lo más n vértices. Para ello, el algoritmo genera todos los coárboles con i hojas para cada $1 \leq i \leq n$ [10]. Éste determina si cada uno de estos coárboles representa una gráfica en (α, β) - M_2 . Si la gráfica no pertenece a la clase, esto significa que es una obstrucción de la misma, por lo que evalúa cada una de sus subgráficas inducidas para determinar si es una obstrucción mínima.

Algoritmo 16: Generar_obstrucciones

Entrada: α , β y n , enteros mayores a uno

Salida: S , un conjunto cuyos elementos son coárboles de gráficas de a lo más n vértices que son obstrucciones mínimas de la clase (α, β) - M_2

```

1 para cada  $1 \leq i \leq n$  hacer
2    $\tau \leftarrow$  todos los coárboles con  $i$  hojas
3   para cada coárbol  $T$  en  $\tau$  con raíz  $r$  que representa a una gráfica
      $G$  hacer
4     si Pertenece_a_Alfa_Beta_M2( $r, \alpha, \beta$ ) = falso entonces
5       si Toda subgráfica inducida de  $G$  está en  $(\alpha, \beta)$ - $M_2$ 
6         entonces
           Agregar  $T$  a  $S$ 
7 devolver  $S$ 

```

4.3.5. Resultados del algoritmo

En esta subsección mostramos las obstrucciones mínimas de algunas clases (α, β) - M_2 encontradas con el Algoritmo 16. Las obstrucciones se buscaron en coárboles de hasta 15 hojas.

Clases $(1, \beta)-M_2$

Las Figuras 4.13,4.14,4.15,4.16 y 4.17 muestran los conjuntos de obstrucciones mínimas encontradas con el Algoritmo 16 para las clases $(1, 1)-M_2$, $(1, 2)-M_2$, $(1, 3)-M_2$, $(1, 4)-M_2$ y $(1, 5)-M_2$ respectivamente.

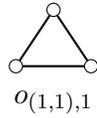


Figura 4.13: Obstrucción mínima de la clase $(1, 1)-M_2$.

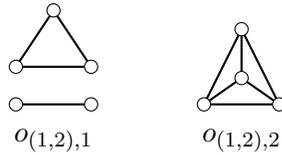


Figura 4.14: Algunas obstrucciones mínimas de la clase $(1, 2)-M_2$.

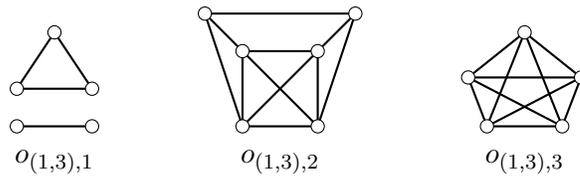


Figura 4.15: Algunas obstrucciones mínimas de la clase $(1, 3)-M_2$.

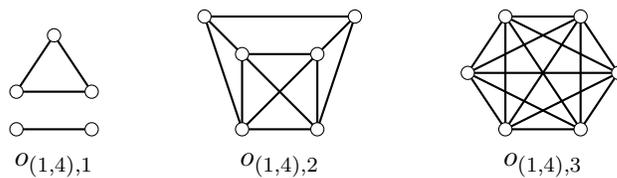


Figura 4.16: Algunas obstrucciones mínimas de la clase $(1, 4)-M_2$.

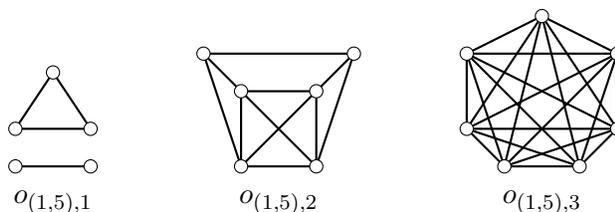


Figura 4.17: Algunas obstrucciones mínimas de la clase $(1, 5)-M_2$.

Podemos observar que el algoritmo encuentra que K_3 es una obstrucción mínima de la clase $(1, 1)-M_2$, lo cual sabemos que es correcto, dado que ésta es la clase de las cográficas bipartitas.

Sea β un entero mayor o igual a uno. Si $2 < \beta \leq 5$, notemos que el algoritmo encuentra tres obstrucciones mínimas para la clase $(1, \beta)-M_2$. Dos de éstas son las obstrucciones mínimas de la clase $(1, \infty)-M_2$. Por otra parte, si $0 \leq \beta \leq 5$, se cumple que $K_{\beta+2}$ es una obstrucción mínima de la clase $(1, \beta)-M_2$. Esto nos lleva a pensar en la clase $(1, \infty)-M_2$ como el límite cuando β tiende a infinito de las clases $(1, \beta)-M_2$. Es decir que esperamos que, si una gráfica G es una obstrucción mínima de cada una de las clases $(\alpha, \beta)-M_2$ con α fijo y $\beta > n$ para un entero n , entonces G sea una obstrucción mínima de la clase $(\alpha, \infty)-M_2$.

Clases $(2, \beta)-M_2$

Las Figuras 4.18, 4.19 y 4.20 muestran los conjuntos de obstrucciones mínimas encontradas con el Algoritmo 16 para las clases $(2, 2)-M_2$, $(2, 3)-M_2$ y $(2, 4)-M_2$ respectivamente. Además de éstas, enlistamos las obstrucciones mínimas de las clases $(2, 5)-M_2$, $(2, 6)-M_2$ y $(2, 7)-M_2$.

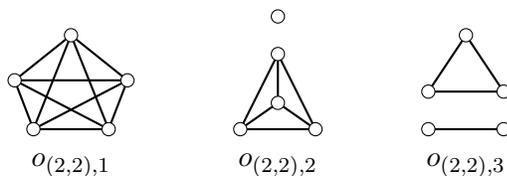


Figura 4.18: Algunas obstrucciones mínimas de la clase $(2, 2)-M_2$.

Las obstrucciones mínimas de la clase $(2, 2)-M_2$, ilustradas en la Figura 4.18 se pueden expresar de la siguiente manera:

- $o_{(2,2),1} = K_5$.
- $o_{(2,2),2} = K_1 + K_4$.
- $o_{(2,2),3} = K_2 + K_3$.

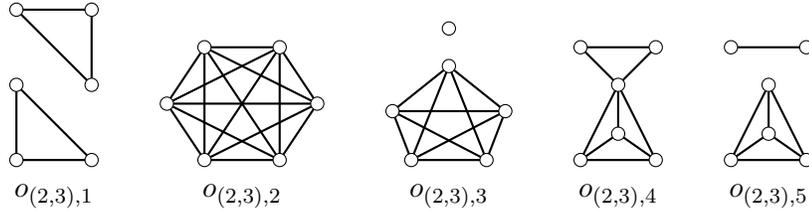


Figura 4.19: Algunas obstrucciones mínimas de la clase $(2, 3)$ - M_2 .

Las obstrucciones mínimas de la clase $(2, 3)$ - M_2 , ilustradas en la Figura 4.19, se pueden expresar de la siguiente manera:

- $o_{(2,3),1} = 2K_3$.
- $o_{(2,3),2} = K_6$.
- $o_{(2,3),3} = K_1 + K_5$.
- $o_{(2,3),4} = K_1 \oplus (K_2 + K_3)$.
- $o_{(2,3),5} = K_2 + K_4$.

Las obstrucciones mínimas de la clase $(2, 4)$ - M_2 , ilustradas en la Figura 4.20, se pueden expresar de la siguiente manera:

- $o_{(2,4),1} = 2K_3$.
- $o_{(2,4),2} = \overline{P_3} \oplus \overline{P_3} \oplus \overline{P_3}$.
- $o_{(2,4),3} = K_7$.
- $o_{(2,4),4} = K_2 \oplus (K_2 + K_3)$.
- $o_{(2,4),5} = K_1 + K_6$.
- $o_{(2,4),6} = K_1 \oplus (K_2 + K_4)$.

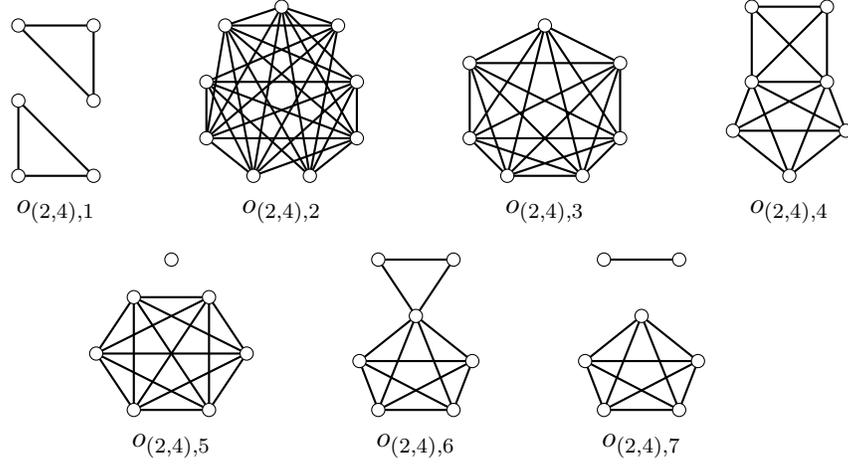


Figura 4.20: Algunas obstrucciones mínimas de la clase $(2, 4)-M_2$.

- $o_{(2,4),7} = K_2 + K_5$.

Las obstrucciones mínimas de la clase $(2, 5)-M_2$ encontradas con el Algoritmo 16 se pueden expresar de la siguiente manera:

- $o_{(2,5),1} = 2K_3$.

- $o_{(2,5),2} = \overline{P_3} \oplus \overline{P_3} \oplus \overline{P_3}$.

- $o_{(2,5),3} = \overline{P_3} \oplus (K_2 + K_3) = (K_1 + K_2) \oplus (K_2 + K_3)$.

- $o_{(2,5),4} = K_8$.

- $o_{(2,5),5} = K_3 \oplus (K_2 + K_3)$.

- $o_{(2,5),6} = K_2 \oplus (K_2 + K_4)$.

- $o_{(2,5),7} = K_1 + K_7$.

- $o_{(2,5),8} = K_1 \oplus (K_2 + K_5)$.

- $o_{(2,5),9} = K_2 + K_6$.

Las obstrucciones mínimas de la clase $(2, 6)-M_2$ encontradas con el Algoritmo 16 se pueden expresar de la siguiente manera:

- $o_{(2,6),1} = 2K_3.$
- $o_{(2,6),2} = \overline{P_3} \oplus \overline{P_3} \oplus \overline{P_3}.$
- $o_{(2,6),3} = \overline{P_3} \oplus (K_2 + K_3) = (K_1 + K_2) \oplus (K_2 + K_3).$
- $o_{(2,6),4} = K_9.$
- $o_{(2,6),5} = K_4 \oplus (K_2 + K_3).$
- $o_{(2,6),6} = K_3 \oplus (K_2 + K_4).$
- $o_{(2,6),7} = K_2 \oplus (K_2 + K_5).$
- $o_{(2,6),8} = K_1 + K_8.$
- $o_{(2,6),9} = K_1 \oplus (K_2 + K_6).$
- $o_{(2,6),10} = K_2 + K_7.$

Las obstrucciones mínimas de la clase $(2, 7)$ - M_2 encontradas con el Algoritmo 16 se pueden expresar de la siguiente manera:

- $o_{(2,7),1} = 2K_3.$
- $o_{(2,7),2} = \overline{P_3} \oplus \overline{P_3} \oplus \overline{P_3}.$
- $o_{(2,7),3} = \overline{P_3} \oplus (K_2 + K_3) = (K_1 + K_2) \oplus (K_2 + K_3).$
- $o_{(2,7),4} = K_{10}.$
- $o_{(2,7),5} = K_5 \oplus (K_2 + K_3).$
- $o_{(2,7),6} = K_4 \oplus (K_2 + K_4).$
- $o_{(2,7),7} = K_3 \oplus (K_2 + K_5).$
- $o_{(2,7),8} = K_2 \oplus (K_2 + K_6).$
- $o_{(2,7),9} = K_1 + K_9.$
- $o_{(2,7),10} = K_1 \oplus (K_2 + K_7).$
- $o_{(2,7),11} = K_2 + K_8.$

Como podemos ver, la gráfica $2K_3$ es una obstrucción mínima de las clases $(2, \beta)$ - M_2 para $3 \leq \beta \leq 7$, la gráfica $\overline{P_3} \oplus \overline{P_3} \oplus \overline{P_3}$ es una obstrucción mínima de las clases $(2, \beta)$ - M_2 para $4 \leq \beta \leq 7$ y la gráfica $\overline{P_3} \oplus (K_2 + K_3)$ es una obstrucción mínima de las clases $(2, \beta)$ - M_2 para $5 \leq \beta \leq 7$. Esto nos lleva a pensar que estas tres gráficas pueden ser obstrucciones mínimas de la clase $(2, \infty)$ - M_2 . Además, podemos identificar algunas posibles familias de obstrucciones para cualquier clase $(2, \beta)$ - M_2 .

Obstrucciones de algunas otras clases (α, β) - M_2

4.4. Particiones en más de dos partes

En esta sección tomamos como base los resultados obtenidos en nuestro estudio de la clase M_2 para estudiar a las clases M_i , para enteros i mayores a 2. Empezamos por caracterizar a la clase M_3 a través de su conjunto de obstrucciones mínimas. El conjunto de obstrucciones mínimas de la clase M_2 y el de la clase M_3 nos ayudan a distinguir dos familias de obstrucciones mínimas para cualquier clase M_i . Posteriormente presentamos dos familias de obstrucciones mínimas para cualquier clase M_i .

4.4.1. Obstrucciones mínimas de la clase M_3

En esta subsección caracterizamos a la clase M_3 a través de su conjunto de obstrucciones mínimas considerando 10 casos diferentes y exhaustivos. Las obstrucciones mínimas de la clase M_3 se presentan agrupadas en 5 familias, cada una de las cuales contiene gráficas con elementos en común, que son utilizados en casos diferentes de la demostración del Teorema 4.4.1. Las familias O_3 y P_3 son descritas en las siguientes subsecciones, ya que se pueden generalizar para cualquier clase M_i . Las gráficas de las familias O_3 y Q_3 se muestran en las Figuras 4.21 y 4.22 respectivamente.

Teorema 4.4.1. *Sean G una cográfica, H , I y J las gráficas de la Figura 4.4 y H' y J' las gráficas de la Figura 4.12, G es un elemento de la clase M_3 si y sólo si es libre de cada una de las siguientes gráficas:*

- $O_{3,1} = K_1 + K_2 + K_3 + K_4$.
- $O_{3,2} = K_2 + (K_1 \oplus (K_1 + K_2)) + K_4$.

- $O_{3,3} = K_2 + K_3 + (K_1 \oplus (K_1 + K_3))$.
- $O_{3,4} = K_2 + K_3 + (K_2 \oplus (K_1 + K_2))$.
- $O_{3,5} = K_1 + K_3 + (K_1 \oplus (K_2 + K_3))$.
- $O_{3,6} = K_3 + (K_1 \oplus (K_1 + K_2 + K_3))$.
- $O_{3,7} = K_3 + (K_1 \oplus (K_2 + (K_1 \oplus (K_1 + K_2))))$.
- $O_{3,8} = (K_1 \oplus (K_1 + K_2)) + (K_1 \oplus (K_2 + K_3))$.
- $P_{3,1} = K_1 + (H \oplus H)$.
- $P_{3,2} = K_1 + (H \oplus I)$.
- $P_{3,3} = K_1 + (H \oplus J)$.
- $P_{3,4} = K_1 + (I \oplus I)$.
- $P_{3,5} = K_1 + (I \oplus J)$.
- $P_{3,6} = K_1 + (J \oplus J)$.
- $Q_{3,1} = K_1 + K_3 + (\overline{P_3} \oplus \overline{P_3})$.
- $Q_{3,2} = (K_1 \oplus (K_1 + K_2)) + (\overline{P_3} \oplus \overline{P_3})$.
- $R_{3,1} = K_1 + K_2 + (\overline{P_3} \oplus H') = K_1 + K_2 + (\overline{P_3} \oplus \overline{P_3} \oplus \overline{P_3})$.
- $R_{3,2} = K_1 + K_2 + (\overline{P_3} \oplus J')$.
- $S_{3,3} = K_2 + (\overline{P_3} \oplus H)$.
- $S_{3,4} = K_2 + (\overline{P_3} \oplus I)$.
- $S_{3,5} = K_2 + (\overline{P_3} \oplus J)$.

Demostración. Supongamos primero que G tiene a alguna de las gráficas listadas como subgráfica inducida y veamos que G no es un elemento de M_3 . Para ello basta con mostrar que ninguna de las gráficas listadas está en M_3 .

Sea i un entero tal que $1 \leq i \leq 8$. Veamos que $O_{3,i}$ no está en la clase M_3 . La Figura 4.21 muestra la representación gráfica de $O_{3,i}$. Notemos que algunos de sus vértices se encuentran coloreados indicando que $O_{3,i}$ tiene

a H (color amarillo) o a I (color anaranjado) como subgráficas inducidas. Notemos también que los vértices no coloreados inducen una gráfica que contiene una copia de K_3 , cada uno de cuyos vértices no es adyacente a un vértice coloreado alguno. Sea (A, B, C) una partición de los vértices de $O_{3,i}$, veamos que (A, B, C) no es una M_3 -partición. Procedamos por contradicción. Supongamos que (A, B, C) es una M_3 -partición de $O_{3,i}$. Como $O_{3,i}$ tiene a K_4 como subgráfica inducida, entonces no es 3-coloreable. Luego, los vértices del K_3 no coloreado de $O_{3,i}$ no pueden estar cada uno en una parte diferente de (A, B, C) . Es decir que al menos dos de ellos deben de estar en la misma parte. Supongamos sin pérdida de generalidad que A contiene dos vértices del K_3 no coloreado. Luego, ninguno de los vértices coloreados de $O_{3,i}$ puede estar en A , o A contendría los vértices de un \overline{P}_3 y por lo tanto no induciría una gráfica multipartita completa. Como los vértices coloreados inducen una obstrucción mínima de la clase M_2 , éstos no se pueden repartir en las dos partes restantes de manera que cada una induzca una gráfica multipartita completa. Así, (A, B, C) no es una M_3 -partición de $O_{3,i}$, y por lo tanto $O_{3,i} \notin M_3$.

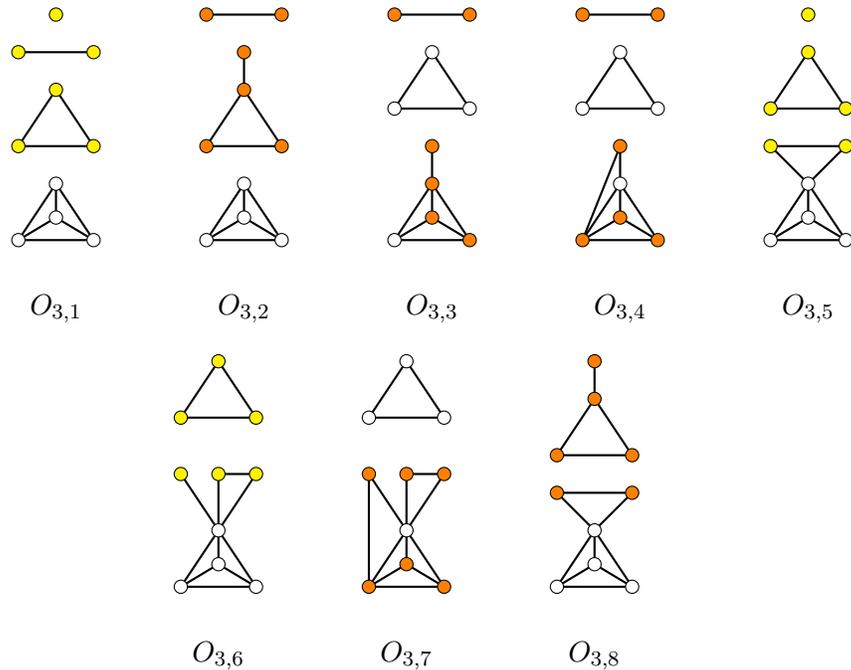


Figura 4.21: Familia O_3 de obstrucciones mínimas de la clase M_3 .

Sea i un entero tal que $1 \leq i \leq 6$. Veamos que $P_{3,i}$ no está en la clase M_3 . Sea (A, B, C) una partición de los vértices de $P_{3,i}$, veamos que (A, B, C) no es una M_3 -partición de $P_{3,i}$. Procedamos por contradicción. Supongamos que (A, B, C) es una M_3 -partición de $P_{3,i}$. Supongamos sin pérdida de generalidad que A contiene al vértice aislado de $O_{3,i}$. Luego, A debe de ser un conjunto independiente. Notemos que los vértices restantes de $P_{3,i}$ inducen la unión completa de dos obstrucciones mínimas X y Y de la clase M_2 . Como $X \cup Y$ induce una gráfica que no está en M_2 , alguno de sus vértices debe de estar en A . Supongamos sin pérdida de generalidad que uno de los vértices de X está en A . Luego, ninguno de los vértices de Y puede estar en A también. Como Y es una obstrucción mínima de M_2 , sus vértices no se pueden repartir entre B y C de manera que ambas induzcan gráficas multipartitas completas. Luego, $P_{3,i} \notin M_3$. Sea i un entero tal que $1 \leq i \leq 2$. Veamos que $Q_{3,i}$ no está en la clase M_3 . La Figura 4.22 muestra la representación gráfica de $Q_{3,i}$. Notemos que algunos de sus vértices se encuentran coloreados con rojo indicando que $Q_{3,i}$ tiene a J como subgráfica inducida. Notemos también que los vértices no coloreados inducen ya sea a la gráfica $K_3 + K_1$ o a la gráfica *Paw*. El argumento para probar que $Q_{3,i}$ no está en M_3 es análogo al utilizado para la familia O_3 .

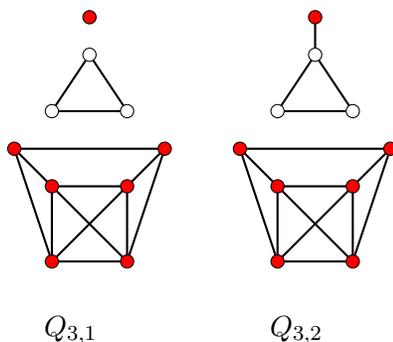


Figura 4.22: Familia Q_3 de obstrucciones mínimas de la clase M_3 .

Sea i un entero tal que $1 \leq i \leq 2$. Veamos que $R_{3,i}$ no está en la clase M_3 . Sea (A, B, C) una partición de los vértices de $R_{3,i}$, veamos que (A, B, C) no es una M_3 -partición de $R_{3,i}$. Procedamos por contradicción. Supongamos que (A, B, C) es una M_3 -partición de $R_{3,i}$. Si los dos vértices del K_2 están en la misma parte, el resto de los vértices deben de poder repartirse en las partes restantes. Como estos vértices inducen una gráfica que contiene una de las

obstrucciones mínimas de la clase M_2 , esto no es posible. Así, ambos vértices del K_2 deben de estar en partes diferentes. Es decir que $R_{3,i}$ debe de aceptar una partición en dos conjuntos independientes y una gráfica multipartita completa. Supongamos sin pérdida de generalidad que A contiene un vértice del K_2 y B el otro. Notemos que $R_{3,i}$ no es 3-coloreable, por lo que el vértice del K_1 debe estar ya sea en A o en B . Es claro que uno de los vértices del $\overline{P_3}$ de $R_{3,i}$ debe de estar ya sea en A o en B para que C pueda inducir una gráfica multipartita completa. Supongamos sin pérdida de generalidad que A contiene un vértice del $\overline{P_3}$ de $R_{3,i}$. Luego, el resto de los vértices deben de poder repartirse en B y C . Como B debe de ser un conjunto independiente y el resto de los vértices contiene una de las obstrucciones mínimas de la clase $(1, \infty)$ - M_2 , esto no es posible. Así, $R_{3,i} \notin M_3$.

Sea i un entero tal que $1 \leq i \leq 3$. Es posible usar un argumento análogo al anterior para ver que $S_{3,i}$ no está en la clase M_3 .

Recíprocamente, supongamos que G es libre de cada una de las gráficas listadas y probemos que $G \in M_3$. Para ello consideramos diez casos que son exhaustivos.

Caso 1: G es la unión ajena de gráficas bipartitas completas.

Es claro que G es bipartita. Luego, $G \in M_3$.

Caso 2: G tiene al menos 4 componentes y al menos 3 no son bipartitas.

Como G es libre de $O_{3,1} = K_1 + K_2 + K_3 + K_4$, todas las componentes de G deben de ser 3-coloreables. Luego, G es 3-coloreable y $G \in M_3$.

Caso 3: G tiene exactamente 3 componentes y ninguna es bipartita.

Si todas las componentes de G son 3-coloreables, G también lo es y por lo tanto $G \in M_3$. En el caso contrario, al menos una de sus componentes tiene un K_4 . Como G es libre de $O_{3,1} = K_1 + K_2 + K_3 + K_4$, todas las demás componentes deben de ser libres de $K_1 + K_2 = \overline{P_3}$. Como G es libre de $O_{3,3}$ y $O_{3,4}$, la componente con el K_4 también debe de ser multipartita completa. Así, G es la unión ajena de 3 gráficas multipartitas completas, y por lo tanto $G \in M_3$.

Caso 4: G tiene exactamente 2 componentes no bipartitas y al menos una bipartita no trivial.

Si todas las componentes de G son 3-coloreables, G también lo es, y por lo tanto $G \in M_3$. En el caso contrario, una de las componentes no bipartitas de G contiene un K_4 . Como G tiene una componente con un K_4 , una con un K_3 y una con un K_2 , y es libre de $O_{3,1} = K_1 + K_2 + K_3 + K_4$, entonces G no tiene más que estas tres componentes. Como la componente bipartita de G es conexa, ésta es multipartita completa. Como G es libre de $O_{3,2}$, entonces

la componente con el K_3 debe de ser multipartita completa. Como G es libre de $O_{3,3}$ y $O_{3,4}$, entonces la componente con el K_4 es multipartita completa. Así, G es la unión ajena de 3 gráficas multipartitas completas, y por lo tanto $G \in M_3$.

Caso 5: G tiene al menos 3 componentes. Exactamente 2 de ellas no son bipartitas y el resto son triviales.

Como G es libre de $O_{3,5}$, cada una de las componentes no bipartitas es libre de $K_2 + K_3$. Como G es libre de $Q_{3,1}$, estas mismas componentes son libres de $(\overline{P}_3 \oplus \overline{P}_3)$. Así, estas componentes están en la clase $(1, \infty)$ - M_2 . Como las demás componentes de G forman un conjunto independiente, G acepta una partición en dos gráficas multipartitas completas y un conjunto independiente. Luego, $G \in M_3$.

Caso 6: G tiene sólo 2 componentes y ninguna es bipartita.

Dado que G es libre de $O_{3,6}$, ninguna de las componentes de G puede contener un $K_1 \oplus (K_1 + K_2 + K_3)$. Luego, como son conexas, éstas deben de ser libres de $K_1 + K_2 + K_3$. Dado que G es libre de $O_{3,7}$, ninguna de las componentes de G puede contener un $K_1 \oplus (K_2 + Paw)$. Luego, como son conexas, éstas deben de ser libres de $K_2 + Paw$. Dado que G es libre de $Q_{3,1}$, ninguna de las componentes de G puede contener un $K_1 + (\overline{P}_3 \oplus \overline{P}_3)$. Así, ambas componentes de G son elementos de M_2 . Si las dos componentes de G están en la clase $(1, \infty)$ - M_2 , entonces G acepta una partición en dos gráficas multipartitas completas y un conjunto independiente. Luego, $G \in M_3$. En el caso contrario, una de las componentes debe de tener ya sea a H' o a J' .

Si una de las componentes de G contiene a J' , dado que G es libre de $Q_{3,2}$, la otra componente debe de ser libre de Paw . Es decir que debe ser una gráfica multipartita completa o una gráfica bipartita. En cualquiera de los casos es una gráfica multipartita completa. Luego, como una componente de G es una gráfica multipartita completa y la otra es un elemento de M_2 , se sigue que $G \in M_3$.

Si una de las componentes de G contiene a $H' = K_2 + K_3$, dado que es conexa, debe contener a $K_1 \oplus (K_2 + K_3)$. Luego, dado que G es libre de $O_{3,8}$, la otra componente debe de ser libre de Paw . Es decir que debe ser una gráfica multipartita completa o una gráfica bipartita. En cualquiera de los casos es una gráfica multipartita completa. Luego, como una componente de G es una gráfica multipartita completa y la otra es un elemento de M_2 , se sigue que $G \in M_3$.

Caso 7: G tiene al menos 3 componentes. Exactamente una no es bipartita y al menos una de las demás componentes no es trivial.

Si la componente de G que no es bipartita es multipartita completa, entonces G acepta una partición en dos conjuntos independientes y una gráfica multipartita. Luego, $G \in M_3$. En el caso contrario. Sea G' la componente de G que no es bipartita, w la raíz de su coárbol, x un nodo en el coárbol de G' con etiqueta 1 tal que contiene un Paw , z el descendiente de w más profundo que contiene un Paw y D la distancia desde w hasta x , mostremos por inducción sobre la distancia d desde w hasta x que $G[x]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa.

Caso base: $d = D$. O bien, $x = z$. Dado que G es libre de $R_{3,1}$, ningún nodo del coárbol de G' con etiqueta 1 puede tener 3 hijos que representen gráficas que no sean multipartitas completas. Si x , que tiene etiqueta 1, tiene 2 hijos x_1 y x_2 que representan gráficas que no son multipartitas completas, dado que G es libre de $R_{3,1}$ y $R_{3,2}$, ninguno de estos puede contener ni a H' ni a J' , por lo que cada uno representa una gráfica de la clase $(1, \infty)$ - M_2 . Es decir que $G[x_1]$ acepta una M_2 -partición (A_1, B_1) tal que $G[A_1]$ es un conjunto independiente y $G[B_1]$ es una gráfica multipartita completa. De manera análoga, $G[x_2]$ acepta una M_2 -partición (A_2, B_2) tal que $G[A_2]$ es un conjunto independiente y $G[B_2]$ es una gráfica multipartita completa. Luego, $(G[x] - A_1) - A_2$ es la unión completa de múltiples gráficas multipartitas completas. Es decir, es una gráfica multipartita completa. Luego, $(A_1, A_2, (V(G[x]) - A_1) - A_2)$ es una M_3 -partición de $G[x]$ en dos conjuntos independientes y una gráfica multipartita completa. Notemos que los argumentos utilizados hasta ahora en el caso base funcionan para cualquier nodo del árbol de G con etiqueta 1 que contengan un Paw

Si x tiene un solo hijo y que representa una gráfica que no es multipartita completa, por el caso base del *Caso 2* de la demostración del Teorema 4.2.4, podemos concluir que todos los hijos de y representan gráficas multipartitas completas. Dado que G es libre de $O_{3,1}$, si y tiene al menos dos hijos que representan gráficas no bipartitas, dichos hijos deben de representar gráficas 3 coloreables. Luego, $G[y]$ es 3-coloreable. Así $G[x]$ es la unión completa de una gráfica 3-coloreable y varias gráficas multipartitas completas. Luego, $G[x]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Por otra parte, si y tiene un solo hijo que representa una gráfica que no es bipartita, recordemos que dicho hijo representa necesariamente una gráfica multipartita completa. Luego, $G[y]$ es la unión de una gráfica multipartita completa y varias gráficas bipartitas. Es decir que acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Luego, como todos los demás hijos de x representan gráficas multi-

partitas completas, $G[x]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa.

Finalmente, si todos los hijos de y representan gráficas bipartitas, $G[y]$ es bipartita. Luego, $G[x]$ es la unión completa de una gráfica bipartita y varias gráficas multipartitas completas. Así, $G[x]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa.

Paso inductivo: $d \geq 2$.

Notemos que d siempre es par. Supongamos como hipótesis inductiva que si x' es un descendiente de w con etiqueta 1 tal que contiene un Paw y su distancia hasta w es menor a d , entonces $G[x']$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa.

Sabemos que w no puede tener tres hijos que representen gráficas que no sean multipartitas completas, y que si tiene 2 de estos, acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Si x tiene un solo hijo y , que representa una gráfica que no es multipartita completa, veamos que $G[y]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Si y tiene al menos 2 hijos que representan gráficas no bipartitas, por el argumento utilizado en el caso base, sabemos que $G[y]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Por otra parte, si y tiene un solo hijo x' que no representa una gráfica bipartita, veamos que éste acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Sabemos que todos los hijos de y diferentes de x' representan gráficas bipartitas conexas. Es decir, representan gráficas multipartitas completas. Como x no es el nodo más profundo con un Paw y todos sus hijos con excepción de y representan gráficas multipartitas completas, entonces x' debe de contener un Paw . Luego, como x' cumple con las condiciones de la hipótesis inductiva, $G[x]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Como todos los hijos de y diferentes de x' representan gráficas bipartitas, $G[y]$ también acepta una partición de este tipo. Finalmente, como todos los hijos de x diferentes de y representan gráficas multipartitas completas, $G[x]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa.

Caso 8: G tiene sólo dos componentes. Una de ellas es bipartita no trivial y la otra no es bipartita.

Sea x la raíz del córbol de G y y el hijo de x que representa a una gráfica no bipartita. Si todos los hijos de y representan gráficas en M_2 , entonces $G[y]$ es la unión completa de varias gráficas en M_2 . Así, $G[y] \in M_2$

y por lo tanto $G \in M_3$, dado que el otro hijo de x representa una gráfica bipartita conexa. Es decir, una gráfica bipartita completa. Si y tiene un hijo x' (con etiqueta 0) tal que $G[x']$ no es una gráfica en M_2 , veamos que $G[y]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Dado que G es libre de $S_{3,1}$, $S_{3,2}$ y $S_{3,1}$, todos los hijos de y diferentes de x' representan gráficas multipartitas completas. Si todos los hijos de x' representan gráficas 3-coloreables, entonces $G[x']$ es 3-coloreable, y por lo tanto $G[y]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Abordemos el caso en el que un hijo de x' tiene un hijo y' que contiene un K_4 . Veamos que todos los demás hijos de x' deben de representar gráficas bipartitas. Procedamos por contradicción. Supongamos que un hijo de x' diferente de y' tiene un K_3 . Como G es libre de $O_{3,1}$, x' no puede tener más hijos. Como G es libre de $O_{3,2}$, $O_{3,3}$ y $O_{3,4}$, los dos hijos de x' deben de representar gráficas multipartitas completas. Luego, $G[x'] \in M_2$. Lo cual es una contradicción. Así, todos los hijos de x' menos y' representan gráficas bipartitas. Notemos que x con etiqueta 0 tiene un hijo diferente de y con un K_2 y que x' , que es descendiente de x y tiene etiqueta 0, tiene al menos un hijo diferente de y' . Luego, podemos aplicar el caso 7 de esta demostración a x' , por lo que $G[x']$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Luego, como todos los hijos de y diferentes de x' representan gráficas multipartitas completas, entonces $G[y]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. De eso se sigue que G acepta una partición de este tipo. Así, $G \in M_3$.

Caso 9: G tiene una componente que no es bipartita completa y al menos una componente trivial.

Notemos que la única posible M_3 -partición de G es en un conjunto independiente y dos gráficas multipartitas completas. Sea w la raíz del coárbol de G y x el hijo de w que no es una hoja. Si $G[x] \in M_2$, entonces es claro que $G \in M_3$. En el caso contrario, dado que $G[x]$ es una gráfica conexa, $G[x]$ debe tener a alguna de las siguientes gráficas como subgráfica inducida: $\mathcal{H} = K_1 \oplus H, \mathcal{I} = K_1 \oplus I, \mathcal{J} = K_1 \oplus J$. Sea y el descendiente de x más profundo que contiene a \mathcal{H} , a \mathcal{I} o a \mathcal{J} , d la distancia desde x hasta y , y' un nodo del coárbol de G con etiqueta 1 que contiene a \mathcal{H} , a \mathcal{I} o a \mathcal{J} y d' la distancia desde y' hasta y , mostremos por inducción sobre d' que $G[y']$ acepta una partición en un conjunto independiente y una gráfica multipartita completa.

Caso base: $d' = 0$. O bien, $y' = y$. Como G es libre de $P_{3,i}$ para todo

entero $1 \leq i \leq 6$, a lo más un hijo de y' representa una gráfica que no está en M_2 . Si todos los hijos de y' representan gráficas en M_2 , entonces $G[y'] \in M_2$. Si y' tiene un hijo z que representa una gráfica que no está en M_2 , todos los hijos de z representan gráficas en M_2 dado que y' es el nodo más profundo que contiene a \mathcal{H} , a \mathcal{I} o a \mathcal{J} . Si todos los hijos de z representan gráficas 3-coloreables, $G[z]$ es 3-coloreable y por lo tanto $G[y']$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas.

Si no todos los hijos de z representan gráficas 3-coloreables, uno de ellos contiene un K_4 . Si además de esto, otro hijo de z contiene un K_3 , dado que G es libre de $O_{3,1}$, todos los demás hijos de z son hojas. Como G es libre de $Q_{3,1}$, los dos hijos de z que no son hojas no contienen a $(\overline{P}_3 \oplus \overline{P}_3 = J')$. Como G es libre de $O_{3,5}$ los hijos de z representan gráficas conexas, los dos hijos de z que no son hojas no contienen a $K_2 + K_3 = H'$. Así, los dos hijos de z que no son hojas representan gráficas en $(1, \infty)$ - M_2 . Como z tiene etiqueta 0 y todos sus demás hijos son hojas, $G[z]$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas. Como y' tiene etiqueta 1 y todos sus hijos diferentes de z representan gráficas en M_2 , $G[y']$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completa.

Si además del hijo de z con un K_4 , z no tiene ningún otro hijo con un K_3 , pero sí tiene un hijo que contiene un K_2 , dado que w tiene etiqueta 0, es ancestro de z que también tiene etiqueta 0 y tiene al menos un hijo diferente de x , se puede aplicar el argumento del *Caso 8* de esta misma demostración a z . Luego, $G[z]$ acepta una partición en dos conjuntos independientes y una gráfica multipartita completa. Luego, como todos los hijos de y' diferentes de z representan gráficas en M_2 , $G[y']$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas.

Si además del hijo de z con un K_4 , todos los demás hijos de z son hojas, como todos los hijos de z representan gráficas en M_2 , se sigue directamente que $G[z]$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas. Luego, $G[y']$ también acepta una partición en un conjunto independiente y dos gráficas multipartitas completas.

Paso inductivo: $d' \geq 2$

Sea x' un nodo con etiqueta 1 descendiente de x tal que contiene a \mathcal{H} , a \mathcal{I} o a \mathcal{J} . Supongamos como hipótesis inductiva que $G[x']$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas.

Por los argumentos presentados en el caso base, sabemos que a lo más un hijo de y' puede representar una gráfica que no esté en la clase M_2 , y que si

todos los hijos de y' representan gráficas en M_2 , entonces $G[y']$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas. Si y' tiene un hijo z que no representa una gráfica en M_2 , sabemos por los argumentos del caso base que si todos los hijos de z representan gráficas 3-coloreables o si uno de los hijos de z contiene un K_4 y otro un K_2 , entonces $G[y']$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas.

El último caso que no hemos abordado es en el que z tiene un sólo hijo x' que contiene un K_4 mientras que el resto de los hijos de z son hojas. Aplicando la hipótesis inductiva, podemos ver que $G[x']$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas. Como todos los hijos de z diferentes de x' son hojas, entonces $G[z]$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas. Finalmente, como todos los hijos de y' diferentes de z representan gráficas en M_2 , entonces $G[y']$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas.

Como x es un nodo con etiqueta 1 y contiene a \mathcal{H} , a \mathcal{I} o a \mathcal{J} , entonces $G[x]$ acepta una partición en un conjunto independiente y dos gráficas multipartitas completas y todos los hijos de w diferentes de x son hojas, se sigue que G acepta una partición en un conjunto independiente y dos gráficas multipartitas completas. Así, $G \in M_3$.

Caso 10: G es conexa.

Si G es isomorfa a K_1 , es claro que $G \in M_3$. En el caso contrario, G es la unión completa de varias gráficas inconexas libres de cada una de las gráficas listadas. Es decir que G es la unión completa de varias gráficas en M_3 . Como la clase es cerrada bajo la unión completa, $G \in M_3$.

□

4.4.2. Familia O de obstrucciones

En esta subsección presentamos a la familia O de obstrucciones mínimas para cualquier clase M_i .

O -componentes

Sean G una cográfica y l un entero mayor o igual a cero, decimos que G es una *O -componente* si su conjunto de vértices acepta una partición

$(A, B_1, B_2, \dots, B_l)$ tal que, para cualesquiera dos enteros $1 \leq i, j \leq l$, se cumplen las siguientes condiciones :

- $G[A]$ es una gráfica completa con m_0 vértices.
- $G[B_i]$ es una gráfica completa con m_i vértices en donde $1 \leq m_i < m_0 - 1$.
- Si $i \neq j$, entonces $m_i \neq m_j$.
- Cada uno de los vértices de B_i es adyacente a los mismos n_i vértices de A , con $0 < n_i < m_0 - m_i$.
- Si $i \neq j$, entonces ningún vértice de B_i es adyacente a algún vértice de B_j .

Decimos que $(A, B_1, B_2, \dots, B_l)$ es una O -partición de G y que G incluye a la gráfica completa K_m si m es igual a m_k para algún $0 \leq k \leq l$. Decimos que $G[A]$ es el cuerpo de G y que, para cualquier $0 \leq k \leq l$, $G[B_k]$ es una extremidad de G .

Sea G una O -componente y $(A, B_1, B_2, \dots, B_l)$ una O -partición de G , notemos que el cuerpo de G es necesariamente el clan más grande en G .

En la Figura 4.23 podemos ver algunos ejemplos de gráficas que son O -componentes y otros ejemplos de gráficas que no son O -componentes. Las gráficas G_1, G_2, G_3 y G_4 son O -componentes. La gráfica G_1 acepta la O -partición $(\{a, b, c\}, \{d\})$. La gráfica G_2 acepta la O -partición $(\{a, b, c, d\}, \{e\})$. La gráfica G_3 acepta la O -partición $(\{a, b, c, d\}, \{e\})$. La gráfica G_4 acepta la O -partición $(\{a, b, c, d\}, \{e\}, \{f, g\})$.

Por el contrario, las gráficas G_5, G_6, G_7 y G_8 no son O -componentes. La gráfica G_5 tiene dos extremidades del mismo orden. El cuerpo de la gráfica G_6 es un K_3 , pero ésta tiene una extremidad de orden mayor a 1. La gráfica G_7 tiene una extremidad que se conecta a más de 2 vértices del cuerpo de G . La gráfica G_8 cumple con las características listadas, pero los vértices e, a, d y f forman un P_4 , por lo que G_8 no es una cográfica, y por lo tanto, tampoco es una O -componente. Esto nos lleva a realizar la siguiente observación.

Sean G una O -componente, l un entero mayor o igual a 0, i, j enteros diferentes entre sí tales que $1 \leq i, j \leq l$, (A, B_1, \dots, B_l) una O -partición de G , x un vértice de B_i y y un vértice de B_j , notemos que no existen vértices diferentes $v, w \in A$ tales que v es adyacente a los vértices en B_i , pero no a los vértices en B_j y w es adyacente a los vértices en B_j , pero no a los vértices

en B_i , ya que, de ser así, x, v, w y y formarían un P_4 . De esto se sigue que si $|B_i| < |B_j|$, los vértices en B_i serán adyacentes a cada uno de los vértices de A a los que sean adyacentes los vértices de B_j .

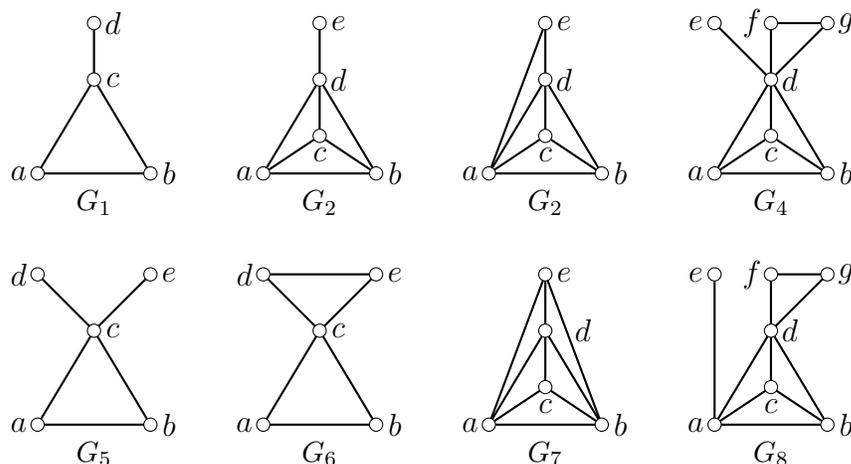


Figura 4.23: Ejemplos de O -componentes (G_1, G_2, G_3 y G_4) y de gráficas que no son O -componentes (G_5, G_6, G_7 y G_8).

Lema 4.4.2. *Sea G una O -componente, G es una gráfica completa o G no es una gráfica multipartita completa.*

Demostración. Sean l un entero mayor o igual a cero y $(A, B_1, B_2, \dots, B_l)$ una O -partición de G . Si $l = 0$, entonces G es igual a $G[A]$, y por lo tanto es una gráfica completa. En el caso contrario, veamos que, dado un entero $1 \leq i \leq l$, A tiene al menos dos vértices que no son adyacentes a ningún B_i . Sea $v \in B_i$, sabemos por la definición de O -componente que v es adyacente a máximo $|A| - |B_i| - 1$ vértices de A . Como i es mayor o igual a 1, entonces v es adyacente a lo más a $|A| - 2$ vértices de A . Luego, existen dos vértices $w, x \in A$ tales que no son adyacentes a v , pero que sí son adyacentes el uno al otro. Así, v, w y x forman un $\overline{P_3}$. De esto se sigue que G no es una gráfica multipartita completa. \square

O -obstrucciones

Sean G una cográfica y n, m enteros mayores o iguales a uno, decimos que G es una O_n -**obstrucción** si G es la unión ajena de algunas O -componentes

G_1, G_2, \dots, G_m tales que, para cualesquiera enteros $1 \leq i \leq m$ y $1 \leq j \leq n+1$, existe una única O -componente G_i que contiene a K_j .

Sea n un entero mayor o igual a uno, la **familia O_n de obstrucciones**, denotada simplemente por O_n , es el conjunto de todas las O_n -obstrucciones.

Teorema 4.4.3. *Sean n un entero mayor o igual a uno y G una cográfica. Si G es una O_n obstrucción, entonces G es una obstrucción mínima de la clase M_n .*

Demostración. Mostremos primero que G no está en la clase M_n . Procedamos por inducción sobre n .

Si $n = 1$, entonces tenemos que la única O_1 -obstrucción que hay es $K_1 + K_2$. Luego, $G = K_1 + K_2 = \overline{P}_3$. Como \overline{P}_3 es la única obstrucción de las gráficas multipartitas completas, se sigue que $G \notin M_1$.

Supongamos como hipótesis inductiva que si una cográfica G' es una O_{n-1} -obstrucción, entonces G' es una obstrucción mínima de la clase M_{n-1} , y consideremos a G una O_n -obstrucción.

Sea $P = (A_1, A_2, \dots, A_n)$ una partición de los vértices de G , veamos que P no es una M_n -partición. Recordemos que, como G es una O_n -obstrucción, ésta es la unión ajena de hasta $n + 1$ O -componentes. Y que exactamente una de estas O -componentes H contiene a K_{n+1} . Como G tiene a K_{n+1} como subgráfica inducida, G no es n -coloreable, y por lo tanto al menos dos vértices de dicho K_{n+1} están en la misma parte de P . Supongamos sin pérdida de generalidad que A_1 contiene al menos dos vértices del K_{n+1} de G .

Como H es una O -componente, ésta acepta una O -partición en i partes $Q = (C, B_1, B_2, \dots, B_i)$ para algún entero $0 \leq i \leq n - 2$. Notemos que C es el conjunto de los vértices del K_{n+1} de H . Denotemos por m_j la cardinalidad de B_j para todo $1 \leq j \leq i$. Supongamos sin pérdida de generalidad que $m_k < m_{k+1}$ para todo $1 \leq k < i$. Recordemos que A_1 contiene al menos dos vértices que están en C y que hay al menos dos vértices de C que no son adyacentes a vértice alguno de B_l para todo $1 \leq l \leq i$.

Si A_1 contiene sólo vértices de C , los vértices de $H - A_1$ inducen en G la unión ajena de un conjunto de gráficas que tienen como subgráficas inducidas O -componentes que, en conjunto, contienen a las gráficas completas $G[B_1], G[B_2], \dots, G[B_i]$. Los vértices de $H - A_1$ y las O -componentes de G diferentes de H deben de repartirse entre las $n - 1$ partes de P diferentes de A_1 . Notemos que $G - A_1$ tiene como subgráfica inducida a un conjunto de O -componentes que contienen todas las gráficas completas desde K_1 hasta K_n . Es decir que $G - A_1$ tiene como subgráfica inducida una O_{n-1} -obstrucción

que, por hipótesis inductiva, no está en la clase M_{n-1} . Luego, G no está en la clase M_n . En la Figura 4.24 se muestra un ejemplo de esto.

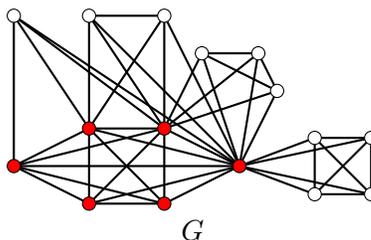


Figura 4.24: La gráfica G es una O -componente que contiene a K_1 , K_2 , K_3 , K_4 y K_6 . En rojo se marcan los vértices que se agregan a A_1 . Se puede apreciar que el resto de los vértices inducen la gráfica $K_1 + K_2 + K_3 + K_4$.

Abordemos el caso en el que A_1 contiene, además de dos vértices de C , a lo más un vértice de al menos una parte de Q diferente de C . Sean j, k enteros tales que A_1 contiene un vértice de B_j y un vértice de B_k . Si A_1 contiene un vértice v de C tal que v es adyacente a los vértices de B_j pero no es adyacente a los vértices de B_k , entonces el subconjunto de $V(A_1)$ que contiene a v , a un vértice de B_j y a un vértice de B_k induce un \overline{P}_3 , con lo que $G[A_1]$ no es una gráfica multipartita completa y P no es una O -partición.

En el caso contrario, un elemento de C está en A_1 si y sólo si es adyacente a todos los elementos de $A_1 - C$ o, en su defecto, no es adyacente a elemento alguno de $A_1 - C$. Si A_1 contiene dos elementos de C que no son adyacentes a elemento alguno de $A_1 - C$, entonces $G[A_1]$ no es una gráfica multipartita completa y por lo tanto P no es una M_n -partición.

Si todos los elementos de C que están en A_1 , excepto a lo más uno, son adyacentes a todos los vértices de $A_1 - C$, veamos que $H - A_1$ tiene como subgráfica inducida a la unión ajena de algunas O -componentes que, en conjunto, contienen a K_{m_l} para cualquier entero $1 \leq l \leq i$. Sea m el máximo entero tal que A_1 tiene un vértice de B_m . Notemos que para cualquier entero $m < l \leq i$, se cumple que $B_l \subset V(H) - A_1$, ya que no hay vértices de B_l en A_1 . Por otra parte, para cualquier entero $1 < l \leq m$, se cumple que $B_l - A_1$ tiene como subgráfica inducida a la gráfica completa con m_{l-1} vértices, ya que $m_l > m_{l-1}$ y a lo más uno de los vértices de B_l está en A_1 . Como ya hemos verificado que todas las gráficas completas con m_1, m_2, \dots, m_i vértices menos la que tiene $|B_m|$ vértices son subgráficas inducidas de $H - A_1$ y no

tienen vértices en común, sólo nos falta verificar que la gráfica completa con $|B_m|$ vértices también está en $H - A_1$, que no tiene vértices en común con las otras gráficas completas mencionadas, y que éstas forman O -obstrucciones.

Dado que todos los vértices de C que están en A_1 , excepto a lo más uno, son adyacentes a todos los vértices de $A_1 - C$, y m es el mayor entero tal que A_1 tiene un vértice de B_m , se cumple que A_1 contiene a lo más $n - |B_m| + 1$ vértices de C . Luego, $C - A_1$ debe de tener al menos $|B_m|$ vértices, por lo que $H - A_1$ contiene a la gráfica completa con $|B_m|$ vértices como subgráfica inducida. Como C es ajeno a B_l para cualquier entero $1 \leq l \leq i$, entonces esta gráfica completa con $|B_m|$ vértices no tiene vértices en común con ninguna de las otras gráficas completas encontradas. Ahora veamos que estas gráficas completas forman un conjunto ajeno de O -obstrucciones.

Sean C' un subconjunto de C tal que contiene $|B_m|$ elementos de C que no son adyacentes a elemento alguno de B_m , B'_l un subconjunto de B_{l+1} con $|B_l|$ elementos para cada entero $1 \leq l < m$, veamos que $Q' = (C', B'_1, B'_2, \dots, B'_{m-1})$ es una O -partición. Sabemos que $G[C']$ es una gráfica completa con m vértices. De igual manera, sabemos que, para cualquier entero $1 \leq l < m$, $G[B'_l]$ es una gráfica completa con m_l vértices con $1 \leq m_l < m$. De igual manera, sabemos que ninguna de las partes de Q tiene la misma cardinalidad y que vértices de diferentes partes diferentes de C' no son adyacentes. Sólo falta por mostrar que, para cualquier entero $1 \leq m_l < m$, cada uno de los vértices de B'_l es adyacente a los mismos a_l vértices de C' con $0 < a_l < m - m_l$. Como H es una O -componente, sabemos que cada uno de los vértices de B'_l es adyacente a los mismos a_l vértices de C' . Como cada vértice de C en A_1 es adyacente a cada uno de los vértices de B_{l+1} , entonces se mantiene la propiedad de que $0 < a_l < m - m_l$. Luego, Q' es una O -partición. Notemos que para cualquier entero $m < l \leq 1$ se cumple que B_l no tiene vértices adyacentes a vértice alguno de C' . Así, $H - A_1$ es la unión ajena de una O -componente y varias gráficas completas que, en conjunto, contienen a todas las gráficas completas con B_1, B_2, \dots y B_i vértices. En la Figura 4.25 se muestra un ejemplo de esto.

Finalmente, $G - A_1$ tiene como subgráfica inducida a la unión de algunas O -componentes que, en conjunto, contienen a todas las gráficas completas desde 1 hasta n vértices. Luego $G - A_1$ tiene como subgráfica inducida una O_{n-1} -onstrucción, que, por hipótesis inductiva, no está en la clase M_{n-1} . Así, P no es una M_n -partición.

Abordemos el caso en el que A_1 contiene, además de dos vértices de C , al menos dos vértices de B_j para algún $1 \leq j \leq i$. Si A_1 contiene un vértice de

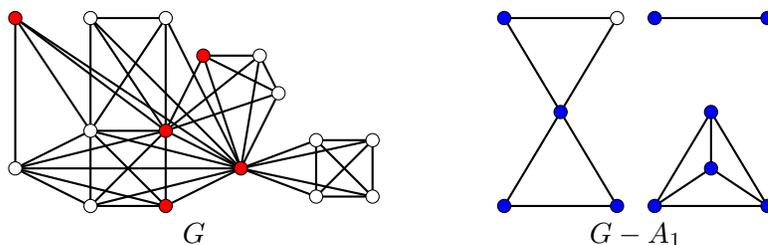


Figura 4.25: La gráfica G es una O -componente que contiene a K_1 , K_2 , K_3 , K_4 y a K_6 . En rojo se marcan los vértices que se agregan a A_1 . A la derecha se muestra $G - A_1$. En azul se muestran los vértices de una O -obstrucción que contiene a K_1 , K_2 , K_3 y a K_4 .

C que no es adyacente a los vértices de B_j o algún vértice que no es elemento ni de C ni de B_j , entonces $G[A_1]$ no es una gráfica multipartita completa. Y por lo tanto, P no es una M_n -partición. Si A_1 contiene sólo vértices de B_j y vértices de C adyacentes a los vértices de B_j , como los vértices de B_j son adyacentes a máximo $n - m_j$ vértices de C , entonces al menos $m_j + 1$ vértices de C no son adyacentes a los vértices de B_j . Luego, al menos m_j vértices de C no están en A_1 . Así, los vértices de H que no están en A_1 inducen una gráfica que tiene como subgráfica inducida la unión ajena de algunas O -componentes que, en conjunto, contienen a las gráficas completas con m_1, m_2, \dots y m_{i-1} vértices. Luego, $G - A_1$ tiene como subgráfica inducida una O_{n-1} -obstrucción que, por hipótesis inductiva, no está en la clase M_{n-1} . Luego, G no está en la clase M_n . En la Figura 4.26 se muestra un ejemplo de esto.

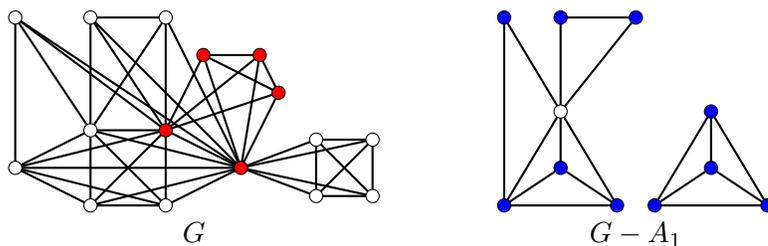


Figura 4.26: La gráfica G es una O -componente que contiene a K_1 , K_2 , K_3 , K_4 y a K_6 . En rojo se marcan los vértices que se agregan a A_1 . A la derecha se muestra $G - A_1$. En azul se muestran los vértices de una O -obstrucción que contiene a K_1 , K_2 , K_3 y a K_4 .

Para mostrar que G es una obstrucción mínima, basta con notar que en cada uno de los casos anteriores, si se sustrae un vértice de G , entonces la gráfica resultante está en M_n .

□

O -obstrucciones conocidas

A lo largo de este documento, podemos identificar varias O -obstrucciones. Notemos que $\overline{P_3}$ es una O -obstrucción. En la Figura 4.4, las gráficas H e I son O -obstrucciones. Todas las gráficas de la Figura 4.21 son O -obstrucciones. En el apéndice C podemos observar a la familia O_4 de obstrucciones.

4.4.3. Familia P de obstrucciones

En esta subsección presentamos a la familia P de obstrucciones mínimas para cualquier clase M_i .

Sean G una cográfica y n un entero mayor a uno, decimos que G es una P_n -**obstrucción** si, para algunas obstrucciones mínimas H_1, H_2 de la clase M_{n-1} , $G = K_1 + (H_1 \oplus H_2)$.

Sea n un entero mayor a uno, la **familia P_n de obstrucciones**, denotada simplemente por P_n , es el conjunto de todas las P_n -obstrucciones.

Teorema 4.4.4. *Sea G una cográfica y n un entero mayor a uno. Si G es una P_n -obstrucción, entonces G es una obstrucción mínima de la clase M_n .*

Demostración. Como G es una P_n -obstrucción, entonces existen dos obstrucciones mínimas H_1 y H_2 de la clase M_{n-1} tales que $G = K_1 + (H_1 \oplus H_2)$.

Veamos primero que G no está en la clase M_n . Sea $P = (A_1, A_2, \dots, A_n)$ una partición de los vértices de G , veamos que P no es una M_n -partición. Supongamos sin pérdida de generalidad que el K_1 de G está en A_1 . Si A_1 contiene un par de vértices adyacentes entre sí, entonces $G[A_1]$ no es una gráfica multipartita completa y P no es una M_n -partición. En el caso contrario, A_1 debe de ser un conjunto independiente. Como A_1 es un conjunto independiente y cada vértice de H_1 es adyacente a cualquier vértice de H_2 , entonces A_1 no puede contener vértices tanto de H_1 como de H_2 . Supongamos sin pérdida de generalidad que ningún vértice de H_1 está en A_1 . Luego, los vértices de H_1 se deben de repartir en las $n - 1$ partes de P distintas de A_1 . Como H_1 es una obstrucción mínima de la clase M_{n-1} , cualquier partición

de sus vértices en $n - 1$ partes no es una M_{n-1} -partición. De esto se sigue que P no es una M_n -partición. Así, G no está en la clase M_n .

Sea x un vértice de G , veamos que $G - x$ sí está en la clase M_n . Si x es el vértice que conforma al K_1 de G , entonces existe una partición $P = (A_1, A_2, \dots, A_n)$ tal que A_1 contiene un vértice y_1 de H_1 y un vértice y_2 de H_2 . Como H_1 y H_2 son obstrucciones mínimas de la clase M_{n-1} , entonces $H_1 - \{y_1\}$ y $H_2 - \{y_2\}$ están en la clase M_{n-1} . Luego, $((H_1 - \{y_1\}) \oplus (H_2 - \{y_2\}))$ está en M_{n-1} , por lo que sus vértices se pueden repartir entre las partes de P diferentes de A_1 de forma que cada parte induzca una gráfica multipartita completa. Notemos que $G[A_1] = K_2$ también es una gráfica multipartita completa. Así $G - x$ está en M_n .

Si x no es el vértice que conforma al K_1 de G , supongamos sin pérdida de generalidad que x es un vértice de H_1 . Luego, existe una partición $Q = (B_1, B_2, \dots, B_n)$ tal que B_1 contiene al vértice aislado de G y a un vértice de H_2 . Por el argumento anterior, sabemos que $G - B_1$ está en la clase M_{n-1} . Por este motivo, y porque $G[B_1] = 2K_1$ es una gráfica multipartita completa, se sigue que $G - x$ está en M_n . Así, G es una obstrucción mínima de la clase M_n .

□

P -obstrucciones conocidas

A lo largo de este documento podemos identificar algunas P -obstrucciones. En la Figura 4.4, la gráfica J es una P -obstrucción. En la lista de obstrucciones mínimas de la clase M_3 , la gráfica $P_{3,i}$ es una P -obstrucción para todo $1 \leq i \leq 6$.

Capítulo 5

Conclusiones y trabajo futuro

En esta tesis estudiamos a la clase de cográficas que aceptan una partición en un número i de gráficas multipartitas completas, para distintos valores de i . Denotamos a estas clases por M_i . Nuestra investigación tiene como punto de partida la clase M_2 , a la cual estudiamos a profundidad tomando como referencia la investigación realizada sobre las cográficas polares.

A partir del estudio de la clase M_2 , encontramos un algoritmo (Algoritmo 7) que es capaz de determinar si una cográfica G , representada a través de su coárbol, es elemento de una clase hereditaria de cográficas fija en tiempo lineal con respecto al orden de G . Este algoritmo funciona utilizando una variación de los coárboles, los coárboles binarios y el concepto de subcoárbol binario, que introducimos en este trabajo.

Se puede caracterizar a la clase M_2 a través de su conjunto de obstrucciones mínimas que se muestra en la Figura 4.4. De igual manera, se presentan dos algoritmos para determinar si una cográfica G , representada a través de su coárbol, pertenece a la clase M_2 . El primero de éstos (Algoritmo 8) es una instancia del algoritmo mencionado en el párrafo anterior. Éste funciona generando un coárbol binario B de G y determinando si alguno de los coárboles binarios de las obstrucciones mínimas de M_2 es subárbol binario de B . El segundo algoritmo (Algoritmo 12) es un algoritmo certificador. Es decir que no sólo determina si G pertenece o no a la clase M_2 , sino que devuelve un sí-certificado o un no-certificado. En el caso de que G , pertenezca a la clase M_2 , el algoritmo colorea las hojas del coárbol de G de dos colores que conforman una M_2 -partición de G . En el caso contrario, el algoritmo colorea algunas hojas del coárbol de G que representan vértices de G que forman una obstrucción mínima de la clase. Este algoritmo se realizó siguiendo la

demostración del Teorema 4.2.4, en el que se presentan las obstrucciones mínimas de la clase M_2 . La ejecución de ambos algoritmos toma una cantidad de tiempo que crece de forma lineal con respecto al orden de G . Sin embargo, una vez implementados, el algoritmo certificador tuvo un desempeño mucho mejor. Se comprobó con éxito que los resultados de ambos algoritmos son coherentes.

Siguiendo la investigación realizada sobre las cográficas polares, estudiamos las clases (α, β) - M_2 , que son las subclases de M_2 que contienen a una gráfica si ésta acepta una partición en dos gráficas multipartitas completas tales que una de ellas está formada por a lo más α conjuntos independientes y la otra está formada por a lo más β conjuntos independientes con α y β enteros mayores o iguales a cero tales que $\alpha \leq \beta$. Presentamos un algoritmo (Algoritmo 16) que, dados dos enteros α y β , genera todas las obstrucciones mínimas de la clase (α, β) - M_2 hasta cierto orden. Este algoritmo utiliza el algoritmo certificador de la clase M_2 . Utilizando este algoritmo, generamos las obstrucciones mínimas de hasta 15 vértices de varias clases (α, β) - M_2 . Si fijamos α , podemos observar que las obstrucciones generadas para clases consecutivas siguen un comportamiento predecible que podría ser descrito definiendo familias de obstrucciones como lo hicimos con la clase $(1, \infty)$ - M_2 .

El siguiente paso en nuestra investigación fue caracterizar a la clase M_3 a través de su conjunto de obstrucciones mínimas (Teorema 4.4.1). Esto nos ayudó a encontrar dos familias de obstrucciones mínimas para cualquier clase M_i . Llamamos a estas familias la familia O y la familia P de obstrucciones. Todas las obstrucciones mínimas de las clases M_1 y M_2 pertenecen a las familias O y P , mientras que 14 de 21 obstrucciones mínimas de la clase M_3 pertenecen a estas dos familias. Las otras 7 fueron agrupadas en otras 3 familias que podrían generalizarse.

5.1. Trabajo a futuro

En nuestra tesis encontramos varios temas con los que se puede continuar el estudio de las clases M_i .

Uno de los resultados principales de nuestra tesis, el Algoritmo 7, funciona determinando si cada uno de los coárboles binarios de un conjunto es subcoárbol binario de otro coárbol binario. El número de coárboles binarios de una gráfica (en este caso la obstrucción mínima de alguna clase) crece de forma exponencial, por lo que, aunque el algoritmo funcione en tiempo lineal,

éste es poco eficiente. A partir de esto, nos preguntamos si una definición más general de sucoárbol pueda dar lugar a un algoritmo más eficiente.

El Algoritmo 16 recibe como entrada tres enteros α , β y n y genera todas las cográficas de hasta n vértices que son obstrucciones mínimas de la clase (α, β) - M_2 . Los resultados que generamos con este algoritmo son obstrucciones mínimas de hasta 15 vértices. Si bien, creemos que los conjuntos de obstrucciones mínimas generados para clases que no tienen obstrucciones mínimas ni con 13 ni con 14 vértices son exhaustivos, no podemos estar seguros de ello ya que no contamos con una cota superior para el número de vértices de una obstrucción mínima lo suficientemente pequeña. Así, un resultado que tendría gran impacto en la investigación de las clases M_i sería encontrar dicha cota superior.

En el Apéndice A se listan las obstrucciones mínimas de varias clases (α, β) - M_2 . Si fijamos α , podemos observar un comportamiento bastante predecible para valores de β consecutivos. Tomando como ejemplo el análisis que realizamos de las clases $(1, \beta)$ - M_2 , nos preguntamos si podemos encontrar resultados parecidos para valores de α diferentes de 1.

Así como encontrar el conjunto de obstrucciones mínimas de la clase M_3 nos llevó a identificar familias de obstrucciones mínimas para cualquier clase M_i , seguir el estudio de la clase M_3 como lo hicimos con la clase M_2 podría llevarnos a resultados más generales. Así, quedan pendientes las siguientes tareas:

- Encontrar un algoritmo para identificar a los elementos de M_3 .
- Encontrar un algoritmo certificador para la clase M_3 .
- El estudio de las clases (α, β, γ) - M_3 .

Otra forma de darle continuidad a los resultados presentados en esta tesis es encontrar un algoritmo para computar a los elementos de la familia O_n de obstrucciones para un entero n .

Finalmente, un resultado importante sería identificar otras familias de obstrucciones para cualquier clase M_i .

Apéndice A

Obstrucciones mínimas de algunas clases (α, β) - M_2

A continuación enlistamos las obstrucciones mínimas de hasta 15 vértices encontradas con el Algoritmo 16 para algunas clases (α, β) - M_2 .

A.1. Algunas obstrucciones mínimas de la clase $(3, 3)$ - M_2

- $o_{(3,3),1} = K_1 + K_5$.
- $o_{(3,3),2} = K_2 + K_4$.
- $o_{(3,3),3} = K_7$.
- $o_{(3,3),4} = K_2 \oplus (K_2 + K_3)$.
- $o_{(3,3),5} = K_1 \oplus (K_3 + K_3)$.
- $o_{(3,3),6} = (K_1 + K_2) \oplus (K_1 + K_4)$.

A.2. Algunas obstrucciones mínimas de la clase $(3, 4)$ - M_2

- $o_{(3,4),1} = K_1 + K_6$.

- $o_{(3,4),2} = K_2 + K_5$.
- $o_{(3,4),3} = K_8$.
- $o_{(3,4),4} = K_3 \oplus (K_2 + K_3)$.
- $o_{(3,4),5} = K_2 \oplus (K_2 + K_4)$.
- $o_{(3,4),6} = K_2 \oplus (K_3 + K_3)$.
- $o_{(3,4),7} = K_1 \oplus (K_3 + K_4)$.
- $o_{(3,4),8} = K_4 + K_4$.
- $o_{(3,4),9} = (K_1 + K_2) \oplus (K_1 + K_5)$.

A.3. Algunas obstrucciones mínimas de la clase $(3, 5)$ - M_2

- $o_{(3,5),1} = K_1 + K_7$.
- $o_{(3,5),2} = K_2 + K_6$.
- $o_{(3,5),3} = K_4 + K_4$.
- $o_{(3,5),4} = K_9$.
- $o_{(3,5),5} = K_4 \oplus (K_2 + K_3)$.
- $o_{(3,5),6} = K_3 \oplus (K_2 + K_4)$.
- $o_{(3,5),7} = K_3 \oplus (K_3 + K_3)$.
- $o_{(3,5),8} = K_2 \oplus (K_2 + K_5)$.
- $o_{(3,5),9} = K_2 \oplus (K_3 + K_4)$.
- $o_{(3,5),10} = K_1 \oplus (K_3 + K_5)$.
- $o_{(3,5),11} = (K_1 + K_2) \oplus (K_3 + K_3)$.
- $o_{(3,5),12} = (K_1 + K_2) \oplus (K_1 + K_6)$.

A.4. ALGUNAS OBSTRUCCIONES MÍNIMAS DE LA CLASE $(3, 6)-M_2$ 113

- $o_{(3,5),13} = (K_1 + K_4) \oplus (K_1 + K_4)$.
- $o_{(3,5),14} = (K_2 + K_3) \oplus (K_2 + K_3)$.
- $o_{(3,5),15} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_2)$.

A.4. Algunas obstrucciones mínimas de la clase $(3, 6)-M_2$

- $o_{(3,6),1} = K_4 + K_4$.
- $o_{(3,6),2} = K_1 + K_8$.
- $o_{(3,6),3} = K_2 + K_7$.
- $o_{(3,6),4} = (K_1 + K_2) \oplus (K_3 + K_3)$.
- $o_{(3,6),5} = K_{10}$.
- $o_{(3,6),6} = K_5 \oplus (K_2 + K_3)$.
- $o_{(3,6),7} = K_4 \oplus (K_2 + K_4)$.
- $o_{(3,6),8} = K_4 \oplus (K_3 + K_3)$.
- $o_{(3,6),9} = K_3 \oplus (K_2 + K_5)$.
- $o_{(3,6),10} = K_3 \oplus (K_3 + K_4)$.
- $o_{(3,6),11} = K_2 \oplus (K_2 + K_6)$.
- $o_{(3,6),12} = K_2 \oplus (K_3 + K_5)$.
- $o_{(3,6),13} = K_1 \oplus (K_3 + K_6)$.
- $o_{(3,6),14} = (K_2 + K_3) \oplus (K_2 + K_3)$.
- $o_{(3,6),15} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_2 + K_3)$.
- $o_{(3,6),16} = (K_1 + K_2) \oplus (K_1 + K_7)$.
- $o_{(3,6),17} = (K_1 + K_4) \oplus (K_1 + K_5)$.
- $o_{(3,6),18} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_2)$.

A.5. Algunas obstrucciones mínimas de la clase $(3, 7)$ - M_2

- $o_{(3,7),1} = K_4 + K_4.$
- $o_{(3,7),2} = (K_1 + K_2) \oplus (K_3 + K_3).$
- $o_{(3,7),3} = K_1 + K_9.$
- $o_{(3,7),4} = K_2 + K_8.$
- $o_{(3,7),5} = (K_2 + K_3) \oplus (K_2 + K_3).$
- $o_{(3,7),6} = K_{11}.$
- $o_{(3,7),7} = K_6 \oplus (K_2 + K_3).$
- $o_{(3,7),8} = K_5 \oplus (K_2 + K_4).$
- $o_{(3,7),9} = K_5 \oplus (K_3 + K_3).$
- $o_{(3,7),10} = K_4 \oplus (K_2 + K_5).$
- $o_{(3,7),11} = K_4 \oplus (K_3 + K_4).$
- $o_{(3,7),12} = K_3 \oplus (K_2 + K_6).$
- $o_{(3,7),13} = K_3 \oplus (K_3 + K_5).$
- $o_{(3,7),14} = K_2 \oplus (K_2 + K_7).$
- $o_{(3,7),15} = K_2 \oplus (K_3 + K_6).$
- $o_{(3,7),16} = K_1 \oplus (K_3 + K_7).$
- $o_{(3,7),17} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_2 + K_3).$
- $o_{(3,7),18} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_2).$
- $o_{(3,7),19} = (K_1 + K_2) \oplus (K_1 + K_8).$
- $o_{(3,7),20} = (K_1 + K_4) \oplus (K_1 + K_6).$
- $o_{(3,7),21} = (K_1 + K_5) \oplus (K_1 + K_5).$

A.6. Algunas obstrucciones mínimas de la clase $(4, 4)$ - M_2

- $o_{(4,4),1} = K_1 + K_6$.
- $o_{(4,4),2} = K_2 + K_5$.
- $o_{(4,4),3} = K_9$.
- $o_{(4,4),4} = K_4 \oplus (K_2 + K_3)$.
- $o_{(4,4),5} = K_3 \oplus (K_2 + K_4)$.
- $o_{(4,4),6} = K_3 \oplus (K_3 + K_3)$.
- $o_{(4,4),7} = K_2 \oplus (K_3 + K_4)$.
- $o_{(4,4),8} = K_1 \oplus (K_4 + K_4)$.
- $o_{(4,4),9} = K_1 \oplus (K_2 + K_5)$.
- $o_{(4,4),10} = (K_1 + K_2) \oplus (K_2 + K_4)$.
- $o_{(4,4),11} = (K_2 + K_3) \oplus (K_2 + K_3)$.
- $o_{(4,4),12} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_4)$.

A.7. Algunas obstrucciones mínimas de la clase $(4, 5)$ - M_2

- $o_{(4,5),1} = K_1 + K_7$.
- $o_{(4,5),2} = K_2 + K_6$.
- $o_{(4,5),3} = K_{10}$.
- $o_{(4,5),4} = K_5 \oplus (K_2 + K_3)$.
- $o_{(4,5),5} = K_4 \oplus (K_2 + K_4)$.
- $o_{(4,5),6} = K_4 \oplus (K_3 + K_3)$.

- $o_{(4,5),7} = K_3 \oplus (K_2 + K_5)$.
- $o_{(4,5),8} = K_3 \oplus (K_3 + K_4)$.
- $o_{(4,5),9} = K_2 \oplus (K_3 + K_5)$.
- $o_{(4,5),10} = K_2 \oplus (K_4 + K_4)$.
- $o_{(4,5),11} = K_1 \oplus (K_4 + K_5)$.
- $o_{(4,5),12} = (K_1 + K_2) \oplus (K_1 + K_6)$.
- $o_{(4,5),13} = (K_1 + K_2) \oplus (K_2 + K_5)$.
- $o_{(4,5),14} = K_5 + K_5$.
- $o_{(4,5),15} = (K_2 + K_3) \oplus (K_2 + K_3)$.
- $o_{(4,5),16} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_5)$.

A.8. Algunas obstrucciones mínimas de la clase $(4, 6)$ - M_2

- $o_{(4,6),1} = K_1 + K_8$.
- $o_{(4,6),2} = K_2 + K_7$.
- $o_{(4,6),3} = K_5 + K_5$.
- $o_{(4,6),4} = K_{11}$.
- $o_{(4,6),5} = K_6 \oplus (K_2 + K_3)$.
- $o_{(4,6),6} = K_5 \oplus (K_2 + K_4)$.
- $o_{(4,6),7} = K_5 \oplus (K_3 + K_3)$.
- $o_{(4,6),8} = K_4 \oplus (K_2 + K_5)$.
- $o_{(4,6),9} = K_4 \oplus (K_3 + K_4)$.
- $o_{(4,6),10} = K_3 \oplus (K_2 + K_6)$.

A.8. ALGUNAS OBSTRUCCIONES MÍNIMAS DE LA CLASE $(4, 6)$ - M_2 117

- $o_{(4,6),11} = K_3 \oplus (K_3 + K_5)$.
- $o_{(4,6),12} = K_3 \oplus (K_4 + K_4)$.
- $o_{(4,6),13} = K_2 \oplus (K_3 + K_6)$.
- $o_{(4,6),14} = K_2 \oplus (K_4 + K_5)$.
- $o_{(4,6),15} = K_1 \oplus (K_2 + K_3) \oplus (K_2 + K_3)$.
- $o_{(4,6),16} = K_1 \oplus (K_4 + K_6)$.
- $o_{(4,6),17} = (K_1 + K_2) \oplus (K_1 + K_7)$.
- $o_{(4,6),18} = (K_1 + K_2) \oplus (K_2 + K_6)$.
- $o_{(4,6),19} = (K_1 + K_2) \oplus (K_4 + K_4)$.
- $o_{(4,6),20} = (K_1 + K_4) \oplus (K_2 + K_4)$.
- $o_{(4,6),21} = (K_2 + K_3) \oplus (K_2 + K_4)$.
- $o_{(4,6),22} = (K_2 + K_3) \oplus (K_3 + K_3)$.
- $o_{(4,6),23} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_3 + K_3)$.
- $o_{(4,6),24} = (K_1 + K_5) \oplus (K_1 + K_5)$.
- $o_{(4,6),25} = (K_1 + K_2) \oplus (K_1 + K_2) \oplus (K_1 + K_6)$.
- $o_{(4,6),26} = (K_1 + K_2) \oplus (K_1 + K_4) \oplus (K_1 + K_4)$.

Apéndice B

Clase	Número de obstrucciones encontradas	Menor número de vértices de una obstrucción	Mayor número de vértices de una obstrucción
$(1, 1)-M_2$	1	3	3
$(1, 2)-M_2$	2	4	5
$(1, 3)-M_2$	3	5	6
$(1, 4)-M_2$	3	5	6
$(1, 5)-M_2$	3	5	7
$(1, 6)-M_2$	3	5	8
$(2, 2)-M_2$	3	5	5
$(2, 3)-M_2$	5	6	6
$(2, 4)-M_2$	7	6	9
$(2, 5)-M_2$	9	6	9
$(2, 6)-M_2$	10	6	9
$(2, 7)-M_2$	11	6	10
$(3, 3)-M_2$	6	6	8
$(3, 4)-M_2$	9	7	9
$(3, 5)-M_2$	15	8	12
$(3, 6)-M_2$	18	8	12
$(3, 7)-M_2$	21	8	12
$(4, 4)-M_2$	12	7	11
$(4, 5)-M_2$	16	8	12
$(4, 6)-M_2$	26	9	13

Tabla B.1: Resumen de las obstrucciones mínimas encontradas con el Algoritmo 16

Apéndice C

Familias O_n de obstrucciones

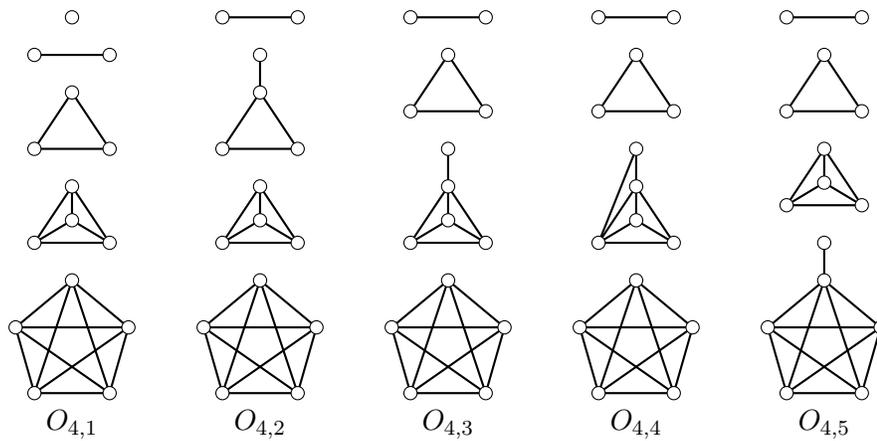
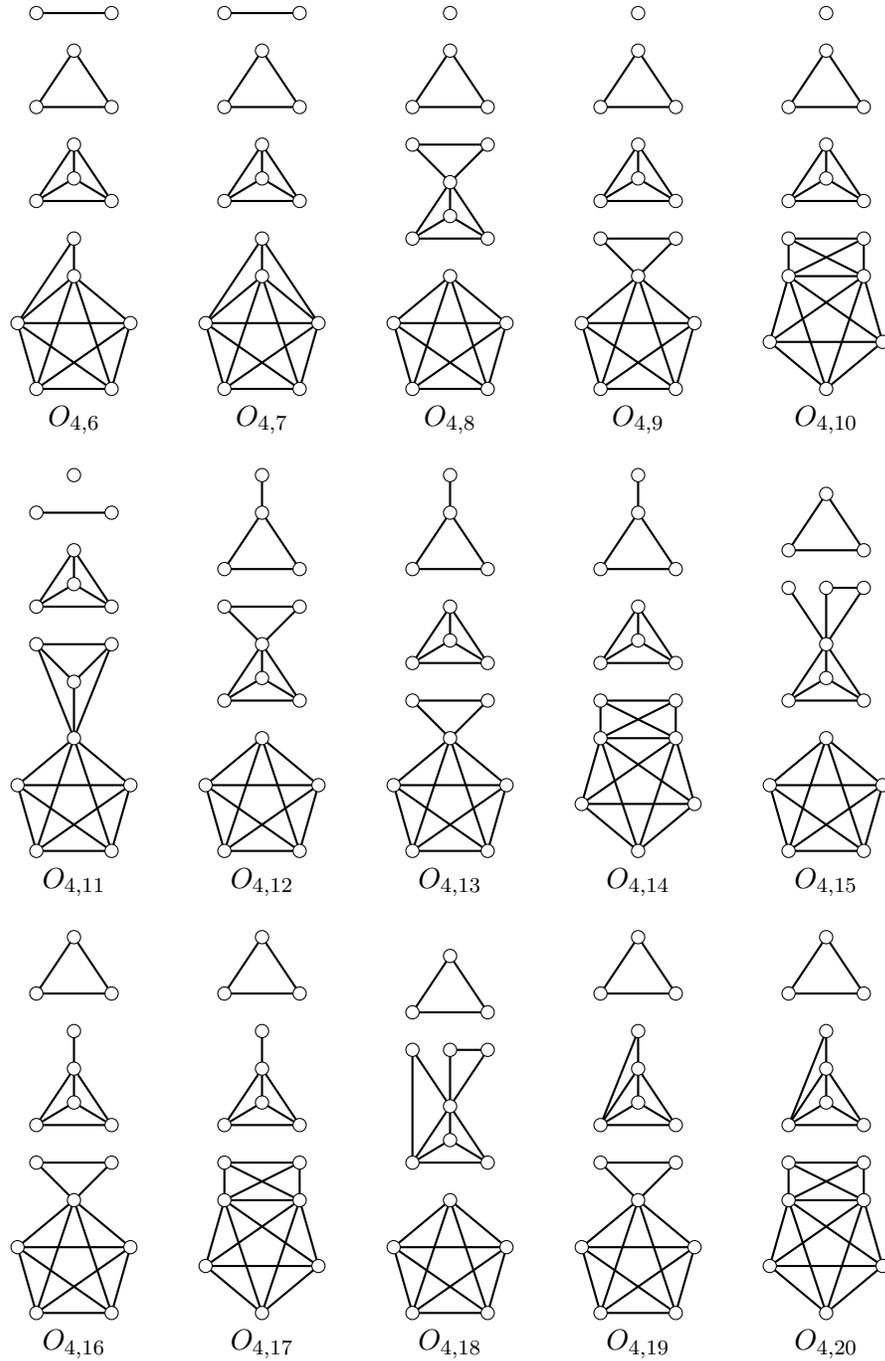


Figura C.1: Familia O_4 de obstrucciones. Parte 1.

Figura C.2: Familia O_4 de obstrucciones. Parte 2.

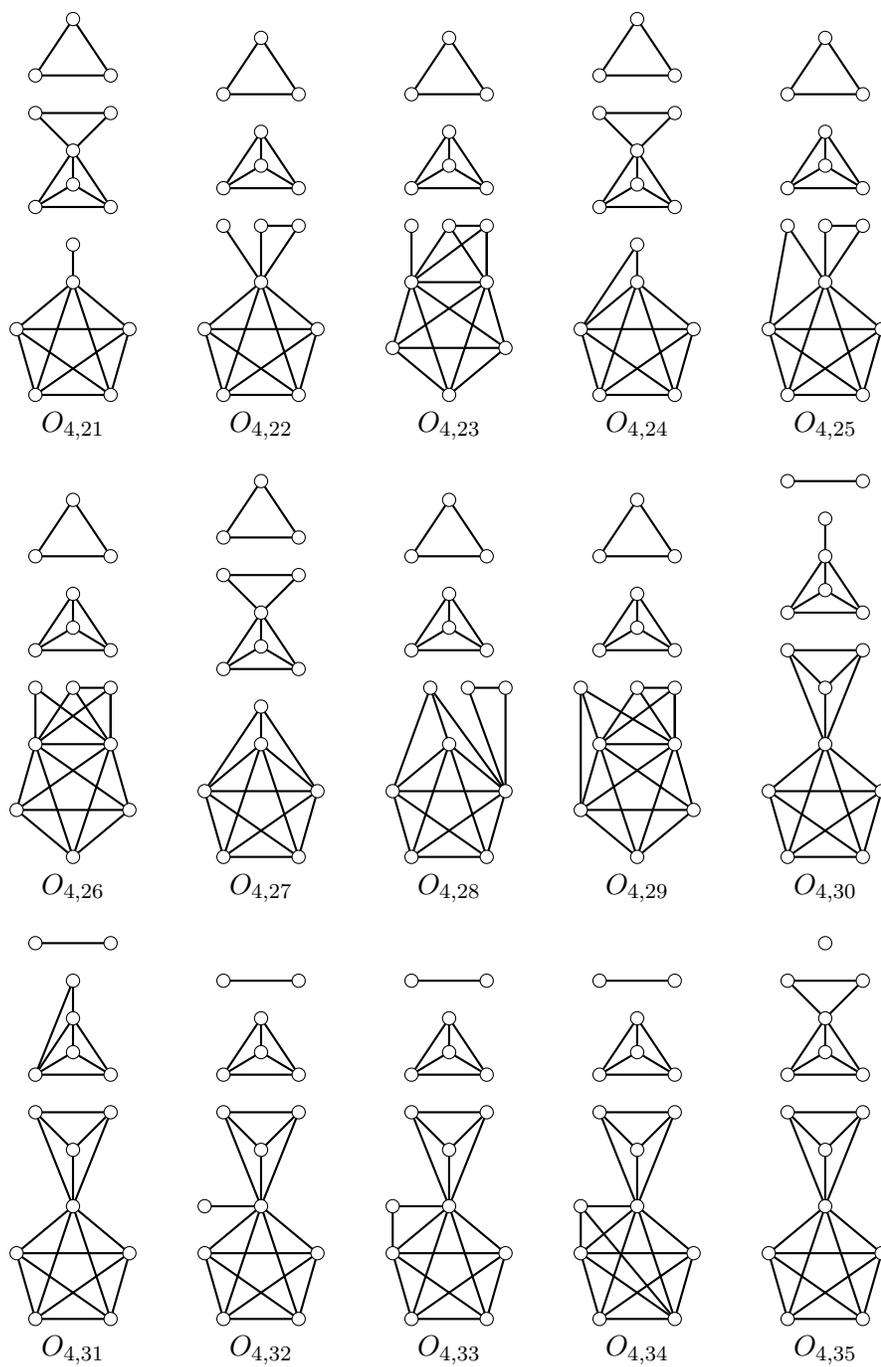
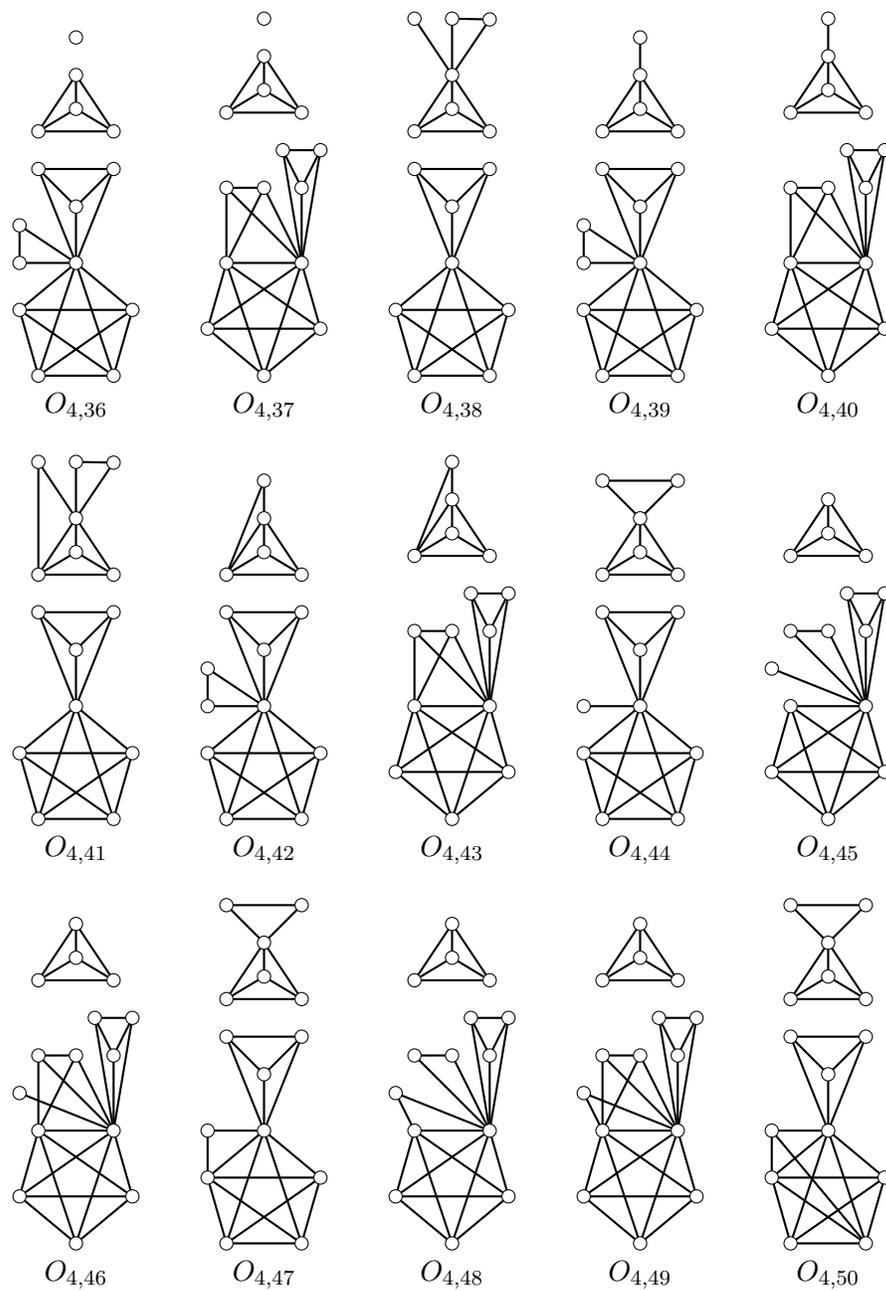


Figura C.3: Familia O_4 de obstrucciones. Parte 3.

Figura C.4: Familia O_4 de obstrucciones. Parte 4.

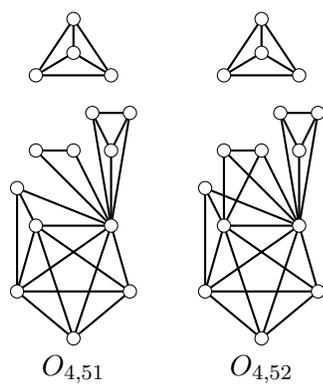


Figura C.5: Familia O_4 de obstrucciones. Parte 5.

Bibliografía

- [1] Peter Damaschke. Induced subgraphs and well-quasi-ordering. *Journal of Graph Theory*, 14:427–435, 1990.
- [2] Béla Bollobás. *Modern Graph Theory*. Springer Science+Business Media, Inc., 1998.
- [3] Sergey Kitaev and Vadim Lozin. *Words and Graphs*. Springer, 1 edition, 2015.
- [4] D.G. Corneil, H. Lerchs, and L. Stewart Burlingham. Complement reducible cographs. *Discrete Applied Mathematics*, 3:163–174, 1981.
- [5] D.G. Corneil, Y. Perl, and L.K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14:926–934, 1985.
- [6] T. Ekim, N.V.R. Mahadev, and D. de Werra. Polar cographs. *Discrete Applied Mathematics*, 156:1652–1660, 2008.
- [7] Fernando Esteban Cotreras-Mendoza and César Hernández Cruz. Minimal obstructions to $(s,1)$ -polarity in cographs. *Discrete Applied Mathematics*, 281:111–117, 2020.
- [8] Pavol Hell, César Hernández Cruz, and Cláudia Linhares Sales. Minimal obstructions to 2-polar cographs. *Discrete Applied Mathematics*, 261:219–228, 2019.
- [9] Stephan Olariu. Paw-free graphs. *Information Processing Letters*, 28:53–54, 1988.
- [10] Átila A. Jones, Fábio Protti, and Renata R. Del-Vecchio. Cograph generation with linear delay. *Theoretical Computer Science*, 713:1–10, 2018.