



UNIVERSIDAD NACIONAL DE ROSARIO

Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales
Departamento de Matemática

Tesina de Licenciatura en Matemática

Dominación Romana en Grafos

María Gracia Cornet

Director: Dr. Pablo Torres
Codirectora: Dra. Gabriela Argiroffo

26 de marzo de 2021
ROSARIO

Índice general

Agradecimientos	5
Introducción	7
1. Preliminares	11
1.1. Definiciones y Notación	11
1.2. El estado del arte	15
1.2.1. Presentación del problema	16
1.2.2. Algunos algoritmos eficientes para calcular γ_R	20
1.2.3. Complejidad computacional de problemas	25
2. Primeros Resultados	31
2.1. Primeras propiedades	31
2.2. γ_R bajo algunas operaciones	34
2.2.1. Borrado de un vértice	35
2.2.2. Generación de mellizos	38
2.2.3. Unión disjunta	41
2.2.4. Join	42
2.3. Un algoritmo para calcular γ_R	43
3. Dominación romana en grafos araña	47
3.1. Grafos araña	47
3.1.1. Arañas finas	48
3.1.2. Arañas gruesas	49
3.1.3. Arañas sin cabeza	50
3.2. Grafos quasi-araña	50
3.3. Grafos araña débilmente etiquetados	51
4. Dominación romana en grafos con pocos P_4's	61
4.1. Familias de grafos con pocos P_4 's	61
4.2. Un algoritmo para calcular γ_R en grafos con pocos P_4 's	63
4.3. Un resultado de NP-completitud	68
Conclusión	71
A. Un algoritmo usando Sage	73
A.1. Grafos P_4 -tidy	73
A.2. Grafos P_4 -sparse	75

Bibliografía

77

Agradecimientos

En primer lugar a mi familia. Por el apoyo incondicional. Por brindarme siempre las herramientas necesarias y la posibilidad de estudiar. Gracias!

A mis compañeras y compañeros a lo largo de toda la carrera. El aprendizaje siempre es más rico cuando es compartido. Gracias!

A las amigas y amigos que me regaló la facultad. Sin duda, de lo más lindo que hay. Por cada almuerzo en el *comedero*. Por cada lucha compartida. Por todo. Gracias!

A Pablo y Gabriela, mis directores. Por el acompañamiento y la guía. Especialmente en estos últimos meses. Gracias!

A Graciela, Paola y Patricia, evaluadoras de este trabajo. Por todos los comentarios, sugerencias y aportes. Por haberle dedicado tiempo a la corrección de mi tesina. Gracias!

A todas aquellas personas que lucharon (y siguen luchando) para que la Universidad sea Pública, Gratuita y de Calidad. Gracias!

Introducción

Durante la segunda guerra mundial, cuando el General Douglas MacArthur estaba conduciendo operaciones militares en el Pacífico, adoptó una estrategia de salto en islas que consistía en movilizar tropas de una isla a otra vecina, pero solo cuando se pudiera dejar atrás una guarnición lo suficientemente grande para mantener la primer isla segura. Un problema similar enfrentó el emperador romano Constantino en el siglo IV d.C., solo que su tarea era mantener la seguridad de todo el imperio. Él decidió lo que parece ser el primer uso registrado de la estrategia que MacArthur adoptó más tarde en el Pacífico.

Así comienza el artículo *Defend the Roman Empire!* [37], donde Ian Stewart nos propone el siguiente juego.

Consideremos una versión simplificada del Imperio Romano de la época de Constantino, dada por el siguiente mapa:



Figura 1: Mapa del Imperio Romano en la época del emperador Constantino

En esta gráfica se presentan ocho ciudades o regiones desde Asia Menor hasta Bretaña, y las rutas o caminos que las unen.

En el siglo III las fuerzas del Imperio dominaban la mayor parte de Europa, y las tropas disponibles llegaron a ser 50 legiones. Sin embargo, ya en el siglo IV, ese número se había

reducido a la mitad, a 25 legiones. Constantino las organizó en cuatro grupos, cada uno formado por al menos seis legiones. Estableció además algunas reglas para el movimiento de las tropas, buscando mejorar la seguridad.

Las reglas de Constantino son las siguientes:

- una ciudad está segura si en ella hay alguna tropa o bien una tropa puede moverse a ella en un solo paso desde una ciudad adyacente;
- deben haber al menos dos tropas ocupando una ciudad antes de sacar una de allí.

Imaginemos entonces que el emperador Constantino nos encarga la tarea de asignar estas tropas a las distintas ciudades del Imperio de manera que podamos garantizar que todas las ciudades estén *seguras*.

Pensemos en cada grupo de legiones como una ficha a ser colocada en uno de los círculos del tablero de la Figura 1.

Claramente, colocando una de estas fichas en cada ciudad, todas están seguras. Pero ante la reducción del ejército romano, nuestro objetivo es el de minimizar la cantidad de tropas asignadas.

Con la disposición de las ciudades en el mapa de la Figura 1, observamos que podemos garantizar la seguridad del Imperio ubicando dos fichas en Roma, una en Bretaña y una en Asia Menor. El resto de las ciudades tienen un camino directo que las conecta a Roma, con lo cual están seguras.

Ahora podemos preguntarnos: ¿es esta la cantidad mínima de fichas necesarias? ¿Podemos encontrar una asignación de tropas a las ciudades del imperio de manera que la cantidad de tropas a utilizar sea menor a cuatro? Si cambiamos el mapa, y queremos asignar tropas con las mismas reglas, ¿cómo hacemos? ¿Existirá algún algoritmo, alguna receta, que podamos usar para distribuir al ejército? Estas son algunas de las preguntas que vamos a intentar responder a lo largo de este trabajo.

El *Problema de Dominación Romana* fue formalizado por Cockayne et al. en [8]. Allí, se modeliza el problema del emperador Constantino utilizando la Teoría de Grafos. Representamos cada ciudad del imperio con un vértice y cada camino directo entre dos ciudades con una arista uniendo los vértices correspondientes. Asignar las tropas a las ciudades se traduce en dar una función que a cada vértice le asigna la cantidad de tropas apostadas en dicha ciudad. Este número debe estar en $\{0, 1, 2\}$ y se debe verificar que si a un vértice se le asigna el valor 0, entonces existe un vértice vecino con el valor 2. El *peso* de f es la suma de los valores de f en todos los vértices del grafo, es decir, la cantidad de tropas utilizadas por esa asignación. El *número de dominación romana* γ_R de un grafo es el mínimo peso entre todas las asignaciones posible.

Se sabe que el Problema de Dominación Romana es NP-completo en general. En esta tesina vamos a abordar el análisis de la complejidad del problema cuando nos restringimos a ciertas familias de grafos.

Entre las familias de grafos a estudiar, están aquellas que se pueden definir por tener una cantidad restringida de P_4 's en un sentido local. Por ejemplo, los *cografos* son aquellos grafos que no tienen ningún P_4 como subgrafo inducido. Un grafo es *P_4 -reducible* si todo vértice del grafo pertenece a lo sumo a un P_4 inducido. Un grafo es *P_4 -sparse* si todo conjunto de 5 vértices contiene a lo sumo un P_4 inducido. Así, hay varias familias que se definen de esta manera. Estas clases de grafos tienen propiedades estructurales que permiten, para ciertos problemas que se conocen que son NP-completos en general, obtener algoritmos que los resuelven en tiempo polinomial si restringimos el problema a algunas de ellas [3].

Una estrategia de abordaje para el problema es descomponer el grafo en subgrafos más pequeños, mediante ciertas operaciones. Si conocemos el comportamiento del parámetro γ_R bajo estas operaciones y podemos calcular el valor del parámetro para estos subgrafos más pequeños, y todo esto lo podemos hacer en tiempo polinomial, entonces podemos recuperar el valor del número de dominación romana en el grafo original.

Muchas de estas familias *con pocos* P_4 se caracterizan de alguna manera con este tipo de descomposiciones. Esto nos permite abordarlos siguiendo la estrategia antes mencionada.

En el primer Capítulo vamos a introducir algunas definiciones y nociones generales que nos van a servir de base para definir formalmente todas las herramientas que vamos a utilizar. A su vez haremos un resumen de los principales resultados que conocemos de la literatura y que tenemos como punto de partida en el tema.

En el Capítulo 2 presentaremos los primeros resultados obtenidos, incluyendo el estudio del comportamiento del parámetro γ_R bajo ciertas operaciones de grafos. Además, usando esos resultados, damos un algoritmo para calcular el número de dominación romana basándonos en la descomposición modular.

En el Capítulo 3 calcularemos el valor de γ_R para ciertas familias de *grafos araña*. Se sabe que el Problema de Dominación Romana sigue siendo NP-completo incluso si nos restringimos a la familia de los grafos split [36]. Sin embargo, probaremos que para los grafos araña sin cabeza, una subclase de los split, el problema es “fácil de resolver”. Extenderemos los resultados a grafos araña más generales que también son importantes en la descomposición de otras familias con pocos P_4 's.

Finalmente en el Capítulo 4 analizaremos las familias con pocos P_4 's dando un algoritmo específico para muchas de ellas. Además, daremos un resultado de NP-completitud para una de esas familias

Parte de los resultados presentados en esta tesina fueron presentados en

- XIII Jornada de Ciencia y Tecnología | Divulgación de la Producción Científica y Tecnológica de la Universidad Nacional de Rosario, exposición del Póster “*Dominación Romana en Grafos*”. Coautores: Argiroffo, Gabriela y Torres, Pablo. (Octubre 2019)
- LXIX Reunión de Comunicaciones Científicas de la Reunión Anual Virtual de la Unión Matemática Argentina (virtUMA 2020), presentación de la Comunicación “*Dominación Romana en Grafos*”. Coautores: Argiroffo, Gabriela y Torres, Pablo. (Septiembre 2020)

Capítulo 1

Preliminares

En este Capítulo presentamos las definiciones que usaremos a lo largo de esta tesina. Introducimos el Problema de Dominación Romana, y resumimos los resultados principales que usaremos como punto de partida para este trabajo.

1.1. Definiciones y Notación

Grafos

Un *grafo* G es un par $G = (V, E)$ que consiste de un conjunto finito de *vértices* V y un conjunto E de pares no ordenados de vértices llamados *aristas*. En caso de que V y E no se expliciten, notaremos con $V(G)$ y $E(G)$ al conjunto de vértices y de aristas de un grafo G respectivamente. Para un grafo $G = (V, E)$, el número $n = |V|$ es el *orden* de G . Todos los grafos de este trabajo son *simples*, esto es, las aristas aparecen a lo sumo una vez en el conjunto de aristas y no hay aristas que van de un vértice a sí mismo.

Si $e = \{u, v\}$ es una arista de G , diremos que u y v son los *extremos* de e y que u y v son *adyacentes* o *vecinos*. A la arista $\{u, v\}$ la notamos también uv .

Representamos al grafo $G = (V, E)$ con un diagrama donde a cada vértice en V le corresponde un nodo o punto grueso, y cuando dos vértices son adyacentes, unimos los nodos correspondientes con una línea que representa la arista.

Por ejemplo, el grafo G con conjunto de vértices $\{1, 2, 3, 4\}$ y conjunto de aristas $\{12, 13, 14, 23\}$, es un grafo de orden 4 y lo podemos representar con el siguiente diagrama:

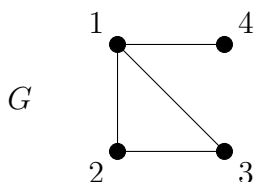


Figura 1.1: Ejemplo de representación de un grafo

Para un grafo $G = (V, E)$, el grafo *complemento* de G , denotado por \overline{G} , es el grafo cuyo conjunto de vértices es $V(\overline{G}) = V$ y $uv \in E(\overline{G})$ si y solo si $uv \notin E$.

Dados dos grafos G y H , una asignación $f : V(G) \rightarrow V(H)$ es un *isomorfismo de grafos* si f es biyectiva y para todos los vértices $u, v \in V(G)$ se cumple que $f(u)f(v) \in E(H)$ si y solo si $uv \in E(G)$. Si existe un isomorfismo de grafos entre G y H , diremos que éstos son *isomorfos*, y lo notamos $G \sim H$. Identificaremos los grafos isomorfos como iguales.

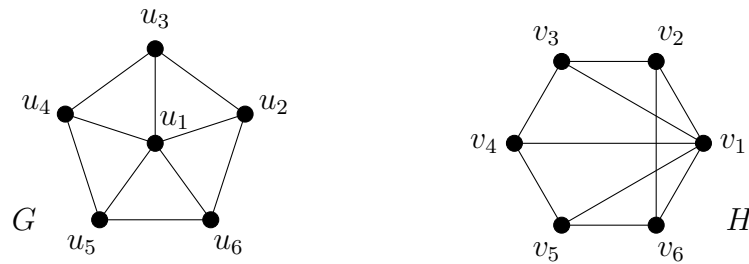


Figura 1.2: Ejemplo de un par de grafos isomorfos

Un grafo $H = (W, F)$ es un *subgrafo* del grafo $G = (V, E)$ si $W \subseteq V$, $F \subseteq E$. Si $G = (V, E)$ y $W \subseteq V$, el subgrafo de G *inducido* por W , que denotamos con $G[W]$ es el subgrafo con conjunto de vértices W y conjunto de aristas $\{uv : u, v \in W, uv \in E\}$.

En un grafo $G = (V, E)$, el *grado* de un vértice v , denotado por $gr(v)$ (o $gr_G(v)$ si es necesario especificar el grafo G), es el número de vértices adyacentes a v . El grado mínimo y máximo de G se denotan por $\delta(G) = \min\{gr(v) : v \in V\}$ y $\Delta(G) = \max\{gr(v) : v \in V\}$, respectivamente. Podemos distinguir algunos vértices según su grado, un vértice v se dice *aislado* si $gr(v) = 0$, *pendiente* si $gr(v) = 1$, *universal* si $gr(v) = |V| - 1$ y *quasi-universal* si $gr(v) = |V| - 2$. Para cualquier vértice $v \in V$, la *vecindad abierta de v* es el conjunto $N(v) = N_G(v) = \{u \in V : uv \in E\}$ y la *vecindad cerrada de v* es el conjunto $N[v] = N_G[v] = N(v) \cup \{v\}$. La vecindad abierta (resp. cerrada) de un conjunto $S \subseteq V$ es $N(S) = \cup_{v \in S} N(v)$ (resp. $N[S] = \cup_{v \in S} N[v] = N(S) \cup S$).

Para un conjunto de vértices S y un vértice $v \in S$, un vértice u es un *vecino privado* de v con respecto a S (y escribimos u es un S -*pn* de v por sus siglas en inglés *private neighbor*) si $u \in N[v] - N[S - \{v\}]$. Es decir, si $u = v$ y v no es adyacente a ningún vértice en S , o bien si v es el único vecino de u en S . El conjunto $pn(v, S) = N[v] - N[S - \{v\}]$ de todos los S -*pn* de v es la *vecindad privada* de v con respecto a S . Un conjunto S es *irredundante* si para cada $v \in S$, $pn(v, S) \neq \emptyset$.

En un grafo, un *camino* de longitud k de un vértice v_0 a un vértice v_k es una colección de aristas $P = \{v_0v_1, v_1v_2, \dots, v_{k-1}v_k\}$. Por simplicidad en la notación, representaremos un camino usando solo sus vértices, $P = v_0v_1\dots v_{k-1}v_k$. Diremos que P es un v_0v_k -*camino*. Un camino es *simple* si todos sus vértices son distintos, y un camino es *cerrado* si $v_0 = v_k$. Un *ciclo* de longitud k es un camino cerrado $C = v_0v_1\dots v_k$ donde v_0, v_1, \dots, v_{k-1} son distintos.

Llamamos P_n al camino de n vértices, $v_1v_2\dots v_n$.

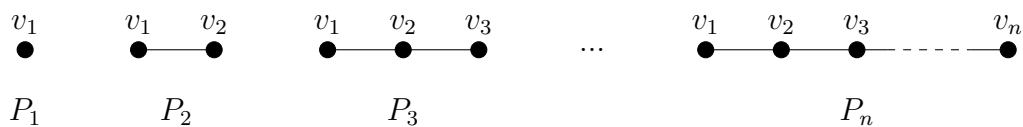


Figura 1.3: Ejemplos de Caminos

Llamamos C_n al ciclo de n vértices (para $n \geq 3$), $v_1v_2\dots v_nv_1$.

Dado un grafo $G = (V, E)$, decimos que un subconjunto de vértices $Q \subseteq V$ es un *conjunto completo* si para todo par de vértices $u, v \in Q$ se verifica $uv \in E$. Una *clique* es un conjunto completo maximal. Llamamos *grafo completo* de n vértices, y notamos K_n , al grafo de orden n en el cual todo par de vértices son adyacentes. Observemos que en K_n todo vértice es universal.

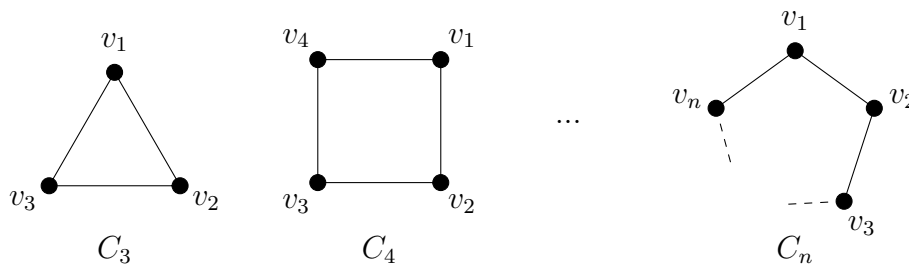


Figura 1.4: Ejemplos de Ciclos

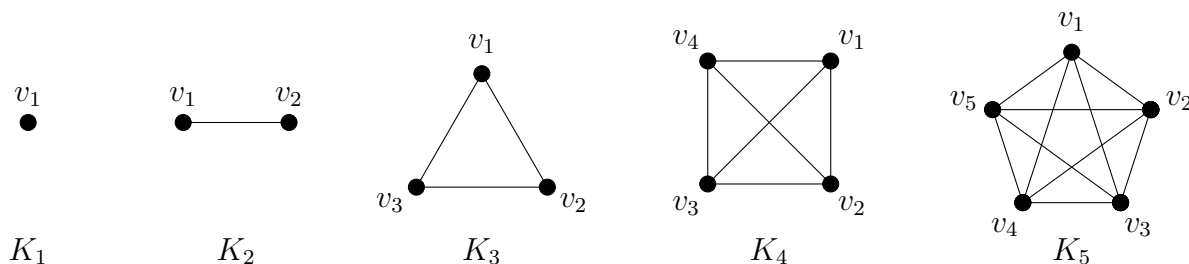


Figura 1.5: Ejemplos de grafos completos

Dado un grafo $G = (V, E)$, decimos que un subconjunto de vértices $S \subseteq V$ es un *conjunto estable* o *independiente* si para todo par de vértices $u, v \in S$ se verifica $uv \notin E$. Llamamos *grafo trivial* de n vértices, y notamos S_n , al grafo complemento de K_n .

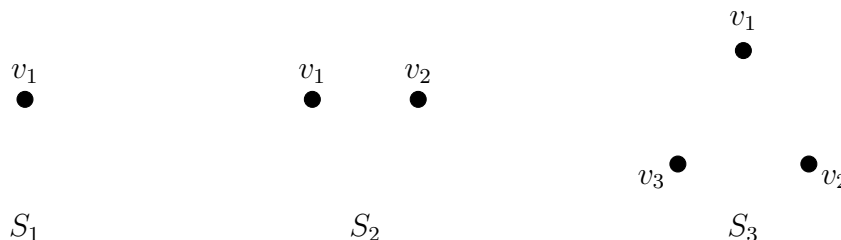


Figura 1.6: Ejemplos de grafos triviales

Un grafo G es *conexo* si para todo par de vértices distintos $u, v \in V$ existe un uv -camino en G . Diremos que un grafo es *disconexo* si no es conexo. Llamamos *componente conexa* de G a los subgrafos conexos maximales de G .

Si un grafo G es conexo y \bar{G} también lo es, diremos que G es un grafo *modular*.

Un grafo $G = (V, E)$ se dice P_4 -conexo, o p -conexo, si para toda partición de V en dos conjuntos disjuntos $U \cup W$ existe un P_4 con vértices en U y en V . Las *componentes p -conexas* de un grafo son los subgrafos inducidos maximales que son p -conexos.

Clases de grafos

Decimos que un grafo $G = (V, E)$ es *bipartito* si existe una partición de V en dos conjuntos estables U_1 y U_2 . Decimos que un grafo bipartito de partición $V = U_1 \cup U_2$ es *bipartito completo*, y notamos $K_{|U_1|, |U_2|}$ si $uv \in E(G)$ para todo $u \in U_1$ y $v \in U_2$. Llamamos *estrella de k puntas* al grafo $K_{1,k}$, con $k \in \mathbb{N}$.

Un grafo G es un *árbol* si es conexo y no contiene ciclos. En un árbol G , los vértices de grado uno se llaman *hojas* de G . Todos los vértices de G que no son hojas se llaman *vértices internos* de G .

Un grafo G es *split* si su conjunto de vértices se puede particionar en dos conjuntos Q y S tales que Q es un conjunto completo maximal y S es un conjunto independiente. En este caso decimos que G es un grafo split con partición (Q, S) .

Algunas familias de grafos se pueden definir a partir de sus subgrafos prohibidos. Si \mathcal{F} es un conjunto de grafos, un grafo G se dice \mathcal{F} -free si no posee ningún grafo de \mathcal{F} como subgrafo inducido.

Los grafos que son P_4 -free se llaman *cografos*. Los grafos split pueden ser definidos por sus subgrafos prohibidos, ya que G es un grafo split si y solo si es $\{C_4, \overline{C_4}, C_5\}$ -free [15].

Operaciones en grafos

Sea $G = (V, E)$ un grafo y $v \in V$. El grafo obtenido por el *borrado* del vértice v es el subgrafo inducido $G[V - \{v\}]$, que notaremos por $G - v$.

Dado $G' = (V', E')$ tal que $V \cap V' = \emptyset$, el grafo obtenido al *reemplazar* v por G' es el grafo cuyo conjunto de vértices es $(V - \{v\}) \cup V'$ y cuyas aristas son $E(G - v) \cup E'$, donde E' es el conjunto de todas las aristas que conectan cada vértice de V' con cada vértice de $N(v)$.

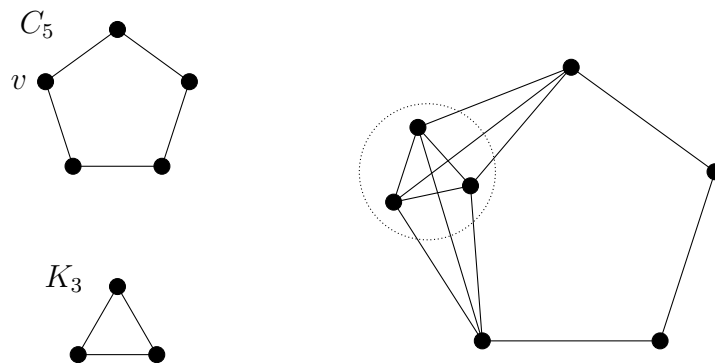


Figura 1.7: Grafo obtenido al reemplazar un vértice de C_5 por K_3

El grafo obtenido al agregar a G un *mellizo verdadero* de v , es el grafo obtenido de reemplazar v por K_2 . De manera similar, el grafo obtenido al agregar a G un *mellizo falso* de v , es el grafo obtenido de reemplazar v por S_2 .

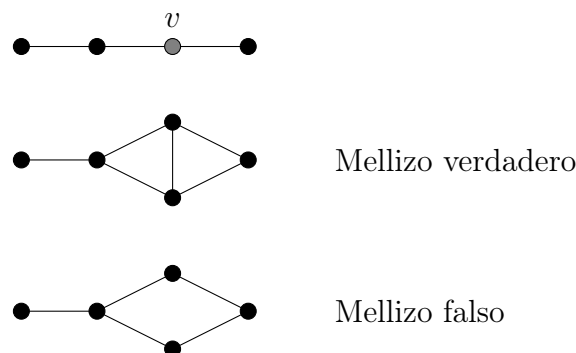


Figura 1.8: Grafos obtenidos al agregarle un mellizo a un vértice de P_4

Dados dos grafos $G_1 = (V^1, E^1)$ y $G_2 = (V^2, E^2)$ con $V^1 \cap V^2 = \emptyset$, el grafo *unión disjunta* de G_1 y G_2 , denotado por $G_1 \cup G_2$, es el grafo con conjunto de vértices $V^1 \cup V^2$ y conjunto de aristas $E^1 \cup E^2$. El *grafo join* de G_1 y G_2 , denotada por $G_1 \vee G_2$, es el grafo con conjunto de vértices $V^1 \cup V^2$ y conjunto de aristas $E^1 \cup E^2 \cup \{vw : v \in V^1, w \in V^2\}$. Resulta inmediato observar que la operación join (resp. unión disjunta) son asociativas, lo que permite definir recursivamente la join (resp. unión disjunta) de un número finito de grafos G_1, \dots, G_k .

Dado un grafo G y un número natural n , el grafo nG es el grafo obtenido por la operación unión disjunta de n copias del grafo G . Esto es

$$nG = \underbrace{G \cup \dots \cup G}_{n \text{ veces}}$$

1.2. El estado del arte

En esta sección presentamos los resultados preliminares sobre el Problema de Dominación Romana, que utilizaremos y en los que nos basaremos a lo largo de esta tesina.

El Problema de Dominación Romana tal como lo estudiaremos aparece por primera vez en [8], donde Cockayne et al. (2004), inspirados en el artículo de Stewart en la revista *Scientific American* [37], presentan un modelo para el problema de asignar tropas a las ciudades del Imperio Romano utilizando Teoría de Grafos. El mismo consiste en una variación del Problema de Dominación clásico.

Una variante del Problema de Dominación clásico

Dado un grafo $G = (V, E)$, un *conjunto dominante* de G es un subconjunto de vértices $D \subseteq V$ tal que todo vértice de $V - D$ es adyacente a algún vértice en D . El *número de dominación* de un grafo G , que notamos $\gamma(G)$, es el mínimo cardinal de un conjunto dominante de G . Una *función dominante* f sobre un grafo G es una función $f : V \rightarrow \{0, 1\}$ tal que si $f(v) = 0$, existe $w \in N(v)$ tal que $f(w) = 1$. El valor $\gamma(G)$ es el mínimo peso entre todas las funciones dominantes sobre el grafo G .

El *Problema de Dominación* clásico es el problema de optimización que consiste en hallar el valor del número de dominación de un grafo $G = (V, E)$, $\gamma(G)$.

El Problema de Dominación aparece en la ubicación de instalaciones o de servicios, donde por ejemplo se busca minimizar la cantidad de instalaciones necesarias de algún servicio para que todos puedan acceder a las mismas. También aparecen conceptos de dominación en problemas que consisten en encontrar algún conjunto de representantes, en el monitoreo de comunicación o las redes eléctricas, y en la toma de medidas de tierras, por ejemplo, buscando minimizar el número de lugares en los cuales ubicar un topógrafo en pos de tomar medidas de altura de una región entera. Otro campo interesante de aplicación son las redes sociales. Si consideramos que los vértices de un grafo son personas, y conectamos dos vértices siempre que esas personas sean amigas, entonces un conjunto dominante es un conjunto de personas tal que cada persona tiene un amigo o amiga en ese conjunto.

Numerosas aplicaciones pueden ser modeladas como problemas de dominación en grafos. La mayoría de estos problemas son NP-difíciles, desde el punto de vista de la complejidad computacional, sin embargo, el enfoque en términos de grafos ha sido de gran importancia para la resolución de problemas del área como así también, ha contribuido al diseño y mejora de algoritmos específicos de resolución de algunas instancias.

Habitualmente, cada aplicación específica impone restricciones adicionales a los conjuntos dominantes, surgiendo de esta manera diferentes variaciones del problema. Una de estas es el *Problema de Dominación Romana*.

1.2.1. Presentación del problema

Formalmente, dado un grafo $G = (V, E)$, una *función de dominación romana* (FDR) sobre G es una función $f : V \rightarrow \{0, 1, 2\}$ que satisface la condición que cada vértice u para el cual $f(u) = 0$ es adyacente al menos a un vértice v para el cual $f(v) = 2$.

Existe una correspondencia uno a uno entre las funciones $f : V \rightarrow \{0, 1, 2\}$ y las particiones ordenadas $V = (V_0, V_1, V_2)$, donde $V_i = \{u \in V : f(u) = i\}$. Luego, a una FDR f la podemos escribir como $f = (V_0, V_1, V_2)$. Si definimos $n_i = |V_i|$, tenemos que $n_0 + n_1 + n_2 = n = |V|$.

Dado un conjunto $S \subseteq V$, $f(S) = \sum_{u \in S} f(u)$. El *peso* de una FDR es el valor $f(V) = \sum_{u \in V} f(u) = n_1 + 2n_2$. Y dado un grafo G , el mínimo peso que puede asumir una FDR sobre G se llama el *número de dominación romana* de G , al que notaremos $\gamma_R(G)$.

Si f es una FDR sobre un grafo G de peso $\gamma_R(G)$, entonces diremos que f es una γ_R -función.

El *Problema de Dominación Romana* es un problema de optimización que consiste en, dado un grafo $G = (V, E)$, hallar el valor $\gamma_R(G)$.

Primeras propiedades

Vamos a enunciar algunas propiedades del número de dominación romana de un grafo. Primero veremos cómo se relacionan el número de dominación romana con el número de dominación de un grafo.

Proposición 1 ([8]). *Para todo grafo G , vale $\gamma(G) \leq \gamma_R(G) \leq 2\gamma(G)$.*

Demostración. Sean $G = (V, E)$ un grafo y $f = (V_0, V_1, V_2)$ una γ_R -función sobre G . Sea $D = V_1 \cup V_2$. Probaremos que D es un conjunto dominante de G . En efecto, sea $u \in V$. Si $u \in V - D = V_0$, entonces existe $v \in V_2 \cap N(u) \subseteq D \cap N[u]$. Y resulta $N[u] \cap D \neq \emptyset$. Por lo tanto tenemos que D es un conjunto dominante, y entonces

$$\gamma(G) \leq |D| = |V_1| + |V_2| \leq |V_1| + 2|V_2| = \gamma_R(G) \quad (1.2.1)$$

Por otro lado, sea D un conjunto dominante mínimo de G . Definimos $\hat{f} = (\hat{V}_0, \hat{V}_1, \hat{V}_2)$ con $\hat{V}_0 = V - D$, $\hat{V}_1 = \emptyset$, y $\hat{V}_2 = D$. Veamos que \hat{f} resulta una FDR. En efecto, sea $u \in \hat{V}_0 = V - D$, entonces como $N(u) \cap D \neq \emptyset$, existe un vértice $v \in N(u) \cap D = N(u) \cap \hat{V}_2$. Por lo tanto tenemos que \hat{f} es una FDR, y entonces

$$\gamma_R(G) \leq \hat{f}(V) = 2|D| = 2\gamma(G) \quad (1.2.2)$$

De (1.2.1) y (1.2.2) tenemos que

$$\gamma(G) \leq \gamma_R(G) \leq 2\gamma(G)$$

como queríamos demostrar. \square

Las cotas dadas por la Proposición 1 son ajustadas. Es decir, para ambas desigualdades existen grafos en los cuales éstas se dan por igualdad.

Para caracterizar los grafos donde se alcanza la igualdad $\gamma_R(G) = \gamma(G)$ tenemos el siguiente resultado:

Proposición 2 ([8]). *Sea G un grafo de orden n . Entonces $\gamma(G) = \gamma_R(G)$ si y solo si $G = S_n$.*

Aquellos grafos para los cuales se verifica $\gamma_R(G) = 2\gamma(G)$ reciben el nombre de grafos romanos y hablamos de ellos más adelante en esta misma sección.

En [8] se comienzan a estudiar propiedades que debe cumplir una γ_R -función sobre un grafo G . Algunas de ellas son:

Proposición 3 ([8]). *Sea $f = (V_0, V_1, V_2)$ una γ_R -función sobre un grafo $G = (V, E)$. Entonces*

(a) $G[V_1]$, el subgrafo inducido por V_1 , tiene grado máximo 1.

(b) Ninguna arista de G conecta V_1 y V_2 .

(c) Cada vértice de V_0 es adyacente a lo sumo a dos vértices de V_1 .

(d) V_2 es un conjunto dominante de $G[V_0 \cup V_2]$.

(e) Sea $H = G[V_0 \cup V_2]$. Entonces cada vértice $v \in V_2$ tiene a lo sumo dos V_2 -pn's en H .

Demostración. Sea G un grafo y $f = (V_0, V_1, V_2)$ una γ_R -función sobre G

(a) Queremos ver que $\Delta(G[V_1]) \leq 1$. Supongamos, por el contrario, que existe un vértice $u \in V_1$ con $gr_{G[V_1]}(u) = k \geq 2$. Sean entonces v_1, \dots, v_k los k vecinos de $u \in V_1$ en V_1 .

Definimos $\hat{f} : V \rightarrow \{0, 1, 2\}$ como sigue

$$\hat{f}(v) = \begin{cases} 2 & \text{si } v = u \\ 0 & \text{si } v = v_i \text{ para algún } i = 1, \dots, k \\ f(v) & \text{en otro caso} \end{cases}$$

Observemos que \hat{f} es una FDR. En efecto, sea w tal que $\hat{f}(w) = 0$. Entonces $w = v_i$ para algún i o bien $f(w) = 0$

- Si $w = v_i$ para algún $i = 1, \dots, k$, entonces como $v_i \in N(u)$ y $\hat{f}(u) = 2$. Entonces existe un vecino u de w para el cual $\hat{f}(u) = 2$.
- Si $f(w) = 0$, $w \in V_0$, y existe un vértice $v \in V_2$ tal que $f(v) = 2$. Como $u, v_1, \dots, v_k \in V_1$, resulta que $\hat{f}(v) = f(v) = 2$. Entonces existe un vecino v de w para el cual $\hat{f}(v) = 2$.

Por lo tanto, \hat{f} es una FDR sobre G y entonces

$$\begin{aligned} \gamma_R(G) &\leq \hat{f}(V) = f(V - \{u, v_1, \dots, v_k\}) + 2 = \\ &= f(V) - |\{u, v_1, \dots, v_k\}| + 2 = f(V) \underbrace{-k + 1}_{<0} < f(V) = \gamma_R(G) \end{aligned}$$

Y llegamos al absurdo $\gamma_R(G) < \gamma_R(G)$, que surge de suponer que $\Delta(G[V_1]) \geq 2$. Por lo tanto concluimos que $\Delta(G[V_1]) \leq 1$, como queríamos.

- (b) Supongamos que existe una arista uv , con $u \in V_1$ y $v \in V_2$. Sea $\hat{f} : V \rightarrow \{0, 1, 2\}$ dada por

$$\hat{f}(v) = \begin{cases} 0 & \text{si } v = u \\ f(v) & \text{en otro caso} \end{cases}$$

\hat{f} es una FDR con $\hat{f}(V) = f(V) - 1$, lo que contradice que f es una γ_R -función. Luego, no puede existir ninguna arista que una vértices de V_1 y V_2 .

- (c) Sea $u \in V_0$, y $A = \{v \in V_1 : uv \in E\}$. Supongamos que $|A| > 2$. Sea $\hat{f} : V \rightarrow \{0, 1, 2\}$ dada por

$$\hat{f}(v) = \begin{cases} 2 & \text{si } v = u \\ 0 & \text{si } v \in A \\ f(v) & \text{en otro caso} \end{cases}$$

\hat{f} así definida es una FDR sobre G y tenemos

$$\gamma_R(G) \leq \hat{f}(V) = \hat{f}(V - A) + \hat{f}(A) = f(V - A) + 2 = f(V) - \underbrace{|A|}_{<0} + 2 < f(V) = \gamma_R(G)$$

Y llegamos al absurdo $\gamma_R(G) < \gamma_R(G)$, que surge de suponer que $|A| > 2$. Por lo tanto concluimos que cada vértice de V_0 es asyacente a lo sumo a dos vértices en V_1 .

- (d) Sea $H = G[V_0 \cup V_2]$. Queremos ver que V_2 es un conjunto dominante de H . Sea $u \in V(H) - V_2 = V_0$. Entonces, como f es una FDR, existe $w \in V_2$ que es vecino de u en G . Entonces, es vecino de u en H . Por lo tanto V_2 es un conjunto dominante en H .
- (e) Sea $H = G[V_0 \cup V_2]$ y $v \in V_2$. Queremos ver que la cantidad de vecinos privados de v respecto de V_2 en H es a lo sumo dos. Sea

$$k = |pn_H(v, V_2)| = |u \in V_0 \cup V_2 : u \in N_H[v] - N_H[V_2 - \{v\}]|$$

Vamos a ver que si $k < 2$, entonces podemos encontrar una γ_R -función sobre G de peso menor a $f(V)$, lo cual es una contradicción.

En efecto, si $k = 0$, entonces v no tiene vecinos privados con respecto a V_2 en H . Es decir, si $u \in V_0 \cap N(v)$, entonces existe otro vértice $w \neq v$ que es vecino de u y que está en V_2 . Entonces \hat{f} definida por

$$\hat{f}(w) = \begin{cases} 1 & \text{si } w = v \\ f(w) & \text{en otro caso} \end{cases}$$

\hat{f} así definida es una FDR con $\hat{f}(V) = f(V) - 1$.

Si $k = 1$, sea u el único vecino privado de v respecto de V_2 en H . Es decir $pn_H(v, V_2) = \{u\}$.

- Si $u \in V_0$. Entonces como $v \notin pn_H(v, V_2)$, existe $v' \in V_2$ tal que $vv' \in E$. Luego definimos \hat{f} como

$$\hat{f}(w) = \begin{cases} 0 & \text{si } w = v \\ 1 & \text{si } w = u \\ f(w) & \text{en otro caso} \end{cases}$$

- Si $u \in V_2 - \{v\}$. Entonces definimos \hat{f} como

$$\hat{f}(w) = \begin{cases} 0 & \text{si } w = v \\ f(w) & \text{en otro caso} \end{cases}$$

- Si $u = v$. Entonces definimos \hat{f} como

$$\hat{f}(w) = \begin{cases} 1 & \text{si } w = v \\ f(w) & \text{en otro caso} \end{cases}$$

En cualquiera de los tres casos, \hat{f} es una FDR sobre G de peso menor a $f(V) = \gamma_R(G)$, lo cual es una contradicción. Concluimos entonces que $k \geq 2$. Es decir, cada vértice $v \in V_2$ tiene al menos dos vecinos privados respecto a V_2 en H .

□

Observación 1. Si G es un grafo y $f = (V_0, V_1, V_2)$ es una FDR sobre G , entonces todos los vértices aislados de G están en V_1 . Entonces, nos basta con estudiar el problema sobre grafos sin vértices aislados.

Proposición 4 ([8]). Si G es un grafo de orden $n \geq 2$ que contiene un vértice universal, entonces $\gamma_R(G) = 2$.

Demostración. Si $V(G) \geq 2$ y G tiene un vértice universal, digamos \hat{v} , tenemos $V = N[\hat{v}]$.

Como $V(G) \geq 2$, resulta

$$\gamma_R(G) \geq 2 \tag{1.2.3}$$

Además $\hat{f} = (\hat{V}_0, \hat{V}_1, \hat{V}_2)$ dada por

$$\hat{V}_2 = \{\hat{v}\}, \quad \hat{V}_1 = \emptyset, \quad \hat{V}_0 = V - \{\hat{v}\} = N(\hat{v})$$

es una FDR de G . Luego,

$$\gamma_R(G) \leq 2 \tag{1.2.4}$$

Luego, de (1.2.3) y (1.2.4), tenemos que

$$\gamma_R(G) = 2$$

□

Ejemplo 1. Para cualquier $k \in \mathbb{N}$, el grafo estrella $K_{1,k}$ y el grafo K_k tienen un vértice universal, y por lo tanto $\gamma_R(K_{1,k}) = \gamma_R(K_k) = 2$.

El número de dominación romana en familias especiales de grafos

En [8] se calcula el valor del número de dominación romana para grafos pertenecientes a algunas familias específicas, así como caminos, ciclos, grafos n -partitos, entre otras.

Proposición 5 ([8]). Si G es un camino P_n o un ciclo C_n , vale

$$\gamma_R(P_n) = \gamma_R(C_n) = \left\lceil \frac{2n}{3} \right\rceil$$

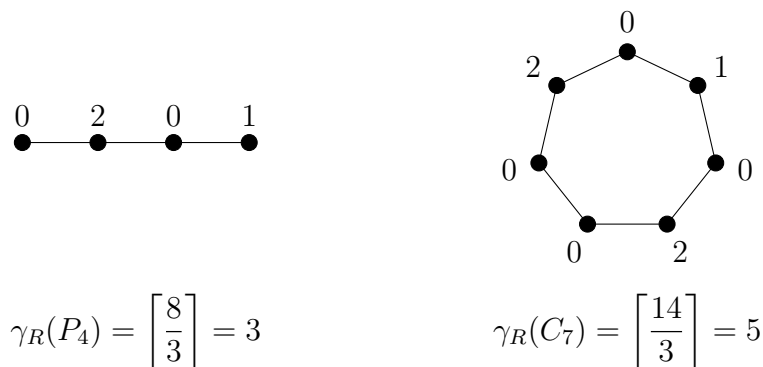


Figura 1.9: Ejemplos de γ_R -función en caminos y ciclos

Proposición 6 ([8]). Sea $G = K_{m_1, \dots, m_n}$ un grafo n -partito completo con $m_1 \leq m_2 \leq \dots \leq m_n$. Entonces

$$\gamma_R(G) = \begin{cases} 4 & \text{si } m_1 \geq 3 \\ 3 & \text{si } m_1 = 2 \\ 2 & \text{si } m_1 \leq 1 \end{cases}$$

Proposición 7 ([8]). Si G es un grafo de orden n sin vértices aislados, entonces $\gamma_R(G) = n$ si y solo si n es par y $G = \binom{n}{2} K_2$.

Grafos Romanos

Los grafos G para los cuales $\gamma_R(G) = 2\gamma(G)$ reciben el nombre de *grafos romanos*. Estos son grafos para los cuales existe una γ_R -función donde V_2 es un conjunto dominante de G de cardinal mínimo, como establece el siguiente resultado.

Proposición 8 ([8]). Un grafo G es romano si y solo si existe una γ_R -función $f = (V_0, V_1, V_2)$ con $n_1 = |V_1| = 0$.

En la Proposición 4 se presenta una clase de grafos romanos, aquellos que tienen un vértice universal. De la Proposición 5 se pueden deducir los caminos y ciclos que son grafos romanos: P_{3k} , C_{3k} , P_{3k+2} , C_{3k+2} . Mientras que de la Proposición 6 se pueden identificar aquellos grafos bipartitos completos que son grafos romanos, i.e., $G = K_{m,n}$ con $\min\{m, n\} \neq 2$ donde o bien $\gamma(G) = 1$ y $\gamma_R(G) = 2$, o $\gamma(G) = 2$ y $\gamma_R(G) = 4$. Por su parte, Henning en [22] da una caracterización constructiva de los árboles romanos (grafos romanos que son a su vez árboles).

Un problema que aún permanece abierto es el de caracterizar la familia de los grafos romanos.

1.2.2. Algunos algoritmos eficientes para calcular γ_R

Liedloff et al. [30] proponen algoritmos eficientes para resolver el Problema de Dominación Romana sobre los grafos de intervalos, grafos AT -free y los cografos. En el caso de las dos primeras familias, el algoritmo se deriva del estudio de las propiedades de las funciones óptimas en estas familias. En el caso de los cografos, se usa la descomposición modular de un grafo. A continuación, definimos estas clases de grafos.

Un grafo $G = (V, E)$ es un *grafo de intervalos* si existe un conjunto $F = \{I_v : v \in V\}$ de intervalos de la recta real tales que $I_u \cap I_v \neq \emptyset$ si y solo si $uv \in E$.

Tres vértices x, y, z de un grafo $G = (V, E)$ forman un *asteroide triple* (AT) si tomando dos cualesquiera existe un camino que los une sin pasar por la vecindad del tercero. Un grafo se dice *AT-free* si no contiene ningún AT.

La familia de los *cografos* puede definirse de manera recursiva de la siguiente manera:

1. K_1 es un cografo.
2. Si G_1 y G_2 son cografos, entonces también lo son $G_1 \cup G_2$ y $G_1 \vee G_2$.
3. Todo cografo puede ser construido a partir de copias de K_1 aplicando las operaciones en (2) un número finito de veces.

Todo cografo puede ser representado por su árbol de descomposición modular $T[G]$ [19]. En $T[G]$, cada hoja representa un vértice de G y cada vértice interno representa o bien una operación *unión disjunta* o bien una operación *join*. Para dos vértices u y v tales que uv es una arista de G , el ancestro común más cercano de u y v en $T[G]$ es un vértice *join*. Si suponemos que G es conexo, la raíz de $T[G]$ es un vértice *join*. Más aún, podemos suponer que $T[G]$ es un árbol binario.

El siguiente resultado nos da el número de dominación romana para un cografo conexo. En caso que G sea desconexo, $\gamma_R(G)$ es igual a la suma de los números de dominación romana de sus componentes conexas.

Teorema 1 ([30]). *Sea G un cografo conexo que es join de dos cografos $G_l = (V_l, E_l)$ y $G_r = (V_r, E_r)$ tales que $|V_l| + |V_r| \geq 3$. Entonces*

$$\gamma_R(G) = \begin{cases} 2 & \text{si y solo si } G_l \text{ o } G_r \text{ tienen un vértice universal,} \\ 3 & \text{si y solo si } G_l \text{ o } G_r \text{ tienen un vértice casi universal y } \gamma_R(G) \neq 2, \\ 4 & \text{en otro caso.} \end{cases}$$

Este resultado permite obtener un algoritmo lineal para calcular el número de dominación romana para un cografo conexo G .

Así también, a partir del Teorema 1 se desprende que un cografo conexo G es romano si y solo si $\gamma_R(G) \neq 3$, con lo cual obtenemos que los cografos romanos pueden ser reconocidos en tiempo lineal [30].

Observación 2. *En [30], el Teorema 1 se refiere a la operación join en el caso particular de los cografos. En el Capítulo 4 mostraremos que este resultado puede generalizarse para el grafo join de cualquier par de grafos.*

Grafos con cliquewidth acotado

En [10] Courcelle establece la existencia de algoritmos eficientes para resolver problemas que admiten una expresión particular, cuando nos restringimos a familias de grafos para las cuales el parámetro *cliquewidth* está acotado por una constante.

El *cliquewidth* de un grafo se define como el mínimo número de etiquetas que se necesitan para construir el grafo G con las siguientes 4 operaciones:

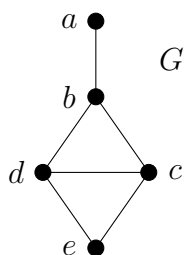
1. crear un nuevo vértice v con etiqueta i (notamos $i(v)$);
2. unión disjunta de dos grafos etiquetados G y H (notamos $G \cup H$);

3. conectar con una arista todo vértice etiquetado con i a todo vértice etiquetados con j (notamos $\eta_{i,j}$); y
4. cambiar a j la etiqueta de todos los que tienen etiqueta i (notamos $\rho_{i \rightarrow j}$).

Todo grafo G puede ser construido usando estas cuatro operaciones, lo que da lugar a una expresión algebraica. Una tal expresión que utiliza k etiquetas distintas, se denomina k -expresión.

El cliquewidth de G , denotado $cw(G)$, es el mínimo k para el cual existe una k -expresión que construya G .

Ejemplo 2. Consideremos el grafo G de la figura siguiente



Una posible 3-expresión para G es

$$\eta_{2,3} \left(\rho_{3 \rightarrow 2} \left(\eta_{2,3} (\eta_{1,2} (1(a) \oplus 2(b)) \oplus 2(e) \oplus 3(c)) \right) \oplus 3(d) \right)$$

Y se construye de la siguiente manera:

1 ●

2 ●

$$1(a) \oplus 2(b)$$

1 ●

2 ●

$$\eta_{1,2} (1(a) \oplus 2(b))$$

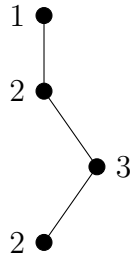
1 ●

2 ●

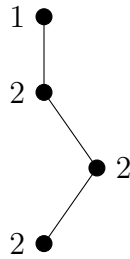
● 3

2 ●

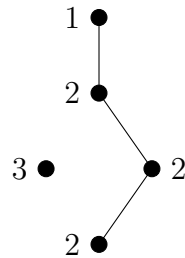
$$\eta_{1,2} (1(a) \oplus 2(b)) \oplus 2(e) \oplus 3(c)$$



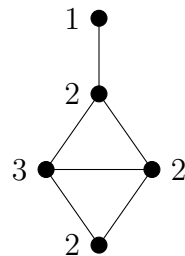
$$\eta_{2,3}(\eta_{1,2}(1(a) \oplus 2(b)) \oplus 2(e) \oplus 3(c))$$



$$\rho_{3 \rightarrow 2}(\eta_{2,3}(\eta_{1,2}(1(a) \oplus 2(b)) \oplus 2(e) \oplus 3(c)))$$



$$\rho_{3 \rightarrow 2}(\eta_{2,3}(\eta_{1,2}(1(a) \oplus 2(b)) \oplus 2(e) \oplus 3(c))) \oplus 3(d)$$



$$\eta_{2,3}(\rho_{3 \rightarrow 2}(\eta_{2,3}(\eta_{1,2}(1(a) \oplus 2(b)) \oplus 2(e) \oplus 3(c))) \oplus 3(d))$$

Si una fórmula (que exprese alguna propiedad en grafos) puede ser definida usando vértices y conjuntos de vértices de un grafo, los operadores lógicos **or**, **and**, **not** (\vee, \wedge, \neg), los cuantificadores \forall y \exists sobre vértices y conjuntos de vértices, la relación de pertenencia

\in , el operador $=$ para vértices y la relación binaria de adyacencia adj , entonces la fórmula es expresable en *lógica monádica de segundo orden* τ_1 ($\text{MSOL}(\tau_1)$).

Un problema de optimización sobre grafos \mathcal{P} es un problema $\text{LinEMSOL}(\tau_1)$ si puede escribirse de la siguiente manera:

Dado un grafo G y funciones f_1, f_2, \dots, f_m a valores enteros sobre los vértices de G , encontrar una asignación z al conjunto de variables libres en θ tal que

$$\sum_{\substack{1 \leq i \leq l \\ 1 \leq j \leq m}} a_{ij} |z(X_i)|_j = \text{opt}_{z'} \left\{ \sum_{\substack{1 \leq i \leq l \\ 1 \leq j \leq m}} a_{ij} |z'(X_i)|_j : \theta(X_1, X_2, \dots, X_l) \right\}$$

donde a_{ij} son números enteros para cada $i = 1, \dots, l, j = 1, \dots, m$, θ es una fórmula $\text{MSOL}(\tau_1)$ con conjunto de variables libres X_1, X_2, \dots, X_l y $|z(X_i)|_j := \sum_{a \in z(X_i)} f_j(a)$

Ejemplo 3. *Expresemos el Problema de Dominación Romana de esta manera.*

Podemos considerar las variables como los posibles conjuntos V_0, V_1, V_2 . Estos deben formar una partición del conjunto de vértices que además verifique que cada vértice en V_0 tiene un vértice vecino en V_2 .

Para expresar la primera parte, podemos escribir:

$$\begin{aligned} \text{Part}(V_0, V_1, V_2) \equiv & \forall u (V_0(u) \vee V_1(u) \vee V_2(u)) \wedge \\ & \neg \exists v ((V_0(v) \wedge V_1(v)) \vee (V_0(v) \wedge V_2(v)) \vee (V_1(v) \wedge V_2(v))) \end{aligned}$$

Es decir, estamos pidiendo que todo vértice u pertenezca a alguno de los tres conjuntos y, además, que no puede haber un vértice en dos de ellos.

Por otro lado, como las FDR's son particiones que además verifican 'algo más', podemos definir las como:

$$\text{Rom}(V_0, V_1, V_2) \equiv \text{Part}(V_0, V_1, V_2) \wedge \forall u (V_0(u) \rightarrow \exists v (V_2(v) \wedge \text{adj}(u, v)))$$

Observemos que definida de esta manera, $\text{Rom}(V_0, V_1, V_2)$ es verdadera si y solo si la función g dada por $g(v) = i, \forall v \in V_i$ es una FDR. El peso de esta función, lo podemos expresar como:

$$g(V) = \sum_{v \in V} g(v) = \sum_{v \in V_0} 0 + \sum_{v \in V_1} 1 + \sum_{v \in V_2} 2 = 0|V_0| + 1|V_1| + 2|V_2|$$

Así, considerando $l = 3, m = 1, f_1 \equiv 1, X_i = V_{i-1}, a_{i1} = i - 1$ y $\theta(X_1, X_2, X_3) \equiv \text{Rom}(X_1, X_2, X_3)$, tenemos que

$$|z(V_i)|_1 = \sum_{a \in z(V_i)} f_1(a) = \sum_{a \in z(V_i)} 1 = |z(V_i)|$$

donde la última expresión denota el cardinal del conjunto que le asigna z a la variable V_i . El Problema de Dominación Romana, entonces, puede plantearse como encontrar una asignación z a V_0, V_1, V_2 tal que

$$\gamma_R(G) = \sum_{1 \leq j \leq 3} a_{i1} |z(V_i)|_1 = \min \left\{ \sum_{1 \leq j \leq 3} a_{i1} |z'(V_i)|_1 : \text{Rom}(V_0, V_1, V_2) \right\}$$

Si \mathcal{F} es una familia de grafos con cliquewidth a lo sumo un valor constante k y existe un algoritmo eficiente para encontrar una k -expresión para un grafo en \mathcal{F} , entonces todo problema de optimización que puede expresarse como un problema $\text{LinEMSOL}(\tau_1)$ se resuelve eficientemente para grafos en \mathcal{F} [10]. La complejidad del problema depende de la complejidad del algoritmo que da la k -expresión. Por ejemplo, si la k -expresión se encuentra en tiempo lineal, entonces el problema se resuelve en tiempo lineal.

Por otro lado, en [34] se presenta, para k fijo, un algoritmo que, dado un grafo G de n vértices, devuelve una $(2^{3k+2} - 1)$ -expresión para G o el mensaje $cw(G) \geq k + 1$ en un tiempo $\mathcal{O}(n^9 \log n)$. Así, dada una familia de grafos con cliquewidth acotado por una constante k , podemos afirmar que todo problema $\text{LinEMSOL}(\tau_1)$ se resuelve en tiempo polinomial en esa familia, sin necesidad de dar un algoritmo específico que encuentre una k -expresión.

En [30] se muestra que el Problema de Dominación Romana se puede expresar de esta manera, y en consecuencia, se demuestra el siguiente resultado:

Proposición 9 ([30]). *El Problema de Dominación Romana es un problema de optimización $\text{LinEMSOL}(\tau_1)$*

De la Proposición 9 se desprende el siguiente corolario:

Corolario 1 ([30]). *Sea k una constante y \mathcal{F} una familia de grafos con cliquewidth a lo sumo k . Entonces el Problema de Dominación Romana se resuelve en tiempo polinomial para grafos en \mathcal{F} . Más aún, si existe un algoritmo que dado un grafo $G \in \mathcal{F}$ da una k -expresión en tiempo lineal, el Problema de Dominación Romana se resuelve en tiempo lineal para grafos en \mathcal{F} .*

En particular, resulta que el número de dominación romana puede ser calculado en tiempo lineal sobre cografos, y grafos distancia hereditarios, dado que en estas familias, el cliquewidth está acotado por 2 y 3 respectivamente, y existe un algoritmo lineal para calcular una k -expresión en cada caso.

1.2.3. Complejidad computacional de problemas

Uno de los aspectos que vamos a analizar en esta tesina, es la complejidad computacional del Problema de Dominación Romana.

Un *problema* π consiste en un conjunto de instancias D_π , que contiene todas las posibles entradas para el problema, y una pregunta sobre esas instancias. Decimos que un algoritmo resuelve un problema π si ese algoritmo puede ser aplicado a cualquier instancia $I \in D_\pi$ y se garantiza que siempre producirá la respuesta correcta para dicha instancia I .

Un *problema de decisión* es aquel cuyas respuestas son “sí” o “no”. Un *problema de optimización* es aquel cuya respuesta es una solución óptima para el problema planteado. Todo problema de optimización tiene asociado un problema de decisión equivalente.

Por ejemplo, si el problema de optimización pregunta por una estructura de cierto tipo que minimice cierto costo (como puede ser un camino de longitud mínima), podemos asociarlo con el problema de decisión que incluya adicionalmente un número c como parámetro y pregunte si existe una estructura de ese tipo que tenga costo menor o igual a c (en el ejemplo, un camino de longitud menor o igual a c).

El Problema de Dominación Romana tal como lo presentamos es un problema de optimización, buscamos una FDR de peso mínimo. Su problema de Decisión asociado es:

Problema de Dominación Romana (DROM)

Instancia: Grafo $G = (V, E)$, $k \in \mathbb{N}$

Pregunta: ¿Existe una FDR de peso a lo sumo k ?

Los problemas se agrupan según cuán costoso es resolverlos.

La *función de complejidad temporal* de un algoritmo expresa sus requerimientos temporales. Intuitivamente, podemos verla como la cantidad de pasos que efectúa hasta obtener la respuesta. Habitualmente es expresada como una función en términos de una sola variable: el tamaño de una instancia del problema.

Una instancia puede describirse como una cadena finita de símbolos elegidos de un alfabeto finito de entrada. Aunque hay varias formas distintas en que las instancias de un problema dado pueden ser descriptas, vamos a suponer que elegimos por adelantado una en particular. Así, el tamaño de una instancia I de un problema π se define como el número de símbolos en la descripción de I .

Una función $f(n)$ es $O(g(n))$ si existe una constante c tal que $|f(n)| \leq c \cdot |g(n)|$ para todo $n > 0$. Un algoritmo *de tiempo polinomial* (o simplemente *polinomial*) se define como aquel que tiene una función de complejidad temporal que es $O(p(n))$ para algún polinomio p , donde n denota el tamaño de la instancia. Todo algoritmo que no pueda ser acotado de esta manera se llama algoritmo *de tiempo exponencial*, aunque esta definición incluya algunas funciones de complejidad temporal no polinomiales que normalmente no se consideran exponenciales, como $n^{\log n}$.

Una *transformación (o reducción) polinomial* de un problema de decisión π_1 a un problema de decisión π_2 es una función $f : D_{\pi_1} \rightarrow D_{\pi_2}$ que satisface las siguientes dos condiciones:

1. f es computable por un algoritmo de tiempo polinomial; y
2. para toda instancia $I \in D_{\pi_1}$, la respuesta a I en π_1 es “sí” si y solo si la respuesta a $f(I)$ en π_2 es “sí”.

Los problemas de decisión que tienen algoritmos de tiempo polinomial que los resuelven se agrupan en la *clase P*. Para todos los demás problemas existen varias clases de complejidad definidas, de las cuales solo nos interesan unas pocas en este trabajo.

La *clase NP* contiene a los problemas de decisión que pueden ser resueltos polinomialmente por una máquina de Turing no determinística. Para no entrar en detalles que van más allá de esta tesina, podemos decir que la clase NP es aquella que contiene a los problemas de decisión para los cuales, para toda instancia en que la respuesta es “sí”, existe un *certificado* de la misma y un algoritmo polinomial que lo verifica. Por ejemplo, si el problema es determinar si existe una ruta de longitud menor a 10 km en un mapa, un certificado es una ruta posible y la verificación es comparar su longitud con 10 km; si esta verificación se puede hacer en tiempo polinomial, entonces el problema pertenece a NP. En particular, $P \subset NP$.

En 1971, Cook [9] probó que un problema particular de la clase NP, el llamado *problema de satisfactibilidad booleana* (o SAT), tiene la propiedad de que todo otro problema en NP puede ser reducido polinomialmente a él. Esto significa que si SAT pudiera ser resuelto con un algoritmo polinomial, entonces todos los otros problemas en NP también, y si algún problema en NP es intratable, es decir que es tan difícil que ningún algoritmo polinomial podría resolverlo, también lo es SAT.

Cook sugirió que otros problemas en NP debían compartir con SAT esta propiedad de ser los “más difíciles” de la clase NP. Karp [28] y luego Garey y Johnson [16] presentaron más

problemas como este y se definió entonces la clase de problemas *NP-completos*. Formalmente, un problema π es NP-completo si

1. π pertenece a la clase NP; y
2. para todo otro problema π' en NP, existe una transformación polinomial de π' en π .

Si omitimos el primer ítem, entonces decimos que π es *NP-hard* (o *NP-difícil*). Un problema es *intratable* si no existen algoritmos que lo resuelvan en tiempo polinomial. La pregunta de si los problemas NP-completos son intratables es considerada una de las preguntas abiertas más importantes de la matemática y las ciencias de la computación contemporáneas. A pesar de los intentos de los investigadores por probar o refutar esta conjetura, poco se ha avanzado en el tema.

Para probar que un problema π es NP-completo, procederemos de la siguiente manera:

1. verificamos que π pertenezca a la clase NP,
2. elegimos un problema π' que sea NP-completo,
3. construimos una transformación f de π' a π ,
4. probamos que f es una transformación polinomial.

En [11] se muestra que el problema DROM es NP-completo, construyendo una transformación polinomial desde el *Problema de 3-satisfacibilidad booleana* (3-SAT), que es NP-completo.

Un literal es o bien una variable $x_j \in X$, o bien $\neg x_j$. Una cláusula C_i , formada por exactamente 3 literales, es verdadera o se satisface por una asignación de valores de verdad $A : X \rightarrow \{TRUE, FALSE\}$ si a al menos uno de los literales de C_i , A le asigna el valor *TRUE*.

El Problema de 3-Satisfacibilidad Booleana (3-SAT) consiste en:

Problema de 3-Satisfacción Booleana (3-SAT)

Instancia: Colección $C = \{C_1, \dots, C_m\}$ de cláusulas (cada una conteniendo exactamente 3 literales) sobre un conjunto finito $X = \{x_1, \dots, x_n\}$ de variables

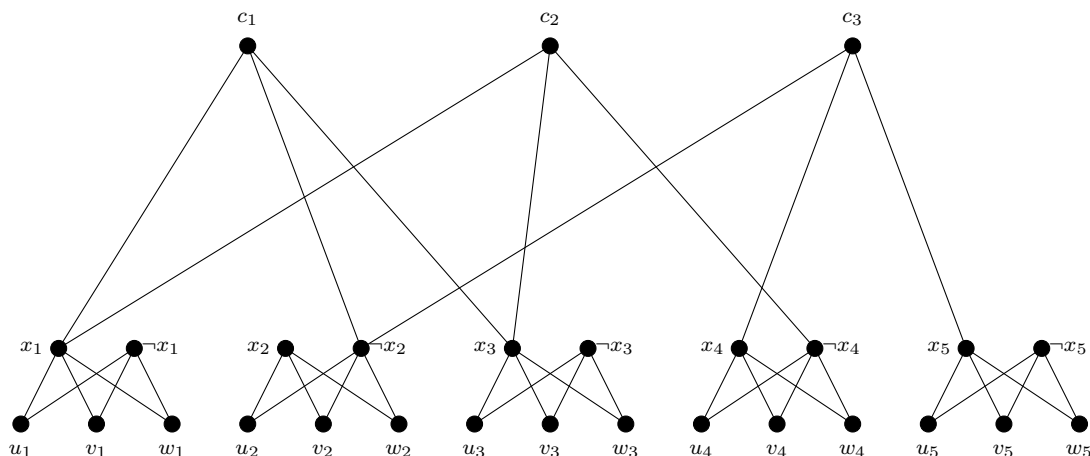
Pregunta: ¿Existe una asignación de valores de verdad para X tal que todas las cláusulas de C sean verdaderas?

Sea I una instancia de 3-SAT, como la definimos. Vamos a construir un grafo $G(I)$ y dar un natural k (una instancia para DROM) tales que I se satisface por una asignación de valores de verdad si y solo si $G(I)$ admite una FDR f con $f(V) \leq k$.

Consideremos el grafo $G(I)$ dado por: para cada cláusula C_i construimos un vértice c_i . Para cada variable $x_j \in X$ agregamos una copia del grafo bipartito completo $K_{2,3}$, con partición $R_j = \{x_j, \neg x_j\}$ y $S_j = \{u_j, v_j, w_j\}$.

En cada $K_{2,3}$, agregamos la arista $x_j \neg x_j$, y desde cada c_i agregamos tres aristas a los vértices correspondientes a los tres literales en la cláusula C_i .

Ejemplo 4. Si la instancia I del problema 3-SAT está dada por $X = \{x_1, x_2, x_3, x_4, x_5\}$ y $C = \{C_1, C_2, C_3\}$ con $C_1 = (x_1 \vee \neg x_2 \vee x_3)$, $C_2 = (x_1 \vee x_3 \vee \neg x_4)$, $C_3 = (\neg x_2 \vee x_4 \vee x_5)$, entonces el grafo $G(I)$ nos queda



Con esta construcción resulta que I admite una asignación de verdad A de manera que todas las cláusulas de C se verifican si y solo si $G(I)$ admite una FDR f de peso $f(V) \leq k = 2n$, donde n es la cantidad de variables en X .

En efecto, sea A una asignación de valores de verdad para X , de manera que la instancia dada I de 3-SAT se satisfaga. Sea $f = (V_0, V_1, V_2)$ definida por $V_1 = \emptyset$, $V_2 = \{x_j : A(x_j) = TRUE\} \cup \{\neg x_j : A(x_j) = FALSE\}$, $V_0 = V(G(I)) - V_2$. f así definida es una FDR de peso $f(V(G(I))) = 2n$ ya que para cada j , o bien x_j o bien $\neg x_j$ está en V_2 , con lo que los vértices de $R_j \cup S_j$ están defendidos. Además, cada vértice c_i tiene al menos un vecino en V_2 pues A asigna al menos a un literal de C_i el valor $TRUE$.

Recíprocamente, supongamos que f es una FDR sobre $G(I)$ de peso $f(V(G(I))) \leq 2n$. Cada subgrafo inducido por $R_j \cup S_j$, debe tener o bien $f(x_j) = 2$ o bien $f(\neg x_j) = 2$. Por lo tanto, f tiene peso al menos $2n$. Se sigue entonces que el conjunto de vértices V_2 defiende a los vértices correspondientes a las cláusulas. Por lo tanto, la instancia I dada de 3-SAT admite una asignación de valores de verdad A que la satisface, dada por $A(x_j) = TRUE$ si y solo si $f(x_j) = 2$.

Con esta transformación, junto al hecho de que DROM es un problema NP, tenemos que DROM es NP-completo.

Usando esta misma construcción, se puede probar que DROM es NP-completo incluso restringiendo la instancia a grafos bipartitos o grafos planares, ya que el grafo construido pertenece también a estas familias.

Pero como ya hemos visto, existen familias de grafos (como los cografos, los grafos de intervalo, los árboles), para las cuales existen algoritmos que resuelven el Problema de Dominación Romana (y por lo tanto el problema de decisión DROM) en tiempo lineal.

En la siguiente tabla resumimos algunas familias de grafos y sus correspondientes resultados en cuanto a la complejidad del problema DROM restringido a ellas. En la última columna, lo comparamos con la complejidad del Problema de Dominación clásico.

Familia	Problema de Dominación Romana	Problema de Dominación Clásico
Árboles	Lineal [8]	Lineal [13]
Intervalos	Lineal [30]	Lineal [7]
Cliquewidth acot.	Polinomial [30]	Polinomial [10]
Cografos	Lineal [30]	Lineal [20]
Block	Lineal [24]	Lineal [33]
Dist. hereditarios	Lineal [30]	Lineal [33]
AT-free	Polinomial [30]	Polinomial [29]
Cordales	NP-completo [31]	NP-completo [6]
Fuertemente cordales	Lineal [31]	Lineal [13]
Bipartitos	NP-completo [31]	NP-completo [12]
Split	NP-completo [36]	NP-completo [4]
Planares	NP-completo [36]	NP-completo [16]

En la tabla podemos observar que en estas familias conocidas, la complejidad de ambos problemas es la misma. A su vez, aún no se conocen familias en las cuales difiera.

Capítulo 2

Primeros Resultados

En este Capítulo vamos a estudiar cómo se comporta el parámetro γ_R ante ciertas operaciones de grafos. Luego, calcularemos el parámetro de dominación romana para algunas familias de grafos particulares, entre ellas los grafos araña y quasi-araña, que usaremos más adelante en el próximo Capítulo para estudiar la complejidad del Problema de Dominación Romana en familias de grafos con pocos P_4 's.

2.1. Primeras propiedades

Primero, vamos a presentar algunos resultados a los que llegamos y que usaremos en las demostraciones de este Capítulo.

Lema 1. *Sea $f = (V_0, V_1, V_2)$ una FDR en un grafo G con $V_2 = \emptyset$, entonces $V_0 = \emptyset$.*

Demostración. Basta probar que si $V_0 \neq \emptyset$ entonces $V_2 \neq \emptyset$. Supongamos que $v \in V_0$. Por definición de FDR, existe $u \in N(v) \cap V_2$. Resulta entonces que $V_2 \neq \emptyset$. \square

Observación 3. *Si $V_2 = \emptyset$, entonces $f(v) = 1$ para todo vértice v . En ese caso, lo notaremos $f \equiv 1$.*

Lema 2. *Sean $G = (V, E)$ un grafo con $E \neq \emptyset$ y $k \in \mathbb{N}$ tal que existe una FDR sobre G de peso k . Entonces existe una FDR $f = (V_0, V_1, V_2)$ sobre G de peso k tal que $V_2 \neq \emptyset$.*

Demostración. Sea f una FDR sobre G tal que $f(V) = k$. Si existe algún vértice v para el cual $f(v) = 2$, entonces ya tenemos el resultado.

En caso contrario, como consecuencia del Lema 1, tenemos que $f \equiv 1$. Sea uw una arista de G . Definimos $\hat{f} : V \rightarrow \{0, 1, 2\}$ como sigue:

$$\hat{f}(v) = \begin{cases} 2 & \text{si } v = u, \\ 0 & \text{si } v = w, \\ f(v) = 1 & \text{en otro caso.} \end{cases}$$

\hat{f} así definida es una FDR, ya que si $f(v) = 0$ entonces $v = w$ y existe un vecino u de w tal que $f(u) = 2$.

Por lo tanto, \hat{f} es una FDR de peso

$$\hat{f}(V) = \hat{f}(V - \{u, w\}) + \underbrace{\hat{f}(\{u, w\})}_2 = f(V) = k.$$

\square

Lema 3. Sean $G = (V, E)$ un grafo, $v \in V$ y f una FDR sobre G . Entonces $f(N[v]) \geq 1$. Más aún, si $f(N[v]) = 1$ entonces $f(v) = 1$ y $f(u) = 0$ para todo vértice $u \in N(v)$.

Demostración. Queremos ver que

$$f(N[v]) = \sum_{u \in N[v]} f(u) \geq 1 \quad (2.1.1)$$

Supongamos, por el contrario, que no. Entonces

$$f(N[v]) = \sum_{u \in N[v]} f(u) = 0$$

Luego resulta

$$f(u) = 0 \quad \forall u \in N[v] \quad (2.1.2)$$

En particular, $f(v) = 0$ y como f es una FDR existe un vértice $u \in N(v)$ con $f(u) = 2$. Pero esto contradice (2.1.2). La contradicción surge de suponer que no vale (2.1.1). Entonces concluimos que $f(N[v]) \geq 1$.

Supongamos ahora que (2.1.1) se da por igualdad. Esto es, existe un único vértice en $N[v]$ al cual f le asigna 1 y al resto les asigna 0. Tenemos entonces que $N(v) \subseteq V_0 \cup V_1$. Luego $f(v) \neq 0$ pues $N(v) \cap V_2 = \emptyset$.

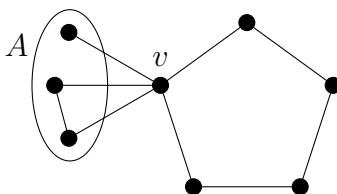
Concluimos entonces que

$$f(v) = 1, \quad f(u) = 0 \quad \forall u \in N(v)$$

como queríamos ver. □

Para el siguiente resultado, veamos antes un ejemplo

Ejemplo 5. Consideremos el siguiente grafo G



Observemos el conjunto de vértices A marcado. De todo el resto de vértices del grafo, los vértices de A tienen un único vecino común: el vértice v .

Supongamos que f es una FDR sobre el grafo G .

Si para algún vértice $a \in A$ se verifica $f(a) = 0$, entonces o bien para algún otro vértice $a' \in A$ se tiene $f(a') = 2$, o bien $f(v) = 2$. De todos modos, solo debemos mirar al conjunto $A \cup \{v\}$.

Entonces, si encontramos un conjunto A , con $|A| \geq 2$, que resulta ser la vecindad privada de un vértice v respecto al resto de los vértices $V - A$, es decir, todos los vértices de A son adyacentes a un único vértice $v \in V - A$, tendremos que $f(A \cup \{v\}) \geq 2$.

Si $|A| = 1$, $A = \{a\}$ esto no es necesariamente cierto. Pues podría ser $f(a) = 1$, $f(v) = 0$, $f(w) = 2$ para algún w adyacente a v fuera de A .

Esta situación se puede generalizar y tenemos el siguiente resultado.

Lema 4. Sean $G = (V, E)$ un grafo, $v \in V$ y $A \subseteq V$ tales que $v \notin A$ y $A = pn(v, V - A)$, con $|A| \geq 2$, y f una FDR sobre G , entonces $f(A \cup \{v\}) \geq 2$.

Demostración. Sea $A = \{a_1, a_2, \dots, a_m\}$ como en el enunciado. Como para cada i tenemos que $a_i \in pn(v, V - A)$, recordando que $pn(v, V - A) = N[v] - N[V - A]$, resulta $N[a_i] \subseteq A \cup \{v\}$ para cada $i = 1, \dots, m$.

Observemos que en particular por el Lema 3 tenemos que $f(N[a_1]) \geq 1$. Así, resulta

$$f(A \cup \{v\}) \geq f(N[a_1]) \geq 1. \quad (2.1.3)$$

Veamos que la cota dada en la ecuación (2.1.3) se puede ajustar un poco más.

Supongamos entonces que $f(A \cup \{v\}) = 1$. Así, como la ecuación (2.1.3) se da por igualdad, resulta $f(N[a_1]) = 1$ y por el Lema 3 tenemos que $f(a_1) = 1$. Como $f(A \cup \{v\}) = 1$, resulta que $f(v) = 0$ y $f(a_i) = 0$ para cada $i \neq 1$. Entonces,

$$(A \cup \{v\}) \cap V_2 = \emptyset. \quad (2.1.4)$$

Pero como $f(a_2) = 0$, debe existir un vértice $u \in N(a_2) \subseteq A \cup \{v\}$ tal que $f(u) = 2$, lo que contradice la ecuación (2.1.4).

La contradicción surge al suponer $f(A \cup \{v\}) = 1$. Luego, tenemos que $f(A \cup \{v\}) \neq 1$. Con esto junto a la ecuación (2.1.3), llegamos a la conclusión

$$f(A \cup \{v\}) \geq 2$$

que es lo que queríamos demostrar. \square

Lema 5. *Sea $G = (V, E)$ un grafo de orden al menos 3. Entonces $\gamma_R(G) = 3$ si y solo si $G = S_3$ o $\Delta(G) = |V| - 2$ (G tiene un vértice quasi-universal).*

Demostración. Supongamos que $\gamma_R(G) = 3$, y sea $f = (V_0, V_1, V_2)$ una γ_R -función sobre G . Tenemos dos opciones posibles:

$$|V_1| = 3, |V_2| = 0 \quad \text{o bien} \quad |V_1| = 1, |V_2| = 1$$

En el primer caso, como $V_2 = \emptyset$, resulta $V_0 = \emptyset$, y $|V| = |V_0| + |V_1| + |V_2| = 3$. Los únicos dos grafos de 3 vértices que admiten una γ_R -función $f \equiv 1$ son S_3 y \overline{P}_3 . El último tiene dos vértices quasi-universales.



En el segundo caso, tenemos un vértice $u \in V_1$ y un vértice $v \in V_2$. El resto de los vértices de V pertenecen a V_0 . Como $V_0 \subseteq N(V_2)$, tenemos que $V_0 \subseteq N(v)$. Además $u \notin N(v)$, pues de lo contrario G tendría un vértice universal y $\gamma_R(G) = 2$. Luego, resulta que $gr(v) = |N(v)| = |V_0| = |V - \{u, v\}| = |V| - 2$. Es decir, v es un vértice quasi-universal de G .

Recíprocamente, ya sabemos que $\gamma_R(S_3) = 3$. Supongamos ahora que el vértice de mayor grado en G es un vértice quasi-universal $v \in V$. Es decir, v es adyacente a cada vértice en $V - \{v\}$ excepto un vértice u .

Por un lado, como $|V| \geq 3$, tenemos que $\gamma_R(G) > 1$. Además, de la Proposición 4 y la Proposición 10 sabemos que $\gamma_R(G) = 2$ si y solo si G tiene un vértice universal. Luego, como G no tiene un vértice universal, tenemos

$$\gamma_R(G) \geq 3 \quad (2.1.5)$$

Veamos ahora que existe una FDR f de peso $f(V) = 3$. Definimos f de la siguiente manera:

$$f(w) = \begin{cases} 2, & \text{si } w = v \\ 1, & \text{si } w = u \\ 0, & \text{en otro caso,} \end{cases}$$

f así definida es un FDR. En efecto, si $f(w) = 0$, entonces $w \notin \{u, v\}$ y como $N(v) = V - \{u, v\}$, w es adyacente a v en G y $f(v) = 2$. Luego, tenemos que

$$\gamma_R(G) \leq f(V) = 3 \quad (2.1.6)$$

Entonces, de las ecuaciones (2.1.5) y (2.1.6), tenemos que

$$\gamma_R(G) = 3.$$

□

La recíproca de la Proposición 4 también es cierta si G es un grafo de orden mayor que 2, y tenemos:

Proposición 10. *Sea $G = (V, E)$ un grafo de orden $n \geq 3$ tal que $\gamma_R(G) = 2$, entonces G tiene un vértice universal.*

Demostración. Sea $f = (V_0, V_1, V_2)$ una γ_R -función de G . Sabemos que f es una FDR con $f(V) = 2$, y tenemos dos casos posibles:

1. $|V_1| = 2$ y $|V_2| = 0$,
2. $|V_1| = 0$ y $|V_2| = 1$.

En el primer caso, $V_2 = \emptyset$ y, por el Lema 1, $V_0 = \emptyset$. Entonces tenemos que $|V| = 2$, lo que contradice que el orden de G es mayor a 2.

En el segundo caso, $V_2 = \{\hat{v}\}$ para algún vértice \hat{v} . Como $V_1 = \emptyset$, resulta $V = V_0 \cup \{\hat{v}\}$. Además, $V_0 \neq \emptyset$ pues por hipótesis G es de orden al menos 3. Sea entonces $u \in V - \{\hat{v}\} = V_0$. Luego, $u \in N(V_2) = N(\hat{v})$. Por lo tanto, $V_0 \subseteq N(\hat{v})$, y resulta

$$V = V_0 \cup \{\hat{v}\} \subseteq N[\hat{v}].$$

Por lo tanto, Tengo que \hat{v} es el unico en V_2 y que todos los otros vertices de V estan en V_0 . Como todos los de V_0 tienen que ser adyacente a uno de V_2 , todos los de V_0 son adyacentes a \hat{v} y \hat{v} es un vértice universal de G .

□

2.2. γ_R bajo algunas operaciones

En esta sección vamos a analizar el comportamiento del número de dominación romana cuando realizamos ciertas operaciones en grafos, tales como unión disjunta, join, generación de un mellizo, y borrado de un vértice.

2.2.1. Borrado de un vértice

Dado un grafo $G = (V, E)$, y $v \in V$, vamos a intentar responder cómo influye en $\gamma_R(G)$ el borrado del vértice v . Comenzaremos mostrando que el número de dominación romana no puede disminuir en más de una unidad al eliminar un vértice.

Teorema 2. Sean $G = (V, E)$ un grafo y $v \in V$. Entonces $\gamma_R(G - v) \geq \gamma_R(G) - 1$

Demostración. Sea f una γ_R -función sobre $G - v$. Definiendo $\hat{f} : V \rightarrow \{0, 1, 2\}$ como

$$\hat{f}(u) = f(u) \quad u \in V - \{v\}, \quad \hat{f}(v) = 1$$

tenemos que \hat{f} es una FDR sobre G . En efecto, si $\hat{f}(u) = 0$ entonces $f(u) = 0$ y existe un vértice w vecino a u en $G - v$ tal que $f(w) = 2$, y por lo tanto $\hat{f}(w) = 2$. Entonces, resulta que

$$\gamma_R(G) \leq \hat{f}(V) = \hat{f}(V - \{v\}) + \hat{f}(v) = f(V - \{v\}) + 1 = \gamma_R(G - v) + 1.$$

Es decir,

$$\gamma_R(G - v) \geq \gamma_R(G) - 1.$$

□

Esto es, si borramos un vértice v de un grafo G , el número de dominación romana no puede disminuir en más de una unidad. Sin embargo, podemos preguntarnos ¿en qué casos disminuye efectivamente? ¿en cuáles permanece igual?. En caso que aumente su valor, ¿es posible acotar el crecimiento del parámetro? Estos interrogantes motivan los siguientes resultados:

Teorema 3. Sean $G = (V, E)$ un grafo y $v \in V$ tales que existe una γ_R -función $f = (V_0, V_1, V_2)$ con $v \in V_1$. Entonces $\gamma_R(G - v) = \gamma_R(G) - 1$

Demostración. Sea f una γ_R -función sobre G con $f(v) = 1$.

Observemos que la función f restringida a $V - \{v\}$ es una FDR sobre el grafo $G - v$. En efecto, cada vértice $u \in V - \{v\}$ para el cual $f(u) = 0$ tiene un vecino $w \in V_2$. Como $v \notin V_2$, tenemos que $w \neq v$ y resulta que u tiene un vecino $w \in (V - \{v\}) \cap V_2$. Así, tenemos

$$\gamma_R(G - v) \leq f(V - \{v\}) = f(V) - f(v) = f(V) - 1 = \gamma_R(G) - 1. \quad (2.2.1)$$

Finalmente, juntando el Teorema 2 con la ecuación (2.2.1) obtenemos

$$\gamma_R(G - v) = \gamma_R(G) - 1.$$

□

Observación 4. La recíproca del Teorema 3 también es cierta. Es decir, si $\gamma_R(G - v) = \gamma_R(G) - 1$, entonces existe una γ_R -función f sobre G tal que $f(v) = 1$.

Teorema 4. Sean $G = (V, E)$ un grafo y $v \in V$ tales que existe una γ_R -función $f = (V_0, V_1, V_2)$ con $v \in V_0$. Entonces

$$\gamma_R(G) - 1 \leq \gamma_R(G - v) \leq \gamma_R(G).$$

Demostración. Sea f una γ_R -función sobre G con $f(v) = 0$. Observemos que la función f restringida a $V - \{v\}$ es una FDR sobre el grafo $G - v$. En efecto, cada vértice $u \in V - \{v\}$ para el cual $f(u) = 0$ tiene un vecino $w \in V_2$. Como $v \notin V_2$, tenemos que $w \neq v$ y resulta que u tiene un vecino $w \in (V - \{v\}) \cap V_2$. Así, tenemos

$$\gamma_R(G - v) \leq f(V - \{v\}) = f(V) - f(v) = f(V) - 0 = \gamma_R(G).$$

Es decir,

$$\gamma_R(G - v) \leq \gamma_R(G).$$

Finalmente, juntando el Teorema 2 con la cota obtenida tenemos

$$\gamma_R(G) - 1 \leq \gamma_R(G - v) \leq \gamma_R(G).$$

□

Veamos que las cotas dadas por el Teorema 4 son ajustadas.

Ejemplo 6. Consideremos el grafo $G = K_{1,n}$, la estrella de n puntas y vértice central w , con $n \geq 2$.

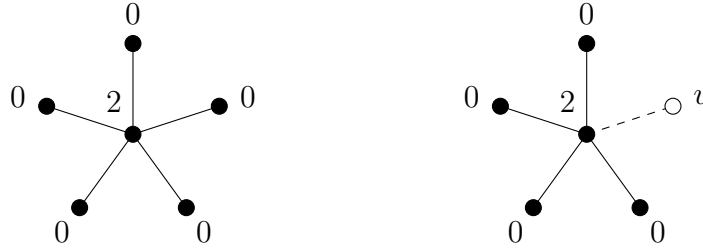


Figura 2.1: Borrado de un vértice en la estrella $K_{1,5}$

El número de dominación romana de G es $\gamma_R(G) = 2$ para cada $n \geq 2$, ya que w es un vértice universal de G . Una γ_R -función viene dada por $f = (V - \{w\}, \emptyset, \{w\})$. Al eliminar un vértice pendiente v de G , obtenemos una nueva estrella de $n - 1$ puntas y vértice central w , y vale $\gamma_R(G - v) = 2 = \gamma_R(G)$. Es decir

$$\gamma_R(G - v) = \gamma_R(G)$$

Ejemplo 7. Consideremos $G = P_4$, el camino de 4 vértices con conjunto de vértices $\{v, w, u, x\}$

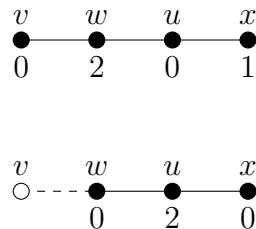


Figura 2.2: Borrado de un vértice en el camino P_4

El número de dominación romana de G es $\gamma_R(G) = 3$, y una posible γ_R -función es $f = (\{v, w\}, \{x\}, \{u\})$. Al eliminar el vértice v , el grafo obtenido es un camino de tres vértices, y $\gamma_R(G - v) = 2 = \gamma_R(G) - 1$. Es decir

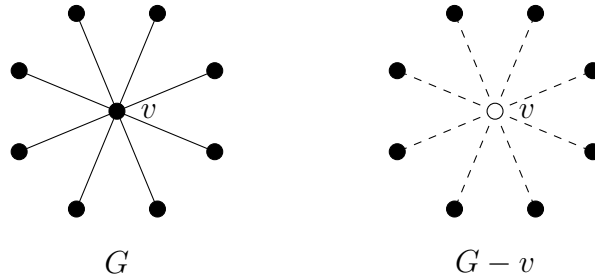
$$\gamma_R(G - v) = \gamma_R(G) - 1.$$

Hasta acá, hemos visto casos en los cuales $\gamma_R(G)$ nunca crece por borrado. Veamos que si el vértice borrado verifica $v \in V_2$ para alguna γ_R -función, el número de dominación romana de G puede crecer tanto como se quiera.

el enunciado del sgte teorema es confuso para mi... yo diria que existen grafos para los cuales la diferencia entre el numero de dom romana entre este y el numero de dom romana de un subgrafo obtenido por borrado de un vertice (no arbitrario), es arbitrariamente grande. Asi como esta escrito dice que para todo grafo G que tiene una γ_R -función con $V_2 \neq \emptyset$, la diferencia entre $\gamma_R(G - v)$ y $\gamma_R(G)$ puede ser arbitrariamente grande

Teorema 5. *Existen grafos G para los cuales la diferencia entre número de dominación romana de G y el número de dominación romana de un subgrafo obtenido por borrado de un vértice (no arbitrario) es arbitrariamente grande.*

Demostración. Basta considerar como ejemplo el grafo $G = K_{1,n}$, una estrella de n puntas y vértice central v . $f = (V - \{v\}, \emptyset, \{v\})$ es una γ_R -función, siendo $\gamma_R(G) = 2$. Si quitamos el vértice v del grafo G , el nuevo grafo $G - v$ consta de n vértices aislados, con lo cual $\gamma_R(G - v) = n$.

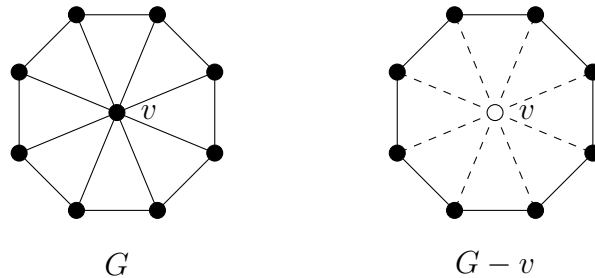


Luego, para n arbitrariamente grande, podemos encontrar un grafo G tal que

$$\gamma_R(G - v) - \gamma_R(G) = n - 2 \tag{2.2.2}$$

□

Ejemplo 8. *Veamos otro ejemplo en el que se da esta situación. Consideremos el grafo $G = W_n$, la rueda formada por un ciclo C_n agregando un vértice v adyacente a todos los vértices del ciclo. Nuevamente, como v es un vértice universal de G , la función $f = (V - \{v\}, \emptyset, \{v\})$ es una γ_R -función, y $\gamma_R(G) = 2$. Al borrar el vértice v del grafo G , $G - v$ es un ciclo de n vértices, y $\gamma_R(G - v) = \gamma_R(C_n) = \lceil \frac{2n}{3} \rceil$*



Luego, para n arbitrariamente grande, podemos encontrar un grafo G conexo tal que

$$\gamma_R(G - v) - \gamma_R(G) = \left\lceil \frac{2n}{3} \right\rceil - 2 \tag{2.2.3}$$

Tanto en la ecuación (2.2.2) como en la ecuación (2.2.3), las diferencias pueden ser tan grandes como queramos.

2.2.2. Generación de mellizos

En la sección anterior analizamos cómo influye en $\gamma_R(G)$ la acción de borrar un vértice del grafo. En esta sección, vamos a analizar que sucede al agregar un vértice, mediante la generación de un mellizo falso (o un mellizo verdadero) a un vértice del grafo. Es decir, agregar un vértice que comparta la misma vecindad abierta (o cerrada) con un vértice v del grafo G .

Primero, probaremos que al agregar un mellizo a un grafo G , el número de dominación romana no disminuye, y si aumenta lo hace en una unidad. Después analizaremos distintos casos, según el valor que una γ_R -función sobre el grafo original le asigne al vértice al cual le agregamos un mellizo.

Lema 6. Sean $G = (V, E)$, $v \in V$ y $G' = (V', E')$ el grafo que obtenemos al agregarle a G un vértice v' mellizo (falso o verdadero) de v . Entonces

$$\gamma_R(G) \leq \gamma_R(G') \leq \gamma_R(G) + 1$$

Demostración. Consideremos una γ_R -función \tilde{f} sobre G' . Observemos que $G = G' - v'$. Separemos la situación en tres casos posibles:

- $\tilde{f}(v') = 0$: por Teorema 4, $\gamma_R(G) \leq \gamma_R(G')$.
- $\tilde{f}(v') = 1$: por Teorema 3, $\gamma_R(G) < \gamma_R(G')$.
- $\tilde{f}(v') = 2$: definamos $f : V \rightarrow \{0, 1, 2\}$ de la siguiente manera:

$$f(u) = \tilde{f}(u) \quad \forall u \neq v, \quad f(v) = 2.$$

Así definida, f resulta una FDR sobre el grafo G . En efecto, si $f(u) = 0$ y $u \notin N(v)$, entonces como $\tilde{f}(u) = f(u) = 0$ y \tilde{f} es una FDR, existe un vértice $w \neq v$ vecino de u tal que $f(w) = \tilde{f}(w) = 2$. Si $u \in N(v)$, entonces v es adyacente a u y por cómo definimos la función f , $f(v) = 2$.

Por lo tanto, f es efectivamente una FDR sobre el grafo G y tenemos

$$\gamma_R(G) \leq f(V) = \underbrace{f(v)}_2 + f(V - \{v\}) = \underbrace{\tilde{f}(v')}_2 + \tilde{f}(V - \{v\}) \leq \tilde{f}(V') = \gamma_R(G').$$

En cualquiera de los tres casos (a., b., c.) tenemos que

$$\gamma_R(G) \leq \gamma_R(G'). \tag{2.2.4}$$

Por otro lado, vemos que G se obtiene a partir de G' borrando el vértice v' . Luego, por el Teorema 2 tenemos que

$$\gamma_R(G') - 1 \leq \gamma_R(G). \tag{2.2.5}$$

Juntando la ecuación (2.2.4) y la ecuación (2.2.5) tenemos el resultado. \square

Teorema 6. Sean $G = (V, E)$ un grafo, $v \in V$ tal que existe una γ_R -función f sobre G con $f(v) = 0$, y $G' = (V', E')$ el grafo que obtenemos al agregarle a G un mellizo (falso o verdadero) v' al vértice v . Entonces

$$\gamma_R(G') = \gamma_R(G).$$

Demostración. Se verifica que $N_{G'}(v) = N_{G'}(v')$ (si v' es un mellizo falso) o bien $N_{G'}[v] = N_{G'}[v']$ (si v' es un mellizo verdadero).

Por el Lema 6 tenemos que

$$\gamma_R(G) \leq \gamma_R(G'). \quad (2.2.6)$$

Consideremos ahora una γ_R -función en G f tal que $f(v) = 0$. Tal función existe por hipótesis. Definiendo $f' : V' \rightarrow \{0, 1, 2\}$ como

$$f'(u) = f(u) \quad \forall u \in V, \quad f'(v') = 0$$

tenemos que f' es una FDR sobre el grafo G' . En efecto, debemos ver que para cada vértice $u \in V'$ con $f'(u) = 0$, existe w adyacente a u en G' tal que $f'(w) = 2$. Sea entonces u tal que $f'(u) = 0$.

Si $u \in V$, como $f(u) = f'(u) = 0$ y f es una FDR, entonces existe w adyacente a u en G (y por lo tanto también en G') con $f(w) = f'(w) = 2$.

Si $u \notin V$, i.e. $u = v'$, tenemos que como $f(v) = 0$, existe w adyacente a v en G con $f(w) = 2$. Tanto si v' es un mellizo falso, ($N_{G'}(v') = N_G(v)$) como si es un mellizo verdadero ($N_{G'}[v'] = N_G[v]$) w es adyacente a v' en G' y $f'(w) = f(w) = 2$.

Por lo tanto, f' es una FDR sobre G' y resulta

$$\gamma_R(G') \leq f'(V') = \underbrace{f'(v')}_0 + \underbrace{f'(V)}_{f(V)} = f(V) = \gamma_R(G)$$

y tenemos entonces

$$\gamma_R(G') \leq \gamma_R(G). \quad (2.2.7)$$

Por las ecuaciones 2.2.6 y 2.2.7 resulta entonces

$$\gamma_R(G) = \gamma_R(G').$$

□

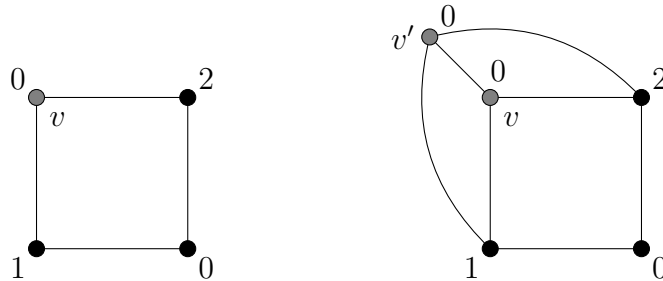


Figura 2.3: Agregarlo de un mellizo verdadero a un vértice de C_4

Teorema 7. Sean $G = (V, E)$ un grafo, $v \in V$ tal que existe una γ_R -función f sobre G con $f(v) = 2$, y $G' = (V', E')$ el grafo que obtenemos al agregarle a G un mellizo verdadero v' al vértice v . Entonces

$$\gamma_R(G') = \gamma_R(G).$$

Demostración. Se verifica que $N_{G'}[v] = N_{G'}[v']$

Por el Lema 6 tenemos que

$$\gamma_R(G) \leq \gamma_R(G'). \quad (2.2.8)$$

Consideremos ahora una γ_R -función en G f tal que $f(v) = 2$. Tal función existe por hipótesis. Definiendo $f' : V' \rightarrow \{0, 1, 2\}$ como

$$f'(u) = f(u) \quad \forall u \in V, \quad f'(v') = 0$$

tenemos que f' es una FDR sobre el grafo G' . En efecto, debemos ver que para cada vértice $u \in V'$ con $f'(u) = 0$, existe w adyacente a u en G' tal que $f'(w) = 2$. Sea entonces u tal que $f'(u) = 0$.

Si $u \in V$, como $f(u) = f'(u) = 0$ y f es una FDR, entonces existe w adyacente a u en G (y por lo tanto también en G') con $f(w) = f'(w) = 2$.

Si $u \notin V$, i.e. $u = v'$, tenemos que como $f(v) = 2$, y $v \in N_{G'}[v] = N_{G'}[v']$, v es adyacente a v' en G' y $f(v) = f'(v) = 2$.

Por lo tanto, f' es una FDR sobre G' y resulta

$$\gamma_R(G') \leq f'(V') = \underbrace{f'(v')}_0 + \underbrace{f'(V)}_{f(V)} = f(V) = \gamma_R(G)$$

y tenemos entonces

$$\gamma_R(G') \leq \gamma_R(G). \quad (2.2.9)$$

Por las ecuaciones 2.2.8 y 2.2.9 resulta entonces

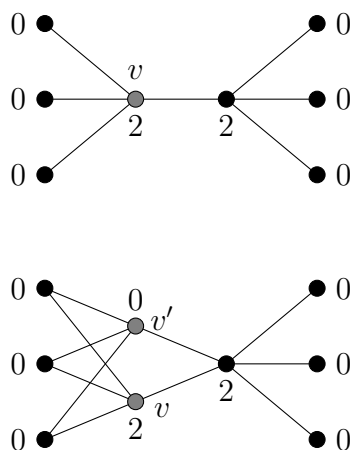
$$\gamma_R(G) = \gamma_R(G').$$

□

En el caso en que agregamos un mellizo falso a un vértice v de un grafo G para el cual existe una γ_R -función f con $f(v) = 2$, pueden existir distintos comportamientos. En efecto, en la Figura 2.4 se ve cómo al agregar un mellizo falso al vértice central de una estrella $K_{1,4}$, el número de dominación romana aumenta en una unidad, mientras que en el grafo de la Figura 2.5 al agregar un mellizo falso al vértice v , el número de dominación romana del grafo se mantiene igual.



Figura 2.4: Agregado de un mellizo falso de $v \in V_2$

Figura 2.5: Agregado de un mellizo falso de $v \in V_2$

2.2.3. Unión disjunta

Recordemos que dados dos grafos G_1 y G_2 con conjuntos de vértices disjuntos, el grafo unión disjunta de G_1 y G_2 , $G_1 \cup G_2$, es aquel cuyo conjunto de vértices es $V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$ y cuyo conjunto de aristas es $E(G_1 \cup G_2) = E(G_1) \cup E(G_2)$.

Teorema 8. *Dados dos grafos cualesquiera G_1 y G_2 tales que $V(G_1) \cap V(G_2) = \emptyset$, tenemos que*

$$\gamma_R(G_1 \cup G_2) = \gamma_R(G_1) + \gamma_R(G_2).$$

Demostración. Sean f_1 y f_2 dos γ_R -funciones sobre G_1 y G_2 respectivamente. Definimos $f : (V(G_1) \cup V(G_2)) \rightarrow \{0, 1, 2\}$ por

$$f(v) = \begin{cases} f_1(v) & \text{si } v \in V(G_1), \\ f_2(v) & \text{si } v \in V(G_2). \end{cases}$$

Es fácil ver que f es una FDR sobre $G_1 \cup G_2$. En efecto, sea $v \in V(G_1 \cup G_2)$ tal que $f(v) = 0$. Si $v \in V(G_1)$, entonces $f_1(v) = 0$ y existe un vértice w vecino a v en G_1 tal que $f_1(w) = 2$, y por lo tanto $f(w) = 2$. Esto es, v tiene un vecino en $G_1 \cup G_2$ al cual f le asigna el valor dos. Un razonamiento análogo vale si $v \in G_2$. Entonces f es una FDR sobre $G_1 \cup G_2$ y tenemos que

$$\gamma_R(G_1 \cup G_2) \leq f(V(G_1 \cup G_2)) = f_1(V(G_1)) + f_2(V(G_2)) = \gamma_R(G_1) + \gamma_R(G_2). \quad (2.2.10)$$

Consideremos ahora una γ_R -función g sobre $G_1 \cup G_2$. Observemos que la función g restringida a cada grafo G_1 o G_2 es una FDR, pues si hay un vértice v , digamos en G_1 , al cual g le asigna el valor 0, entonces existe un vértice w vecino a v al cual g le asigna el valor 2. Pero por construcción de $G_1 \cup G_2$, w debe ser un vértice de G_1 . Y análogamente si v es un vértice de G_2 . Entonces, tenemos que $g(V(G_1)) \geq \gamma_R(G_1)$ y $g(V(G_2)) \geq \gamma_R(G_2)$. Entonces

$$\gamma_R(G_1 \cup G_2) = g(V(G_1 \cup G_2)) = g(V(G_1)) + g(V(G_2)) \geq \gamma_R(G_1) + \gamma_R(G_2). \quad (2.2.11)$$

A partir de la ecuación (2.2.10) y la ecuación (2.2.11) tenemos que

$$\gamma_R(G_1 \cup G_2) = \gamma_R(G_1) + \gamma_R(G_2).$$

□

Observación 5. Este resultado se puede generalizar para la unión disjunta de una cantidad finita de grafos. Si G_1, \dots, G_k son k grafos cualesquiera cuyos conjuntos de vértices son disjuntos dos a dos, entonces

$$\gamma_R(G_1 \cup \dots \cup G_k) = \gamma_R(G_1) + \dots + \gamma_R(G_k).$$

2.2.4. Join

Recordemos que dados dos grafos G_1 y G_2 con conjuntos de vértices disjuntos, el grafo join de G_1 y G_2 , $G_1 \vee G_2$, es el grafo con conjunto de vértices $V(G_1 \vee G_2) = V(G_1) \cup V(G_2)$ y conjunto de aristas $E(G_1 \vee G_2) = E(G_1) \cup E(G_2) \cup \{\{u, v\} : u \in V(G_1), v \in V(G_2)\}$. Para calcular el número de dominación romana del grafo join de dos grafos, solo basta con conocer los grados máximos de cada uno de los grafos.

Este resultado ya se conoce para el caso en que los grafos pertenecen a la familia de los cografos, pero logramos probar esta generalización para cualquier par de grafos:

Teorema 9. Dados dos grafos cualesquiera G_1 y G_2 con $V(G_1) \cap V(G_2) = \emptyset$, tenemos que

$$\gamma_R(G_1 \vee G_2) = \begin{cases} 2 & \text{si } \Delta(G_i) = |V(G_i)| - 1 \text{ p.a. } i, \\ 3 & \text{si } \Delta(G_i) < |V(G_i)| - 1 \forall i \text{ y } \Delta(G_i) = |V(G_i)| - 2 \text{ p.a. } i, \\ 4 & \text{en otro caso.} \end{cases}$$

Demostración. **Caso 1.** Supongamos que $\Delta(G_1) = |V(G_1)| - 1$ (el caso $\Delta(G_2) = |V(G_2)| - 1$ es análogo). Entonces G_1 tiene un vértice universal v . En el grafo $G_1 \vee G_2$, v es adyacente a cada vértice en G_2 , luego resulta que v es un vértice universal de $G_1 \vee G_2$ y por la Proposición 4 resulta que $\gamma_R(G_1 \vee G_2) = 2$.

Caso 2. Supongamos ahora que tanto G_1 como G_2 no tienen un vértice universal. Entonces, tampoco lo tiene $G_1 \vee G_2$. Resulta entonces de la Proposición 10 que $\gamma_R(G_1 \vee G_2) \geq 3$. Ahora bien, supongamos que $\Delta(G_1) = |V(G_1)| - 2$ (el caso $\Delta(G_2) = |V(G_2)| - 2$ es análogo). Es decir, existen un vértice v que es quasi-universal en G_1 y un único vértice en $w \in V(G_1)$ que no es adyacente a v . En el grafo $G_1 \vee G_2$, v es adyacente a cada vértice en G_2 , luego resulta que v es un vértice quasi-universal de $G_1 \vee G_2$, ya que $N_{G_1 \vee G_2}[v] = V(G_1 \vee G_2) - \{w\}$.

Observemos que $|V(G_1 \vee G_2)| \geq 3$ ya que de lo contrario, tendríamos que como

$$|V(G_1 \vee G_2)| = |V(G_1)| + |V(G_2)| \geq 2$$

entonces $|V(G_1 \vee G_2)| = 2$ y tanto G_1 como G_2 resultan copias de K_1 y el único vértice de cada uno de los grafos sería universal, con lo cual estaríamos en el caso 1.

Entonces, por el Lema 5 resulta que $\gamma_R(G_1 \vee G_2) = 3$.

Caso 3. Supongamos ahora que tanto G_1 como G_2 no tienen ningún vértice universal ni quasi-universal. Entonces, tampoco lo tendrá $G_1 \vee G_2$. Luego, por la Proposición 10 y el Lema 5 tenemos que

$$\gamma_R(G_1 \vee G_2) \geq 4. \quad (2.2.12)$$

Por otro lado, sea $v_1 \in V(G_1)$ y $v_2 \in V(G_2)$. La función $f = (V(G_1 \vee G_2) - \{v_1, v_2\}, \emptyset, \{v_1, v_2\})$ es una FDR sobre $G_1 \vee G_2$. En efecto, sea v tal que $f(v) = 0$. Si $v \in V(G_1)$, entonces es adyacente en $G_1 \vee G_2$ a v_2 , y $f(v_2) = 0$; y si $v \in V(G_2)$, entonces es adyacente en $G_1 \vee G_2$ a v_1 , y $f(v_1) = 0$. Luego tenemos que

$$\gamma_R(G_1 \vee G_2) \leq f(V(G_1 \vee G_2)) = 4. \quad (2.2.13)$$

Entonces, de la ecuación (2.2.12) y la ecuación (2.2.13) tenemos que $\gamma_R(G_1 \vee G_2) = 4$. \square

Observación 6. Este resultado se puede generalizar para el grafo join de una cantidad finita de grafos. Si G_1, \dots, G_k son k grafos cualesquiera cuyos conjuntos de vértices son disjuntos dos a dos, entonces

$$\gamma_R(G_1 \vee \dots \vee G_k) = \begin{cases} 2 & \text{si } \Delta(G_i) = |V(G_i)| - 1 \text{ p.a. } i, \\ 3 & \text{si } \Delta(G_i) < |V(G_i)| - 1 \forall i \text{ y } \Delta(G_i) = |V(G_i)| - 2 \text{ p.a. } i, \\ 4 & \text{en otro caso.} \end{cases}$$

2.3. Un algoritmo para calcular γ_R

En esta sección vamos a dar un algoritmo general para calcular el número de dominación romana de un grafo G , $\gamma_R(G)$.

Para todo grafo G se verifica exactamente uno de los siguientes enunciados:

1. G es desconexo
2. \overline{G} es desconexo
3. G es modular

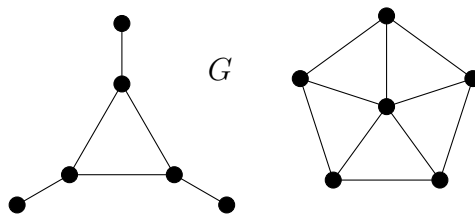
Todo grafo G desconexo puede pensarse como unión disjunta de sus componentes conexas. Todo grafo G cuyo complemento es desconexo es el resultado de hacer la operación join entre los subgrafos de G inducidos por los vértices de cada componente conexa de \overline{G} .

Por otro lado, las familias de grafos con pocos P_4 's presentadas en la sección anterior, son familias hereditarias. Una familia \mathcal{F} se dice *hereditaria* si para todo grafo $G \in \mathcal{F}$ y H subgrafo inducido de G , se verifica $H \in \mathcal{F}$.

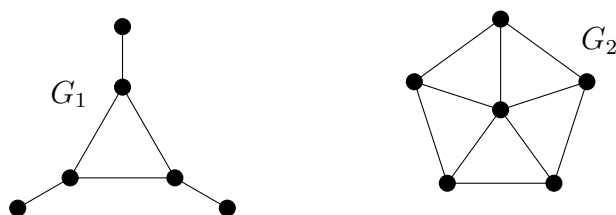
Entonces, podemos reescribir la clasificación dada más arriba en los siguientes términos. Para todo grafo $G \in \mathcal{F}$, donde \mathcal{F} es una familia de grafos hereditaria, se verifica exactamente uno de los siguientes enunciados:

1. G es unión disjunta de una cantidad finita de grafos en \mathcal{F}
2. G es join de una cantidad finita de grafos en \mathcal{F}
3. G es un grafo modular en \mathcal{F}

Ejemplo 9. Consideremos el siguiente grafo G



G es desconexo, y puede expresarse como unión disjunta de los grafos G_1 y G_2 .



Observando cada uno de estos dos grafos, vemos que G_1 es un grafo modular. Es decir, es conexo y su complemento también. Por otro lado G_2 es desconexo, y podemos ver que G_2 puede expresarse como la join de K_1 y C_5 , cada uno de los cuales, a su vez, es modular.

$$G = G_1 \cup (K_1 \vee C_5)$$

Si conocemos cuáles son los grafos modulares de una familia de grafos determinada \mathcal{F} , y podemos calcular el número de dominación romana γ_R en ellos, como ya conocemos la respuesta del parámetro γ_R a las operaciones *join* y *unión disjunta* y la misma se calcula en tiempo lineal conociendo el valor del parámetro en los grafos involucrados, podemos usar esta clasificación para calcular $\gamma_R(G)$ en un grafo $G \in \mathcal{F}$.

Ejemplo 10. El grafo G del ejemplo 9 puede pensarse como unión disjunta de los grafos G_1 y G_2 . Entonces, por el Teorema 8 tenemos que

$$\gamma_R(G) = \gamma_R(G_1) + \gamma_R(G_2)$$

G_1 es un grafo modular, y podemos ver que se trata de una araña fina sin cabeza de 3 patas. Luego, por el Teorema 10 sabemos que

$$\gamma_R(G_1) = 3 + 1 = 4$$

Por otro lado, G_2 es join de K_1 y C_5 . Por el Teorema 9, para determinar $\gamma_R(G_2)$ solo basta con chequear los grados máximos de K_1 y C_5 . Vemos que se verifica $\Delta(K_1) = 0 = |V(K_1)| - 1$. Entonces tenemos que

$$\gamma_R(G_2) = \gamma_R(K_1 \vee C_5) = 2$$

Entonces, concluimos que

$$\gamma_R(G) = \underbrace{\gamma_R(G_1)}_4 + \underbrace{\gamma_R(G_2)}_2 = 6$$

Basados en el procedimiento de descomponer el grafo mediante las operaciones join y unión disjunta, hasta llegar a grafos modulares (descomposición modular) y analizando el comportamiento de γ_R para estas operaciones, presentamos el siguiente algoritmo:

Algoritmo 2.1: Algoritmo para obtener γ_R para un grafo G

```

1  input G grafo
2  output drom(G) #numero de dominacion romana de G
3
4  if G es desconexo then:
5      d = 0
6      for H componente conexa de G do:
7          d += drom(H)
8      return d
9  else if complemento(G) desconexo then:
10     G join de G1, ..., Gk
11     if algun Gi tiene un vertice universal then:
12         return 2
13     else if algun Gi tiene un vertice quasi-universal then:
14         return 3
15     else:
16         return 4
17 else:
18     d = numero de dominacion romana de G grafo modular
19     return d

```

De los Teoremas 8 y 9 tenemos que este algoritmo resuelve el Problema de Dominación Romana para un grafo G cualquiera, y la complejidad del mismo dependerá de la complejidad del cálculo de $\gamma_R(G)$ para los grafos modulares, ya que el cálculo para la unión disjunta o join de dos grafos se consigue en tiempo lineal (conociendo $\gamma_R(G_i)$ para los grafos que intervienen en el caso de la unión disjunta).

Observemos además, que en esta descomposición del grafo con las operaciones join y unión disjunta no es necesario desarrollar todo el árbol de descomposición modular del grafo, ya que si en algún punto \overline{G} es desconexo, y se expresa $G = H_1 \vee \dots \vee H_k$, no hace falta continuar la descomposición de cada grafo H_i , sino solamente analizar los grados máximos de cada uno de ellos y compararlos.

Capítulo 3

Dominación romana en grafos araña

Ya vimos que el Problema de Dominación Romana es difícil de resolver. Su problema de decisión asociado es NP-completo, incluso cuando nos restringimos a algunas familias de grafos como los grafos split.

En este Capítulo vamos a estudiar el problema en una subfamilia de los grafos split, y mostraremos que si nos restringimos a esta clase de grafos, el problema es resoluble en tiempo lineal. Se trata de los *grafos araña sin cabeza*.

Además, extenderemos el estudio a familias de arañas más generales, que juegan un rol importante en la descomposición de algunas familias de grafos con pocos P_4 's.

3.1. Grafos araña

Vamos a presentar una familia particular de grafos *split*, dada por los llamados *grafos araña sin cabeza*.

Una *araña sin cabeza* es un grafo split G con partición (C, S) tal que $|S| = |C| = r$, con $r \geq 2$ y si $S = \{s_1, \dots, s_r\}$ y $C = \{c_1, \dots, c_r\}$, exactamente una de las siguientes condiciones se verifica:

- s_i es adyacente a c_j si y solo si $i = j$, en este caso diremos que es una *araña fina*;
- s_i es adyacente a c_j si y solo si $i \neq j$ en este caso diremos que es una *araña gruesa*.

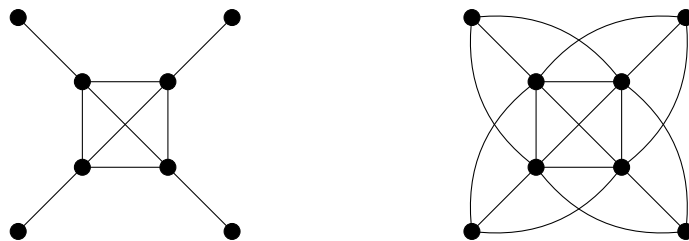


Figura 3.1: Ejemplos de araña fina sin cabeza (izq.) y araña gruesa sin cabeza (der.)

La clase de los grafos araña sin cabeza se generaliza definiendo la clase de los *grafos araña*.

Un *grafo araña* es un grafo G cuyo conjunto de vértices puede ser particionado en tres conjuntos (S, C, H) , tales que $G[S \cup C]$ es una araña sin cabeza, $N(S) \cap H = \emptyset$ y $G[C \cup H] = G[C] \vee G[H]$. Si $G[S \cup C]$ es una araña sin cabeza fina (resp. gruesa) decimos que G es una araña fina (resp. gruesa).

Las aristas que tienen un extremo en S se llaman *patas* y H se llama *cabeza*. Además, notaremos a una araña por su partición $G = (S, C, H)$.

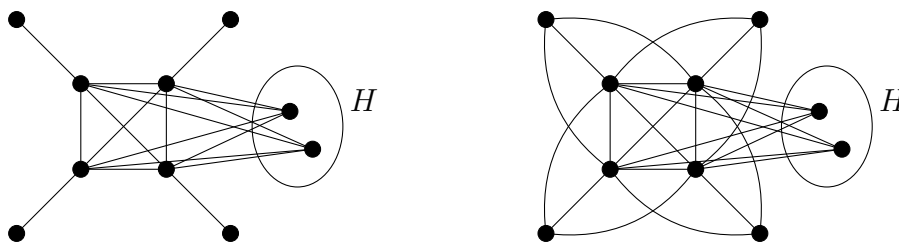


Figura 3.2: Ejemplos de araña fina (izq.) y araña gruesa (der.)

Observación 7. Si G es una araña fina, entonces \overline{G} es una araña gruesa, y viceversa.

3.1.1. Arañas finas

Teorema 10. Sea $G = (S, C, H)$ una araña fina con $|S| = |C| = r$. Entonces $\gamma_R(G) = r + 1$.

Demostración. Sea f una FDR sobre G . Observemos que para cada $i = 1, 2, \dots, r$ tenemos que $N[s_i] = \{s_i, c_i\}$. Por el Lema 3,

$$f(s_i) + f(c_i) \geq 1. \quad (3.1.1)$$

Entonces

$$f(V) = \sum_{v \in V} f(v) = \sum_{i=1}^r \underbrace{f(s_i) + f(c_i)}_{\geq 1} + \sum_{v \in H} \underbrace{f(v)}_{\geq 0} \geq r. \quad (3.1.2)$$

Veamos ahora que $f(V) > r$. Como ya hemos probado que $f(V) \geq r$, basta solo ver que $f(V) \neq r$.

Supongamos por el contrario que $f(V) = r$. Esto es, (3.1.2) se da por igualdad. En consecuencia (3.1.1) se da por igualdad para cada $i = 1, 2, \dots, r$, y vale $f(v) = 0$ para cada $v \in H$. Por el Lema 3 resulta que $f(s_i) = 1$ y $f(c_i) = 0$ para cada $i = 1, 2, \dots, r$. Luego tenemos que $V_0 = C \cup H \neq \emptyset$, $V_1 = S$, $V_2 = \emptyset$, lo que contradice el Lema 1. La contradicción surge de suponer que $f(V) = r$. Concluimos entonces que $f(V) > r$. Por lo tanto

$$\gamma_R(G) \geq r + 1. \quad (3.1.3)$$

Veamos ahora que existe una FDR f de G con $f(V) = r + 1$.

Definimos $\hat{f} : V = (S, C, H) \rightarrow \{0, 1, 2\}$ como sigue:

$$\hat{f}(s_i) = \begin{cases} 0 & \text{si } i = 1, \\ 1 & \text{si } i \neq 1. \end{cases} \quad \hat{f}(c_i) = \begin{cases} 2 & \text{si } i = 1, \\ 0 & \text{si } i \neq 1. \end{cases} \quad \text{y} \quad \hat{f}(v) = 0 \quad \forall v \in H$$

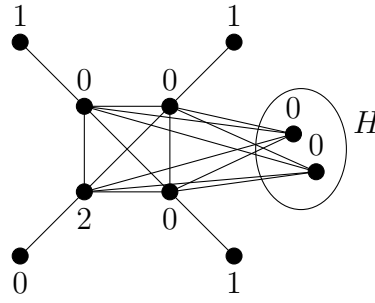
Así, como $V_0 = H \cup \{s_1\} \cup (C - \{c_1\}) = N(c_1)$, con $c_1 \in V_2$, resulta \hat{f} una FDR de G , con $\hat{f}(V) = r + 1$. Por lo tanto

$$\gamma_R(G) \leq r + 1. \quad (3.1.4)$$

De (3.1.3) y (3.1.4) concluimos

$$\gamma_R(G) = r + 1.$$

□

Figura 3.3: Ejemplo de γ_R -función para una araña fina

Corolario 2. Sean $G = (C, S, H)$ una araña fina y $v \in S \cup C$. Entonces existe una γ_R -función f tal que $f(v) = 0$.

Demostración. Sean $S = \{s_1, s_2, \dots, s_r\}$, $C = \{c_1, c_2, \dots, c_r\}$ y s_i es adyacente a c_j si y solo si $i = j$.

Si $v = s_k$, para algún $k = 1, \dots, r$, entonces definimos $f : V \rightarrow \{0, 1, 2\}$ como sigue:

$$f(s_i) = \begin{cases} 0 & \text{si } i = k, \\ 1 & \text{si } i \neq k. \end{cases} \quad f(c_i) = \begin{cases} 2 & \text{si } i = k, \\ 0 & \text{si } i \neq k. \end{cases} \quad \text{y} \quad f(v) = 0 \quad \forall v \in H$$

Como $V_0 = H \cup \{s_k\} \cup (C - \{c_k\}) = N(c_k)$, con $c_k \in V_2$, resulta f una FDR de G , con $f(V) = r + 1 = \gamma_R(G)$.

Es decir, f es una γ_R -función con $f(v) = 0$.

Si ahora $v = c_k$, para algún $k = 1, \dots, r$, entonces consideramos $j \in \{1, \dots, r\}$, $j \neq k$, y definimos $f : V \rightarrow \{0, 1, 2\}$ de la siguiente manera:

$$f(s_i) = \begin{cases} 0 & \text{si } i = j, \\ 1 & \text{si } i \neq j. \end{cases} \quad f(c_i) = \begin{cases} 2 & \text{si } i = j, \\ 0 & \text{si } i \neq j. \end{cases} \quad \text{y} \quad f(v) = 0 \quad \forall v \in H$$

Como $V_0 = H \cup \{s_j\} \cup (C - \{c_j\}) = N(c_j)$, con $c_j \in V_2$, resulta f una FDR de G , con $f(V) = r + 1 = \gamma_R(G)$.

Y nuevamente f es una γ_R -función con $f(v) = 0$. □

3.1.2. Arañas gruesas

Teorema 11. Sea $G = (S, C, H)$ una araña gruesa. Entonces $\gamma_R(G) = 3$.

Demostración. Observemos que $\Delta(G) = |V| - 2$, ya que cualquier vértice c_i del cuerpo verifica $N[c_i] = C \cup H \cup (S - \{s_i\}) = V - \{s_i\}$. Luego, por el Lema 5 tenemos que $\gamma_R(G) = 3$. □

Corolario 3. Sean $G = (C, S, H)$ una araña gruesa y $v \in S \cup C$. Entonces existe una γ_R -función f tal que $f(v) = 0$.

Demostración. Sean $S = \{s_1, \dots, s_r\}$, $C = \{c_1, \dots, c_r\}$ y s_i adyacente a c_j si y solo si $i \neq j$.

Si $v = s_k$, para algún $k = 1, \dots, r$, entonces definimos $f : V \rightarrow \{0, 1, 2\}$ como sigue:

$$f(s_i) = \begin{cases} 1 & \text{si } i = k, \\ 0 & \text{si } i \neq k. \end{cases} \quad f(c_i) = \begin{cases} 2 & \text{si } i = k, \\ 0 & \text{si } i \neq k. \end{cases} \quad \text{y} \quad f(v) = 0 \quad \forall v \in H$$

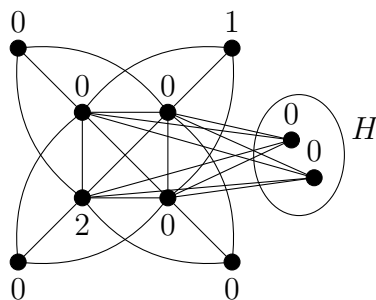


Figura 3.4: Ejemplo de dominación romana óptima para una araña gruesa

Como $V_0 = H \cup (S - \{s_k\}) \cup (C - \{c_k\}) = N(c_k)$, con $c_k \in V_2$, resulta f una FDR de G , con $f(V) = 3 = \gamma_R(G)$.

Es decir, f es una γ_R -función con $f(v) = 0$.

Si ahora $v = c_k$, para algún $k = 1, \dots, r$, entonces consideramos $j = 1, \dots, r$, $j \neq k$, y definimos $f : V \rightarrow \{0, 1, 2\}$ de la siguiente manera:

$$f(s_i) = \begin{cases} 1 & \text{si } i = j, \\ 0 & \text{si } i \neq j. \end{cases} \quad f(c_i) = \begin{cases} 2 & \text{si } i = j, \\ 0 & \text{si } i \neq j. \end{cases} \quad \text{y} \quad f(v) = 0 \quad \forall v \in H$$

Como $V_0 = H \cup (S - \{s_j\}) \cup (C - \{c_j\}) = N(c_j)$, con $c_j \in V_2$, resulta f una FDR de G , con $f(V) = 3 = \gamma_R(G)$.

Y nuevamente f es una γ_R -función con $f(v) = 0$. □

3.1.3. Arañas sin cabeza

Las arañas sin cabeza forman una subfamilia de los grafos split, para los cuales el Problema de Dominación Romana es NP-completo [36]. La partición (S, C) de una araña se calcula en tiempo lineal [35]. Como para calcular $\gamma_R(G)$ para G araña sin cabeza solo se necesita conocer el valor de $|S| = |C| = r$, tenemos el siguiente resultado:

Corolario 4. *La familia de grafos araña sin cabeza son una subfamilia de los grafos split para la cual el Problema de Dominación Romana se resuelve en tiempo lineal.*

Observación 8. *Más aún, si sabemos que un grafo es un grafo araña, basta con chequear la cantidad de vértices pendientes. Si no tiene ninguno, se trata de una araña gruesa y $\gamma_R(G) = 3$. Si tiene alguno, se trata de una araña fina y la cantidad de vértices pendientes coincide con r y $\gamma_R(G) = r + 1$, es la cantidad de vértices pendientes más uno.*

3.2. Grafos quasi-araña

Giakoumakis et al. definen en [38] una familia de grafos que generalizan a los grafos araña: la familia de los *grafos quasi-araña*.

Un grafo *quasi-araña* es un grafo obtenido a partir de una araña (S, C, H) (fina o gruesa) donde a lo sumo un vértice en $S \cup C$ es reemplazado por un grafo con dos vértices.

Esta misma definición puede reestablecerse en función de la operación de agregar un mellizo (falso o verdadero). Un *grafo quasi-araña* se obtiene a partir de un grafo araña (S, C, H) agregándole un mellizo a lo sumo a un vértice en $S \cup C$.



Figura 3.5: Grafos con dos vértices

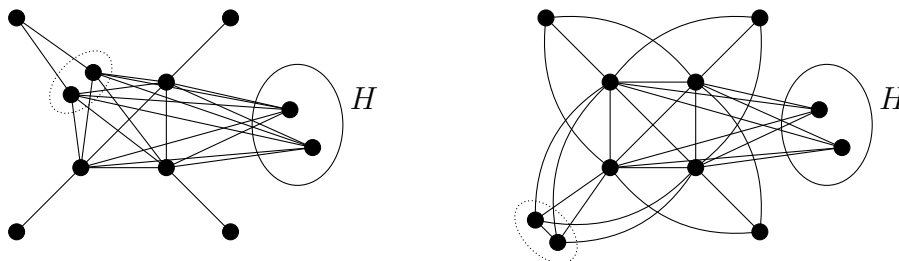


Figura 3.6: Ejemplos de grafos quasi-araña

Teorema 12. *Sea G un grafo quasi-araña que se obtiene a partir de una araña $G' = (S, C, H)$. Entonces $\gamma_R(G) = \gamma_R(G')$*

Demostración. G es una quasi-araña que se obtiene añadiendo un mellizo (falso o verdadero) a un vértice v de $S \cup C$.

Por los corolarios 2 y 3, vemos que siempre es posible encontrar una γ_R -función sobre G' que le asigne al vértice v el valor 0. De esta manera, utilizando el Teorema 6 tenemos que $\gamma_R(G') = \gamma_R(G)$.

□

3.3. Grafos araña débilmente etiquetados

Roussel et al. [35] introducen la clase de *grafos araña débilmente etiquetados*, que generalizan a los grafos quasi-araña.

Dado un grafo araña fina (S, C, H) , y una función $L : S \cup C \rightarrow \mathbb{N}$, un *grafo araña etiquetado* (S, C, H, L) es un grafo obtenido reemplazando todo vértice $v \in S \cup C$ por un grafo G_v tal que $|V(G_v)| = L(v)$. Dado un grafo araña etiquetado (S, C, H, L) y una arista uv del grafo araña (S, C, H) , la etiqueta de uv es $L(u) + L(v)$.

Un *grafo araña débilmente etiquetado* es un grafo araña etiquetado (S, C, H, L) tal que $L(v) \in \{1, 2, 3\}$ para cada $v \in S \cup C$ y además, se verifica una de las siguientes condiciones:

- I. exactamente una pata tiene etiqueta 4 y todas las patas restantes tienen etiqueta 2, o bien
- II. toda pata tiene etiqueta a lo sumo 3.

Observación 9. *Los grafos araña fina son grafos araña débilmente etiquetados con $L(v) = 1$ para todo $v \in S \cup C$.*

En [1] se reestablece esta última definición en términos de las siguientes operaciones sobre un grafo araña fina (S, C, H) :

R1: reemplazar un único vértice $v \in S \cup C$ por un grafo G con tres vértices.

R2: reemplazar ambos extremos de una única pata de (S, C, R) por grafos con dos vértices.

R3: reemplazar a lo sumo un extremo de cada pata de (S, C, R) por un grafo con dos vértices

Las operaciones R1 y R2 corresponden a las posibles sustituciones que verifican el ítem I en la definición de Roussel et al. y la operación R3 corresponde a la sustitución que verifica el ítem II, y recíprocamente.

Así, un *grafo araña débilmente etiquetado* es aquel que se obtiene a partir de un grafo araña fina (S, C, H) realizando una de las operaciones R1, R2, R3 una vez.

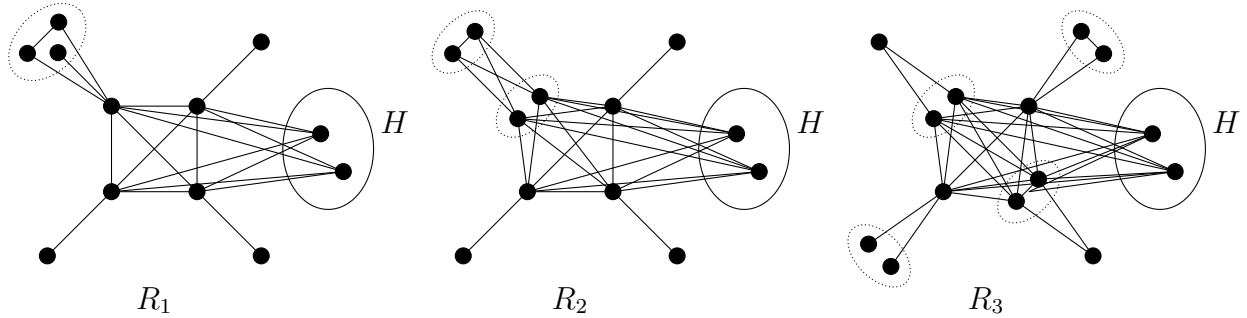


Figura 3.7: Ejemplos de grafos araña débilmente etiquetados

Al realizar una de estas operaciones (R1, R2, R3), es indistinto cual sea el vértice o la pata que se use. Los grafos obtenidos resultan isomorfos. Esto permite las siguientes notaciones:

Un grafo araña débilmente etiquetado obtenido por la operación R1 se denomina *grafo 3-araña débilmente etiquetado* y lo identificamos por su partición denotada $(S \leftrightarrow G, C, H)$ o $(S, C \leftrightarrow G, H)$ si el vértice reemplazado v pertenece a S o C respectivamente, y G es el grafo que reemplaza a v . En la Figura 3.8 se ven todos los posibles grafos con tres vértices, salvo isomorfismos.

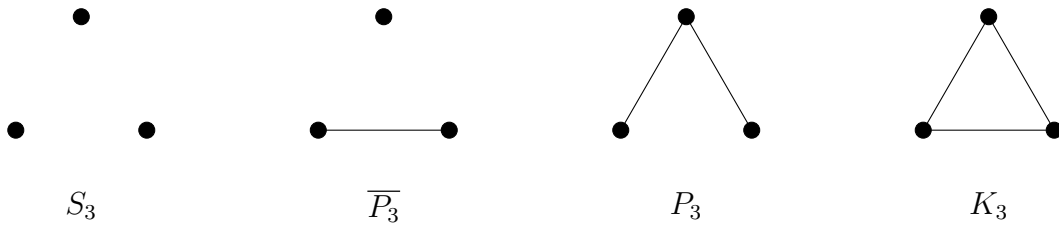


Figura 3.8: Grafos con tres vértices

Un grafo araña débilmente etiquetado obtenido por la operación R2 se denomina *grafo (2,2)-araña débilmente etiquetado* y se identifica por su partición denotada $(S \leftrightarrow G_s, C \leftrightarrow G_c, H)$, donde G_s y G_c son los grafos que reemplazan los extremos de una pata.

Finalmente, para aquellos grafos obtenidos por la operación R3, diremos que tenemos un *grafo araña múltiple débilmente etiquetado* y se identifica con su partición denotada $(S_{\sigma, \kappa}, C_{\gamma, \eta}, H)$ donde $\sigma, \kappa, \gamma, \eta$ denotan el número de vértices en S reemplazados por S_2 , el número de vértices en S reemplazados por K_2 , el número de vértices en C reemplazados por S_2 , el número de vértices en C reemplazados por K_2 , respectivamente.

Teorema 13. *Sea $G(S \leftrightarrow G', C, H)$ o $G = (S, C \leftrightarrow G', H)$ un grafo 3-araña débilmente etiquetado con $|S| = |C| = r$. Entonces $\gamma_R(G) = r + 1$, que coincide con el número de dominación romana de la araña subyacente.*

Demostración. Sea $v \in S \cup C$ el vértice que es reemplazado por el grafo G' de orden 3. Observemos que este reemplazo puede pensarse como agregar iterativamente un mellizo, dos veces, a un mismo vértice $v \in S \cup C$.

- Si $G' = S_3$, el reemplazo de v por G' es equivalente a agregar un mellizo falso a v , y después agregar otro mellizo falso a v .
- Si $G' = \overline{P}_3$, el reemplazo de v por G' es equivalente a agregar un mellizo falso a v , y después agregar un mellizo verdadero a v .
- Si $G' = P_3$, el reemplazo de v por G' es equivalente a agregar un mellizo verdadero a v , y después agregar un mellizo falso a v .
- Si $G' = K_3$, el reemplazo de v por G' es equivalente a agregar un mellizo verdadero a v , y después agregar otro mellizo verdadero a v .

De cualquier manera, tenemos que G se obtiene a partir de \tilde{G} mediante agregado de mellizos, y luego por el Lema 6 tenemos que

$$\gamma_R(G) \geq \gamma_R(\tilde{G}) = r + 1. \tag{3.3.1}$$

Veamos ahora que existe una FDR f de G con $f(V) = r + 1$. Analicemos primero el caso en que el vértice reemplazado es $v \in S$.

Definimos $\hat{f} : V(G) \rightarrow \{0, 1, 2\}$ dada por:

$$\hat{f}(s_i) = 1 \quad \forall i = 2, \dots, r \quad \hat{f}(c_i) = \begin{cases} 2 & \text{si } i = 1, \\ 0 & \text{si } i \neq 1. \end{cases} \quad \hat{f}(v) = 0 \quad \forall v \in H \cup \{u_1, u_2, u_3\}$$

Así, como $V_0 = H \cup \{u_1, u_2, u_3\} \cup (C - \{c_1\}) = N(c_1)$, con $c_1 \in V_2$, resulta \hat{f} una FDR de G , con $\hat{f}(V) = r + 1$. Por lo tanto

$$\gamma_R(G) \leq r + 1. \tag{3.3.2}$$

A partir de la ecuación (3.3.1) y la ecuación (3.3.2) concluimos que

$$\gamma_R(G) = r + 1.$$

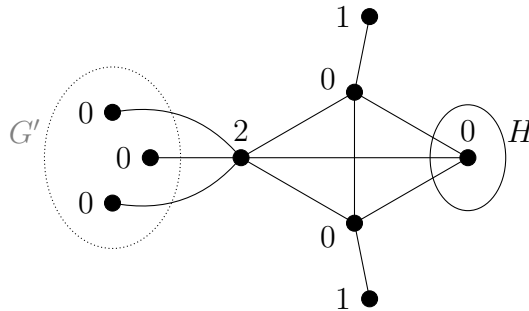


Figura 3.9: Ejemplo de FDR sobre un grafo 3-araña

De manera similiar, si el vértice reemplazado es $v \in C$, definimos $\hat{f} : V(G) \rightarrow \{0, 1, 2\}$ dada por:

$$\hat{f}(s_i) = 1 \quad \forall i = 2, \dots, r \quad \hat{f}(c_i) = \begin{cases} 2 & \text{si } i = 2, \\ 0 & \text{si } i \neq 2. \end{cases} \quad \hat{f}(v) = 0 \quad \forall v \in H \cup \{u_1, u_2, u_3\}$$

Así, como $V_0 = H \cup \{u_1, u_2, u_3\} \cup (C - \{c_1, c_2\}) = N(c_2)$, con $c_2 \in V_2$, resulta \hat{f} una FDR de G , con $\hat{f}(V) = r + 1$. Por lo tanto

$$\gamma_R(G) \leq r + 1. \quad (3.3.3)$$

De la ecuación (3.3.1) y la ecuación (3.3.3) concluimos

$$\gamma_R(G) = r + 1.$$

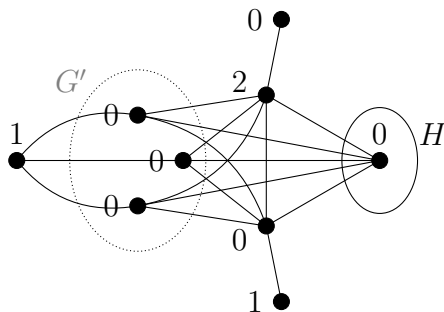


Figura 3.10: Ejemplo de FDR sobre un grafo 3-araña

□

Teorema 14. Sea G un grafo $(2, 2)$ -araña débilmente etiquetado ($S \leftrightarrow G_s, C \leftrightarrow G_c, H$) con $|S| = |C| = r$. Entonces

$$\gamma_R(G) = \begin{cases} r + 1, & \text{si } G_c = K_2 \\ r + 2, & \text{si } G_c = S_2 \end{cases}$$

Demostración. G se obtiene a partir de la araña fina \tilde{G} de partición (S, C, H) donde ambos extremos de una única pata $s_i c_i$ son reemplazados por grafos de 2 vértices G_s y G_c respectivamente. Observemos que estos reemplazos pueden pensarse como agregar un mellizo a s_i (verdadero si $G_s = K_2$, falso si $G_s = S_2$) y un mellizo a c_i (verdadero si $G_s = K_2$, falso si $G_s = S_2$).

Tenemos entonces que G se obtiene a partir de \tilde{G} mediante agregado de mellizos, y luego por el Lema 6 tenemos que

$$\gamma_R(G) \geq \gamma_R(\tilde{G}) = r + 1 \quad (3.3.4)$$

Veamos que si $G_c = K_2$ es posible encontrar una FDR \hat{f} con $\hat{f}(V) = r + 1$. Sin perder generalidad, podemos pensar que la pata modificada es $s_1 c_1$, y reemplazamos s_1 por un grafo G_s con conjunto de vértices $\{u_1, u_2\}$ y c_1 por un K_2 con conjunto de vértices $\{v_1, v_2\}$. Consideremos el conjunto de vértices $U = \{u_1, u_2, v_1, v_2\}$. Definimos $\hat{f} : V(G) \rightarrow \{0, 1, 2\}$ dada por la partición

$$V_0 = (U - \{v_1\}) \cup \{c_2, \dots, c_r\} \cup H, \quad V_1 = \{s_2, \dots, s_r\}, \quad V_2 = \{v_1\}$$

Como $V_0 = N(v_1)$ y $\hat{f}(v_1) = 2$, \hat{f} es efectivamente una FDR, y

$$\gamma_R(G) \leq \hat{f}(V(G)) = r + 1 \quad (3.3.5)$$

Así, por las ecuaciones la ecuación (3.3.4) y la ecuación (3.3.5) tenemos que si $G_c = K_2$ entonces

$$\gamma_R(G) = r + 1 \quad (3.3.6)$$

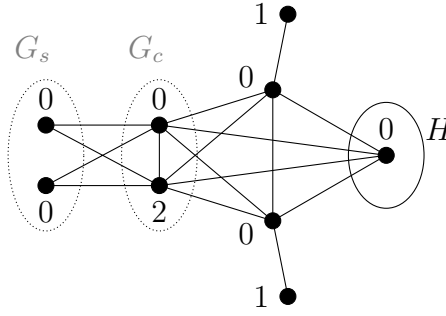


Figura 3.11: Ejemplo de FDR sobre un grafo (2, 2)-araña

Analicemos ahora qué pasa si $G_c = S_2$. Nuevamente, sin perder generalidad, podemos pensar que la pata modificada es s_1c_1 , y reemplazamos s_1 por un grafo G_s con conjunto de vértices $\{u_1, u_2\}$ y c_1 por S_2 con conjunto de vértices $\{v_1, v_2\}$. Por la ecuación (3.3.4) sabemos que

$$\gamma_R(G) \geq r + 1.$$

Sea f una γ_R -función sobre G , y supongamos que $f(V) = r + 1$.

Consideremos el conjunto de vértices $U = \{u_1, u_2, v_1, v_2\}$. Observemos que $N[u_1] \subseteq U$, con lo cual por el Lema 3

$$f(U) \geq f(N[u_1]) \geq 1$$

Si $f(U) = f(N(u_1)) = 1$, nuevamente por el Lema 3 tenemos que $f(u_1) = 1$ y $f(w) = 0$ para cada $w \in U - \{u_1\}$. De esta manera, $f(u_2) = 0$; pero como $N(u_2) \subseteq U$ y $U \cap V_2 = \emptyset$, no existe ningún vecino de u_2 al que $f = (V_0, V_1, V_2)$ le asigne el valor 2, contradiciendo que f es una FDR. Entonces, concluimos que

$$f(U) \geq 2 \quad (3.3.7)$$

Para $j = 2, \dots, r$, $N[s_j] = \{s_j, c_j\}$ y por el Lema 3 tenemos que

$$f(s_j) + f(c_j) \geq 1 \quad (3.3.8)$$

Entonces juntando las desigualdades obtenidas en la ecuación (3.3.7) y la ecuación (3.3.8)

$$f(V) = \underbrace{f(u_1) + f(u_2) + f(v_1) + f(v_2)}_{\geq 2} + \sum_{i=2}^r \underbrace{f(s_i) + f(c_i)}_{\geq 1} + \sum_{v \in H} \underbrace{f(v)}_{\geq 0} \geq r + 1 \quad (3.3.9)$$

Como $f(V) = r + 1$, la ecuación (3.3.9) se da por igualdad, y en consecuencia, la ecuación (3.3.7) se da por igualdad, la ecuación (3.3.8) se da por igualdad para cada $j = 2, \dots, r$ y $f(w) = 0$ para cada $v \in H$. Por el Lema 3 resulta $f(s_j) = 1$, $f(c_j) = 0$ para cada $j = 2, \dots, r$.

Como $c_2 \in V_0$, debe existir un vértice en $N(c_2) \cap V_2$. Observemos que $N(c_2) = \{s_2\} \cup \{v_1, v_2\} \cup (C - \{c_1, c_2\}) \cup H$. Como ya hemos visto que $(C - \{c_1, c_2\}) \cup H \subseteq V_0$ y $s_1 \in V_1$, entonces $f(v_1) = 2$ o bien $f(v_2) = 2$. En cualquiera de los dos casos, nos queda que $f(s_i) = 1$ para $i = 2, \dots, r$, $f(v_1) = 2$ (o $f(v_2) = 2$) y el resto de los vértices en 0, en particular $f(v_2) = 0$ (o $f(v_1) = 0$). La vecindad de v_2 (o v_1) está contenida en V_0 , lo que contradice que f es una FDR.

Concluimos entonces que si $G_c = S_2$, no existe ninguna FDR f con $f(V(G)) = r + 1$. Entonces,

$$\gamma_R(G) \geq r + 2 \quad (3.3.10)$$

Finalmente, veamos que es posible encontrar una FDR \hat{f} con $\hat{f}(V) = r + 2$. En efecto, consideremos $\hat{f} : V(G) \rightarrow \{0, 1, 2\}$ definida por la partición

$$V_0 = \{u_1, u_2\} \cup \{c_2, \dots, c_r\} \cup H, \quad V_1 = \{v_2\} \cup \{s_2, \dots, s_r\}, \quad V_2 = \{v_1\}$$

Como $V_0 = N(v_1)$ y $\hat{f}(v_1) = 2$, \hat{f} es efectivamente una FDR, y

$$\gamma_R(G) \leq \hat{f}(V(G)) = r + 2 \quad (3.3.11)$$

Así, por las ecuaciones la ecuación (3.3.10) y la ecuación (3.3.11) tenemos que si $G_c = S_2$ entonces

$$\gamma_R(G) = r + 2 \quad (3.3.12)$$

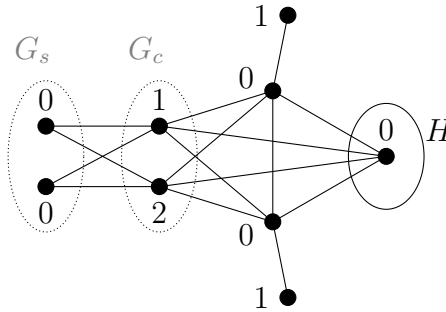


Figura 3.12: Ejemplo de FDR sobre un grafo (2,2)-araña

Juntando la ecuación (3.3.6) y la ecuación (3.3.12) tenemos que, si G es una (2,2)-araña débilmente etiquetada $G = (S \leftrightarrow G_s, C \leftrightarrow G_c, H)$, entonces

$$\gamma_R(G) = \begin{cases} r + 1, & \text{si } G_c = K_2 \\ r + 2, & \text{si } G_c = S_2 \end{cases}$$

□

Teorema 15. Sea G un grafo araña múltiple débilmente etiquetado $(S_{\sigma, \kappa}, C_{\gamma, \eta}, H)$ con $|S| = |C| = r$. Entonces

$$\gamma_R(G) = \begin{cases} r + 2, & \text{si } \gamma = r \\ r + 1, & \text{si } \gamma < r \text{ y } \sigma = \kappa = 0 \\ r + \kappa + \sigma, & \text{en otro caso} \end{cases}$$

Demostración. G es obtenido a partir de la araña fina $G' = (S, C, H)$ con $|C| = |S| = r$. Es decir, a lo sumo un extremo de cada pata $s_i c_i$ se reemplaza por un grafo G_i de dos vértices ($V(G_i) = \{u_i, v_i\}$), y $\sigma, \kappa, \gamma, \eta$ denotan el número de vértices en S reemplazados por S_2 , el número de vértices en S reemplazados por K_2 , el número de vértices en C reemplazados por S_2 , el número de vértices en C reemplazados por K_2 , respectivamente.

Sea f una γ_R -función sobre el grafo G .

Llamemos Q_i , para $i = 1, \dots, r$, al grafo inducido por la pata $s_i c_i$ después de haber hecho el reemplazo correspondiente. Y diremos que Q_i es

- de tipo 0 si no se realiza ningún reemplazo.
- de tipo 1 si se reemplaza s_i por $G_i = S_2$;
- de tipo 2 si se reemplaza s_i por $G_i = K_2$;
- de tipo 3 si se reemplaza c_i por $G_i = S_2$;
- de tipo 4 si se reemplaza c_i por $G_i = K_2$.

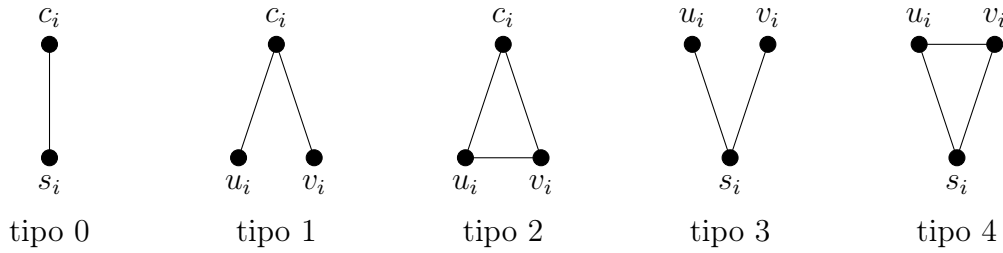


Figura 3.13: Tipos de patas en una araña múltiple débilmente etiquetada

Observemos que la cantidad de Q_i 's de tipo 1, 2, 3 y 4 son respectivamente σ, κ, γ y η , y definimos los subconjuntos $I_j = \{i = 1, \dots, r : Q_i \text{ es de tipo } j\}$, para $j = 0, 1, \dots, 4$.

Si Q_i es de tipo 0, tenemos que $V(Q_i) = \{s_i, c_i\}$, y como $N[s_i] = V(Q_i)$, por el Lema 3 tenemos que

$$f(V(Q_i)) \geq 1 \quad (3.3.13)$$

Si Q_i es de tipo 1 o de tipo 2, tenemos que $V(Q_i) = \{c_i, u_i, v_i\}$, y como $V(G_i) = \{u_i, v_i\} = pn(c_i, V(G) - V(G_i))$ y $|V(G_i)| = 2$, por el Lema 4 tenemos que

$$f(V(Q_i)) \geq 2 \quad (3.3.14)$$

Si Q_i es de tipo 3 o de tipo 4, tenemos que $V(Q_i) = \{s_i, u_i, v_i\}$, y como $N[s_i] = V(Q_i)$, por el Lema 3 tenemos que

$$f(V(Q_i)) \geq 1 \quad (3.3.15)$$

Luego, por las ecuaciones (3.3.13), (3.3.14) y (3.3.15) tenemos

$$\gamma_R(G) = f(V(G)) = \sum_{i=1}^r f(V(Q_i)) + \sum_{v \in H} \underbrace{f(v)}_{\geq 0} \geq 2(\sigma + \kappa) + (\gamma + \eta + |I_0|) \quad (3.3.16)$$

Veamos que si $\sigma \neq 0$ o $\kappa \neq 0$ entonces $\gamma_R(G) = 2(\sigma + \kappa) + (\gamma + \eta + |I_0|)$. Es decir, que existe una FDR \hat{f} sobre G tal que $\hat{f}(V(G)) = 2(\sigma + \kappa) + (\gamma + \eta + |I_0|)$. En efecto, definiendo $\hat{f} = (V_0, V_1, V_2)$ con la partición

$$V_2 = \{c_i : i \in I_1 \cup I_2\}, \quad V_1 = \{s_i : i \in I_0 \cup I_3 \cup I_4\}, \quad V_0 = V - (V_1 \cup V_2)$$

vemos que $V_2 \subseteq N(V_0)$, con lo cual es una FDR y además $\hat{f}(V(G)) = 2(\sigma + \kappa) + (\gamma + \eta + |I_0|)$. Es decir, si $\sigma \neq 0$ o $\kappa \neq 0$, entonces

$$\gamma_R(G) = 2(\sigma + \kappa) + (\gamma + \eta + |I_0|) \quad (3.3.17)$$

Y como $\gamma + \kappa + \sigma + \eta + |I_0| = r$, tenemos

$$\gamma_R(G) = r + \sigma + \kappa \quad (3.3.18)$$

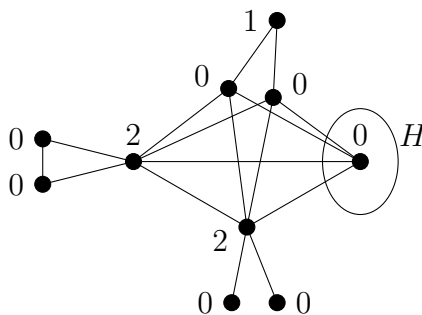


Figura 3.14: Ejemplo de FDR sobre un grafo araña múltiple

Si $\sigma = \kappa = 0$, la ecuación (3.3.16) nos queda

$$\gamma_R(G) \geq \gamma + \eta + |I_0| = r \quad (3.3.19)$$

Observemos que G se obtiene a partir de una araña fina de r patas mediante la operación de agregar mellizos, falsos o verdaderos, a vértices en SUC . Entonces, por el Lema 6 tenemos que

$$\gamma_R(G) \geq r + 1 \quad (3.3.20)$$

Si $\gamma < r$ entonces $\gamma_R(G) = r + 1$. Es decir, que existe una FDR \hat{f} sobre G tal que $\hat{f}(V(G)) = r + 1$. En efecto, si $\eta > 0$, elegimos k tal que Q_k es de tipo 4, y definimos $\hat{f} = (V_0, V_1, V_2)$ con la partición

$$V_2 = \{u_k\}, \quad V_1 = \{s_i : i \neq k\}, \quad V_0 = V - (V_1 \cup V_2)$$

Si $\eta = 0$, entonces elegimos k tal que Q_k es de tipo 0, y definimos $\hat{f} = (V_0, V_1, V_2)$ con la partición

$$V_2 = \{c_k\}, \quad V_1 = \{s_i : i \neq k\}, \quad V_0 = V - (V_1 \cup V_2)$$

En cualquiera de los dos casos, como $V_2 \subseteq N(V_0)$, \hat{f} resulta una FDR y además $\hat{f}(V(G)) = r + 1$. Es decir, si $\gamma < r$, entonces

$$\gamma_R(G) = r + 1 \quad (3.3.21)$$

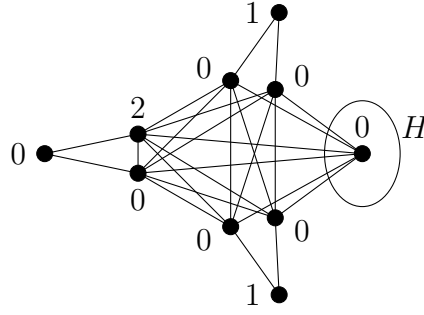


Figura 3.15: Ejemplo de FDR sobre un grafo araña múltiple

Finalmente, nos queda analizar el caso en que $\gamma = r$. Sea f una γ_R -función. Ya hemos visto que $f(V(Q_i)) \geq 1$ para cada $i = 1, \dots, r$. Si $f(V(G)) = r + 1$, entonces a lo sumo una pata Q_j verifica $f(V(Q_j)) = 2$ y para el resto vale $f(V(Q_i)) = 1$.

Para cada $i \neq j$ tenemos que $N[s_i] = V(Q_i)$, y entonces por el Lema 3 tenemos que $f(s_i) = 1$ y $f(u_i) = f(v_i) = 0$. Como $V_0 \neq \emptyset$, debe existir al menos un vértice $w \in V_2$. Tal w no puede ser un vértice de H ya que de lo contrario tendríamos que

$$r + 1 = \gamma_R(G) = f(V(G)) = \sum_{i \neq j} \underbrace{f(V(Q_i))}_{=1} + \underbrace{f(V(Q_j))}_{\geq 1} + \underbrace{f(H)}_{\geq 2} \geq r + 2$$

que es un absurdo. Entonces resulta $w \in V(Q_j)$. Además

$$f(V(Q_j)) = \underbrace{f(V(G))}_{\gamma_R(G)} - \underbrace{f(V(G) - V(Q_j))}_{r-1} = 2$$

Entonces, exactamente un vértice en $V(Q_j)$ está en V_2 y el resto está en V_0 . Pero

- Si $f(u_j) = 2$, entonces $f(v_j) = 0$ y $N(v_j) \cap V_2 = \emptyset$;
- Si $f(v_j) = 2$, entonces $f(u_j) = 0$ y $N(u_j) \cap V_2 = \emptyset$;
- Si $f(s_j) = 2$, entonces $f(v_j) = 0$ y $N(v_j) \cap V_2 = \emptyset$.

En cualquier caso, se contradice que f es una FDR. Esta contradicción viene de suponer que $\gamma_R(G) = r + 1$. Luego, concluimos que si $\gamma = r$ entonces

$$\gamma_R(G) \geq r + 2 \tag{3.3.22}$$

Para terminar esta demostración, veamos que existe una FDR \hat{f} sobre G tal que $\hat{f}(V(G)) = r + 2$. En efecto, definimos $\hat{f} = (V_0, V_1, V_2)$ con la partición

$$V_2 = \{u_1\}, \quad V_1 = \{s_i : i \neq 1\} \cup \{v_1\}, \quad V_0 = V - (V_1 \cup V_2)$$

Como $V_0 = N(u_1)$ y $u_1 \in V_2$, \hat{f} resulta una FDR y además $\hat{f}(V(G)) = r + 2$. Es decir, si $\gamma = r$, entonces

$$\gamma_R(G) = r + 2 \tag{3.3.23}$$

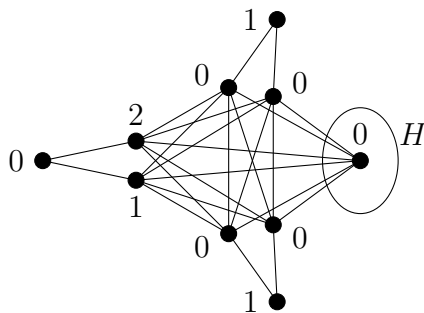


Figura 3.16: Ejemplo de FDR sobre un grafo araña múltiple

Juntando la ecuación (3.3.18), la ecuación (3.3.21) y la ecuación (3.3.23), hemos probado entonces que si G es una araña múltiple débilmente etiquetada $G = (S_{\sigma, \kappa}, C_{\gamma, \eta}, H)$, entonces

$$\gamma_R(G) = \begin{cases} r + 2, & \text{si } \gamma = r \\ r + 1, & \text{si } \gamma < r \text{ y } \sigma = \kappa = 0 \\ r + \kappa + \sigma, & \text{en otro caso} \end{cases}$$

□

Conclusión del capítulo

En este capítulo hemos analizado cuál es el valor del número de dominación romana γ_R para algunas familias de grafos araña. Éstas juegan un papel fundamental en la descomposición de ciertas familias con pocos P_4 's que veremos en el capítulo a continuación.

Cabe destacar que tanto en las arañas finas o gruesas, quasi-arañas como en las arañas débilmente etiquetadas, el valor de γ_R no depende del grafo generado por la cabeza. Y la complejidad computacional del cálculo de $\gamma_R(G)$ depende de la cantidad de operaciones requeridas para obtener la partición de la araña subyacente y los reemplazos realizados.

Capítulo 4

Dominación romana en grafos con pocos P_4 's

Algunas familias de grafos que se definen por contener, en un sentido local, solo un número restringido de P_4 's. Estas familias han ido cobrando relevancia en los últimos años [2, 35, 38].

Una de las razones que motivaron su estudio es que estos grafos tienen propiedades estructurales que han sido de gran ayuda para resolver eficientemente algunos problemas que son difíciles en general.

En este Capítulo, estudiaremos el Problema de Dominación Romana para las familias de grafos llamadas con *pocos* P_4 's, a fin de analizar la complejidad del problema en cada una de ellas, y buscar algoritmos específicos eficientes para algunas de estas familias, como los grafos P_4 -sparse y P_4 -tidy.

4.1. Familias de grafos con pocos P_4 's

En [26] Jamison y Olariu extendieron el análisis de los cografos a familias de grafos con un número limitado de P_4 's. Entre ellas los grafos P_4 -reducibles definidos por la condición de que todo vértice pertenece a lo sumo a un P_4 . Claramente los cografos forman una subfamilia propia de los grafos P_4 -reducibles, ya que, por ejemplo, el mismo P_4 es P_4 -reducible pero no es un cografo.

Hoáng [23] introdujo los grafos P_4 -sparse como aquellos para los cuales todo conjunto de cinco vértices induce a lo sumo un P_4 . Los grafos P_4 -sparse extienden estrictamente a los grafos P_4 -reducibles.

Giakoumakis y Vanherpe [18] estudiaron clases extendidas de los grafos P_4 -reducibles y P_4 -sparse. Estas clases son obtenidas a partir de las familias P_4 -reducible y P_4 -sparse respectivamente, permitiendo C_5 como componente p -conexa. Llamaron a estas clases de grafos P_4 -reducible extendidos y P_4 -sparse extendidos respectivamente. Todo grafo P_4 -reducible extendido es P_4 -sparse extendido, pero no recíprocamente. Por ejemplo, la araña fina sin cabeza de 3 patas es P_4 -sparse (extendido) pero no P_4 -reducible (extendido). Además, por ejemplo C_5 es P_4 -reducible extendido pero no es P_4 -sparse.

Otra de las familias estudiadas en la literatura son los grafos P_4 -extendible [21] y P_4 -lite [25]. Un grafo es P_4 -extendible si para todo subconjunto U de vértices de G que induce un P_4 existe a lo sumo un vértice v en $V - U$ tal que v induce un P_4 junto con vértices en U . Esta clase de grafos generaliza la clase de los grafos P_4 -reducible extendidos. Además esta

inclusión es propia. Los grafos P_4 -lite son grafos tales que todo subgrafo inducido con a lo sumo 6 vértices contiene a lo sumo dos P_4 's o es isomorfo a una araña sin cabeza. La familia de grafos P_4 -lite contiene estrictamente a los grafos P_4 -sparse [25].

Sea U un subconjunto de vértices que inducen un P_4 en G . Un *partner* de U es un vértice $v \in V - U$ tal que $U \cup \{v\}$ induce al menos dos P_4 en G . Un grafo es P_4 -tidy si todo P_4 inducido tiene a lo sumo un partner. Los grafos P_4 -tidy fueron definidos por Giakoumakis en [38] y representan una superclase de los grafos P_4 -lite, P_4 -extendibles y P_4 -sparse extendidos.

En [35], Roussel et al. llamaron grafos *partner-limited* a los grafos G para los cuales todo P_4 en G tiene a lo sumo dos partners. La clase de grafos partner-limited generaliza la clase de grafos P_4 -tidy. Además, esta inclusión es propia.

Giakoumakis [17] definió las siguientes clases de grafos con pocos P_4 's. Un grafo G es P_4 -laden si todo subgrafo inducido de G con a lo sumo 6 vértices que contiene más de dos P_4 's inducidos es un grafo split. En particular, la familia de los grafos split es una subclase de los grafos P_4 -laden. De manera similar, un grafo G es P_4 -laden extendido si todo subgrafo inducido de G con a lo sumo 6 vértices contiene a lo sumo dos P_4 's inducidos o es un grafo $\{C_4, \overline{C_4}\}$ -free. Como un grafo es split si y solo si es $\{C_4, \overline{C_4}, C_5\}$ -free [15], resulta que todo grafo P_4 -laden es P_4 -laden extendido. Por definición, tenemos que los grafos P_4 -lite son P_4 -laden.

En [3], Babel y Olariu generalizaron estas familias de grafos y definieron a un grafo (q, t) si ningún conjunto de a lo sumo q vértices induce más de t distintos P_4 's. En particular, los grafos $(q, q - 4)$ extienden y generalizan varias clases de grafos con pocos P_4 's. por ejemplo, los cografos son los grafos $(4, 0)$, y los P_4 -sparse son los $(5, 1)$. Además los grafos $(6, 2)$ generalizan a los grafos P_4 -reducibles, y los grafos $(7, 3)$ a los P_4 -lite.

En la Figura 4.1 se muestran algunas de estas familias, ordenadas por la inclusión estricta.

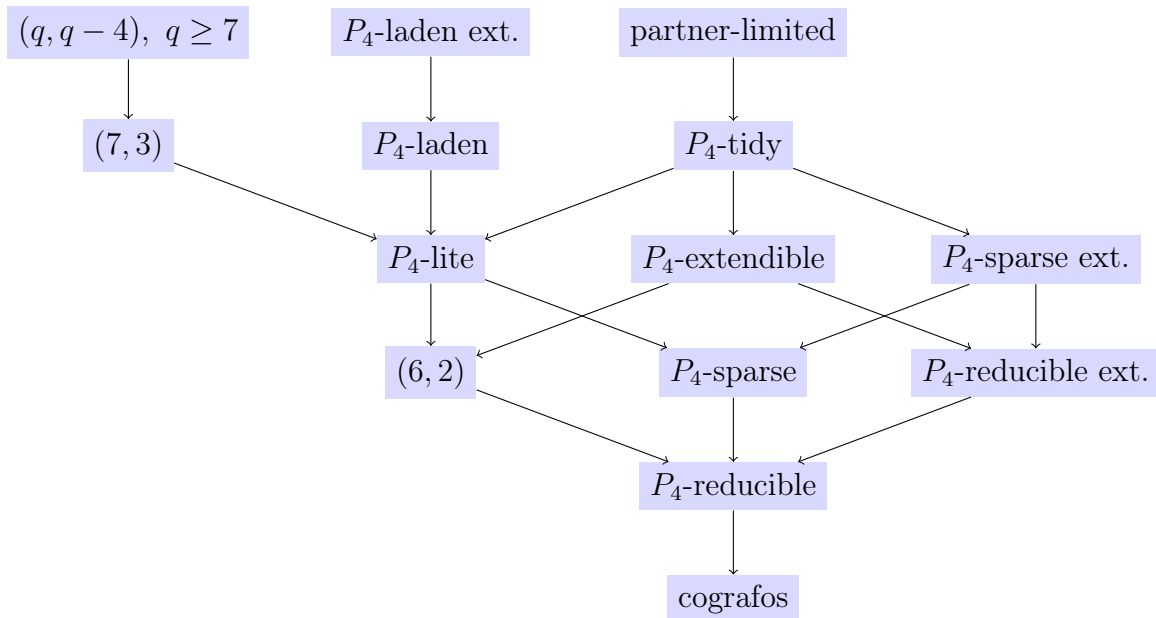


Figura 4.1: Familias con pocas P_4 's

En [10] se muestra que, para q fijo, los grafos de la familia $(q, q - 4)$ tienen cliquewidth acotado por la constante q , y la q -expresión necesaria para describir un grafo de esa familia se obtiene en tiempo lineal. En [39] se prueba que la familia de grafos *partner-limited* tiene

cliquewidth acotado por 4, y la 4-expresión necesaria para describir un grafo de esa familia se obtiene en tiempo lineal.

Como el Problema de Dominación Romana se puede expresar como un problema de optimización $\text{LinEMSOL}(\tau_1)$, por el Corolario 1, resultará que el Problema de Dominación Romana se resuelve en tiempo polinomial si nos restringimos a dichas familias.

Tenemos entonces el siguiente resultado

Corolario 5. *Existe un algoritmo que resuelve el Problema de Dominación Romana en tiempo lineal, si nos restringimos a las familias de grafos $(q, q - 4)$ y partner-limited.*

Claramente, el Problema de Dominación Romana se resuelve también eficientemente para todas las familias que están incluidas dentro de estas dos últimas. Sin embargo, como se menciona en [5], los algoritmos lineales basados en la expresión del problema en $\text{MSOL}(\tau_1)$ [10] en general no brindan algoritmos útiles en la práctica pues su dependencia de la descomposición cliquewidth involucra constantes multiplicativas muy grandes, incluso cuando el clique-width está acotado por 2 (ver [14]).

Entonces cabe preguntarnos si no podemos dar algún algoritmo específico que resuelva el problema eficientemente en alguna de estas familias.

4.2. Un algoritmo para calcular γ_R en grafos con pocos P_4 's

Recordemos que en la Sección 2.3 hemos dado un algoritmo para calcular $\gamma_R(G)$ inspirados en la descomposición modular del grafo G .

En esta sección vamos a dar un algoritmo específico para calcular el número de dominación romana γ_R para algunas familias de grafos con pocos P_4 's. En particular, analizaremos las familias de los grafos P_4 -sparse y P_4 -tidy.

Para ciertas familias con pocos P_4 's que hemos presentado se pueden caracterizar justamente por cuáles son sus subgrafos modulares. De esta manera, el paso en la línea 18 del Algoritmo puede especificarse aún más.

Por ejemplo, los cografos se pueden caracterizar de la siguiente manera:

Teorema 16 ([19]). *Dado un grafo G , son equivalentes:*

1. G es un cografo
2. G verifica una y solo una de las siguientes afirmaciones:
 - a) G es desconexo
 - b) \overline{G} es desconexo
 - c) G es K_1

En general, si G es una araña de partición (S, C, H) y el grafo $G[H]$ pertenece a una familia de grafos \mathcal{F} , diremos que G es una \mathcal{F} -araña. De la misma manera, si G es una quasi-araña que se obtiene a parti de una araña de partición (S, C, H) y el grafo $G[H]$ pertenece a una familia de grafos \mathcal{F} , diremos que G es una \mathcal{F} -quasi-araña.

En [27], Jamison y Olariu, caracterizan los grafos modulares de la familia P_4 -sparse:

Teorema 17 ([27]). *Dado un grafo G , son equivalentes:*

1. G es un grafo P_4 -sparse
2. G verifica una y solo una de las siguientes afirmaciones:
 - a) G es desconexo
 - b) \overline{G} es desconexo
 - c) $G = K_1$
 - d) G es una P_4 -sparse-araña

En [38] Giakoumakis et al. estudian la estructura de los grafos P_4 -tidy, y caracterizan los subgrafos modulares de esta familia. El resultado se puede resumir como:

Teorema 18 ([38]). *Dado un grafo G , son equivalentes:*

1. G es un grafo P_4 -tidy
2. G verifica una y solo una de las siguientes afirmaciones:
 - a) G es desconexo
 - b) \overline{G} es desconexo
 - c) G es K_1, C_5, P_5 o $\overline{P_5}$
 - d) G es una P_4 -tidy-quasi-araña

Utilizando estos resultados, podemos entonces reescribir el Algoritmo 2.1 cuando nos restringimos a estas familias.

Grafos P_4 -tidy

Analicemos qué sucede si restringimos la entrada del algoritmo a un grafo P_4 -tidy.

En particular, queremos determinar $\gamma_R(G)$ para G un grafo modular P_4 -tidy. Si $G \in \{K_1, \overline{P_5}, P_5, C_5\}$, ya conocemos el valor de $\gamma_R(G)$.

En [38] se muestra que la partición (S, C, H) de una quasi-araña se consigue en tiempo lineal, y por ende la cantidad de patas r de la araña subyacente también, que es la información que necesitamos para obtener $\gamma_R(G)$ (ver el Teorema 12). Vamos a ver que no necesitamos calcular la partición (S, C, H) de la araña subyacente, sino que si G es una quasi-araña, para calcular $r = |S| = |C|$ y si es fina o gruesa (que es la información necesaria para calcular $\gamma_R(G)$) nos alcanza con analizar algunos aspectos relativos a vértices de grados específicos en G .

Si G tiene un vértice quasi-universal, entonces G es una quasi-araña que se obtiene a partir de una araña gruesa y $\gamma_R(G) = 3$ (ver el Teorema 12).

Si G no tiene ningún vértice quasi-universal, entonces $G = (S', C', H)$ es una quasi-araña que se obtiene a partir de una araña fina de partición (S, C, H) , donde a lo sumo un vértice en $S \cup C$ es reemplazado por un grafo de 2 vértices. Observemos además que $r = |S| = |C| \geq 3$, ya que de lo contrario $\gamma_R(G) = r + 1 = 3$ y por el Lema 5, G tendría un vértice quasi-universal, lo cual genera una contradicción. Entonces G se obtiene a partir de una araña fina de $r \geq 3$ patas. Consideremos los conjuntos

$$X_1 = \{v \in V(G) : gr(v) = 1\}$$

$$X_2 = \{v \in V(G) : gr(v) = 2\}$$

Tenemos entonces que $S' = X_1 \cup X_2$.

En efecto, si $v \in H$ entonces $C' \subseteq N(v)$ y $gr(v) \geq |C'| \geq |C| \geq 3$. Si $v \in C'$, entonces $gr(v) \geq r \geq 3$. Es decir $(H \cup C') \cap (X_1 \cup X_2) = \emptyset$. Además, para un vértice $v \in S'$ tenemos varias situaciones posibles. Si v es un extremo de una pata que no fue modificada, o bien fue modificada cambiando un vértice en S por S_2 , entonces $gr(v) = 1$ y $v \in X_1$. Si v es un extremo de una pata que fue modificada cambiando un vértice en S por K_2 , o bien un vértice en C por S_2 o K_2 , entonces $gr(v) = 2$ y $v \in X_2$.

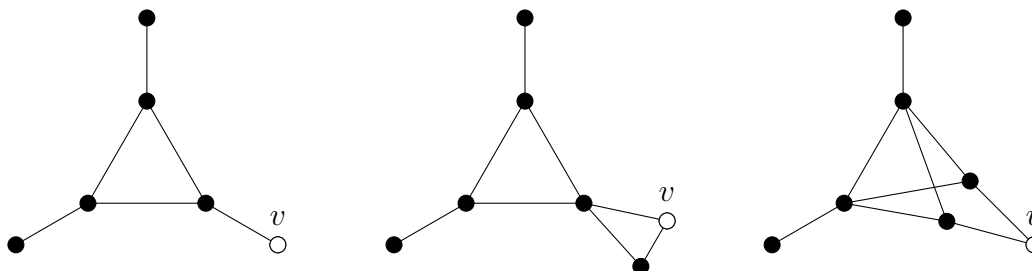


Figura 4.2: Algunos ejemplos de $v \in X_1 \cup X_2$

Entonces $S' = X_1 \cup X_2$ y por ende C' son los vecinos de $X_1 \cup X_2$ que no están en X_2 . Consideremos $r_1 = |X_1|$, $r_2 = |X_2|$ y $r_c = |\{u \in N(X_1 \cup X_2) : u \notin X_2\}|$. Entonces $|S'| = r_1 + r_2$, $|C'| = r_c$ y tenemos que

$$r = \text{mín}\{r_1 + r_2, r_c\},$$

y resulta $\gamma_R(G) = r + 1$

Así, podemos reescribir el algoritmo 4.1 para el caso de los grafos P_4 -tidy de la siguiente manera:

Algoritmo 4.1: Algoritmo para obtener γ_R para un grafo G P_4 -tidy

```

1  input G grafo P4-tidy
2  output drom(G) #número de dominación romana de G
3
4  if G es desconexo then:
5      d = 0
6      for H componente conexa de G do:
7          d += drom(H)
8      return d
9  else if complemento(G) desconexo then:
10     G join de G1, ..., Gk
11     if algún Gi tiene un vértice universal then:
12         return 2
13     else if algún Gi tiene un vértice quasi-universal then:
14         return 3
15     else:
16         return 4
17 else:
18     if G = K1 then:
19         return 1
20     else if G = C5 or G = P5 then:
21         return 4
22     else if G tiene un vértice quasi-universal then:
23         return 3
24     else:

```

```

25     X1 = {vértices de grado 1}
26     X2 = {vértices de grado 2}
27     r1 = |X1|
28     r2 = |X2|
29     rc = |vecindad(X1 U X2) - X2|
30     r = min{r1 + r2, rc}
31     return r + 1

```

De los teoremas 8, 9 y 18, tenemos que este algoritmo resuelve el Problema de Dominación Romana para grafos P_4 -tidy.

Si un grafo P_4 -tidy G es modular pueden pasar tres cosas:

- $G \in \{K_1, C_5, P_5\}$, cuyo reconocimiento y cálculo de γ_R se lleva a cabo en tiempo lineal,
- G tiene un vértice quasi-universal, este es el caso en que $G = \overline{P_5}$ o G es una quasi-araña gruesa, el reconocimiento de este caso se lleva a cabo en tiempo lineal y $\gamma_R(G) = 3$,
- Si G no verifica ninguno de los items anteriores, G es una quasi-araña que se obtiene a partir de una araña fina de $r \geq 3$ patas, y r se calcula contando la cantidad de vértices de grado 1 y 2 en G .

Entonces el cálculo de $\gamma_R(G)$, si G es un grafo P_4 -tidy modular, se logra en tiempo lineal.

Si el grafo de entrada G del algoritmo verifica que su complemento es desconexo, entonces el valor de $\gamma_R(G)$ se consigue analizando los grados máximos de cada uno de los grafos G_i tales que $G = G_1 \vee \dots \vee G_k$. Esto se consigue también en tiempo lineal respecto de la cantidad de vértices en G .

Finalmente, si el grafo G es desconexo, entonces para cada componente conexa H de G debemos correr el algoritmo. Cada una de estas componentes, o bien el complemento es desconexo o bien es un grafo modular. En cualquier caso el valor de $\gamma_R(H)$ se consigue en tiempo lineal respecto al tamaño del grafo H . Por lo tanto $\gamma_R(G)$ se calcula en tiempo lineal respecto al tamaño del grafo G .

Todo esto nos lleva al siguiente resultado:

Teorema 19. *El Algoritmo 4.1 resuelve el Problema de Dominación Romana en tiempo lineal para grafos P_4 -tidy*

El hecho de que exista un algoritmo que resuelve el problema en tiempo lineal para grafos P_4 -tidy no es una novedad. Ya en el Corolario 5, habíamos analizado que para los grafos partner-limited (que contienen a los grafos P_4 -tidy), al tratarse de una familia con cliquewidth acotado por la constante 4, existe un algoritmo lineal que resuelve el problema, usando la 4-expresión del grafo y el problema expresado en $MSOL(\tau_1)$. Se trata de un algoritmo general, sin gran aplicación práctica específica. En cambio, el Algoritmo 4.1, además de ser un algoritmo específico, es sencillo y claro.

Observación 10. *Del Teorema 19 resulta que el algoritmo dado resuelve el Problema de Dominación Romana en tiempo lineal para todas las subfamilias de los grafos P_4 -tidy. En particular, el mismo resuelve el problema eficientemente para las subfamilias que se muestran en la Figura 4.1: P_4 -lite, P_4 -extendible, P_4 -sparse extendidos, P_4 -reducible extendidos, P_4 -sparse, $(6, 2)$, P_4 -reducible, cografos.*

Grafos P_4 -sparse

En el caso de los grafos P_4 -sparse, podemos simplificar aún más el algoritmo, observando la estructura particular de los subgrafos modulares en esta familia. Hemos visto en el Teorema 17 que estos son o bien K_1 , o bien grafos araña cuya cabeza, en caso de tenerla, es a su vez un grafo P_4 -sparse.

Si G es una araña fina, entonces la cantidad de patas (r) coincide con la cantidad de vértices de grado 1. Además, por el Teorema 10, tenemos que el número de dominación romana de G es $\gamma_R(G) = r + 1$. Es decir, para obtener $\gamma_R(G)$ nos basta solamente con contar la cantidad de vértices pendientes, y sumar 1. Si G es una araña gruesa, G no tiene ningún vértice pendiente. Además el valor de $\gamma_R(G)$ es 3, independientemente de la cantidad de patas de la araña. Entonces tenemos el siguiente resultado:

Teorema 20. *Sea $G = (V, E)$ un grafo modular P_4 -sparse. Entonces se verifica una y solo una de estas condiciones:*

1. $|V| = 1$
2. G tiene $r > 0$ vértices pendientes. En este caso G es una araña fina de r patas y $\gamma_R(G) = r + 1$
3. $|V| > 1$ y G no tiene vértices pendientes. En este caso G es una araña gruesa y $\gamma_R(G) = 3$

Así, podemos reescribir el algoritmo 4.1 para el caso de los grafos P_4 -sparse de la siguiente manera:

Algoritmo 4.2: Algoritmo para obtener γ_R para un grafo G P_4 -sparse

```

1  input G grafo P4-sparse
2  output drom(G) --> número de dominación romana de G
3
4  if G es desconexo then:
5      d = 0
6      for H componente conexa de G do:
7          d += drom(H)
8      return d
9  else if complemento(G) desconexo then:
10     G join de G1, ..., Gk
11     if algún Gi tiene un vértice universal then:
12         return 2
13     else if algún Gi tiene un vértice quasi-universal then:
14         return 3
15     else:
16         return 4
17 else:
18     if |V| = 1 then:
19         return 1
20     else if #(vértices pendientes en G) = 0 then:
21         return 3
22     else:
23         return #(vértices pendientes en G) + 1

```

4.3. Un resultado de NP-completitud

En esta sección vamos a mostrar una familia de grafos con pocos P_4 's para la cual el problema de decisión asociado al Problema de Dominación Romana es NP-completo.

En vista de esto, damos una prueba de la NP-completitud del Problema de Dominación Romana para los grafos split, mediante una reducción partiendo del Problema de Dominación Romana para un grafo cualquiera (DROM), que sabemos que es NP-completo.

En [8] se menciona que Mc. Rae mostró que DROM es NP-completo en la familia de grafos split. Sin embargo, no hemos encontrado ninguna publicación de estos resultados. Después de escribir esta prueba, hemos encontrado que en [36], Schnupp prueba efectivamente que el problema DROM es NP-completo para la familia de grafos split, mediante una reducción a partir del problema de Cubrimiento por Vértices.

A continuación, vamos a demostrar el resultado mediante la reducción a partir de DROM. Consideremos entonces los siguientes problemas de decisión:

DROM

Instancia: $G = (V, E)$ grafo, con $E \neq \emptyset$, $k \in \mathbb{N}$

Pregunta: ¿Existe una FDR f sobre G de peso a lo sumo k ?

DROM-Split

Instancia: $G = (V, E)$ grafo split, $k \in \mathbb{N}$

Pregunta: ¿Existe una FDR f sobre G de peso a lo sumo k ?

Recordemos que un grafo $G = (V, E)$ es un grafo split si el conjunto de vértices V puede partitionarse en dos conjuntos disjuntos Q y S , de manera que Q forma una clique y S un conjunto estable.

Teorema 21. *El problema de decisión DROM restringido a la familia de grafos split es NP-completo.*

Demostración. Como DROM es NP-completo, $\text{DROM} \in \text{NP}$, por lo tanto $\text{DROM-Split} \in \text{NP}$. Para demostrar que $\text{DROM-Split} \in \text{NP-completo}$, presentamos una reducción polinomial de DROM a DROM-Split.

Usaremos una transformación adecuada de DROM a DROM-Split para mostrar que este último pertenece también a la clase de los problemas NP-completos.

Sea $I = (G, k)$ una instancia de DROM. Construimos una instancia $I' = (G', k')$ de DROM-Split de la siguiente manera: $k' = k$ y el grafo G' tiene conjunto de vértices $V(G') = \cup_{v \in V(G)} \{s_v, q_v\}$, donde los vértices $S = \{s_v\}_{v \in V(G)}$ forman un estable y $Q = \{q_v\}_{v \in V(G)}$ forman una clique. Además $s_u q_v \in E(G')$ si y solo si $u \in N_G[v]$. En la Figura 4.3 se ve un ejemplo de esta construcción.

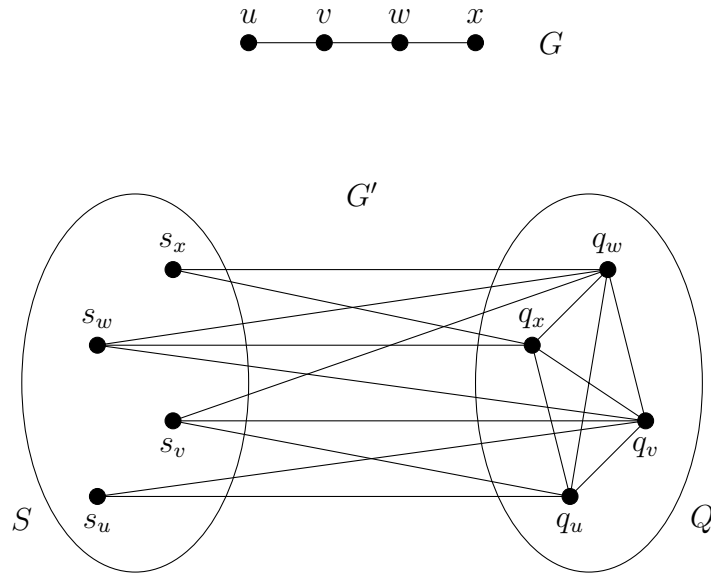


Figura 4.3: Ejemplo de construcción de G'

La instancia I admite una respuesta positiva para DROM si y solo si la instancia I' admite una respuesta positiva para DROM-Split. En efecto, sea $f = (V_0, V_1, V_2)$ una FDR sobre el grafo G , de peso a $f(V) \leq k$. Por el Lema 2 podemos suponer sin pérdida de generalidad que $V_2 \neq \emptyset$. Definimos una función $f' : V(G') \rightarrow \{0, 1, 2\}$ dada por

$$f'(u) = \begin{cases} 1 & \text{si } u = s_v \text{ con } v \in V_1 \\ 2 & \text{si } u = q_v \text{ con } v \in V_2 \\ 0 & \text{en otro caso} \end{cases}$$

En la Figura 4.4 se ilustra la función f' a partir de una posible FDR sobre G .

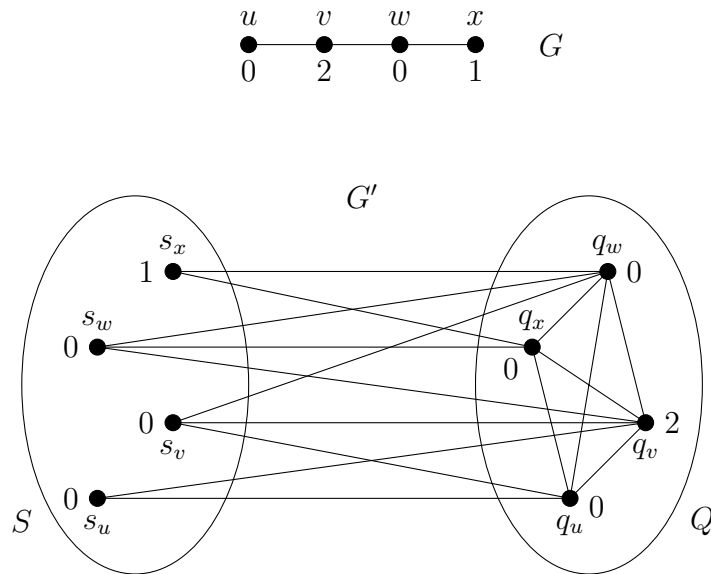


Figura 4.4: FDR sobre G y G'

Esta función f' es una FDR sobre G' de peso a lo sumo k . En efecto, sea $u \in V(G')$ tal que $f(u) = 0$. Si $u = q_v$ para algún $v \in V(G)$, entonces como $V_2 \neq \emptyset$, existe $w \in V_2$ y $f'(q_w) = 2$.

Al ser Q una clique, $q_w u \in E(G')$. Si $u = s_v$, para algún $v \in V(G)$ entonces tenemos dos posibles casos. Si $v \in V_2$, entonces $f'(q_v) = 2$ y $q_v u \in E(G')$. Si no, $v \in V_0$ y existe $w \in V_2 \cap N_G(v)$. Entonces $f'(q_w) = 2$ y por construcción de G' , $q_w u \in E(G')$. Por lo tanto, f' es una FDR sobre G' . Además, para cada $v \in V(G)$ tenemos que $f'(s_v) + f'(q_v) = f(v)$ con lo cual $f'(V(G')) = f(V(G)) \leq k$.

Recíprocamente, sea f' una FDR sobre G' de peso a lo sumo k . Definimos una función $f : V(G) \rightarrow \{0, 1, 2\}$ dada por $f(v) = \max\{f'(s_v), f'(q_v)\}$. Esta función f es una FDR sobre G de peso a lo sumo k . En efecto, si $f(v) = 0$, entonces $f'(s_v) = f'(q_v) = 0$ y como $N(s_v) \subseteq Q$, existe $w \in V(G)$ tal que $vw \in E(G)$ y $f'(q_w) = 2$. Entonces $f(w) \geq f'(q_w) = 2$, tenemos que existe un vecino w de v en G tal que $f(w) = 2$. Además, el peso de f es $f(V(G)) = \sum_{v \in V(G)} \max\{f'(s_v), f'(q_v)\} \leq \sum_{v \in V(G)} f'(s_v) + f'(q_v) = f'(V(G')) \leq k$. \square

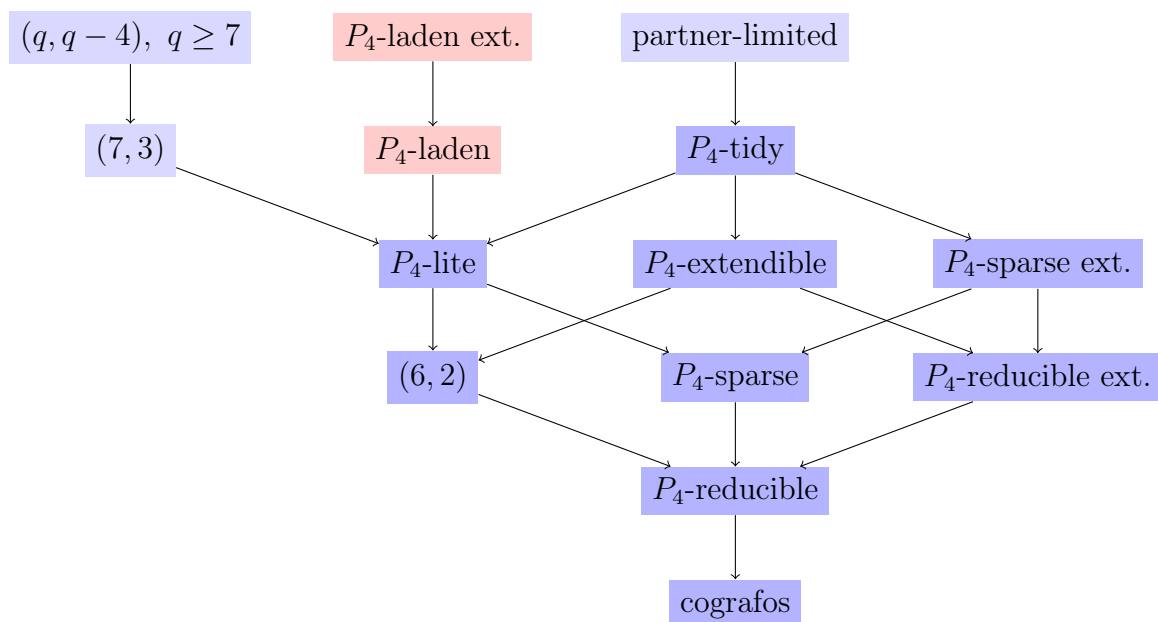
Recordemos que un grafo es P_4 -laden si todo subgrafo inducido de orden al menos 6 que contiene más de dos P_4 inducido es un grafo split. La familia de los grafos split está contenida dentro de la familia P_4 -laden. Entonces, como el problema DROM es NP-completo para los grafos split, tenemos el siguiente resultado:

Teorema 22. *El problema de decisión DROM restringido a la familia de grafos P_4 -laden es NP-completo.*

Observación 11. *Del Teorema 22 resulta que el Problema de Dominación Romana es NP-completo para todas las familias con pocos P_4 's que contengan a los grafos P_4 -laden. En particular, es NP-completo cuando restringimos el problema a los grafos P_4 -laden extendidos.*

Conclusión del capítulo

Los resultados de este capítulo se resumen en el siguiente cuadro, donde pintamos de celeste claro aquellas familias donde conocemos que el problema se resuelve en tiempo lineal, de celeste oscuro aquellas donde se resuelve en tiempo lineal mediante el Algoritmo 4.1 y finalmente de rojo aquellas familias donde el problema es NP-completo.



Conclusión y Trabajo a Futuro

En este trabajo hemos estudiado el Problema de Dominación Romana. En particular, hemos analizado el problema restringido a las familias de grafos *con pocos P_4 's* que se ven en la Figura 4.1.

Hemos mostrado, con un resultado general, que para varias de estas familias el problema se resuelve en tiempo lineal, y hemos diseñado un algoritmo a partir de la descomposición de un grafo mediante las operaciones *join* y *unión disjunta*, que resuelve el problema en tiempo lineal para los grafos P_4 -tidy, y por ende para todas sus subfamilias. Por otro lado, hemos probado que si nos restringimos a los grafos P_4 -laden el problema de decisión asociado al problema de dominación romana es NP-completo.

En el Capítulo 2 presentamos un algoritmo para resolver el Problema de Dominación Romana, con una estrategia basada en la descomposición modular de un grafo. En el Capítulo 3 de este trabajo estudiamos distintas familias de grafos *araña*. Los grafos araña y quasi-araña aparecen en la descomposición modular de los grafos P_4 -tidy, y los resultados obtenidos para estas familias fueron utilizados para adaptar el algoritmo dado en el Capítulo 2 cuando nos restringimos a estas familias.

Completamos con los resultados obtenidos el cuadro de la Sección 1.2.3 de esta manera:

Familia	Problema de Dominación Romana	Problema de Dominación Clásico
Árboles	Lineal [8]	Lineal [13]
Intervalos	Lineal [30]	Lineal [7]
Cliquewidth acot.	Polinomial [30]	Polinomial [10]
Cografos	Lineal [30]	Lineal [20]
Block	Lineal [24]	Lineal [33]
Dist. hereditarios	Lineal [30]	Lineal [33]
AT-free	Polinomial [30]	Polinomial [29]
Cordales	NP-completo [31]	NP-completo [6]
Fuertemente cordales	Lineal [31]	Lineal [13]
Bipartitos	NP-completo [31]	NP-completo [12]
Split	NP-completo [36]	NP-completo [4]
Planares	NP-completo [36]	NP-completo [16]
$(q, q - 4)$	Lineal	Lineal [10]
partner-limited	Lineal	Lineal[39]
P_4-tidy	Lineal	Lineal[39]
P_4-sparse	Lineal	Lineal[32]
P_4-laden	NP-completo	NP-completo[4]

Trabajo a Futuro

Los grafos araña débilmente etiquetados, también analizados en ese capítulo, aparecen en la descomposición modular de la familia de grafos *partner-limited*. Queda pendiente, como trabajo a futuro, estudiar qué sucede con el resto de los grafos modulares de esta familia, entre los que se encuentra una clase de grafos split [1].

Otra pregunta de interés, a partir del análisis del cuadro en la sección 1.2.3 es la siguiente: ¿existe alguna clase de grafos donde los problemas de decisión asociados al Problema de Dominación Romana y al Problema de Dominación clásico tienen diferente complejidad (polinomial - NP-completo)? O bien, ¿existe algún indicio de que los problemas de Dominación y de Dominación Romana tengan la misma complejidad? Hasta dónde conocemos, no se han hallado clases de grafos donde estos problemas difieran en complejidad.

Por otro lado, entre las operaciones que hemos analizado en el capítulo 2, se encuentra el borrado de un vértice. Esto nos lleva a preguntarnos qué características puede o debe tener un grafo γ_R -crítico, es decir, un grafo en el cual el borrado de cualquier vértice implica la disminución (en una unidad) del valor de γ_R .

Apéndice A

Un algoritmo usando Sage

En este apéndice, vamos a presentar una implementación del algoritmo 4.1 utilizando Sage (www.sagemath.org).

A.1. Grafos P_4 -tidy

En esta primera sección implementaremos el Algoritmo 4.1. Introduciremos y explicaremos el funcionamiento de dos funciones auxiliares `Rdom_join` y `Rdom_mod_tidy` para luego presentar la función final `Rdom_P4_tidy`, que dado un grafo G en la familia P_4 -tidy, devuelve el valor de $\gamma_R(G)$.

`Rdom_join`

La función `Rdom_join` toma como entrada una lista de grafos `Glist` y calcula el valor de γ_R para el grafo join de todos los grafos de la lista. Para este cálculo utilizamos el Teorema 9. Lo primero que hacemos es calcular el valor de m , siendo

$$m = \min_{H \in \text{Glist}} \{|V(H)| - \Delta(H)\}$$

Según m sea 1, 2 o > 2 , estaremos en el primer, segundo o tercer caso del teorema 9 respectivamente.

```
1 def Rdom_join(Glist):
2     L = []
3     for g in Glist:
4         L.append(g.order() - max(g.degree()))
5     m = min(L)
6     if m == 1: return 2
7     else:
8         if m == 2: return 3
9         else: return 4
```

`Rdom_mod_tidy`

La función `Rdom_mod_tidy` recibe como entrada un grafo modular P_4 -tidy G , y devuelve el valor de $\gamma_R(G)$. Recordemos que por el Teorema 18, G cumple una y solo una de las siguientes opciones:

1. G es K_1 ,

2. G es C_5 ,
3. G es P_5 ,
4. G es $\overline{P_5}$,
5. G es una P_4 -tidy-quasi-araña

Y además, por lo expuesto antes del Algoritmo 4.1 tenemos que G se puede clasificar según propiedades de los vértices de grados particulares.

Entonces, podemos calcular el parámetro γ_R de un grafo modular P_4 -tidy con la siguiente función:

```

1 def Rdom_mod_tidy(G):
2     if G.order() == 1:
3         return 1
4     else:
5         if G.is_isomorphic(graphs.PathGraph(5)) or G.is_isomorphic(graphs.
6             CycleGraph(5)):
7             return 4
8         else:
9             if max(G.degree()) == G.order() - 2:
10                return 3
11            else:
12                X1 = Set([])
13                X2 = Set([])
14                N = Set([])
15                for v in G.vertices():
16                    if G.degree(v) == 1:
17                        X1 = X1.union(Set([v]))
18                        N = N.union(Set(G.neighbors(v)))
19                    else:
20                        if G.degree(v) == 2:
21                            X2 = X2.union(Set([v]))
22                            N = N.union(Set(G.neighbors(v)))
23                r1 = X1.cardinality()
24                r2 = X2.cardinality()
25                N = N.difference(X2)
26                rc = N.cardinality()
27                r = min([r1+r2, rc])
28                return r + 1

```

Rdom_P4_tidy

Finalmente, la función `Rdom_P4_tidy` recibe como entrada un grafo P_4 -tidy cualquiera, y devuelve el valor del parámetro γ_R para ese grafo.

El funcionamiento es el siguiente. Para cada componente conexa H de G , calculamos $\gamma_R(H)$, y sumamos esos valores en la variable d , que será igual a $\gamma_R(G)$ (ver el Teorema 8). Para calcular $\gamma_R(H)$, separamos en dos casos. Si \overline{H} es desconexo, entonces H puede expresarse como join de los complementos de las componentes conexas de \overline{H} , y usamos la función `Rdom_join`. Si no, H es un subgrafo modular y usamos la función `Rdom_mod_tidy`.

```

1 def Rdom_P4_tidy(G):
2     Glist = G.connected_components_subgraphs()
3     d = 0

```

```

4
5     for H in Glist:
6         if H.complement().connected_components_number() > 1:
7             HList = []
8             for CC in H.complement().connected_components_subgraphs():
9                 HList.append(CC.complement())
10            d += Rdom_join(HList)
11        else:
12            d += Rdom_mod_tidy(H)
13    return d

```

A.2. Grafos P_4 -sparse

De manera similar, vamos a dar una implementación del Algoritmo 4.2. La función `Rdom_join` es la misma ya presentada en la sección anterior. Presentaremos la función auxiliar `Rdom_mod_sparse` para luego presentar la función final `Rdom_P4_sparse`, que dado un grafo G en la familia P_4 -sparse, devuelve el valor de $\gamma_R(G)$.

`Rdom_mod_sparse`

La función `Rdom_mod_sparse` recibe como entrada un grafo modular P_4 -sparse G , y devuelve el valor de $\gamma_R(G)$. Recordemos que por el Teorema 17, G cumple una y solo una de las siguientes opciones:

1. G es K_1 ,
2. G es una P_4 -sparse-araña

Para el primer caso sabemos que el número de dominación romana es $\gamma_R(K_1) = 1$. Si G es una araña de r patas, para calcular su número de dominación romana, basta con identificar sus vértices pendientes. En efecto, observemos que una araña gruesa no tiene vértices pendientes (en este caso $\gamma_R(G) = 3$) mientras que una araña fina tiene exactamente r vértices pendientes (y tenemos $\gamma_R(G) = r + 1$).

Entonces, podemos calcular el parámetro γ_R de un grafo modular P_4 -sparse con la siguiente función:

```

1 def Rdom_mod_sparse(G):
2     if G.order() == 1:
3         return 1
4     else:
5         r = 0
6         for v in G.vertices():
7             if G.degree(v) == 1:
8                 r += 1
9         if r == 0:
10            return 3
11        else:
12            return r + 1

```

`Rdom_P4_sparse`

Finalmente y de manera similar a lo hecho para los grafos P_4 -tidy, la función `Rdom_P4_sparse` recibe como entrada un grafo P_4 -sparse, y devuelve el valor del parámetro γ_R para ese grafo.

El funcionamiento es análogo al de la función `Rdom_P4_tidy`. Para cada componente conexa H de G , calculamos $\gamma_R(H)$, y sumamos esos valores en la variable `d`, que será igual a $\gamma_R(G)$ (ver el Teorema 8). Para calcular $\gamma_R(H)$, separamos en dos casos. Si \overline{H} es desconexo, entonces H puede expresarse como join de los complementos de las componentes conexas de \overline{H} , y usamos la función `Rdom_join`. Si no, H es un subgrafo modular y usamos la función `Rdom_mod_sparse`.

```

1 def Rdom_P4_sparse(G):
2     Glist = G.connected_components_subgraphs()
3     d = 0
4
5     for H in Glist:
6         if H.complement().connected_components_number() > 1:
7             Hlist = []
8             for CC in H.complement().connected_components_subgraphs():
9                 Hlist.append(CC.complement())
10            d += Rdom_join(Hlist)
11        else:
12            d += Rdom_mod_sparse(H)
13    return d

```

Bibliografía

- [1] ARGIROFFO, G., NASINI, G., AND TORRES, P. The packing coloring problem for lobsters and partner limited graphs. *Discrete Applied Mathematics* 164 (2014), 373–382.
- [2] BABEL, L., KLOKS, T., KRATOCHVIL, J., KRATSCH, D., MÜLLER, H., AND OLARIU, S. Efficient algorithms for graphs with few p4's. *Discrete Mathematics* 235, 1-3 (2001), 29–51.
- [3] BABEL, L., AND OLARIU, S. On the structure of graphs with few p4s. *Discrete Applied Mathematics* 84, 1-3 (1998), 1–13.
- [4] BERTOSSI, A. A. Dominating sets for split and bipartite graphs. *Information processing letters* 19, 1 (1984), 37–40.
- [5] BONOMO-BRABERMAN, F., NASCIMENTO, J. R., OLIVEIRA, F. S., SOUZA, U. S., AND SZWARCFITER, J. L. Linear-time algorithms for eliminating claws in graphs. In *International Computing and Combinatorics Conference* (2020), Springer, pp. 14–26.
- [6] BOOTH, K. S., AND JOHNSON, J. H. Dominating sets in chordal graphs. *SIAM Journal on Computing* 11, 1 (1982), 191–199.
- [7] CHANG, M.-S. Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM Journal on computing* 27, 6 (1998), 1671–1694.
- [8] COCKAYNE, E. J., DREYER JR, P. A., HEDETNIEMI, S. M., AND HEDETNIEMI, S. T. Roman domination in graphs. *Discrete Mathematics* 278, 1-3 (2004), 11–22.
- [9] COOK, S. A. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing* (1971), pp. 151–158.
- [10] COURCELLE, B., MAKOWSKY, J. A., AND ROTICS, U. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems* 33, 2 (2000), 125–150.
- [11] CURRÒ, V. *The roman domination problem on grid graphs*. PhD thesis, Università di Catania, 2014.
- [12] DEWDNEY, A. K. *Fast Solvable Reductions Between Problems in NP*. Department of Computer Science, University of Western Ontario, 1981.
- [13] FARBER, M. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Applied Mathematics* 7, 2 (1984), 115–130.

- [14] FLUM, J., AND GROHE, M. Parameterized complexity theory. *Texts Theoret. Comput. Sci. EATCS Ser* (2006).
- [15] FOLDES, S., AND HAMMER, P. L. *Split graphs*. Universität Bonn. Institut für Ökonometrie und Operations Research, 1976.
- [16] GAREY, M. R., AND JOHNSON, D. S. Computers and intractability. *A Guide to the* (1979).
- [17] GIAKOUMAKIS, V. P4-laden graphs: a new class of brittle graphs. *Information processing letters* 60, 1 (1996), 29–36.
- [18] GIAKOUMAKIS, V., AND VANHERPE, J.-M. On extended p4-reducible and extended p4-sparse graphs. *Theoretical Computer Science* 180, 1-2 (1997), 269–286.
- [19] HABIB, M., AND PAUL, C. A simple linear time algorithm for cograph recognition. *Discrete Applied Mathematics* 145, 2 (2005), 183 – 197. Structural Decompositions, Width Parameters, and Graph Labelings.
- [20] HAYNES, T. W., HEDETNIEMI, S., AND SLATER, P. *Fundamentals of domination in graphs*. CRC press, 1998.
- [21] HEESCH, H. *Untersuchungen zum Vierfarbenproblem*, vol. 810. Bibliographisches Institut, 1969.
- [22] HENNING, M. A characterization of roman trees. *Discussiones Mathematicae Graph Theory* 22, 2 (2002), 325–334.
- [23] HOANG, C. T., AND SRITHARAN, R. *Perfect graphs*. CRC Press, 2015.
- [24] HSU, C., LIU, C., AND PENG, S. Roman domination on block graphs. In *Proceedings of the 22nd Workshop on Combinatorial Mathematics and Computation Theory* (2005), pp. 188–191.
- [25] JAMISON, B., AND OLARIU, S. A new class of brittle graphs. *Studies in Applied Mathematics* 81, 1 (1989), 89–92.
- [26] JAMISON, B., AND OLARIU, S. P4-reducible graphs—class of uniquely tree-representable graphs. *Studies in Applied Mathematics* 81, 1 (1989), 79–87.
- [27] JAMISON, B., AND OLARIU, S. A tree representation for p4-sparse graphs. *Discrete Applied Mathematics* 35, 2 (1992), 115–129.
- [28] KARP, R. M. Reducibility among combinatorial problems. In *Complexity of computer computations*. Springer, 1972, pp. 85–103.
- [29] KRATSCH, D. Domination and total domination on asteroidal triple-free graphs. *Discrete Applied Mathematics* 99, 1-3 (2000), 111–123.
- [30] LIEDLOFF, M., KLOKS, T., LIU, J., AND PENG, S.-L. Efficient algorithms for roman domination on some classes of graphs. *Discrete Applied Mathematics* 156, 18 (2008), 3400–3415.

- [31] LIU, C.-H., AND CHANG, G. J. Roman domination on strongly chordal graphs. *Journal of Combinatorial Optimization* 26, 3 (2013), 608–619.
- [32] MAKOWSKY, J. A., AND ROTICS, U. On the clique-width of graph with few p_4 's. *International Journal of Foundations of Computer Science* 10, 03 (1999), 329–348.
- [33] NICOLAI, F., AND SZYMCAK, T. Homogeneous sets and domination: A linear time algorithm for distance-hereditary graphs. *Networks: An International Journal* 37, 3 (2001), 117–128.
- [34] OUM, S.-I., AND SEYMOUR, P. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B* 96, 4 (2006), 514–528.
- [35] ROUSSEL, F., RUSU, I., AND THUILLIER, H. On graphs with limited number of P_4 -partners. *International Journal of Foundations of Computer Science* 10, 01 (1999), 103–121.
- [36] SCHNUPP, M. *Broadcast domination with flexible powers*. PhD thesis, Friedrich Schiller Universität Jena, 2006.
- [37] STEWART, I. Defend the roman empire! *Scientific American* 281, 6 (1999), 136–138.
- [38] THUILLIER, H., ROUSSEL, F., AND GIAKOUMAKIS, V. On P_4 -tidy graphs. *Discrete Mathematics & Theoretical Computer Science* 1 (1997).
- [39] VANHERPE, J.-M. Clique width of partner limited graphs. *Electronic Notes in Discrete Mathematics* 5 (2000), 303–306.