

03063

**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

UNIDAD ACADEMICA DE LOS CICLOS PROFESIONAL Y DE POSTGRADO DEL
COLEGIO DE CIENCIAS Y HUMANIDADES
INSTITUTO DE INVESTIGACIONES EN MATEMATICAS APLICADA Y
SISTEMAS

**GRAFICAS MARCADAS SIMETRICAS CON
MARCAJES VIVOS-SEGUROS**

TESIS
QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACION
PRESENTA
JOAQUIN AGUADO GUADARRAMA
DIRECTOR: DR. SERGIO RAJSBAUM G.

MEXICO D.F. JUNIO 1995

FALLA DE ORIGEN
EN SU TOTALIDAD



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

03063

1
201
2012

**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

UNIDAD ACADÉMICA DE LOS CICLOS PROFESIONAL Y DE POSTGRADO DEL
COLEGIO DE CIENCIAS Y HUMANIDADES
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADA Y
SISTEMAS

**GRAFICAS MARCADAS SIMÉTRICAS CON
MARCAJES VIVOS-SEGUROS**

TESIS
QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACION
PRESENTA
JOAQUIN AGUADO GUADARRAMA
DIRECTOR: DR. SERGIO RAJSBAUM G.

MEXICO D.F. JUNIO 1995

FALLA DE ORIGEN
EN SU TOTALIDAD

En memoria de mi hermano Mauricio Ivan Aguado Guadarrama

Contenido

Introducción	1
1 Preliminares	5
1.1 Conceptos básicos de la Teoría de Gráficas	5
1.2 Modelado de Sistemas Distribuidos.....	12
2 Gráficas Marcadas.....	17
2.1 El Modelo.....	17
2.2 Marcajes vivos y Marcajes vivos-seguros	21
3 Algoritmos para obtener marcajes vivos y vivos-seguros.....	26
3.1 Algoritmo L	26
3.2 Algoritmo LS	29
4 Gráficas Marcadas Simétricas.....	47
4.1 El modelo MGS y las calendarizaciones	47
4.2 Algoritmo $MGGS$	55
5 Número de marcajes vivos-seguros para las MGS	63
5.1 El máximo número de marcajes vivos-seguros	63
5.2 Total de marcajes vivos-seguros para algunas familias de MGS	68
5.3 Colorco de vértices.....	73
5.4 Polinomios cromáticos, orientaciones acíclicas y marcajes vivos-seguros.....	84
Conclusiones	94
Bibliografía.....	96

Introducción

Esta tesis trata de los conceptos, estructuras y mecanismos relacionados con el modelo de gráficas marcadas simétricas. Su objetivo es investigar y presentar, tan clara y completamente como sea posible, las características de dicho modelo, poniendo especial énfasis en cierta clase de marcajes, los llamados marcajes vivos-seguros.

Una *Gráfica Marcada* $MG=(G,M)$ es un modelo formal donde $G=(V,E)$ es una digráfica finita y M es un marcaje. Un *marcaje* es un vector de $|E|$ componentes enteros no negativos, cada uno de los cuales representa el marcaje de cada una de las aristas. El marcaje de una arista es el número de fichas que contiene. El marcaje de una MG cambia de acuerdo a un par de reglas conocidas como el "juego de fichas". En el *juego de fichas* se establece que un vértice está *habilitado* si y sólo si en cada una de las aristas de entrada hay al menos una ficha. Además, un vértice habilitado puede ser "disparado". La *operación de disparo* es atómica y consiste en la eliminación de una ficha de cada arista de entrada y la adición de una ficha en cada arista de salida del vértice.

Las *Gráficas Marcadas Simétricas (MGS)* son una clase especial de MG . En este tipo de gráficas, para todo par de vértices distintos se cumple una de las dos condiciones siguientes: i) existe un par de aristas antiparalelas entre ellos, o ii) no existe arista alguna entre ellos. En las MGS no hay bucles.

Un *marcaje* es *vivo*, si cada vértice de la gráfica está habilitado o puede habilitarse mediante el disparo de una serie de vértices (*secuencia legal de disparos*). Un marcaje vivo es *vivo-seguro* si cumple las siguientes condiciones: i) cada arista de la gráfica tiene 0 o 1 ficha, y ii) no existe secuencia legal de disparos que haga que alguna arista tenga 2 o más fichas.

Si $G=(V,E)$ es un "sistema distribuido", en donde el conjunto V de vértices representa a las unidades computacionales físicamente distribuidas, y en donde el conjunto E los canales de comunicación del tipo "half-duplex". Las unidades computacionales tienen capacidad de procesamiento y almacenamiento. Un canal de comunicación "half-duplex" es bidireccional con la restricción de que la información no puede ser transmitida en ambas direcciones simultáneamente.

Una *calendarización* de G es una secuencia infinita de rondas $[r_1, r_2, r_3, \dots]$, tal que $\forall i, r_i \subset V, r_i \neq \emptyset$ y cualquier par de vértices en r_i cumplen con que no son adyacentes. Un *calendarizador* es un protocolo que determina una calendarización. Este modelo es importante tanto en la teoría como en la práctica.

En general las MGS con marcajes vivos-seguros son útiles para modelar sistemas donde los procesos comparten y compiten por los recursos para llevar a cabo la tarea que tienen asignada. Cada par de aristas antiparalelas entre un par de vértices corresponde a un recurso compartido por dos procesos. Cada proceso requiere todos los recursos que se le pueden asignar para operar.

La importancia de los marcajes vivos-seguros, desde la perspectiva del problema en cuestión radica en que garantizan aspectos de funcionamiento correcto en el sistema. Por ejemplo ausencia de interbloqueo y ausencia de espera infinita. También garantizan que un recurso es asignado a uno y sólo a un proceso.

Un caso particular del problema de los procesos en competencia con recursos compartidos, es el de la calendarización de sistemas distribuidos. En tal caso las MGS con marcajes vivos-seguros ofrecen un modelo de protocolo de operación correcta y libre de colisiones.

El modelo de las MGS con marcajes vivos-seguros es discutido y estudiado en esta tesis. Así como, su relación con el problema de las calendarizaciones de sistemas distribuidos.

Con el propósito de hacer el trabajo autocontenido, se ofrece en el **Capítulo 1** una introducción a los conceptos básicos de la teoría de gráficas y a los sistemas distribuidos. Posteriormente en el **Capítulo 2**, son tratados en su totalidad los temas de *MG*, marcajes vivos, marcajes vivos-seguros y el disparo de vértices.

Los problemas referentes a la obtención de marcajes vivos y vivos-seguros, son caracterizados y definidos formalmente. Así como, un par de algoritmos para resolverlos (llamados en este trabajo algoritmo *L* y *LS*) sugeridos por F. Commoner et al. en [CHEP71] son desarrollados, probados y analizados por completo en el **Capítulo 3**. En donde también se demuestran ciertas propiedades relacionadas con la conexidad que deben cumplir la gráficas para poder contar con un marcaje vivo-seguro.

El modelo *MGS* es presentado en el **Capítulo 4**, con una discusión general de la relación que guarda con las calendarizaciones. Brindando un tratamiento detallado del algoritmo *MGGS* de mejor complejidad computacional que el algoritmo *LS*. El algoritmo *MGGS* sirve para obtener marcajes vivos-seguros para esta clase de gráficas.

En esta tesis también se estudia el problema de hallar el total de marcajes vivos-seguros para una *MGS* dada. La importancia de dicho resultado es que representa el número de posibles asignaciones de los recursos compartidos en el problema de los procesos en competencia. Además da una idea precisa del tamaño del espacio de búsqueda de la mejor asignación. Todo ello es estudiado en el **Capítulo 5**.

El tratamiento para encontrar el total de marcajes vivos-seguros es similar al sugerido por R. P. Stanley en [S73] y [S74] respecto a las orientaciones acíclicas de una gráfica. Si G es una gráfica finita con $|V|$ vértices y ρ_G es su polinomio cromático, se prueba que $(-1)^{|V|}\rho_G(-1)$ es el total de orientaciones acíclicas para G . Dicho resultado es idéntico al número de marcajes vivos-seguros para una *MG* cuya gráfica subyacente simple es G .

Finalmente es posible afirmar que la obtención de un marcaje vivo-seguro para una *MGS* arbitraria, garantiza una ronda inicial para la

calendarización del sistema bajo estudio. Dicha calendarización es *eventualmente periódica*, es decir que una secuencia de rondas se repite un número infinito de veces. Por las propiedades de los marcajes vivos-seguros se asegura que el sistema está libre de colisión en los mensajes, de interbloqueo y de espera infinita de los procesos. Por otra parte, la caracterización del total de marcajes vivos-seguros para esta familia de gráficas (MGS) proporciona una base de la cual partir para continuar un tratamiento más profundo respecto a la eficiencia de las calendarizaciones. Por ejemplo, se puede intentar responder cuestionamientos tales como:

¿ Cómo encontrar un marcaje vivo-seguro para una MGS tal que la calendarización del sistema bajo estudio contenga el mínimo número de rondas ?.

Finalmente en las **Conclusiones** se resumen los resultados más importantes de este trabajo y las direcciones posibles de investigación que se pueden continuar.

Capítulo 1

Preliminares

Debido a que el presente trabajo trata en su totalidad con el modelo de gráficas marcadas (MG), en este capítulo se sintetizan los aspectos fundamentales en los que se apoyan los desarrollos subsecuentes. Para ello se brinda en primer lugar, una introducción a los conceptos de la teoría de gráficas y posteriormente se presentan de manera general los tópicos relacionados con sistemas distribuidos (SD), apegado tal tratamiento a la modelación mediante gráficas.

Finalmente y como resultado natural de la discusión, se establece un nexo directo con la temática del siguiente capítulo al evidenciar la necesidad de contar con un modelo más poderoso que permita abstraer adicionalmente al elemento estructural, las características de comportamiento dinámico (cambios de estado) inherentes en los SD.

1.1 CONCEPTOS BÁSICOS DE LA TEORÍA DE GRÁFICAS

La teoría de gráficas tiene aplicaciones en diversas disciplinas y es por ello que la terminología empleada no ha sido del todo estandarizada. Tal situación ha llevado a describir conceptos diferentes bajo un mismo término por distintos autores, de aquí que, cuando sea posible se indica en las subsecuentes definiciones las diversas alternativas que son más frecuentemente empleadas en la literatura del área de las ciencias de la computación.

Definición 1.1: Una gráfica $G = (V, E)$ es un sistema que consiste en un conjunto no vacío V , de vértices y un multiconjunto E de pares de

vértices llamados aristas, y sin que ello implique que tales pares son ordenados.

En la literatura los vértices son llamados también nodos o puntos.

De la definición 1.1 es directo que cada arista $e \in E$ está asociada con un par de vértices no necesariamente distintos $u, v \in V$; y en tal caso e es incidente a los vértices u y v , o los vértices finales de la arista e son u y v . Cuando un par de vértices son los vértices finales de una arista se les nombra vértices adyacentes o vecinos.

A partir de este momento cuando se establezca que G es una gráfica se asume, a menos que se indique lo contrario, que G es finita, esto es, que ambos conjuntos V y E son finitos.

El grado de un vértice $v \in V$ denotado como $d(v)$, está definido como el número de aristas que son incidentes a v . Un vértice está aislado si su grado es 0. Un bucle en una gráfica se refiere a una arista cuyos vértices finales son el mismo vértice. Cuando dos o más aristas tienen los mismos vértices finales se les conoce como aristas paralelas.

Definición 1.2: *Cualquier gráfica que contenga aristas paralelas es una multigráfica. Una gráfica sin bucles ni aristas paralelas es llamada gráfica simple.*

El término gráfica es usado por algunos autores para referirse a una gráfica simple. Así que para evitar cualquier tipo de confusión los últimos términos definidos son resumizados en la tabla 1.1

Término	¿ Bucles ?	¿ Aristas Paralelas ?
Multigráfica	Permitido	Necesario
Gráfica	Permitido	No
Gráfica Simple	No	No

Tabla 1.1

Ejemplo 1.1

Si consideramos la gráfica G de la figura 1.1, se puede observar que los vértices son esquematizados como círculos y las aristas como líneas; por lo tanto $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ y $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ con lo que G es finita. Debido a que la arista e_5 es incidente a v_2 y v_4 se afirma que tales vértices son los vértices finales de e_5 (v_2 y v_4 son adyacentes). Por otra parte, $d(v_4) = 2$ y $d(v_6) = 0$ (v_6 está aislado). Dado que e_7 es un bucle, la gráfica no es simple y más aún ya que e_3 y e_4 son aristas paralelas G es una multigráfica.

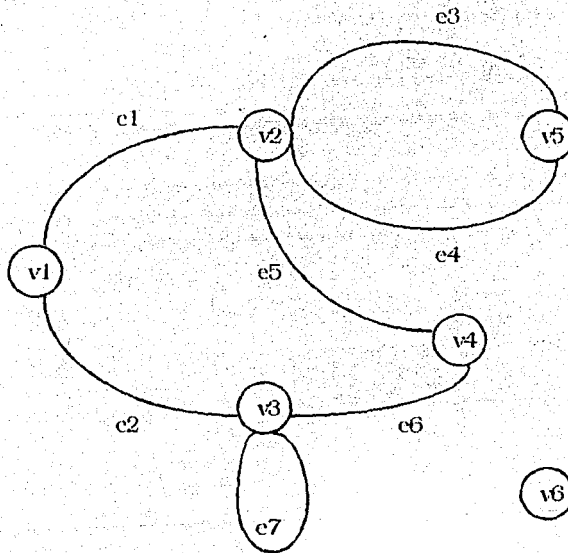


Figura 1.1

Definición 1.3: Un camino es una secuencia de aristas e_1, e_2, e_3, \dots tal que:

- e_j y e_{j+1} son incidentes al mismo vértice.

- Si e_j no es un bucle y no es la primera ni la última arista del camino entonces comparte uno de sus vértices finales con e_{j-1} y otro con e_{j+1} .

Un camino es simple si ningún vértice es repetido. La longitud o distancia de un camino se refiere al número de aristas que contiene. Un ciclo es un camino cuyo vértice inicial y final es el mismo. Un ciclo es simple si ningún vértice a excepción del inicial (final) es repetido.

Un vértice $v \in V$ es alcanzable desde otro vértice $u \in V$ si \exists al menos un camino en G de u a v . Si v es alcanzable desde u entonces u es alcanzable desde v . Una gráfica es conexa si $\forall u, v \in V$ se cumple con que v es alcanzable desde u . Un árbol es una gráfica conexa y sin ciclos.

Definición 1.4: Una gráfica completa con n vértices, denotada como K_n es una gráfica simple en la que $\forall u, v \in V | u \neq v, e = (u, v) \in E$.

La distancia de un camino entre un par de vértices está definida como el número de aristas en el camino. Para un par de vértices cualquiera $u, v \in V$; $\delta(u, v)$ denota la distancia del camino más corto entre ellos. Así de la gráfica de la figura 1.1 tenemos que $\delta(v_1, v_1) = 0, \delta(v_2, v_3) = 2$ y $\delta(v_5, v_6) = \infty$.

Definición 1.5: La vecindad de un vértice $v \in V$ se define como $\Gamma(v) = \{w | w \in V \text{ y } \delta(v, w) \leq 1\}$. Un conjunto $W \subset V$ es 1-esparcido si se cumple con que $\forall u, v \in W \delta(u, v) > 1$.

Ejemplo 1.2

Si $P = e_2, e_6, e_5$, $P' = e_1, e_3, e_4, e_5$, $C = e_1, e_2, e_6, e_5$, y $C' = e_1, e_2, e_6, e_5, e_3, e_4$, son cuatro secuencias de aristas de la gráfica de la figura 1.1, podemos afirmar que:

- P es un camino simple de longitud 3.
- P' es un camino que no es simple de longitud 4.
- C es un ciclo simple de longitud 4.

- C es un ciclo que no es simple de longitud 6.

Sin embargo, la secuencia de aristas e_1, e_3, e_5 no es un camino.

De la misma figura, si $A = \{v_1, v_4, v_5\}$, $B = \{v_2, v_3\}$ y $C = \{v_1, v_2, v_5\}$ son tres subconjuntos de V es fácil ver que A y B son 1-*esparcidos* en tanto C no lo es.



Una gráfica es bipartita si el conjunto V de vértices puede ser dividido en un par de subconjuntos V_1 y V_2 tales que $V_1 \cup V_2 = V$ y $V_1 \cap V_2 = \emptyset$ y además $\forall e = (u, v) \in E$ se cumple con que $u \in V_1$ y $v \in V_2$ o bien $u \in V_2$ y $v \in V_1$.

Una digráfica o gráfica dirigida se define de manera similar a las gráficas, salvo que en el caso de las digráficas las aristas son pares ordenados de vértices. Si $e = (u, v)$ es una arista cualquiera de una digráfica donde u y v no son necesariamente distintos, entonces se dice que v es el vértice final y u es el vértice inicial de la arista. Las aristas de una gráfica dirigida también son conocidos como arcos.

El grado de salida de un vértice $v \in V$ denotado como $d^-(v)$ se refiere al número de aristas para las cuales v es el vértice inicial. De manera similar, el grado de entrada $d^+(v)$ está definido como el número de aristas, para las cuales v es el vértice final.

A una arista de la forma $e = (u, w)$, donde $u = w$ en una digráfica, se le denomina bucle. Todo conjunto de aristas tales que su vértice inicial y final son el mismo, son llamadas aristas paralelas. Un par de aristas e_1 y e_2 son antiparalelas si el vértice inicial de e_1 es el vértice final de e_2 y el vértice final de e_1 es el inicial de e_2 .

Un camino dirigido es una secuencia de aristas e_1, e_2, e_3, \dots tal que el vértice final de e_i es el vértice inicial de e_{i+1} . Un camino dirigido es simple si ningún vértice es repetido.

Un ciclo dirigido o circuito, es un camino dirigido tal que el primer y último vértice del camino son el mismo vértice. Un circuito es simple si el primer (último) vértice es el único repetido. Una digráfica es fuertemente conexa si $\forall u, v \in V, \exists$ al menos un camino dirigido de u a v .

Definición 1.6: La gráfica subyacente de una digráfica G , es la gráfica (no dirigida) resultante de ignorar las direcciones de las aristas de G .

Ejemplo 1.3.

La figura 1.2 ilustra una digráfica de 6 vértices y 11 aristas. Es importante observar que dado que las aristas son pares ordenados, deben ser representadas con dirección, con el fin de mostrar dicha propiedad. Así en $e_1 = (v_1, v_2)$ el vértice inicial es v_1 y el final v_2 lo que en la figura aparece en forma de arista dirigida de v_1 a v_2 . Por otro lado es fácil ver que:

- $d^-(v_3) = 3$ y $d^+(v_3) = 2$
- $e_{10} = (v_5, v_5)$ es un bucle
- e_8 y e_{11} son aristas paralelas
- e_6 y e_7 son aristas antiparalelas
- G no es fuertemente conexa ya que \exists un camino de v_i a $v_j \forall 2 \leq i \leq 6$.
- $P = e_2, e_5, e_8, e_9$, es un camino dirigido simple de v_1 a v_2 de longitud 4 y $P' = e_1$, es también un camino dirigido simple de v_1 a v_2 pero de longitud 1.
- $P'' = e_1, e_4, e_{11}, e_9$, es un camino dirigido de v_1 a v_2 que no es simple de longitud 4.
- $C = e_4, e_8, e_9$, es un circuito simple de longitud 3.
- $C' = e_7, e_{10}, e_6$, es un circuito de G que no es simple de longitud 3.

Finalmente en la figura 1.3 aparece la gráfica subyacente de la digráfica de la figura 1.2.

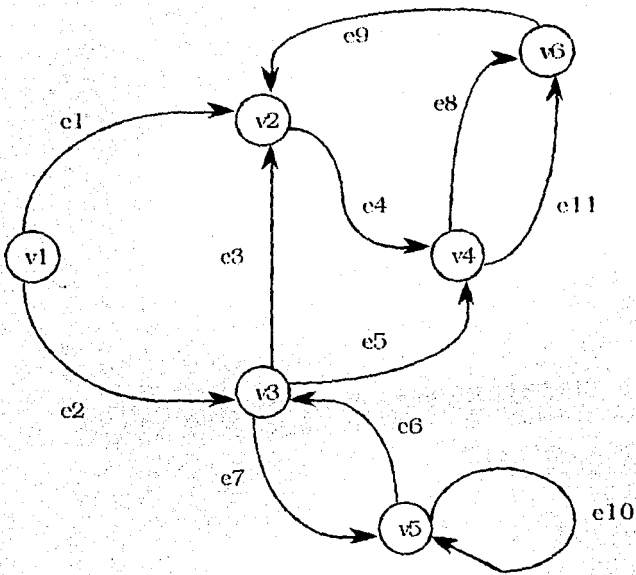


Figura 1.2

Existen diversas maneras para representar una gráfica $G = (V, E)$. La forma más directa de representación es la matriz de adyacencias, en donde una matriz cuadrada de $|V| \times |V|$ es usada y el ij -ésimo elemento es 1 si \exists al menos una arista del vértice i al vértice j , y en cualquier otro caso es 0. La principal ventaja consiste en que se puede saber de manera inmediata si una determinada arista está presente o no en la gráfica, independientemente del número de aristas y de vértices. Sin embargo, la matriz requiere $\theta(|V|^2)$ memoria aún y cuando la gráfica tenga muchas menos aristas. Otro método alternativo de representación de gráficas es la lista de adyacencias, en donde $\forall v \in V \exists$ una lista $ady[v]$ de todos los vértices adyacentes a v con lo que una gráfica puede ser representada con memoria $\theta(|E|)$.

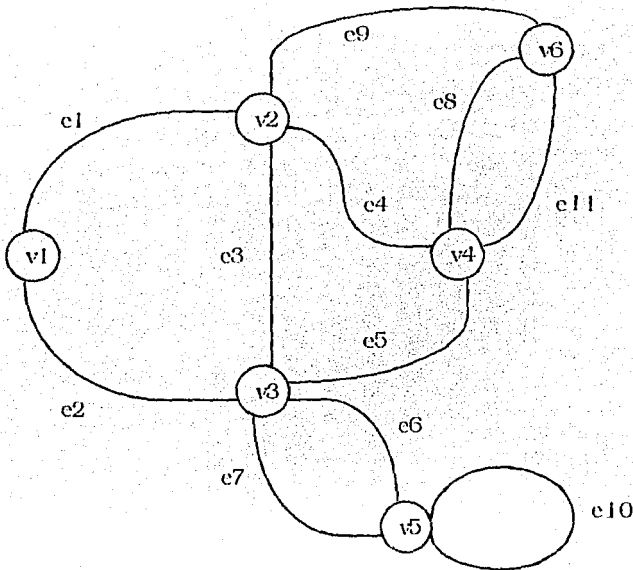


Figura 1.3

1.2 MODELADO DE SISTEMAS DISTRIBUIDOS

Un *SD* está formado por un conjunto de unidades computacionales (también llamadas procesadores) físicamente distribuidas, que cuentan con capacidades de procesamiento y almacenamiento e interconectadas por una red arbitraria de comunicación. Dichas unidades pueden ser programadas para enviar y recibir mensajes sobre los canales de comunicación, así como para ejecutar algoritmos (computaciones locales).

En cuanto a las líneas de comunicación en términos generales se puede afirmar que existen tres tipos.

- "*Simplex*": Una sola dirección de comunicación.
- "*Half - Duplex*": Dos direcciones de comunicación pero con la

restricción de que el envío de mensajes no puede hacerse en ambas direcciones simultáneamente.

- "*Full - Duplex*": Dos direcciones de comunicación con la posibilidad de envío de mensajes en ambas direcciones simultáneamente.

Un *SD* (tal y como se ha definido) puede ser modelado mediante una gráfica no dirigida y conexa $G=(V,E)$; en donde el conjunto V de vértices representa al conjunto de procesadores y el conjunto E de aristas describe los canales de comunicación bidireccional ("*Half - Duplex*" o "*Full - Duplex*") entre los procesadores.

Es necesario observar que, para modelar líneas de comunicación unidireccional ("*Simplex*") debe usarse una gráfica dirigida. Entonces, usando el modelo de gráficas para estudiar *SD* diremos que un vértice es capaz de:

- Ejecutar computaciones locales.
- Procesar los mensajes recibidos de sus vecinos (vértices adyacentes).
- Enviar mensajes a sus vecinos.

Adicionalmente, si un procesador requiere enviar un mensaje a un vértice no adyacente a él, dicho mensaje debe transmitirse a través de un camino (en donde cada vértice intermedio se utiliza como un repetidor).

En el presente trabajo se hacen las siguientes suposiciones respecto al modelo, dichas suposiciones también aparecen en [A85] entre otros.

- La red está descrita por una gráfica simple.
- Todos los vértices tienen identificador (id) distinto.
- La red es asíncrona (no hay reloj global).
- No hay memoria común entre los procesadores.

- No hay fallas.
- Cada vértice procesa los mensajes recibidos de sus vecinos.
- Cada vértice puede ejecutar computaciones locales.
- Cada vértice es capaz de enviar mensajes a sus vecinos.
- La longitud de los mensajes es fija.
- Los mensajes llegan en un tiempo indeterminado pero finito.
- La topología de la gráfica es fija.

Cuando se cuenta con un *SD*, lo primero en lo que se puede pensar, es en aprovechar las capacidades de procesamiento y almacenamiento del sistema en conjunto para resolver un determinado problema. Sin embargo, es evidente que para lograrlo resulta necesario coordinar las actividades de los diversos procesadores por medio del envío y recepción de ciertos mensajes a través de los canales de comunicación.

Definición 1.7: Un algoritmo o protocolo distribuido (o simplemente un protocolo) es el ensamble de un conjunto de algoritmos ejecutándose en los diversos procesadores del *SD* para hallar una solución común a un problema particular.

Partiendo de la base de que un sistema es un conjunto, ensamble o agrupación de elementos interrelacionados de alguna manera que ejecutan una función que es imposible llevar a cabo por cualquiera de las partes, es posible abstraer dos aspectos inherentes a todos los sistemas. Un componente estructural que se refiere a los elementos e interrelaciones entre ellos y un componente de comportamiento dinámico que corresponde a los estados y a los cambios de estado que ocurren en el sistema (función del sistema).

Desde la perspectiva de los sistemas, cuando se diseña un protocolo distribuido es posible pensar en dos diferentes niveles de abstracción:

- *Nivel estructural*: Que corresponde a los procesadores e interconexiones entre ellos (topología del sistema).
- *Nivel de comportamiento dinámico*: Que se refiere a los estados globales del *SD* y a los cambios de un estado a otro (transición).

Respecto al segundo nivel estableceremos que:

El estado global de un protocolo es el conjunto de los estados locales de los procesadores y de los canales de comunicación en un momento dado, en donde el estado local de un procesador es sólo uno de dos posibles:

- Tiene permiso para utilizar las líneas o
- No tiene el permiso.

El estado local de un canal de comunicación queda definido como la secuencia de mensajes enviados a través de dicha línea, menos los mensajes que se han recibido al final del canal.

Dentro de las principales aplicaciones de los protocolos distribuidos encontramos: La transmisión de imágenes, voz y datos entre los procesadores físicamente distribuidos, los cuales deben enviar y coordinar estas transmisiones. Por lo que una de las actividades más importantes de los *SD* consiste en contar con algún mecanismo para asignar permiso a los procesadores de la utilización de los canales, sin violar ciertas restricciones, dicho mecanismo se conoce como calendarización.

Definición 1.8: Una calendarización de una gráfica $G = (V, E)$ consiste de una secuencia de rondas $r = [r_1, r_2, r_3, \dots]$ tal que $\forall i$ se cumple que r_i es un subconjunto de V , en donde si $v \in r_i$ entonces $\forall u \in I(v)$ se tiene que si $v \neq u$ entonces $u \notin r_i$ o en otras palabras $\forall i$, r_i es un subconjunto 1-esparcido de V .

De manera menos formal podemos pensar en cada ronda de la calendarización como el conjunto de procesadores que tienen permiso de transmisión en ese momento. Un ejemplo de aplicación práctica del tipo de coordinación que se modelan con las calendarizaciones, es el de las redes de radio en donde cada estación puede enviar mensajes a otras estaciones vecinas, sin embargo un par de estaciones vecinas no pueden transmitir simultáneamente debido a que dichas transmisiones interferirían entre ellas.

A pesar de que el objetivo de esta tesis se centra en los aspectos de funcionalidad de una calendarización modelada mediante *MG*, en el sentido de que se trata de lograr ciertas características que tienen que ver con la operación correcta del *SD* (por ejemplo ausencia de interbloqueo y de espera infinita). Es necesario mencionar que hay aspectos de la calendarización relacionados con la eficiencia del sistema (por ejemplo maximizar el número de procesadores en cada una de las rondas), que resultan ser de muy alta complejidad, como se muestra en la parte final del capítulo 5, al calcular el número de rondas posibles para un *SD*.

Definición 1.9: Un calendarizador es un protocolo que determina una calendarización.

El nivel estructural de un *SD* puede ser modelado con una gráfica, pero para estudiar el comportamiento dinámico se hace necesario contar con: alguna forma de expresar los diferentes estados globales y una serie de reglas que indiquen como se da el cambio de estados.

Diferentes modelos de gráficas han sido propuestos y usados para estudiar sistemas concurrentes y distribuidos, cada uno de los cuales difiere en generalidad y alcance dependiendo de las propiedades que se requiera analizar. En este trabajo se emplea el modelo de *MG* para tales fines debido a la utilidad que presenta para expresar los estados globales de un *SD*. Además de contar con la herramienta suficiente para modelar las transiciones del sistema con lo que se ajusta de manera muy natural al estudio de las calendarizaciones.

Capítulo 2

Gráficas Marcadas

Una gráfica marcada MG es un modelo formal que permite describir y analizar cierto tipo de sistemas que presentan actividades concurrentes y asíncronas. Debido a que las MG son una subclase del modelo de Redes de Petri (PN) - con la restricción de que cada lugar tiene a lo más una transición de entrada y una de salida -, es importante recalcar que las MG permiten modelar actividades paralelas, pero no así actividades alternativas.

El capítulo comienza con una presentación del modelo ejemplificando, hasta donde es posible, cada uno de los conceptos y la relación que guardan estos con los SD . Posteriormente se analizan los resultados más importantes relacionados con dos clases de marcajes, los llamados marcajes vivos y los marcajes vivos-seguros. Dichos resultados aparecen en su totalidad en [CHEP71] y son tratados también en [R91] entre otros.

La importancia de estudiar los marcajes vivos y los marcajes vivos-seguros, desde la perspectiva de este trabajo, radica en que garantizan las propiedades de funcionamiento correcto de las calendarizaciones mencionadas en el capítulo anterior.

2.1 EL MODELO

No obstante que las MG son un modelo restringido de las PN , en el sentido de que no es posible atacar ciertos problemas de decisión con dependencia de datos. Sí permiten describir las características estructurales y las propiedades dinámicas de los SD , como se describe posteriormente.

El estado global de un *SD* está dado por el conjunto de estados locales de las unidades computacionales y de los canales de comunicación. En el modelo de las *MG* para expresar el estado local de un canal de comunicación, la arista asociada en la *MG* contiene fichas ("tokens") indicadas esquemáticamente como puntos negros, y representan el número de mensajes enviados a través del canal menos el total de mensajes recibidos al final del canal.

El número de fichas de una arista es el marcaje de la arista. Más formalmente el marcaje de las aristas μ , es una función:

$$\mu: E \rightarrow Z^+$$

El marcaje M de una *MG* es un vector de $m = |E|$ números enteros no negativos de la forma $M = \langle \mu(1), \mu(2), \dots, \mu(m) \rangle$, cuyo i -ésimo componente $\mu(i)$ es el marcaje de la arista i , por lo tanto M es una representación de todos los estados locales de los canales de comunicación.

Una *gráfica marcada* es una pareja ordenada de la forma $MG = (G, M_0)$ en donde $G = (V, E)$ es una gráfica dirigida finita y M_0 es un marcaje inicial.

El marcaje inicial M_0 de una *MG* modela únicamente el conjunto de estados locales de las aristas en un momento dado, por lo que se hace necesario:

- Modelar el estado local de los procesadores
- Indicar cómo el modelo evoluciona. Esto es, cómo cambian los estados.

Para ello se cuenta con un par de reglas que se conocen como el juego de fichas ("token - game") y tales reglas son:

1. **Regla de habilitación:** Dada una $MG = (G, M_j)$ un vértice $v \in V$ está habilitado, si $\forall e = (u, v)$ de entrada a v se cumple que $\mu(e) > 0$.

2. Regla de disparo: Un vértice habilitado se dispara tomando una ficha de cada arista de entrada y añadiendo una ficha a cada arista de salida, de tal forma que la adición y la eliminación de marcas se hace simultáneamente (en una operación atómica).

Para los propósitos de este trabajo podemos suponer que las fichas son mensajes de control. Así, un vértice habilitado de acuerdo con la regla 1, corresponde a un procesador en el sistema con permiso de utilización de las líneas, en tanto un vértice no habilitado se refiere al modelado de un procesador sin permiso para establecer comunicación en ese momento.

De la regla 2, debe ser claro que cuando se dispara un vértice, el marcaje de las aristas de entrada y salida cambia (excepto en los bucles) y por lo tanto el marcaje M de la MG también cambia de la siguiente manera.

Sea $M = \langle \mu(1), \mu(2), \dots, \mu(m) \rangle$ el marcaje de una MG antes de disparar un vértice $v \in V$ habilitado en dicho marcaje y sea $\mu'(i)$ el marcaje para la arista $i = (m, n) \in E$, después de disparar v con que:

Si $n = v$ **entonces**

$$\mu'(i) = \mu(i) - 1$$

sino

Si $m = v$ **entonces**

$$\mu'(i) = \mu(i) + 1$$

sino

$$\mu'(i) = \mu(i)$$

y $M' = \langle \mu'(1), \mu'(2), \dots, \mu'(m) \rangle$ es el nuevo marcaje para la MG .

Ejemplo 2.1

Sea una $MG = (G, M_0)$ donde:

$$V(G) = \{a, b, c, d\}$$

$$E(G) = \{e_1 = (a, b), e_2 = (a, c), e_3 = (c, b), e_4 = (b, d), e_5 = (d, a), e_6 = (d, d)\} \text{ y}$$

$$M_0 = \langle 0, 0, 2, 1, 1, 1 \rangle$$

tal y como se ilustra en la figura 2.1.a

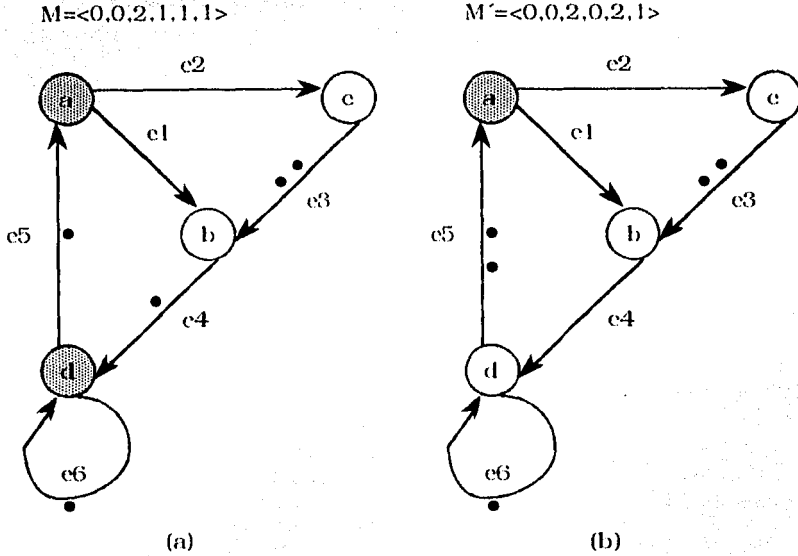


Figura 2.1

Por la regla 1 del juego de fichas tenemos que d y $a \in V$ únicos vértices habilitados. Si disparamos d de acuerdo con la regla 2 obtenemos un nuevo marcaje $M' = \langle 0, 0, 2, 0, 2, 1 \rangle$ en donde por la regla 1 tenemos que a es el único vértice habilitado tal y como se muestra en la figura 2.1.b.



Una observación muy importante es que el número de fichas en una MG puede incrementarse o decrementarse mediante el disparo. Esto es porque si consideramos un vértice $u \in V$ habilitado, $d+(u)$ no es necesariamente igual a $d-(u)$ con lo que al disparar u el total de fichas en la gráfica cambia.

Proposición 2.1: El disparo de un vértice $v \in V$ habilitado en una $MG = (G, M)$ dada, no deshabilita a ningún otro vértice $u \in V \mid u \neq v$.

Demostración:

Sean $u, v \in V$ un par de vértices habilitados en algún marcaje M . Por la regla 1 todas las aristas de entrada a v y u tienen al menos una ficha.

Sin pérdida de generalidad disparemos $v \in V$, por la regla 2 se elimina 1 ficha de cada arista de entrada a v y se adiciona 1 ficha a cada arista de salida. Como no existe ninguna arista que sea de entrada a v y u simultáneamente, entonces $u \in V$ sigue habilitado.



2.2 MARCAJES VIVOS Y MARCAJES VIVOS-SEGUROS

Se dice que un marcaje M es *vivo* si cada vértice de la MG está habilitado o puede ser habilitado a través de una secuencia de disparos.

Un marcaje es *seguro* si $\forall e \in E$ se cumple con que $\mu(e) \leq 1$ y no existe ninguna secuencia de disparos que haga que $\mu(e) \geq 2$.

El *conteo de fichas* de un circuito dirigido C denotado como $(M|C)$ es la suma de fichas de todas las aristas de C en un marcaje M dado.

Proposición 2.2: El conteo de fichas de un circuito dirigido no cambia por el disparo de vértices.

Demostración:

Sea un circuito dirigido C y sea v algún vértice que es disparado.

De la regla 2 del juego de fichas sabemos que las únicas aristas que cambian su marcaje son las de entrada y las de salida al vértice v , con lo que si $v \notin C$ es evidente que C no cambia su conteo de fichas.

Por otra parte si $v \in C$, es fácil ver que únicamente se deben considerar las aristas de entrada y de salida a dicho vértice que pertenecen al circuito, y debido a que cada vez que se visita a v en C se entra y se sale por alguna de tales aristas es directo que el número de veces que se entra es igual al número de veces que se sale del vértice. En general el total de veces que cada arista e del circuito aparece, contribuye con $\mu(e)$ fichas a $\langle M|C \rangle$. Dado que las únicas aristas que cambian su conteo con el disparo son las de entrada y salida a v , es directo que en el nuevo marcaje M' cumple con que $\langle M'|C \rangle = \langle M|C \rangle$

■

Lema 2.1: Dada una $MG = (G, M)$: M es un marcaje vivo si y sólo si \forall circuito dirigido C_i de G se cumple que $\langle M|C_i \rangle > 0$.

Demostración:

Sea C_i algún ciclo dirigido tal que $\langle M|C_i \rangle = 0$, entonces ningún vértice que esté en C_i está habilitado y por la proposición 2.2 sabemos que $\langle M|C_i \rangle$ no cambia, con lo que no existe ninguna secuencia de disparos que habilite al menos un vértice de C_i .

Supongamos que $\forall C_i$ se cumple con que $\langle M|C_i \rangle > 0$ y sea $v \in V$ un vértice cualquiera de la MG . Si v está habilitado todas las aristas de entrada a v tiene al menos una ficha. De otro modo esto es, si v no está habilitado entonces existe un subconjunto no vacío de aristas $E_1 = \{e = (w, v) | e \in E \text{ y } \mu(e) = 0\}$ y existe también un subconjunto no vacío de vértices $V_1 = \{w \in V | e = (w, v) \in E_1\}$. Si todos los vértices de V_1 están habilitados, v se habilita después del disparo de cada uno de los vértices de V_1 . Si al menos uno de los vértices de V_1 no está habilitado entonces debe haber un subconjunto no vacío de aristas $E_2 = \{e = (u, w) | e \in E \text{ y } w \in V_1 \text{ y } \mu(e) = 0\}$ y debe existir también un subconjunto no vacío de vértices $V_2 = \{u \in V | e = (u, w) \in E_2\}$. Continuando con este proceso, finalmente se debe terminar porque G es finita y no existe ningún ciclo dirigido C_i tal que $\langle M|C_i \rangle = 0$, lo que implica que

$$E_1 \cap E_2 \cap \dots = \emptyset \quad \text{y} \quad V_1 \cap V_2 \cap \dots = \emptyset$$

con lo que se obtiene un subgráfica $G' = (V', E')$ de G tal que

$$V' = V_1 \cup V_2 \cup \dots \quad \text{y} \quad E' = E_1 \cup E_2 \cup \dots$$

y es directo que al menos un vértice $z \in V'$ no tiene aristas de entrada en la gráfica G' y $\therefore z$ está habilitado en M . Si disparamos z , entonces la gráfica G' se reduce con un vértice, continuando de esta manera es posible habilitar a $v \in V$.

[CHEP71] ■

Corolario 2.1: Si M es un marcaje vivo para una MG después de un disparo cualquiera, se obtiene un marcaje M' que también es vivo.

Demostración:

Sea una $MG = (G, M)$ tal que M es un marcaje vivo. Por el lema 2.1 tenemos que \forall ciclo dirigido C_i se cumple con que $\langle M / C_i \rangle > 0$. Si disparamos un vértice cualquiera habilitado en M , se obtiene un nuevo marcaje M' , tal que por la proposición 2.2, $\forall C_i$ $\langle M' / C_i \rangle > 0$ con lo que por el lema 2.1, M' es un marcaje vivo.

[CHEP71] ■

La implicación más importante del corolario 2.1 se refiere a que el problema de modelar un sistema libre de interbloqueo mediante una MG , se reduce a encontrar un marcaje inicial M_0 que sea vivo.

Lema 2.2: Un marcaje M vivo, es vivo-seguro si y sólo si toda arista de la MG está en al menos un circuito dirigido C_i tal que $\langle M / C_i \rangle = 1$.

Demostración:

Sea una $MG = (G, M)$ tal que M es un marcaje vivo.

Supongamos que $\forall e \in E \exists$ al menos un circuito dirigido C_i en el que participa e tal que $\langle M/C_i \rangle = 1$ y sea que mediante una secuencia de disparos se obtiene un marcaje M' en donde $\mu'(e) \geq 2 \dots \langle M'/C_i \rangle \geq 2$ (de acuerdo con la proposición 2.2) y M es necesariamente seguro.

Sea que M es un marcaje que además de vivo es vivo-seguro para la MG y sea que \exists al menos una arista $e = (u, v) \in E$ tal que para todo ciclo dirigido C_i en el que la arista participa se tiene que $\langle M/C_i \rangle \geq 2$. Si $\mu(e) = 0$ dado que M es vivo el vértice $u \in V$ se puede habilitar y disparar con lo que $\mu'(e) = 1$. Si consideramos el subconjunto de aristas $E_1 = \{e = (w, u) \in E / \mu'(e) = 0\}$ y consideramos también el subconjunto de vértices $V_1 = \{w \in V / e = (w, u) \in E_1\}$ y así sucesivamente obtenemos una subgráfica $G' = (V', E')$ de G . Donde $V' = V_1 \cup V_2 \dots$ y $E' = E_1 \cup E_2 \dots$ y es claro que $v \notin V'$ porque ello implicaría que \exists al menos un ciclo dirigido C_i en el que participa e tal que $\langle M/C_i \rangle = 1$. Por lo tanto es posible disparar u por segunda vez sin tener que disparar v y $\mu''(e) = 2$ con lo que el marcaje M es seguro !!.

[CHEP71] ■

Corolario 2.2: Dada una $MG = (G, M)$, si M es un marcaje vivo-seguro entonces $\forall e \in E$, existe al menos un circuito dirigido en el que e participa.

Demostración:

Sea $e = (u, v) \in E$ una arista en MG que no participa en ningún circuito y sea también M un marcaje vivo-seguro para MG . Si $\mu(e) = 0$ dado que M es vivo es posible disparar u en algún momento con lo que $\mu'(e) = 1$. Ahora bien si se construye una subgráfica $G' = (V', E')$ de G de forma similar a los lemas 2.1 y 2.2, es fácil ver que $v \notin V'$ porque ello implicaría que e está en algún ciclo dirigido. Así que es posible habilitar por segunda vez al vértice u sin disparar v y obtener un marcaje tal que $\mu''(e) = 2$, con lo que M es vivo-seguro !!.

[CHEP71] ■

El siguiente paso natural consiste en el diseño de algoritmos para obtener marcajes vivos y marcajes vivos-seguros y precisamente de ello trata el siguiente capítulo.

Capítulo 3

Algoritmos para obtener marcajes vivos y vivos-seguros

En el presente capítulo, se desarrollan los algoritmos para obtener marcajes vivos y marcajes vivos-seguros para una MG , sugeridos por F. Commoner et al. [CHEP71], así mismo se demuestra que dichos algoritmos son correctos y se prueba que su complejidad en el peor caso es lineal en E (el número de aristas) para el primero (marcajes vivos) y es $O(|V|^2|E|)$ para el segundo (marcajes vivos-seguros).

Una consideración adicional a nuestro modelo de estudio se refiere a que dada una $MG = (G, M)$ cualquiera, la gráfica subyacente G_s de G se asume conexa. La razón de esta suposición es facilitar el análisis sin afectar la generalidad de los algoritmos en el sentido de que si G_s no es conexa, el algoritmo en cuestión puede ser aplicado a cada componente conexa de la gráfica sin alterar el resultado final.

3.1 ALGORITMO L

Consideremos primero el problema de hallar un marcaje vivo para una MG , definiéndolo formalmente de la siguiente manera:

Problema Π_1 : Hallar un marcaje vivo para una MG .

- *Parámetros:* Una $MG = (G, M)$

• *Propiedades de la respuesta:*

Precondición: G es finita y su gráfica subyacente es conexa

$MG(G, M') \leftarrow L(MG)$

Postcondición: $M' = (\mu'(e_1), \mu'(e_2), \dots)$ es un marcaje vivo para MG .

El siguiente algoritmo resuelve Π_1 de manera directa.

Algoritmo 3.1

gráfica_marcada $\leftarrow L$ (gráfica_marcada γ) {

1. $\forall e \in E(G)$ do

$\mu(e) = 1;$

2. return γ }

Proposición 3.1: El algoritmo L es correcto.

Demostración:

Como G es finita, L siempre termina. Supongamos que al terminar la ejecución del algoritmo existe al menos un ciclo dirigido C_i tal que $\langle M/C_i \rangle = 0$, pero como al final la ejecución del algoritmo tenemos que $\mu(e) = 1 \forall e \in E$ entonces es directo que $\langle M/C_i \rangle = 0$!! y $\therefore M$ es un marcaje vivo.

Proposición 3.2: El algoritmo L es de $O(|E|)$.

Demostración:

Directamente del paso (1) del algoritmo.

Antes de considerar el problema de hallar marcajes vivos-seguros, estudiemos una propiedad relacionada con la conexidad de las MG que tienen dichos marcajes.

Lema 3.1: Si M es un marcaje vivo-seguro para una $MG = (G, M)$, entonces G es fuertemente conexas.

Demostración:

Supongamos que M es un marcaje vivo-seguro para una $MG = (G, M)$ en donde G no es fuertemente conexas, entonces deben existir al menos un par de vértices $u, v \in V$ tales que no hay un camino dirigido de u a v . Sea U el conjunto de vértices alcanzables desde u y desde luego $u \in U$ y $v \notin U$. Como la gráfica subyacente de G es conexas debe haber una arista en G de la forma $e = (v', u')$ tal que $u' \in U$ y $v' \in V - U$ como se muestra en la figura 3.1.

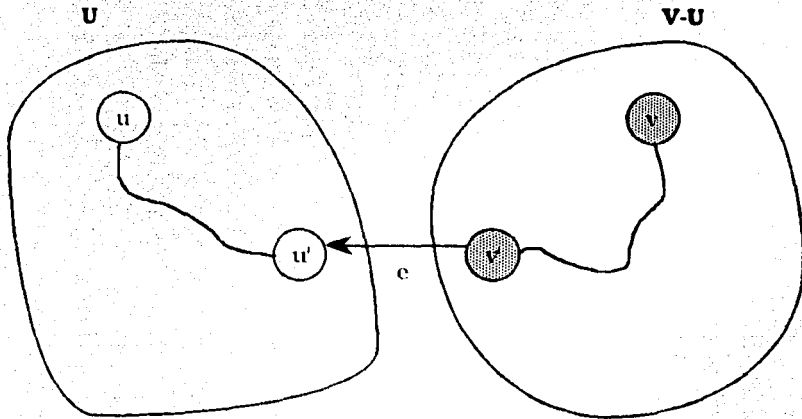


Figura 3.1

Como MG tiene un marcaje vivo-seguro por el corolario 2.2 debe haber un camino de u a v , con lo que $v \notin V - U$ y que G no sea fuertemente conexa !!.

[CHEP71] ■

3.2 ALGORITMO LS

En esta sección se resuelve el problema de encontrar para una $MG = (G, M)$ dada un marcaje vivo-seguro.

Problema Π_2 : Hallar un marcaje vivo-seguro para una MG

- *Parámetros: Una $MG = (G, M)$*
- *Propiedades de la respuesta:*

Precondición: G es finita y fuertemente conexa

$MG(G, M') \leftarrow LS(MG)$

Postcondición: M' es un marcaje vivo-seguro para la MG

■

De acuerdo con F. Commoner et al. es posible resolver Π_2 de la siguiente manera. "Podemos encontrar un marcaje vivo poniendo una ficha en cada arista...podemos usar una técnica... para cambiar el marcaje hasta que sea seguro sin que deje de ser vivo. Sea $k > 1$ el mínimo conteo de fichas para los circuitos en los que una arista participa. Podemos describir una secuencia de disparos para llevar k fichas a la arista en cuestión y eliminando $k - 1$ fichas ningún circuito tiene un conteo de fichas igual a 0 y ahora al menos un circuito en el que la arista participa cumple con que su conteo de fichas es 1. Este proceso puede ser repetido mientras existan aristas que no estén en un circuito con conteo de fichas igual a 1" [CHEP71].

Procedamos a un análisis y desarrollo más concreto de la solución descrita para Π_2 .

La primera cuestión a resolver consiste en encontrar el circuito con menor conteo de fichas en el que participa una arista, y definimos este problema como sigue:

Problema $\Pi_{2,1}$: Encontrar el circuito con menor conteo de fichas en el que participa una arista.

• *Parámetros:* Una $MG = (G, M)$ y una arista $e \in E(G)$

• *Propiedades de la respuesta:*

Precondición: G es finita y fuertemente conexa y M es vivo.

$k \leftarrow \text{MINC}(MG, e)$

Postcondición: $k = \min(\langle M/C_i \rangle \forall C_i / e \in C_i)$

Considerando una arista cualquiera $e = (u, v)$ y siendo $\mu(e)$ su marcaje, es cierto que \forall ciclo dirigido simple que pasa por e , la contribución de dicha arista en el conteo de fichas de cada uno de esos circuitos es $\mu(e)$, tal y como se esquematiza en la figura 3.2.

Proposición 3.3: Si $MG = (G, M)$ es una gráfica marcada tal que M es un marcaje vivo, entonces dada cualquier $e \in E$ se cumple con que el circuito con menor conteo en el que la arista participa es simple.

Demostración:

Supongamos lo contrario, esto es que dada $e = (u, v)$ el circuito C con menor conteo de fichas no es simple. Sea la parte simple de C el circuito C' entonces $\langle M/C' \rangle \geq \langle M/C \rangle$ y esto implica que existe en G al menos un ciclo dirigido C'' tal que $\langle M/C'' \rangle \leq 0$ donde C'' pertenece a C . Pero si $\langle M/C'' \rangle = 0$ implica que M no es un marcaje vivo y que $\langle M/C'' \rangle < 0$ no es posible, porque el marcaje M no está definido para los enteros negativos. $\therefore \langle M/C' \rangle \geq \langle M/C \rangle$!! y se concluye que $\langle M/C' \rangle < \langle M/C \rangle$.

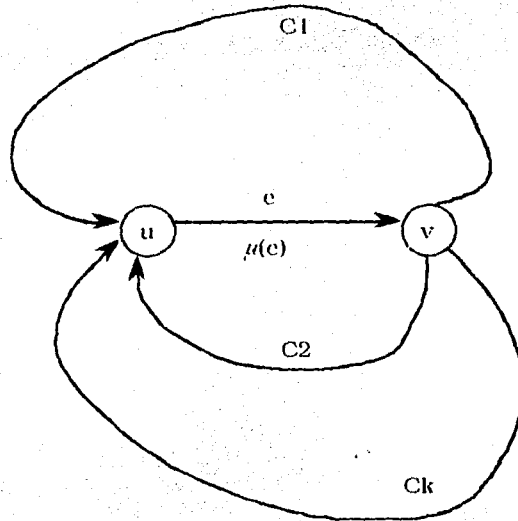


Figura 3.2

La idea principal para encontrar el total de fichas en circuito menor conteo de éstas en el que una arista $e = (u, v)$ participa, consiste entonces en hallar el camino P de v a u con menor conteo y adicionar e a dicho camino para obtener un circuito $C = P + \{e\}$ que cumple con la postcondición de $\Pi_{2,1}$.

Por otra parte, encontrar el camino de v a $u \in V$ con menor número de fichas, es equivalente al problema de hallar el camino más corto entre un par de vértices, si consideramos a $\mu(e)$ como la longitud $\forall e \in E$.

Para resolver el problema $\Pi_{2,1}$ usemos una variante del algoritmo de Dijkstra (camino más corto), y que llamaremos aquí *MINC*. Las pruebas de los lemas para demostrar que el algoritmo es correcto y de análisis de complejidad son similares y están fundamentadas en las demostraciones del algoritmo de Dijkstra que aparecen en el libro de S. Even [E79].

Las estructuras de datos que se utilizan en *MINC* son: En primer lugar una lista de adyacencias (de la manera tradicional), en donde dado un vértice

cualquiera $v \in V$, $\forall e \in \text{ady}[v]$ significa todas las aristas que están en la lista de adyacencias de v , en otras palabras, todas las aristas de la forma $e = (v, x)$ de salida de v . En segundo lugar se cuenta con un conjunto T temporal de vértices que soporta operaciones de inserción, borrado y consulta.

Adicionalmente para cada vértice $v \in V$ existe una etiqueta $\lambda(v)$ que se emplea para contar el número de fichas en un camino desde un vértice inicial hasta v . El algoritmo es pues el siguiente:

Algoritmo 3.2

```

entero  $\leftarrow$  MINC (gráfica_marcada,  $g$ , arista  $e = (t, s)$ ) {
1.  $T \leftarrow V(G)$ ;
2.  $\lambda(s) \leftarrow \mu(e)$ ;
3.  $\forall v \in V / v \neq s$  do  $\lambda(v) \leftarrow \infty$ ;
4.  $u \leftarrow s$ ;
5. while ( $u \neq t$ ) do {
6.    $\forall e \in \text{ady}[u]$  do
7.     if ( $v \in T$ ) and ( $\lambda(v) > \lambda(u) + \mu(e')$ )
           then  $\lambda(v) \leftarrow \lambda(u) + \mu(e')$ ;
8.    $T \leftarrow T - \{u\}$ ;
9.    $u \leftarrow v / \lambda(v) = \min\{\lambda(w) \mid w \in T\}$ ;
10. return  $\lambda(t)$ ;
}

```

Lema 3.2: En el algoritmo MINC: si $\lambda(v)$ es finito para algún vértice v entonces existe un camino P de t a v que pasa por s tal que $\lambda(P) = \lambda(v)$.

Demostración:

Si $\lambda(v)$ es finito entonces por el paso (7) dicha etiqueta fue asignada por otro vértice u de la forma $\lambda(v) \leftarrow \lambda(u) + \mu(e_1)$ donde $e_1 = (u, v)$, después de esto u es eliminado del conjunto T en el paso (8) y $\lambda(u)$ no cambia. Como $\lambda(v)$ es finito entonces $\lambda(u)$ también es finito. Continuando con este proceso se obtiene cada vez un vértice que fue eliminado del conjunto T con anterioridad y por lo tanto ningún vértice es repetido en

el camino, así finalmente se tiene un camino del vértice s al vértice t dado que la etiqueta $\lambda(s)$ es asignada en el paso (2).

Con lo que tenemos que $\lambda(v) \leftarrow \lambda(u) + \mu(e_1)$ donde $e_1 = (u, v)$ pero $\lambda(u) \leftarrow \lambda(w) + \mu(e_2)$ donde $e_2 = (w, u)$ y $\therefore \lambda(v) \leftarrow \lambda(w) + \mu(e_2) + \mu(e_1)$.

Continuando de esta manera finalmente se llega a que $\lambda(v) \leftarrow \lambda(s) + \mu(e_n) + \mu(e_{n-1}) + \dots + \mu(e_2) + \mu(e_1)$ y por el paso (2) sabemos que $\lambda(s) \leftarrow \mu(e)$ donde $e = (t, s)$ y es directo que $P = e, e_n, \dots, e_2, e_1$ es un camino de t a v que pasa por s y $\lambda(v) = \langle M \mid P \rangle$.

[E79] ■

Una suposición que se ha hecho, pero no se ha mencionado es que $\forall e \in E$ tal que M es un vivo-seguro se tiene que $\mu(e)$ es finito, de aquí que:

Lema 3.3: En el algoritmo MINC, cuando un vértice es seleccionado en el paso (9) se cumple con que $\lambda(u) \neq \infty$.

Demostración:

Sea que al momento de ejecutarse el paso (9) el vértice seleccionado u cumple con que $\lambda(u) = \infty$, sin pérdida de generalidad sea también que u es el primer vértice de etiqueta infinita que es seleccionado.

Entonces en el momento que u es seleccionado $\forall v \in T$ se tiene que $\lambda(v) = \infty$ y que $\forall v' \in V - T$, $\lambda(v')$ es finito, por lo tanto una de las dos siguientes condiciones se cumple:

- no existe arista que vaya de un vértice en $V - T$ a un vértice en T
- \forall arista e que vaya de $V - T$ a T , $\mu(e) = \infty$

Pero la primera condición implica que la gráfica no es fuertemente conexa (lo que es inconsistente con el lema 3.1 y la definición de $\Pi_{2,1}$) y la segunda implica que existen aristas tales que $\mu(e) = \infty$!.

[E79] ■

Denotemos $m(v)$ como el total de fichas que hay en el camino con menor número de éstas de t a v que pasa por s , entonces hay que demostrar que al terminar *MINC*, $\lambda(t) = m(t)$.

Lema 3.4: En *MINC*, cuando un vértice u es seleccionado en el paso (9) se cumple que $\lambda(u) = m(u)$.

Demostración:

Por inducción en el orden en que los vértices son eliminados de T .

Paso base: por el paso (4) de *MINC*, tenemos que el primer vértice seleccionado es s y es claro que $\lambda(s) = m(s) = \mu(e)$ donde $e = (t, s)$.

Paso de inducción:

HIPOTESIS: \forall vértice v que es eliminado de T antes que u se cumple que $\lambda(v) = m(v)$.

Cuando u es seleccionado en el paso (9) tenemos por el lema 3.3 que $\lambda(u)$ es finito y por el lema 3.2 \exists un camino P de t a u que pasa por s tal que $\lambda(u) = \langle M | P \rangle$, y $\lambda(u) \geq m(u)$.

Sea $P' = V_0 \xrightarrow{e_1} V_1 \xrightarrow{e_2} V_2 \dots V_{k-1} \xrightarrow{e_k} V_k$ el camino con menor número de fichas de t a u en el que participa $e = (t, s)$, entonces $t = V_0$, $s = V_1$, y también $u = V_k$. Sea también V_i el vértice en P' el último eliminado de T antes de u y por hipótesis:

$$\lambda(V_i) = m(V_i)$$

Si V_{i+1} es el vértice eliminado de T justo después de V_i entonces, hay un par de casos:

i) $V_{i+1} = u$ y

$\lambda(V_{i+1}) \leq \lambda(V_i) + \mu(e_{i+1})$ dado que V_{i+1} pudo haber sido etiquetado desde un vértice distinto a V_i . Por hipótesis $\lambda(V_{i+1}) \leq m(V_i) + \mu(e_{i+1})$. Como después de seleccionar V_i las etiquetas de los vértices que están en T sólo puede decrementarse y como u es seleccionado después de V_i , tenemos que $\lambda(V_{i+1}) \leq m(V_i) + \mu(e_{i+1}) \leq m(u)$, y si $m(u) < \lambda(u)$ entonces $\lambda(V_{i+1}) < \lambda(u)$ y u no puede ser el vértice seleccionado o bien

ii) Si $V_{i+1} = u$ se tiene:

$\lambda(u) \leq \lambda(V_i) + \mu(e_{i+1})$. Por hipótesis $\lambda(u) \leq m(V_i) + \mu(e_{i+1})$ como u es seleccionado después de V_i , $m(V_i) + \mu(e_{i+1}) \leq m(u)$ y dado que al seleccionar V_i , $\lambda(u)$ sólo puede decrementarse $\lambda(u) \leq m(u)$.

Con lo que es imposible que $\lambda(u) > m(u)$ y se concluye que $\lambda(u) = m(u)$.

[E79] ■

Corolario 3.1: Cuando t es seleccionado en el paso (9) de *MlNC* se cumple que $\lambda(t) = m(t)$.

Demostración:

Consecuencia directa del lema 3.4

Proposición 3.4: El algoritmo *MlNC* siempre termina.

Demostración:

Dado que la gráfica es finita, entonces los conjuntos V y E son finitos, con lo que es fácil ver que T es finito y entonces los pasos (1), (3), (6) y (9) terminan. (Por demostrar que el ciclo while del paso (5) termina). Sea que la primera vez que se ejecuta el paso (4) $u = s = t$ entonces el paso (5) no se ejecuta y el algoritmo termina. De otro modo si $u = s \neq t$ se ejecuta el ciclo while y en el paso (8) $u = s$ se elimina de T y en el paso (9) se selecciona un nuevo vértice de T . Como en ningún momento en el ciclo se adicionan vértices a T y debido a que el conjunto T es finito finalmente se cumple que $u = t$, lo que implica que la condición del while es verdadera y podemos concluir que el algoritmo *MINC* siempre termina.

[E79] ■

Lema 3.5: El algoritmo *MINC* es correcto.

Demostración:

Por la proposición 3.4 sabemos que *MINC* termina y por el paso (10) *MINC* regresa $\lambda(t)$. De donde por el corolario 3.1 $\lambda(t) = m(t)$ y dado que $m(t)$ es el total de fichas en el circuito con menor número de éstas de t a t en el que participa la arista $e = (t, s)$, se concluye que *MINC* es correcto.

■

Lema 3.6: El algoritmo *MINC* es de $O(|V|^2)$.

Demostración:

Cada vez que se ejecuta el paso (6) se revisa la lista de adyacencias de algún vértice y como en el peor caso el paso (6) se ejecuta $\forall v \in V - \{t\}$ entonces la complejidad de tiempo es a lo más $O(|E|)$.

Por otro lado el paso (9) consiste en buscar el vértice en T con etiqueta mínima, y es claro que esta operación puede realizarse en $|T|-1$ comparaciones. Como cada vez que se ejecuta el paso (9) el conjunto T ha decrecido en un vértice en relación a la ejecución anterior de (9) y como dicha búsqueda se ejecuta $|V|$ veces a lo más, es fácil ver dado que inicialmente $T = V$ que (9) se ejecuta a lo más:

$$0 + 1 + \dots + V - 1 = \sum_{i=1}^{V-1} i = \frac{(V-1) * V}{2} = \frac{V^2 - V}{2} = O(|V|^2)$$

Como la gráfica no es una multi-gráfica entonces no contiene aristas paralelas así que $|E| \leq |V|^2$ con lo que la complejidad en el peor caso para $MINC$ es $O(|V|^2)$.

[E79] ■

Una vez obtenido $k = MINC(y, e)$, lo que resta es llevar a través de una secuencia de disparos la k fichas a la arista e .

Proposición 3.5: Dada una $MG = (G, M)$ tal que M es una marcaje vivo y dada una arista $e = (u, v)$ de la MG : Si $k = MINC(MG, e)$ entonces u puede ser disparado $k - \mu(e)$ veces sin disparar el vértice v .

Demostración:

Sea C el circuito con menor número de fichas en el que $e = (u, v)$ participa. Por inducción sobre:

$$n = k - \mu(e) = \sum \mu(e_i) \mid e_i \in C \text{ y } e_i \neq e$$

Paso base: Si $C = V_0 \xrightarrow{e_1} V_1 \xrightarrow{e_2} V_2 \dots V_{j-1} \xrightarrow{e_j} V_j \xrightarrow{e} V_0$ donde $V_0 = v$ y $V_j = u$ y si $n = 0$ entonces

$$\sum_{i=1}^j \mu(e_i)$$

\therefore al menos la arista $e_j = (V_{j-1}, V_j) = (V_j, u)$ de entrada a u no contiene fichas con lo que $V_j = u$ no está habilitado: Del mismo modo V_{j-1} no puede estar habilitado porque al menos la arista $e_{j-1} = (V_{j-2}, V_{j-1})$ no contiene fichas. Continuando así es fácil ver que V_1 no está habilitado ya que $\mu(e_1) = 0$ donde $e_1 = (V_0, V_1) = (v, V_1)$ y es directo que para disparar u es necesario disparar v . Con lo que es posible disparar u , $n = 0$ veces sin disparar v .

Paso de inducción:

HIPOTESIS: $\forall 0 \leq i \leq n-1$, es posible disparar u , i veces sin disparar al vértice v .

Como $n > n-1$, se cumple que $n > 0$. Si u está habilitado todas sus aristas de entrada incluyendo e_j contienen al menos una ficha, y u puede ser disparado sin disparar v . De otro modo, si u no está habilitado es posible construir una subgráfica $G' = (V', E')$ de la misma forma que en el lema 2.1 y es claro que $v \notin V'$ porque ello implicaría que $k = \mu(e)$ y $n = 0$ lo que contradice que $n > 0$. \therefore es posible disparar una vez u sin disparar v y justo después de efectuar el disparo de u ocurre que $n = n-1$ lo que por hipótesis nos lleva a que es posible disparar v , $n-1+1 = n$ veces sin disparar u .

Problema $\Pi_{2,2}$: Dada una arista $e = (u, v)$ disparar el vértice u , un cierto número de veces sin disparar al vértice v .

- *Parámetros:* Una $MG = (G, M)$, una arista e y un entero j .
- *Propiedades de la respuesta:*

Precondición: G es finita y fuertemente conexa,

$e \in E(G)$.

M es un marcaje vivo y

$j = k - \mu(e)$ donde $k = \min(\langle M | C_i \rangle \forall C_i | e \in C_i)$.

$MG(G, M') \leftarrow DISP(MG, e, j)$

Postcondición: M' es un marcaje vivo obtenido a partir de M por medio de una secuencia legal de disparos y $\mu'(e) = k$.



Para facilitar el análisis, en el algoritmo $DISP$ adicionalmente a la lista de adyacencias se utiliza una lista de entradas en donde $\forall v \in V \exists$ una lista que contiene al conjunto de aristas de entrada a v , y tal lista se denota como $ent[v]$. Es muy importante observar que en lo referente al espacio de almacenamiento requerido para representar la gráfica, la inclusión de la lista de entradas no incrementa la complejidad de almacenamiento y esto es porque ambas listas emplean $O(|E|)$ espacio y tenemos que asintóticamente $O(2|E|) = O(|E|)$.

El algoritmo $DISP$ está numerado como algoritmo 3.3 y su pseudocódigo es el siguiente.

Algoritmo 3.3

```

gráfica_marcada  $\leftarrow$  DISP(gráfica_marcada  $\gamma = (G, M)$ , arista  $e = (u, v)$ ,
                               entero  $j$ ) {
1.  if ( $j \geq 1$ ) then {
2.       $\forall e' = (w, u) \in ent[u]$  do
3.          if  $\mu(e') = 0$  then  $\gamma \leftarrow DISP(\gamma, e', 1)$ ;
4.       $\forall e \in ent[u]$  do  $\mu'(e) \leftarrow \mu(e) - 1$ ;
5.       $\forall e'' \in ady[u]$  do  $\mu'(e'') \leftarrow \mu(e'') + 1$ ;
6.       $\gamma \leftarrow DISP(\gamma = (G, M'), e, j - 1)$ 
7.  return  $\gamma$  }
    
```

Sea $\sigma = (v_{i_1}, v_{i_2}, \dots, v_{i_k})$ una secuencia de disparos y denotemos σ_i como el número de veces que el vértice v_i aparece en σ , y sea $\xi(v) \subset E(G)$ el conjunto de aristas de entrada a un vértice $v \in V$ tal que $\forall e \in \xi(v)$ se cumple con que $\mu(e) = 0$. Usando esta notación procedamos a demostrar el siguiente:

Lema 3.7: El algoritmo $DISP(\gamma, e, j)$ es correcto.

Demostración:

Por inducción sobre j .

Paso base: Si $j = 0$ entonces la condición del paso (1) no se cumple y únicamente se ejecuta (7) que regresa la gráfica sin modificación en el marcaje. Lo que es correcto ya que por la precondition de $\Pi_2, j = k - \mu(e)$ donde $k = \min(\{M/C_i\} \forall C_i / e \in C_i)$ con lo que $\mu(e) = k$.

Paso de inducción:

HIPOTESIS: $DISP(\gamma, e, i)$ es correcto $\forall 0 \leq i \leq j-1$.

Como $j > 0$, la condición del paso (1) evalúa a verdadero, entonces se ejecutan los pasos (2) y (3) en donde $\forall e \in \xi(u)$ se invoca $\gamma \leftarrow DISP(\gamma, e, 1)$ y dicho circuito debe terminar dado que $E(G)$ es finito.

Sin pérdida de generalidad sea $e = (w, u)$ una de las aristas en $\xi(u)$. Así en una nueva invocación de $DISP(\gamma, e, 1)$ se ejecutan los pasos (2) y (3) ($j > 0$) en donde $\forall e' \in \xi(w)$ se invoca nuevamente $DISP$; y así sucesivamente. Pero dado que M es vivo y G es finita, en algún momento se invoca $DISP$ con una arista $e_i = (x, y)$ tal que $\xi(x) = \emptyset$, en donde $x \neq u$ ya que lo contrario implicaría la existencia de un circuito sin fichas y es directo por el lema 2.1, que M es vivo. Por otra parte $x \neq y$ ya que lo contrario implica que inicialmente $k = \mu(e)$ y $j > 0$!!.

Debido a que $\xi(x) = \emptyset$ entonces se dispara el vértice x en los pasos (4) y (5) del algoritmo para así obtener un nuevo marcaje M' tal que $\mu'(e_i) = \mu(e_i) + 1$. Por el corolario 2.1 M' es un marcaje vivo. Posteriormente, en el paso (6) se invoca $DISP(\gamma = (G, M'), e_i, 0)$ en donde como $j = 0$ por el paso base de la inducción el marcaje M' no se modifica, por lo tanto al ejecutarse el paso (7) se regresa la gráfica $\gamma = (G, M')$ y se tiene que siendo $e_{i-1} = (y, z)$ la arista empleada en la invocación anterior de $DISP$ el conjunto $\xi(y)$ se reduce en una arista. Continuando de esta manera al finalizar los pasos (2) y (3) de la primera

invocación se tiene un marcaje M^m vivo tal que u ni el vértice u , ni el vértice v han sido disparados así que $\mu^m(e) = \mu(e)$.

Al disparar u en los pasos (4) y (5) $\sigma_u = 1, \sigma_v = 0$ y se obtiene un nuevo marcaje M^k que por el corolario 2.1 es vivo y donde:

$$\mu^k(e) = \mu^m(e) + \sigma_u - \sigma_v = \mu^m(e) + 1 - 0 = \mu(e) + 1$$

Por hipótesis al ejecutar el paso (6) desde M^k de la forma $\gamma \leftarrow DISP(\gamma = (G, M^k), e, j-1)$ se obtiene un nuevo marcaje M^j que es vivo en donde:

$$\mu^j(e) = \mu^k(e) + j - 1 = \mu(e) + 1 + j - 1 = \mu(e) + j = \mu(e) + k - \mu(e) = k$$

■

Lema 3.8: El algoritmo *DISP* es de $O(|E|)$.

Demostración:

Cada vez que se ejecuta el paso (2) se revisan todas las aristas $e' = (w, u)$ de entrada a un vértice u y $\forall e' \mu(e') = 0$ se invoca nuevamente *DISP*, con lo que de nueva cuenta se revisan todas las aristas de entrada a un vértice w vecino de u , y así sucesivamente con lo que la complejidad en el peor caso es $O(|E|)$.

Por otra parte, una vez que un vértice ha sido disparado en los pasos (4) y (5) se invoca *DISP* pero el parámetro j es decrementado en 1 y este proceso recursivo continúa hasta que $j = 0$ en donde sólo se ejecuta (7) \therefore en esta última invocación la complejidad es $O(1)$ y tenemos que la complejidad del algoritmo es:

$$O(1) + O(|E|) = O(|E|)$$

■

Una vez diseñados *MINC* y *DISP* podemos resolver el problema Π_2 con el algoritmo *LS* que se emplea tanto a *MINC* como a *DISP*, adicionalmente en *LS* se usa una lista de aristas que soporta los siguientes procedimientos:

- *HEAD(D)*: Regresa el primer elemento de la lista *D* sin modificarla.
- *TAIL(D)*: Regresa la lista *D* sin el primer elemento.
- *IS_EMPTY(D)*: Regresa *true* si la lista *D* está vacía y regresa *false* en cualquier otro caso.

Algoritmo 3.4

```

gráfica_marcada ← LS(gráfica_marcada  $\gamma = (G, M)$ ) {
1.   $D \leftarrow E(G)$ ;
2.   $\forall e \in E(G)$  do  $\mu(e) = 1$ ;
3.  while (IS_EMPTY(D)) do {
4.       $e \leftarrow HEAD(D)$ ;
5.       $k \leftarrow MINC(\gamma, e)$ ;
6.      if ( $k > 1$ ) then {
7.           $\gamma \leftarrow DISP(\gamma, e, k - \mu(e))$ ;
8.           $\mu(e) \leftarrow 1$  }
9.       $D \leftarrow TAIL(D)$  }
10. return  $\gamma$  }
    
```

Proposición 3.6: El algoritmo *LS* siempre termina.

Demostración:

Por la precondition de Π_2 tenemos que *G* es finita y por lo tanto *E(G)* es finito. Así que después de la ejecución de (1) la lista *D* es también finita. Debido a que la ejecución del ciclo while del paso (3) se da mientras *D* no está vacía y dado que en el paso (9) se elimina una arista de *D* en cada iteración y puesto que en ningún lugar del ciclo while se

insertan aristas en D , entonces eventualmente D queda vacía y es directo que LS termina.



Hay un caso que vale la pena revisar y ocurre cuando dada una gráfica dirigida $G = (V, E)$ finita y fuertemente conexa, se cumple con que $|E| = 0$ y $|V| = 1$, pero si sólo existe un procesador en el sistema y no hay canales de comunicación se sigue que el sistema no es distribuido. Así que, para los propósitos de nuestro modelo $|E| \geq 2$, dado que $|V| \geq 2$ y G es fuertemente conexa. La consecuencia de esta suposición es que la condición del paso (3) del algoritmo LS se verifica al menos 3 veces.

Sean I_1, I_2, \dots, I_f las iteraciones del ciclo while de LS , por la proposición 3.6 tenemos que dicho número de iteraciones es finito. Sea I_f el caso cuando D está vacía aunque propiamente no se ejecuta la iteración en I_f esta numeración es útil para demostrar la siguiente.

Proposición 3.7: *En cualquier iteración del ciclo while del algoritmo LS el marcaje de la gráfica es vivo.*

Demostración:

Por inducción sobre el número de iteración I_j .

Paso base: Si $j=1$ entonces antes de ejecutar la primera iteración del ciclo while en el paso (2) se hace que $\forall e \in E(G), \mu(e) = 1$ con lo que \forall circuito C_i de la gráfica $\langle M / C_i \rangle \geq 1$ y por el lema 2.1 el marcaje es vivo.

Dentro del ciclo while los únicos pasos que modifican el marcaje son (7) y (8), así que al ejecutar (5) si $k=1$ entonces (7) y (8) no se ejecutan con lo que el marcaje es vivo en cualquier parte de la iteración. De otro modo, si $k > 1$ se ejecuta (7) en donde dada una arista $e \in E$, su marcaje $\mu(e)$ y un entero $k = \min(\langle M / C_i \rangle \forall C_i / e \in C_i)$ se obtiene a través de una secuencia de disparos un nuevo marcaje M' , que por el lema 3.7 y la definición $\Pi_{2,2}$ es vivo y $\mu'(e) = k$ esto es que el total de fichas del

circuito con menor conteo en que participa la arista e , están precisamente en la arista e .

Supongamos ahora que después de ejecutar (8) el marcaje no es vivo y esto implica que al menos un circuito C es tal que $\langle M/C \rangle = 0$ pero en (8) se hace que $\mu(e) = k - (k - 1) = 1$ con lo que necesariamente $e \in C$ y \therefore antes de ejecutar (8) ocurre que $\langle M'/C \rangle = k - 1$ y se sigue que $k = \min(\langle M/C \rangle \mid \forall C_i, e \in C_i)$. Sin embargo antes de ejecutar (7) y (8) se ejecuta en (5) *MINC* así que por el lema 3.5 necesariamente $k = \min(\langle M'/C \rangle \mid \forall C_i, e \in C_i)$ y \therefore después de ejecutar (8) el marcaje es vivo.

Paso de inducción:

HIPOTESIS: $\forall j, 1 \leq j \leq f - 1$ se cumple con que en la iteración I_j el marcaje es vivo.

Para I_j tenemos que la condición de (3) evalúa a *false* y por lo tanto no se modifica el marcaje pero por hipótesis el marcaje en la iteración I_{j-1} es vivo.

Corolario 3.2: El algoritmo *LS* regresa una gráfica con marcaje vivo.

Demostración:

Consecuencia directa de la proposición 3.7.

Proposición 3.8: Durante la ejecución de alguna iteración del ciclo *while* del algoritmo *LS* ningún circuito dirigido incrementa su conteo de fichas.

Demostración:

Es claro que inicialmente en los pasos (7) y (8) del algoritmo en cuestión se modifica el marcaje. Como en (7) se ejecuta una secuencia

de disparos y por la proposición 2.2 el conteo de fichas de los circuitos no cambia, y antes de ejecutar (8) tenemos que $\mu(e) = k$ por la postcondición de $\Pi_{2,2}$ y el lema 3.7.

Sea que después de ejecutar (8) se obtiene un marcaje M' tal que para algún circuito C se cumple que $\langle M'/C \rangle > \langle M/C \rangle$. Además, debido a que en (8) $\mu(e) \leftarrow k - (k - 1)$ necesariamente la arista pertenece al circuito y $\forall C_i / e \in C_i$ después de ejecutar (8), tenemos que $\langle M'/C \rangle = \langle M/C \rangle - (k - 1)$. Así que, en particular para C , $\langle M'/C \rangle - (k - 1) > \langle M/C \rangle$ lo que implica que $-(k - 1) > 0$, pero dado que el marcaje no está definido para los negativos entonces que $\mu(e) = k < 0$!!

■

Lema 3.9: El algoritmo LS es correcto.

Demostración:

En el paso (4) se toma una arista e cualquiera (la primera de la lista D) y como en (9) dicha arista es eliminada de la lista, entonces por cada arista se ejecuta una iteración del ciclo while. En el paso (5) por el lema 3.5 tenemos que k es el mínimo conteo de fichas de los circuitos en los que la arista participa.

Si $k = 1$ entonces e participa en al menos un circuito C , con $\langle M/C \rangle = 1$ y la condición de (6) evalúa a *false* y se ejecuta únicamente (9) para proceder con las siguientes iteraciones.

Si $k > 1$ la condición de (6) es *true* y se ejecuta (7) que por el lema 3.7 proporciona un nuevo marcaje $M' / \mu'(e) = k$, así al ejecutar (8) resulta un marcaje $M'' / \langle M''/C \rangle = 1$ y $e \in C$, después se ejecuta (9) y se procede así con las siguientes iteraciones.

De este modo al terminar cualquier iteración se asegura que la arista seleccionada está en al menos un ciclo $C / \langle M/C \rangle = 1$, y por la proposición 3.7 es imposible que alguna iteración posterior $\langle M/C \rangle = 0$. Además, por

la proposición 3.8 tenemos que en ninguna iteración posterior $\langle M/C \rangle > 1$, así que una vez ejecutada una iteración para alguna arista e se tiene siempre que $e \in C / \langle M/C \rangle = 1$. Debido a que se ejecuta una iteración para cada arista y siendo que el marcaje regresado por LS es vivo de acuerdo con el corolario 3.2, entonces por el lema 2.2 el marcaje final de la gráfica es vivo-seguro. ■

Una de las consecuencias más importantes resultantes de lo correcto del algoritmo LS se presenta a continuación:

Corolario 3.3: $\forall MG$ tal que G es finita y fuertemente conexa existe al menos un marcaje vivo-seguro.

Demostración:

Como consecuencia directa de la definición de Π_2 y del lema 3.9. ■

Lema 3.10: El algoritmo LS es de $O(|V|^2|E|)$.

Demostración:

En primer lugar es directo que (1) y (2) son de $O(|E|)$ y es claro que siendo las operaciones $HEAD$ y $TAIL$ de $O(1)$, que (4) y (9) son también $O(|E|)$. Por otra parte dado que por el lema 3.6 $MINC$ es de $O(|V|^2)$ entonces (5) es de $O(|V|^2|E|)$. Además por el lema 3.8 sabemos que $DISP$ es de $O(|E|)$ y como (8) se ejecuta l veces donde $l < |E|$ entonces a lo más (8) es de $O(|E|^2)$. Así que no teniendo aristas paralelas $|E| \leq |V|^2$ y por lo tanto $O(|E|^2) \leq O(|V|^2|E|)$ y la complejidad de LS es pues $O(|V|^2|E|)$. ■

Capítulo 4

Gráficas Marcadas Simétricas

En este capítulo nos enfocamos a una clase particular de gráficas marcadas que es bastante útil para estudiar calendarizaciones. Este modelo es el de las gráficas marcadas simétricas (*MGS*). Se presenta un algoritmo para obtener marcajes vivos-seguros para tal familia de gráficas. Sin embargo y pese a que dicho algoritmo no es tan general como *LS* en el sentido de que el último funciona para cualquier gráfica marcada, el nuevo algoritmo *MGS* es más eficiente en términos de complejidad computacional, de hecho es lineal en el número de aristas, es decir es $\mathcal{O}(|E|)$.

4.1 EL MODELO MGS Y LAS CALENDARIZACIONES

Definición 4.1: Una gráfica marcada simétrica *MGS* es una gráfica marcada $MG = (G, M)$ tal que G es finita y fuertemente conexa y donde dados un par de vértices distintos $u, v \in V$ se cumple una de las siguientes condiciones.

- No existen aristas entre u, v o
- Existen un par de aristas antiparalelas $\bar{e} = (u, v)$ y $e' = (v, u) \in E$ entre los vértices u, v .

Y no existen bucles en la gráfica.

Evidentemente una *MGS* cuenta con un marcaje inicial M_0 y debido a que estamos interesados únicamente en marcajes vivos-seguros. Todo marcaje M que sea vivo-seguro es un vector $M = \langle \mu(1), \mu(2), \dots, \mu(m) \rangle$ donde

$m = |E|$, cuyas componentes son binarias debido a que por definición en un marcaje vivo-seguro para una arista particular e se cumple que $\mu(e) = 0$ o $\mu(e) = 1$.

Sin embargo no cualquier cadena binaria de $|E|$ posiciones representa un marcaje vivo-seguro para una MGS.

Ejemplo 4.1

Sea la MGS de la figura 4.1, entonces si consideramos las dos posibles cadenas binarias de 10 posiciones:

$$M' = (0, 1, 0, 1, 0, 1, 0, 1, 1, 0) \text{ y } M'' = (0, 1, 0, 1, 0, 1, 0, 1, 0, 1)$$

que representan dos marcajes distintos para la MGS.

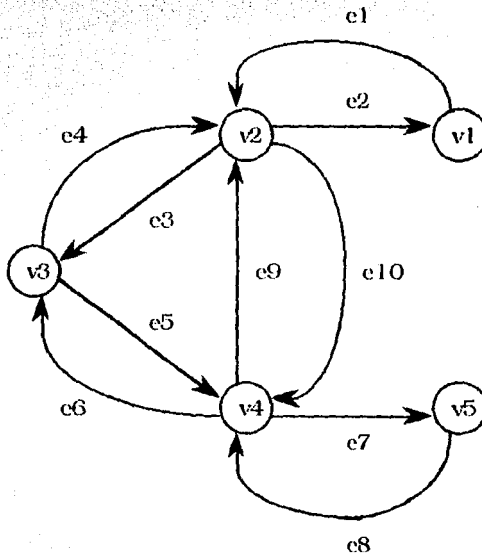


Figura 4.1

Tenemos que M' es un marcaje vivo-seguro y que M'' ni siquiera es un marcaje vivo.

En definitiva, se debe de hacer un análisis más detallado de las propiedades de las MGS que permita evidenciar las características de los marcajes vivos-seguros para esta familia de gráficas.

Lema 4.1: \forall MGS con marcaje vivo-seguro, se cumple con que dado un vértice habilitado u ningún vértice adyacente a u está habilitado en el mismo marcaje.

Demostración:

Sea M un marcaje vivo-seguro para alguna MGS, entonces existe al menos un vértice $u \in V$ que está habilitado, es decir que en cada una de sus aristas de entrada hay exactamente una ficha.

Supongamos que otro vértice $v \in V$ distinto de u tal que u y v son vecinos, está también habilitado en M , entonces en cada una de las aristas de entrada a v hay exactamente una ficha. Por la definición 4.1 existe una arista en cada dirección entre u y v , y debido a que ambos vértices están habilitados en particular para las aristas $e_1 = (u, v)$ y $e_2 = (v, u)$ se cumple que $\mu(e_1) = 1$ y $\mu(e_2) = 1$.

Por lo tanto si se dispara u en el marcaje M se obtiene un marcaje M' tal que $\mu'(e_1) = 0$ y $\mu'(e_2) = 2$, por otra parte si se dispara v en el marcaje M se obtiene un marcaje M'' tal que $\mu''(e_1) = 2$ y $\mu''(e_2) = 0$ con lo que se concluye que M es un marcaje vivo-seguro !!.

La consecuencia más importante del lema 4.1 tiene que ver directamente con el modelado e implantación de una calendarización. En efecto, siendo que un SD es modelado por medio de una gráfica $G = (V, E)$ y siendo que los canales de comunicación son bidireccionales, es posible

convertir a G en una MGS. Esta conversión se realiza mediante una operación bastante simple que consiste en substituir cada arista $e = (u, v)$ de G por un par de aristas dirigidas $e_1 = (u, v)$ y $e_2 = (v, u)$ de la MGS. Es claro que como resultado de esta operación si G es finita y conexa entonces la gráfica de la MGS debe de ser finita y fuertemente conexa.

Por otro lado y de acuerdo al corolario 3.3 es posible obtener un marcaje M vivo-seguro para la MGS. En donde por el lema 4.1 se cumple que $\forall v \in V / v$ está habilitado en M , $\exists u \in \Gamma(v) / u$ está habilitado en M . De aquí que siendo ϑ el conjunto de vértices habilitados en M , dicho conjunto representa una ronda de la calendarización. Esto es que ϑ es un conjunto de procesadores que tienen permiso de transmisión en ese momento.

Considerando que los sistemas bajo estudio son distribuidos, podemos adicionar una regla que se refiere a paralelizar la operación de disparo de las MGS.

Regla de disparo en el bloque: Siendo ϑ el conjunto de vértices habilitados en un marcaje M vivo-seguro para una MGS dada. La operación de bloque se refiere a disparar al mismo tiempo todos los vértices del conjunto ϑ .

Lema 4.2: \forall MGS con marcaje vivo-seguro M ; Si ϑ es el conjunto de vértices habilitados en M entonces la operación de bloque es equivalente a disparar secuencialmente uno por uno y una sola vez los vértices que están en ϑ .

Demostración:

Como M es vivo-seguro entonces inicialmente $\vartheta = \emptyset$ y por el lema 4.1 es directo que no existen aristas entre cualquier par de vértices que esten en ϑ . Adicionalmente y por la definición de habilitación todas las aristas de entrada a cualquier vértice $v \in \vartheta$ contienen una ficha. Ahora supongamos que \exists una arista $e = (v, w) / \mu(e) = 1$ y $v \in \vartheta$ y por el lema 4.1 es claro que $w \in V - \vartheta$ y entonces $w \notin \vartheta$ así es que v está habilitado y w no está habilitado con lo que es posible disparar v y obtener un marcaje

$M^b / \mu^b(e) = 2$ y que M es vivo-seguro !!. \therefore Ninguna arista de salida de cualquier vértice que está en ϑ contiene fichas en M .

Sean pues ι el conjunto de aristas de entrada a los vértices que están en ϑ , ω el conjunto de aristas de salida de los vértices que están en ϑ , y sea $\eta = \{e \in E / e \notin \iota \text{ y } e \notin \omega\}$.

Tenemos que inicialmente $\forall e \in \iota, \mu(e) = 1$ y $\forall e \in \omega, \mu(e) = 0$, así al ejecutar la operación de bloque se obtiene un nuevo marcaje $M^b / \mu^b(e) = 0 \quad \forall e \in \iota$ y $\mu^b(e) = 1 \quad \forall e \in \omega$. Supongamos que alguna arista $e = (a, b) \in \eta$ cambia su marcaje pero $a \notin \vartheta$ y $b \notin \vartheta$ con lo que ni a , ni b fueron disparados y que la arista e cambia su marcaje evidentemente es una contradicción $\therefore \mu^b(e) = \mu(e) \quad \forall e \in \eta$.

Sea ahora que se disparan los vértices que están en ϑ secuencialmente a partir de M . Sin pérdida de generalidad sea $v \in \vartheta$ el primer vértice disparado, entonces se obtiene un marcaje M^v tal que para toda arista e_i^v de entrada a v $\mu^v(e_i^v) = 0$ y para toda arista e_o^v de salida de v $\mu^v(e_o^v) = 1$ pero toda $e_i^v \in \iota$ y toda $e_o^v \in \omega$ y el conjunto ϑ original queda reducido en un vértice $\vartheta' = \vartheta - \{v\}$. Con lo que por la proposición 2.1 siendo ψ el conjunto de vértices habilitados en M^v tenemos que $\vartheta' \subseteq \psi$ y como $\iota \cap \omega = \emptyset$ ninguna otra arista de entrada o de salida a ϑ modifica su marcaje, de este modo es posible disparar a partir de M^v algún vértice $u \in \vartheta' \subseteq \vartheta$. Continuando así es directo que al terminar de disparar una vez cada uno de los vértices que están en el conjunto original ϑ se obtiene un marcaje $M^f / \mu^f(e) = 0 \quad \forall e \in \iota$ y $\mu^f(e) = 1 \quad \forall e \in \omega$ y dado que sólo se disparan los vértices que están en ϑ entonces $\mu^f(e) = \mu(e) \quad \forall e \in \eta$ con lo que $M^f = M^b$.

Corolario 4.1: \forall MGS con marcaje vivo-seguro M después de la operación de bloque se obtiene un marcaje M^b que es vivo-seguro.

Demostración:

Directamente lema 4.2 y del colorario 2.1



El modelado de una calendarización por medio de una MGS con marcaje vivo-seguro M_0 se logra y todos los vértices habilitados en M_0 representan a los procesadores con permiso de transmisión en la ronda r_0 . Después de la operación de bloques se obtiene un nuevo marcaje M_1 que por el colorario 4.1 es vivo-seguro tal que los vértices habilitados son los procesadores de la ronda r_1 y así sucesivamente. Como todos los marcajes sucesivos son de los definidos como vivos-seguros se garantiza la ausencia de interbloqueo y de espera infinita en la calendarización.

Ejemplo 4.2

Sea el sistema distribuido $G = (V, E)$ donde $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ y $E = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_3, v_5), (v_4, v_8), (v_5, v_6), (v_5, v_7), (v_5, v_8), (v_6, v_7)\}$ tal y como se muestra en la figura 4.2.

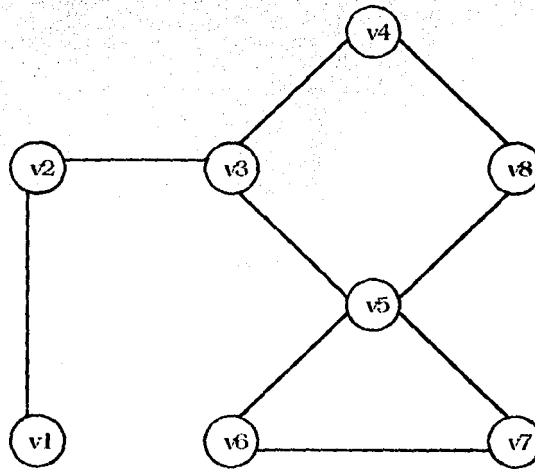


Figura 4.2

Transformado G en una MGS y asignándole un marcaje inicial M_0 el modelo aparece como en la figura 4.3 donde los vértices habilitados son v_1, v_3, v_7, v_8 con lo que la primera ronda de la calendarización está formada precisamente por tales vértices.

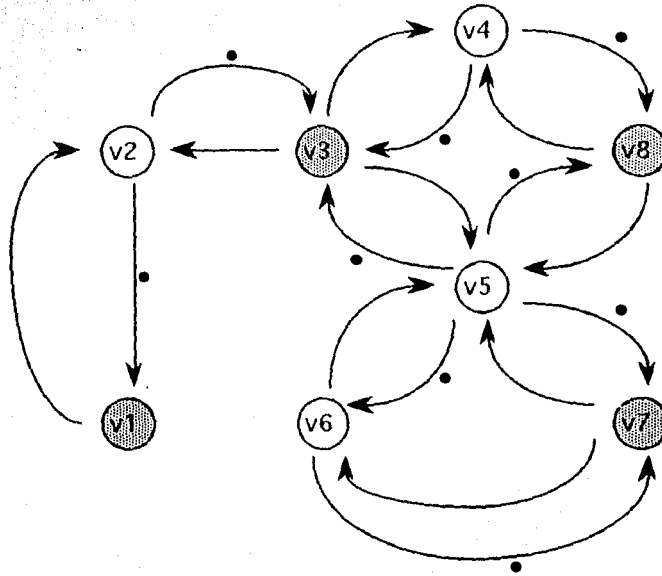


Figura 4.3

Al ejecutar la operación de bloque desde M_0 se obtiene un nuevo marcaje M_1 donde los vértices habilitados son v_2, v_4 y v_6 . Repitiendo esta operación se obtiene un marcaje M_2 tal que los vértices habilitados son v_1 y v_5 , continuando así en M_3 están habilitados v_3, v_7 y v_8 , y si se aplica de nueva cuenta el disparo en bloque a partir de M_3 se regresa a M_1 , por lo tanto la calendarización resultante es:

$$r = \begin{matrix} v_1 & v_2 & v_1 & v_3 & v_2 & v_1 & v_3 \\ v_3 & v_4 & v_5 & v_7 & v_4 & v_5 & v_7 \\ v_7 & v_6 & & v_8 & v_6 & & v_8 \\ v_8 & & & & & & \end{matrix} \dots \text{donde cada columna representa una ronda}$$

Como los marcajes M_1 , M_2 y M_3 se repiten en lo sucesivo, entonces las rondas r_1 , r_2 y r_3 aparecen en ese orden un número infinito de veces en la calendarización y esto se denota de la siguiente manera:

$$r = \begin{matrix} v_1 \\ v_3 \\ v_7 \\ v_8 \end{matrix} \begin{bmatrix} v_2 & v_1 & v_3 \\ v_4 & v_5 & v_7 \\ v_6 & & v_8 \end{bmatrix}^*$$

Lema 4.3: \forall MGS con marcaje vivo-seguro, se cumple siempre que entre cualquier par de aristas antiparalelas una contiene una ficha y la otra no.

Demostración:

Sea M un marcaje vivo-seguro para una MGS y sean $e_1 = (u, v)$ y $e_2 = (v, u)$ cualquier par de aristas antiparalelas de la gráfica.

Supongamos que $\mu(e_1) = \mu(e_2) = 0$ entonces el circuito $C = e_1, e_2$ es tal que $\langle M/C \rangle = 0$ y por el lema 3.1 M no es un marcaje vivo-seguro.

Supongamos ahora que $\mu(e_1) = \mu(e_2) = 1$ y como consecuencia del lema 4.1 u y v no pueden estar habilitados al mismo tiempo en M ; y más aún ni u ni v pueden estar habilitados porque ello implicaría que después del disparo $\mu'(e_1) = 2$ y $\mu'(e_2) = 0$ o $\mu'(e_1) = 0$ y $\mu'(e_2) = 2$ con lo que el marcaje M es vivo-seguro. Sin embargo aún suponiendo que ni u ni v están habilitados en M tenemos por la definición de marcaje vivo-seguro que cualquier vértice está habilitado o puede habilitarse a través de una secuencia de disparos con lo que en algún momento se obtiene un marcaje tal que alguna de las dos aristas e_1 o e_2 contiene dos fichas.

La condición del lema 4.3 es necesaria pero no suficiente de los marcajes vivos-seguros para las MGS; y para ilustrar esto basta con observar el marcaje M'' de la figura 4.1, en donde tal marcaje asocia una ficha entre

FALLA DE ORDEN

cada par de aristas antiparalelas pero M'' no es un marcaje vivo-seguro porque existe un circuito $C = e_3, e_5, e_9$ tal que $\langle M''/C \rangle = 0$.

Corolario 4.2: *Un marcaje M vivo para una MGS es un marcaje vivo-seguro si entre cualquier par de aristas antiparalelas una contiene exactamente una ficha y la otra no contiene fichas.*

Demostración:

Como M es un marcaje vivo y cada arista $e = (u, v)$ participa en un circuito dirigido C formado por la arista y su antiparalela tal que $\langle M/C \rangle = 1$ por el lema 2.2 M es vivo-seguro.

4.2 ALGORITMO MGGS

Problema Π_3 : Encontrar un marcaje vivo-seguro para una MGS

- *Parámetros:* Una $MG = (G, M)$
- *Propiedades de la respuesta:*

Precondición: MG es una gráfica marcada simétrica

$$MG = (G, M') \leftarrow MGGS(MG)$$

Postcondición: M' es un marcaje vivo-seguro para la MG .

La idea, intuitivamente hablando, para resolver Π_3 consiste en ir tomando cada uno de los vértices y revisar las aristas de salida. Si una arista va a un vértice que no ha sido visitado con anterioridad se pone una ficha en tal arista. De otra forma, si la arista va a un vértice que ya ha sido visitado entonces su marcaje se hace 0 y tal proceso termina cuando todos los vértices han sido revisados.

Adicionalmente a la lista de adyacencias se emplea en *MGGS* una lista temporal T de vértices y un predicado *visitado* que tiene como argumento un vértice;

Algoritmo 4.1

```

gráfica_marcada ← MGGS(gráfica_marcada  $\gamma = (G, M)$ ) {
1.   $T \leftarrow V(G)$ ;
2.   $\forall v \in V(G)$  do visitado[ $v$ ] ← false;
3.  while (IS_EMPTY( $T$ )) do {
4.       $v \leftarrow \text{HEAD}(T)$ ;
5.       $\forall e = (v, u) \in \text{ady}[v]$  do
6.          if (visitado[ $u$ ] = false) then  $\mu(e) \leftarrow 1$ 
           else  $\mu(e) \leftarrow 0$ ;
7.       $T \leftarrow \text{TAIL}(T)$ ;
8.      visitado[ $v$ ] ← true }
9.  return  $\gamma$  }
    
```

Proposición 4.1: El algoritmo *MGGS* siempre termina.

Demostración:

De la definición 4.1 tenemos que la gráfica es finita y fuertemente conexa y \therefore la lista T es finita y la lista de adyacencias de cualquier vértice es finita. Así que como en el paso (7) del algoritmo cada vez se elimina un vértice de T y como en ningún paso que este dentro del ciclo se adicionan vértices en T es directo que el ciclo while siempre termina porque es un hecho que en cada iteración el ciclo del paso (5) termina.



Sea que $T = \text{cons}(v, T')$ donde v es un vértice y T' una lista de vértices entonces la especificación de la lista de vértices debe cumplir los dos siguientes axiomas:

$$v = \text{HEAD}(\text{cons}(v, T''))$$

$$T'' = \text{TALL}(\text{cons}(v, T''))$$

Proposición 4.2: En el algoritmo MGGS cuando un vértice v es seleccionado en el paso (4) necesariamente $v \in T$.

Demostración

Supongamos que un vértice $u / u \notin T$ es seleccionado en el paso (4) de alguna iteración del algoritmo, y como se está ejecutando una iteración es directo del paso (3) que $T \neq \emptyset$, así pues en (4) $v \leftarrow \text{HEAD}(T)$ y como $u = v$ entonces $T = \text{cons}(u, T'') \therefore$ que $u \in T$!!



Proposición 4.3: En el algoritmo MGGS en cualquier evaluación del paso (6) se cumple con que $\forall u / u \notin T \text{ visitado}[u] = \text{true}$ y $\forall w / w \in T \text{ visitado}[w] = \text{false}$.

Demostración:

Por inducción en el orden en que los vértices son eliminados de T .

Paso base: Cuando el primer vértice es seleccionado en el paso (4) del algoritmo, de hecho ningún vértice ha sido eliminado de T , y tenemos que debido a los pasos (1) y (2) que $T = V$ y que $\forall v \in V \text{ visitado}[v] = \text{false}$ con lo que al ejecutar los pasos (5) y (6) se cumple que $\forall w / w \in T \text{ visitado}[w] = \text{false}$ y que $V - T = \emptyset$.

Paso de inducción:

HIPOTESIS: $\forall u \in V$ que es eliminado de T antes que v se cumple que $\text{visitado}[u] = \text{true}$ y $\forall w \in T \text{ visitado}[w] = \text{false}$.

Sea la iteración del ciclo while en la que en el paso (4) el vértice $v \leftarrow \text{HEAD}(T)$ es seleccionado, entonces al ejecutar el paso (7) de dicha iteración se elimina un vértice de la lista con la operación $T' \leftarrow \text{TALL}(T)$.

Como $v = \text{HEAD}(\text{cons}(v, T'))$ y $T' = \text{TAIL}(\text{cons}(v, T'))$ el vértice que se elimina de T es v , así en el paso (8) se hace $\text{visitado}[v] \leftarrow \text{true}$ y $v \notin T$. Como valor de visitado de ningún otro vértice es modificado y por la hipótesis tenemos que al ser seleccionado el siguiente vértice v_i de T al ejecutar los pasos (5) y (6) en tal iteración $\forall u \in V$ que es eliminado de T antes que v_i , $\text{visitado}[u] = \text{true}$ y que $\text{visitado}[v] = \text{true}$ y además $\forall w \in T$ incluyendo v_i , $\text{visitado}[w] = \text{false}$.



Lema 4.4: *MGGS produce un marcaje vivo.*

Demostración:

Supongamos lo contrario, esto es que al terminar la ejecución de *MGGS* el marcaje M de la gráfica no es vivo y por lo tanto debe existir al menos un circuito $C = e_1, e_2, \dots, e_n$ tal que $\langle M/C \rangle = 0$, entonces si $e_1 = (v_1, v_2), e_2 = (v_2, v_3), \dots, e_n = (v_n, v_1)$, tenemos que $\mu(e_1) = \mu(e_2) = \dots = \mu(e_n) = 0$.

Dado que $\mu(e_1) = 0$ cuando $v_1 \leftarrow \text{HEAD}(T)$ por la proposición 4.2 $v_1 \in T$ y en (6) se debe cumplir que $\text{visitado}[v_2] = \text{true}$ lo que implica de acuerdo con la proposición 4.3 que en ese momento $v_2 \notin T$. Pero como $\mu(e_2) = 0$ entonces cuando $v_2 \leftarrow \text{HEAD}(T)$ por la proposición 4.2, $v_2 \in T$ y dado que $\mu(e_1) = 0$, $v_1 \in T$ y además se cumple la condición de que $\text{visitado}[v_3] = \text{false}$ al momento de ejecutar el paso (6) por lo tanto según la proposición 4.3 $v_3 \notin T$. Sin embargo y continuando de esta manera podemos concluir dado que $\mu(e_n) = 0$ al momento en que $v_n \leftarrow \text{Head}(T)$ se cumple con que $v_1 \notin T$ y como $\mu(e_2) = 0$ entonces $v_2 \notin T$ y así sucesivamente con lo que $v_n \notin T$ y por la proposición 4.2 que $v_n \leftarrow \text{Head}(T)$!!



Lema 4.6: *El algoritmo MGGS es correcto.*

Demostración:

Sean $e_1 = (u, v)$ y $e_2 = (v, u)$ un par cualquiera de aristas antiparalelas de la gráfica. Es imposible que al finalizar *MGGS* se obtenga un marcaje M tal que $\mu(e_1) = \mu(e_2) = 0$. Ya que ello implicaría que M no es vivo lo que contradice al lema 4.4. Supongamos entonces que en M se cumple $\mu(e_1) = \mu(e_2) = 1$ pero como $\mu(e_1) = 1$ tenemos que cuando $u \leftarrow \text{HEAD}(T)$ al ejecutar el paso (6) $\text{visitado}[v] = \text{false}$ por lo tanto de acuerdo con las proposiciones 4.2 y 4.3 $u, v \in T$ y al terminar esa iteración del ciclo $u \notin T$ y $\text{visitado}[u] = \text{true}$ con lo que posteriormente al ser $v \leftarrow \text{HEAD}(T)$ y al ejecutar el paso (6) como $\mu(e_2) = 1$ entonces $u \in T$. Como el algoritmo en ningún otro lado salvo en el paso (1) se insertan vértices en T y salvo en el paso (2) se hace que $\text{visitado}[v] = \text{false}$ es imposible que $\mu(e_2) = 1$, la demostración es similar si v es seleccionado antes que u ; Por lo tanto dado que M es vivo de acuerdo al lema 4.4 y debido a que entre cualquier par de aristas antiparalelas una contiene una ficha y la otra no, por el corolario 4.2 M es un marcaje vivo-seguro. ■

Lema 4.7: El algoritmo *MGGS* es de $O(|E|)$.

Demostración:

En primer lugar es directo que los pasos (1) y (2) son de $O(|V|)$, por otra parte dado que (4), (7) y (8) son de $O(1)$ y debido a que el ciclo *while* se ejecuta $|V|$ veces entonces la complejidad para tales pasos es también de $O(|V|)$ y es claro que el paso (9) es de $O(1)$.

Como en los pasos (5) y (6) se revisa toda la lista de adyacencias de un vértice y debido a que tales pasos se ejecutan para todos y cada uno de los vértices entonces se revisan las listas de adyacencias de todos los vértices. Es decir, todas las aristas con que la complejidad es $O(|E|)$ y debido a que en una gráfica simétrica que es fuertemente conexa $2|V| - 2 \leq |E|$ la complejidad de *MGGS* es $O(|E|)$. ■

Evidentemente el algoritmo *LS* puede ser aplicado a cualquier gráfica marcada y obtener una marcaje vivo-seguro, sin embargo su complejidad $O(|V|^2|E|)$ es reducida a $O(|E|)$ con el algoritmo *MGGs*. Este último sólo es útil para una clase particular de gráficas marcadas, las llamadas en este trabajo gráficas marcadas simétricas. Para demostrar que *MGGs* no funciona para cualquier *MG* analicemos el siguiente ejemplo.

Ejemplo 4.3

Sea $MG = (G, M)$ tal que:

$$V = \{v_1, v_2, v_3, v_4\} \text{ y}$$

$$E = \{e_1 = (v_1, v_2), e_2 = (v_2, v_3), e_3 = (v_3, v_4), e_4 = (v_4, v_1)\}.$$

tal y como se muestra en la figura 4.4

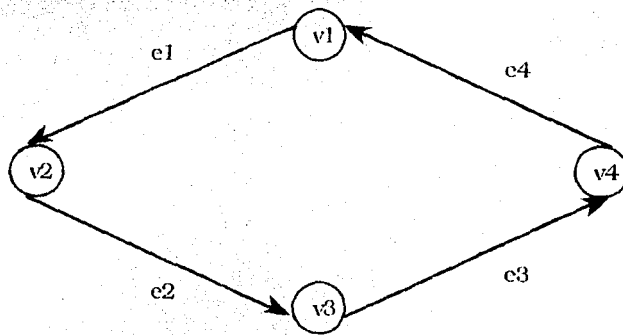
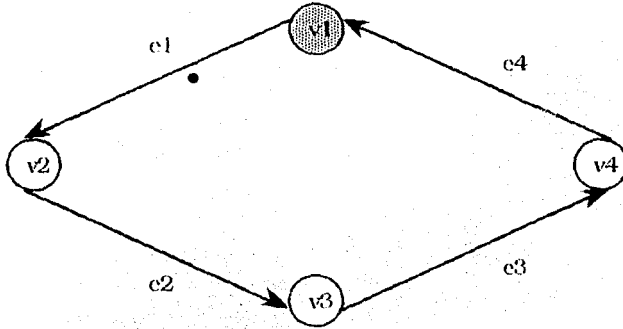


Figura 4.4

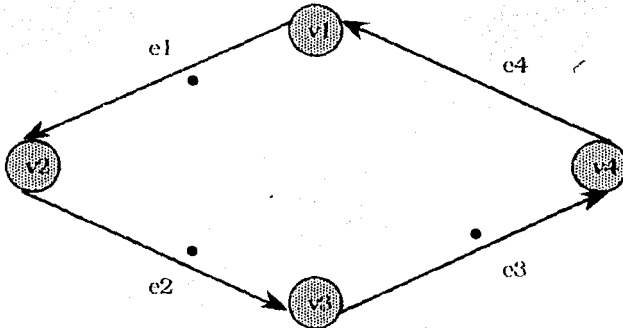
Ejecutemos *MGGs* sobre tal *MG*; así una vez concluidos los pasos (1) y (2) $T = \{v_1, v_2, v_3, v_4\}$ y $\forall v \in T$ se cumple que *visitado*[v] = *false*. Después de la primera iteración $T = \{v_2, v_3, v_4\}$, $\forall v \in T$ *visitado*[v] = *false*, y *visitado*[v_1] = *true* y también se cumple que $\mu(e_1) = 1$ tal como se ilustra en la figura 4.5



■ Vértices visitados

Figura 4.5

Continuando de esta manera hasta la última iteración se obtiene el marcaje $M = \langle 1,1,1,0 \rangle$ como se muestra en la figura 4.6 que es un marcaje vivo, pero no es vivo-seguro.



■ Vértices visitados

Figura 4.6

Sin embargo, y pese a tal restricción de *MGS*, la importancia teórica de contar con una menor complejidad para la obtención de marcajes vivos-seguros para las *MGS* tiene que ver con el modelado de *SD* en donde, siendo los canales de comunicación bidireccionales, se trata de implementar una calendarización.



Capítulo 5

Número de marcajes vivos-seguros para las MGS

Uno de los aspectos a considerar relacionados con la eficiencia de las calendarizaciones para los SD , se refiere en términos del modelo de las MGS al total de marcajes vivos-seguros que es posible asignar a la gráfica.

En el presente capítulo se hace un análisis del total de dichos marcajes para las MGS , estudiando en primer lugar ciertas subclases de gráficas y concluyendo con el resultado de R.P. Stanley [S73] y [S74] respecto al total de orientaciones acíclicas para una gráfica. Sin embargo la demostración del teorema de Stanley (lema 5.2) se hace aquí de manera inductiva, en vez de emplear la prueba combinatoria de Stanley.

La importancia de conocer el total de marcajes vivos-seguros que es posible asignar a una MGS radica en que dicho valor es el tamaño del espacio de búsqueda cuando se trata de hallar un marcaje M vivo-seguro.

5.1 EL MÁXIMO NÚMERO DE MARCAJES VIVOS-SEGUROS

Definición 5.1: El total de marcajes vivos-seguros de una MG se denota como $\alpha(MG)$ y se refiere al número de posibles maneras en las que se puede asignar un marcaje vivo-seguro a la MG , sin que ello implique alguna relación de alcance.

Ejemplo 5.1

Sea la MGS de la figura 5.1, si ejecutamos el algoritmo MGS con diferentes ordenamientos de la lista T podemos ver que $\alpha(MGS) = 6$, sin embargo esto

no quiere decir que los 6 marcajes vivos-seguros posibles son todos alcanzables desde cualquier otro.

El problema que se trata de resolver aquí consiste en hallar $\alpha(MGS)$ para cualquier MGS. Para facilitar el análisis en las subsecuentes secciones se emplea la siguiente.

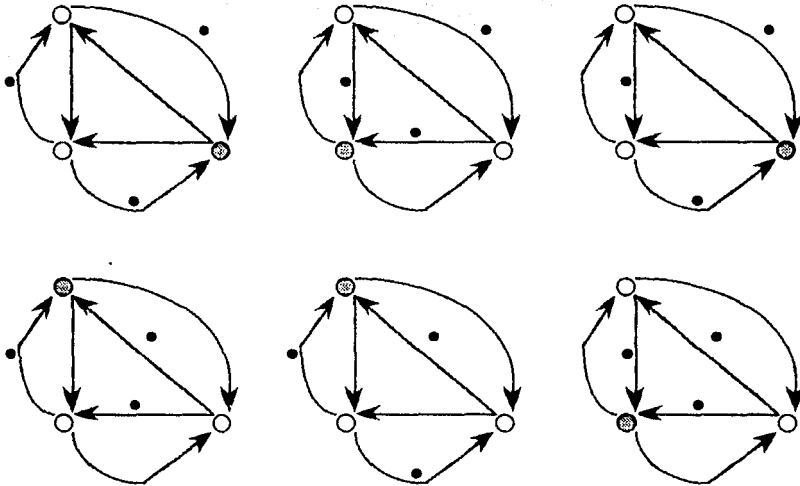


Figura 5.1

Definición 5.2: La gráfica subyacente simple G_S de una $MG = (G, M)$ es la gráfica subyacente G_S de G , pero sin aristas paralelas.

Otra definición que resulta ser de bastante utilidad en los subsecuentes desarrollos es.

Definición 5.3: La gráfica reducida de G con respecto a un subconjunto $V' = \{v_1, v_2, \dots, v_r\} \subset V$ de vértices y denotada como $G \setminus \{v_1, v_2, \dots, v_r\}$ es la gráfica obtenida a partir de G al eliminar el conjunto V' junto con todas las aristas tales que al menos uno de sus vértices finales está en V' .

El primer paso para encontrar $\alpha(MGS)$ de una MGS dada consiste en establecer el valor máximo para el total de marcajes vivos-seguros, así pues procedamos en esa dirección.

Definición 5.4: Sea $\gamma = (G, M)$ una MGS, la MGS reducida denotada como γ_ε respecto a un conjunto $\varepsilon = \{v_1, v_2, \dots, v_r\} \subset V$ de vértices es $\gamma_\varepsilon = (G \setminus \varepsilon, M_\varepsilon)$, donde $G \setminus \varepsilon$ es la gráfica reducida de G con respecto al conjunto ε , y M_ε es un marcaje tal que $\mu_\varepsilon(e_i) = \mu(e_j) \Leftrightarrow e_i = e_j = (u, v)$ donde $e_i \in E(\gamma_\varepsilon)$ y $e_j \in E(\gamma)$.

Proposición 5.1: Si $\gamma = (G, M)$ es una MGS donde M es vivo-seguro y donde el conjunto de vértices habilitados es ϑ , entonces γ_ϑ es una MGS con marcaje vivo-seguro M_ϑ .

Demostración:

Supongamos lo contrario, es decir que M_ϑ no es vivo-seguro, entonces hay un par de casos; o existe un circuito C en γ_ϑ tal que $\langle M_\vartheta / C \rangle = 0$ o existe al menos una arista $e \in E(\gamma_\vartheta)$ tal que no participa en algún circuito C con $\langle M_\vartheta / C \rangle = 1$.

i) Si existe un circuito C en γ_ϑ tal que $\langle M_\vartheta / C \rangle = 0$, es claro que ningún vértice de C está en ϑ . Así que en γ ocurre que el mismo circuito C es tal que $\langle M / C \rangle = 0$ y M no es un marcaje vivo.

ii) Sea $e = (u, v)$ una arista en γ_ϑ tal que \forall circuito en el que e participa se cumple con que $\langle M_\vartheta / C \rangle \geq 2$. Como $e \in E(\gamma_\vartheta)$ evidentemente $u, v \in V - \vartheta$ ya que de lo contrario $e \notin E(\gamma_\vartheta)$, y más aún su arista antiparalela $e' = (v, u) \in E(\gamma_\vartheta)$, de aquí que el circuito $C_0 = u \xrightarrow{e} v \xrightarrow{e'} u$ debe cumplir con que $\langle M_\vartheta / C_0 \rangle \geq 2$, y $\therefore \mu_\vartheta(e) + \mu_\vartheta(e') \geq 2$ lo que implica por la definición 5.4 que $\mu(e) + \mu(e') \geq 2$. Pero como M es vivo-seguro por el lema 4.3 $\mu(e) + \mu(e') = 1$ y que M_ϑ no es vivo-seguro !!

Proposición 5.2: Para cualquier MGS dada con marcaje vivo-seguro M , el algoritmo *MGS* puede producir M .

Demostración:

Consideremos una MGS obviamente finita $\gamma = (G, M)$ donde M es un marcaje vivo-seguro. Podemos proceder a emplear una técnica de reducción de la gráfica de la siguiente manera:

Sea el conjunto ϑ_1 el conjunto de vértices habilitados en M en donde evidentemente $\vartheta_1 \neq \emptyset$. Obtengamos γ_{ϑ_1} entonces por la proposición 5.1 el marcaje M_{ϑ_1} es vivo-seguro, así que debe existir un conjunto ϑ_2 de vértices habilitados en γ_{ϑ_1} . Obtengamos a partir de γ_{ϑ_1} la gráfica $\gamma_{(\vartheta_1)\vartheta_2}$ que por la proposición 5.2 cumple con que $M_{(\vartheta_1)\vartheta_2}$ es un marcaje vivo-seguro. Continuando de esta forma finalmente terminaremos con todos los vértices de la gráfica y tenemos que.

$$\vartheta_1 \cap \vartheta_2 \cap \dots \cap \vartheta_k = \emptyset$$

$$\vartheta_1 \cup \vartheta_2 \cup \dots \cup \vartheta_k = V$$

Del mismo proceso de reducción y respecto a las aristas tenemos que:

$$\forall e = (u, v) / u \in \vartheta_i \text{ y } v \in \vartheta_j$$

si $i > j$ entonces $\mu(e) = 0$

sino

si $i < j$ entonces $\mu(e) = 1$

sino $e \notin E$

Sea que se ejecuta el algoritmo *MGS* sobre γ y se obtiene un marcaje vivo-seguro M' , insertando para ello en el paso (1) del algoritmo los vértices en T de tal forma que los del conjunto ϑ_k estén antes que los vértices del conjunto ϑ_{k-1} , y éstos antes que los del conjunto ϑ_{k-2} y así sucesivamente hasta que finalmente estén los del conjunto ϑ_1 . Sin importar el arreglo relativo de los vértices respecto a los demás vértices que se encuentren en su mismo conjunto $\vartheta_i \forall 1 \leq i \leq k$. Procedamos a

probar por inducción sobre el orden en que dichos conjuntos son eliminados de la lista T .

Paso base: Al finalizar las primeras $|\vartheta_k|$ iteraciones del ciclo while del algoritmo y considerando que $\forall u, v \in \vartheta_k$ no existen aristas entre ellos. Podemos ver que dado que en el paso (2) se hace $\forall v \in V$ *visitado*[v] \leftarrow *false* que en cada una de estas iteraciones en los pasos (5) y (6) siempre se pone una ficha en las aristas de la lista de adyacencias del vértice en cuestión. Al terminar se cumple que $\forall e = (u, v) / u \in \vartheta_k$ $\mu'(e) = \mu(e) = 1$ y dado que en el paso (8) se modifica el valor del predicado *visitado* también es cierto que al finalizar estas iteraciones $\forall u \notin \vartheta_k$ *visitado*[u] = *true*, $\forall v \in \vartheta_k$. Así mismo como en ningún lugar del ciclo se cambia el valor de *visitado* a *false*, entonces en cualquier iteración posterior del while, al ejecutar el paso (6) se cumplirá que $\forall e' = (v, u) / u \in \vartheta_k$ $\mu'(e') = \mu(e') = 0$.

Paso de inducción:

HIPOTESIS: \forall par de conjuntos ϑ_i y ϑ_j que son eliminados de la lista T antes que ϑ_1 se cumple con que: $\forall e = (u, v) / u \in \vartheta_i$ y $v \in \vartheta_j$, $\mu'(e) = \mu(e)$.

Sea el momento en que $T = \vartheta_1$ entonces a las únicas aristas que falta asignar marcaje son aquellas de salida o de entrada a ϑ_1 . Como $\forall v \in \vartheta_1$ en ese momento *visitado*[v] = *false* y $\forall u \notin \vartheta_1$ *visitado*[u] = *true*. Como no existen aristas entre los vértices de ϑ_1 , entonces en cada iteración subsecuente al ejecutar el paso (6) tendremos que $\forall e = (v, u)$ $\mu'(e) = \mu(e) = 0$. Además como las aristas de entrada a ϑ_1 vienen de algún vértice que pertenece a un conjunto que fue eliminado antes entonces $\forall e' = (u, v)$ de entrada a ϑ_1 tenemos que $\mu'(e') = \mu(e') = 1$, y como por hipótesis todas las demás aristas cumplen con que su marcaje es igual al original, se concluye que $M = M'$. ■

Corolario 5.1: \forall MGS con $\eta = |V|$ se cumple que $\alpha(MGS) \leq \eta!$.

Demostración:

De la proposición 5.2 sabemos que el algoritmo *MGS* puede producir cualquier marcaje vivo-seguro para una *MGS* dada. Esto depende del orden en que los vértices son insertados en la lista temporal *T*. Si $\eta = |V|$ es claro que $\exists \eta!$ posibles permutaciones de los vértices, con lo que $\exists \eta!$ maneras de insertar los vértices en *T* y es directo que $\alpha(MGS)$ es a lo más $\eta!$.



5.2 TOTAL DE MARCAJES VIVOS-SEGUROS PARA ALGUNAS FAMILIAS DE MGS

Definición 5.5: Una *MGS* se dice que es completa cuando su *G_s* asociada es completa.

Proposición 5.3: $\forall MGS$ completa con $\eta = |V|$ se cumple que $\alpha(MGS) = \eta!$

Demostración:

Por inducción en el número de vértices de la gráfica.

Paso base: Si $\eta = 3$ es fácil ver que $\alpha(MGS) = 3! = 6$ se cumple.

Paso de inducción:

HIPOTESIS: $\forall MGS$ completa con i vértices, donde $3 \leq i \leq \eta - 1$ es verdadero que $\alpha(MGS) = i!$.

Sea una *MGS* completa con η vértices. Por el lema 4.1 sabemos que si un vértice cualquiera está habilitado en un marcaje vivo-seguro, entonces ningún vértice adyacente puede estar habilitado en el mismo marcaje. Sin pérdida de generalidad sea que v_1 es el vértice habilitado, y es fácil ver que $MGS_{\{v_1\}}$ es una *MGS* completa con $\eta - 1$ vértices con lo que por hipótesis $\exists (\eta - 1)!$ marcajes vivos-seguros en los que v_1 está

habilitado. Continuando de esta manera se puede demostrar que $\forall v \in V$
 $\exists (\eta - 1)!$ marcajes vivos-seguros en los que v está habilitado. Como
 cuando un vértice está habilitado ningún otro lo está y teniendo $\eta = |V|$
 se sigue que $\alpha(MGS) = \eta(\eta - 1)! = \eta!$.

Definición 5.6: Una MGS se dice que es un ciclo cuando su G_{Σ} asociada
 es un ciclo.

Proposición 5.4: $\forall MGS$ que es un ciclo con $\eta = |V|$ vértices se cumple que
 $\alpha(MGS) = 2^{\eta} - 2$.

Demostración:

Del lema 4.3 tenemos que entre cualquier par de aristas antiparalelas
 una debe contener una ficha y la otra no, en cualquier marcaje que sea
 vivo-seguro. Por el lema 2.1 para que un marcaje M sea vivo se requiere
 que en cualquier circuito C_i de la MGS, $\langle M/C_i \rangle > 0$.

Sea una MGS que es un ciclo con η vértices, entonces es evidente que
 existen 2^{η} formas en que se puede asignar a cada par de aristas
 antiparalelas una ficha a una y ninguna a la otra. Por otra parte sea que el
 conjunto de aristas es tal que $\forall 1 \leq i \leq \eta - 1$ $e_i = (v_i, v_{i+1})$ y $e'_i = (v_{i+1}, v_i)$, y
 además $e_{\eta} = (v_{\eta-1}, v_{\eta})$ y $e'_{\eta} = (v_{\eta}, v_{\eta-1})$.

Como $\forall 1 \leq i \leq \eta$ se cumple que entre e_i y e'_i una contiene ficha, en tanto
 la otra no; entonces los únicos circuitos que pueden hacer que el
 marcaje no sea vivo (por su conteo de fichas) son $C = e_1, e_2, \dots, e_{\eta}$ y por
 supuesto $C' = e'_1, e'_2, \dots, e'_{\eta}$. Si $\langle M/C \rangle = 0$ entonces $\forall e \in C'$ tenemos que
 $\mu(e) = 1$ y del mismo modo si $\langle M/C' \rangle = 0$ entonces $\forall e \in C$ $\mu(e) = 1$. Lo que
 implica que de las 2^{η} maneras de asignar fichas entre las arista paralelas
 en un par de ellas no se obtiene un marcaje vivo-seguro (de hecho ni
 siquiera se obtienen marcajes vivos) y se concluye que $\alpha(MGS) = 2^{\eta} - 2$.

Definición 5.7: Una MGS se dice que es un árbol si su G_{Σ} asociada es un árbol.

Proposición 5.5: \forall MGS que es un árbol con η vértices se tiene que $\alpha(MGS) = 2^{\eta-1}$.

Demostración:

Sea una MGS que es un árbol. Para tener un marcaje M vivo-seguro se requiere de acuerdo con el lema 2.1 que para todo circuito C_i de la MGS se cumpla con que $\langle M / C_i \rangle > 0$ y de acuerdo con lema 4.3 tenemos que entre cualquier par de aristas antiparalelas una contenga ficha y la otra no. Supongamos que se asigna un marcaje M_0 a la MGS que satisfaga la condición del lema 4.3 tal que no sea vivo-seguro, entonces debe existir un circuito C en la gráfica tal que $\langle M_0 / C \rangle = 0$.

Sea v un vértice en el circuito, y sea $e_1 = (v, u)$ la arista por la que se sale de v en C , evidentemente $\mu_0(e_1) = 0$ y además el circuito no debe pasar por la arista antiparalela $e'_1 = (u, v)$ de e_1 , ya que $\mu_0(e'_1) = 1$, entonces la arista del circuito por la que se llega a v debe ser de la forma $e_j = (w, v) / w \neq u$. Pero eso implica que \exists al menos un circuito en G_{Σ} con lo que MGS es un árbol \blacksquare , y es claro que cualquier marcaje M que cumpla con la condición del lema 4.3 es un marcaje vivo-seguro.

Por otro lado dado que $|E| = 2(\eta - 1)$ tenemos que:

$$\alpha(MGS) = 2^{\frac{|E|}{2}} = 2^{(\eta-1)}$$

Definición 5.8: Una MGS es un nudo si la G_{Σ} consta de un par de ciclos simples que comparten uno y sólo un vértice.

Proposición 5.6: \forall MGS que es un nudo se cumple con que $\alpha(MGS) = (2^{k_1} - 2)(2^{k_2} - 2)$ donde k_1 y k_2 son el número de vértices de cada uno de los ciclos simples de la G_{Σ} asociada con la MGS.

Demostración:

Sea γ una MGS que es un nudo, entonces existe un vértice $v \in V$ tal que $\gamma_{\{v\}}$ es desconexa y tiene dos componentes fuertemente conexas: γ_1 y γ_2 .

Si consideramos la MGS formada por γ_1 junto con el vértice v y sus aristas de entrada y salida, tenemos una MGS γ_1^v que por la definición 5.6 es un ciclo y de la proposición 5.4 $\alpha(\gamma_1^v) = (2^{k_1} - 2)$. Del mismo modo tenemos que $\alpha(\gamma_2^v) = (2^{k_2} - 2)$, es decir que hay $(2^{k_1} - 2)(2^{k_2} - 2)$ maneras de asignar marcajes a γ en los que γ_1^v y γ_2^v tienen un marcaje vivo-seguro. Ahora supongamos que una de tales asignaciones M no es un marcaje vivo-seguro para γ pero dado que para todo circuito simple C se tiene que $\langle M/C \rangle > 0$ entonces para todo circuito C_w se cumple que $\langle M/C_w \rangle > 0$ y como entre cada par de aristas antiparalelas una contiene una marca y la otra no es directo que: $\alpha(\gamma) = (2^{k_1} - 2)(2^{k_2} - 2)$. ■

Definición 5.9: Una MGS es una división- i si su G_Σ asociada consta de un par de ciclos simples que comparten i aristas con $i \geq 1$.

Proposición 5.7: \forall MGS que es una división-1 entonces $\alpha(MGS) = \frac{(2^{k_1} - 2)(2^{k_2} - 2)}{2}$ donde k_1 y k_2 son el número de vértices de cada uno de los ciclos simples de la G_Σ asociada.

Demostración:

Sea $e = (u, v)$ la arista en la G_Σ común a ambos ciclos, entonces en la MGS existe un par de aristas dirigidas $e_1 = (u, v)$ y $e_2 = (v, u)$ con lo que en cualquier marcaje vivo-seguro alguna de las dos contiene una ficha y la otra no.

Si $\mu(e_1) = 1$ y $\mu(e_2) = 0$ entonces quedan $k_1 - 1$ pares de aristas antiparalelas para uno de los ciclos y $k_2 - 1$ pares de aristas paralelas para el otro de los ciclos de la gráfica. También tenemos que hay $2^{k_1 - 1} - 1$

formas de poner una ficha en una arista y no poner fichas en su antiparalela en cada uno de los pares del primer ciclo sin que se forme un circuito sin fichas. Teniendo así, del mismo modo para el otro ciclo existen $2^{k_2-1}-1$ maneras de asignar un marcaje sin formar un circuito sin fichas, y de este modo no existe forma de que las aristas del primer y segundo ciclo formen un circuito dirigido. Así, que en total tenemos $(2^{k_1-1}-1)(2^{k_2-1}-1)$ marcajes vivos-seguros para la gráfica con $\mu(e_1)=1$ y $\mu(e_2)=0$. De manera similar si $\mu(e_1)=0$ y $\mu(e_2)=1$ entonces $\exists (2^{k_1-1}-1)(2^{k_2-1}-1)$ marcajes vivos-seguros y es directo que:

$$\begin{aligned} \alpha(MGS) &= 2(2^{k_1-1}-1)(2^{k_2-1}-1) \\ &= (2^{k_1}-2)(2^{k_2-1}-1)\frac{2}{2} \\ &= \frac{(2^{k_1}-2)(2^{k_2}-2)}{2} \end{aligned}$$

■

Proposición 5.8: $\forall MGS$ que es una división- b donde k_1 y k_2 son respectivamente el número de vértices de cada uno de los ciclos simples de la G_{Σ} asociada con la gráfica marcada, se cumple que :

$$\alpha(MGS) = \frac{(2^{k_1-b+1}-2)(2^{k_2-b+1}-2)}{2} + (2^b-2)(2^{k_1+k_2-2b}-2)$$

Demostración:

Lo primero que hay que observar es que los b pares de aristas de la MGS que corresponden a las b aristas comunes de la G_{Σ} pueden tener 2^b marcajes que cumplan con la condición del lema 4.3, y en 2 de ellos se forma un camino sin fichas. Si los vértices $u, v \in V$ son los vértices finales del camino formado por las b aristas y consideramos los casos en que se forman los caminos sin fichas de u a v y de v a u , procediendo de manera similar a la proposición 5.7 es fácil ver que hay $\frac{(2^{k_1-b+1}-2)(2^{k_2-b+1}-2)}{2}$ de tales marcajes.

2

Si consideramos por otra parte los casos en que no se forma camino alguno entre u y v (en cualquier dirección) sin fichas, entonces de la proposición 5.4 es directo que hay $(2^b - 2)(2^{k_1+k_2-2b} - 2)$ de tales marcajes vivos-seguros, con lo que:

$$\alpha(MGS) = \frac{(2^{k_1-b+1} - 2)(2^{k_2-b+1} - 2)}{2} + (2^b - 2)(2^{k_1+k_2-2b} - 2)$$

Evidentemente se hace necesario contar con un argumento más general respecto al total de marcajes vivos-seguro para las gráficas marcadas en cuestión. Como se estudia posteriormente dicho problema tiene una relación directa con el coloreo de vértices y más particularmente con el polinomio cromático de una gráfica.

5.3 COLOREO DE VÉRTICES

El coloreo de vértices de una gráfica G , es el proceso de asignación de colores c_1, c_2, c_3, \dots a los vértices $V(G)$ de tal forma que ningún par de vértices adyacentes tengan el mismo color. Si una gráfica puede ser coloreada con k colores distintos se dice que la gráfica es k -coloreable y el número cromático denotado como $\chi^{(G)}$ es el mínimo $i \in \mathbb{Z}^+$ tal que G es i -coloreable.

Ejemplo 5.2

Si consideramos la gráfica de la figura 5.2 podemos asignar el color c_1 al vértice v_1 , c_2 a los vértices v_2, v_4, v_6 y c_3 a v_3, v_5, v_7 . Del mismo modo en la gráfica de la figura 5.3 es posible asignar c_1 a los vértices v_1, v_3, v_5, v_7 y c_2 a v_2, v_4, v_6, v_8 , con lo que podemos concluir que la primera gráfica es 3-coloreable en tanto la segunda es 2-coloreable.

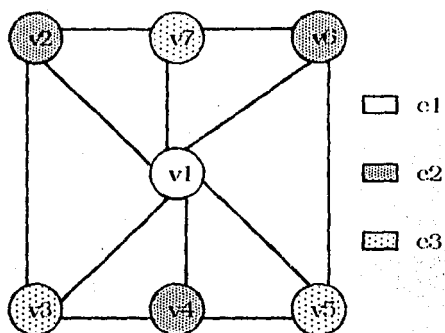


Figura 5.2

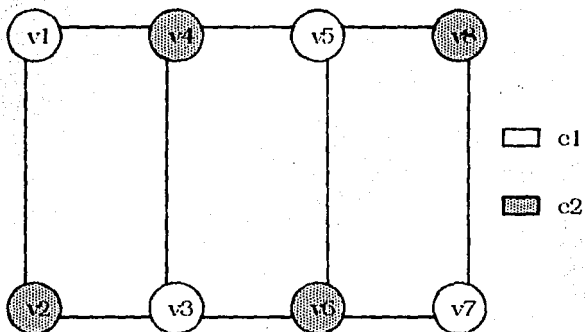


Figura 5.3

Proposición 5.9: Para cualquier gráfica completa K_η se cumple con que $\chi^{(K_\eta)} = d(v) + 1 = \eta \quad \forall v \in V(K_\eta)$.

Demostración:

Por inducción sobre $\eta = |V(G)|$.

Paso base: Cuando $\eta = 1$ es claro que K_1 únicamente puede ser coloreada con un color y dado que $d(v) = 0$ tenemos que $\chi^{(K_1)} = d(v) + 1 = 1$.

Paso de inducción:

HIPÓTESIS: Sea que $\forall j / 1 \leq j \leq \eta - 1$, K_j cumple con que $\chi^{(K_j)} = d(v) + 1 = j$, $\forall v \in V(K_j)$.

Si consideramos la gráfica completa $K_{\eta-1}$ tenemos que $\forall v \in V(K_{\eta-1})$ $d(v) = \eta - 2$ y por hipótesis $\chi^{(K_{\eta-1})} = \eta - 1$. Si adicionamos un nuevo vértice u con una arista a cada uno de los vértices de $K_{\eta-1}$ es directo que el color de u debe ser distinto de los $\eta - 1$ colores de $K_{\eta-1}$. Además, ahora todos los vértices cumplen con que su grado es $\eta - 1$. Como dicha construcción nos lleva a obtener K_η se concluye que $\chi^{(K_\eta)} = d(v) + 1 = \eta$ $\forall v \in V(K_\eta)$. ■

Proposición 5.10: Para cualquier gráfica G que es un ciclo simple con $\eta \geq 3$ vértices, G es 2-coloreable si el número de vértices es par y G es 3-coloreable si el número de vértices es impar.

Demostración:

Sea G un ciclo simple con $\eta \geq 3$ vértices. Sin pérdida de generalidad sean que los vértices están etiquetados v_1, v_2, \dots, v_η en ese orden siguiendo el ciclo en alguna dirección.

Podemos asignar entonces un color c_ξ a cada v_i con $\xi \leftarrow (i \bmod 2) + 1$ y es directo que después de ejecutar dicha asignación se cumple con que el color para todo vértice par es c_1 y para todo vértice impar es c_2 . Con lo que v_1 está coloreado con c_2 , v_2 con c_1 y así sucesivamente y es fácil ver que la siguiente condición se cumple:

- (a) $\forall v_j / 1 < j < \eta$ el color de v_j es distinto que el de v_{j-1} y el de v_{j+1} .

Y hay un par de casos.

i) Cuando η es par, entonces el color de v_{η} es c_1 y el color de v_1 es c_2 y por (a) tenemos que G es 2-coloreable.

ii) Si η es impar entonces el color de v_{η} es igual que el de v_1 (que evidentemente es c_2) y de (a) es claro que el color de $v_{\eta-1}$ es igual que el de v_2 con lo que no es posible cambiar el color de v_1 o v_{η} a c_1 sin que algún par de vértices adyacentes sean del mismo color. Se sigue que se requiere un nuevo color c_3 con el cual colorear v_1 o v_{η} . Sin pérdida de generalidad sea que se asigna c_3 a v_{η} , entonces tenemos que para v_1 su color es distinto que el de sus vértices vecinos y ya que el color de v_{η} es distinto que el de cualquier otro vértice y como (a) se sigue cumpliendo concluimos que G es 3-coloreable. ■

Corolario 5.2: Para cualquier gráfica G que es un ciclo simple, con $\eta \geq 3$ vértices se tiene que $\chi^{(G)} = (\eta \bmod 2) + 2$.

Demostración:

Dada una G que es un ciclo simple con al menos $\eta = 3$ vértices que $\chi^{(G)} > 1$, y de la proposición 5.9 sabemos que si η es par entonces G es 2-coloreable con lo que $\chi^{(G)} = (\eta \bmod 2) + 2 = 0 + 2 = 2$ se cumple, de la misma proposición tenemos que si η es impar G es 3-coloreable con que en tal caso $\chi^{(G)} = (\eta \bmod 2) + 2 = 1 + 2 = 3$. ■

Para determinar cuando una gráfica simple es k -coloreable, podemos desde luego intentar alguna técnica de búsqueda exhaustiva (que claramente es ineficiente), pero también podemos cuestionar si es que existe alguna técnica eficiente (de complejidad polinomial) que determine si dada una gráfica simple ésta es k -coloreable.

Ejemplo 5.3

Si contamos con la gráfica de la figura 5.4 y tenemos k colores, y procedemos a colorear los vértices en orden creciente respecto a su subíndice entonces el vértice v_1 puede ser coloreado con cualquiera de estos k colores, pero v_2 únicamente puede ser coloreado con alguno de los $(k-1)$ colores que no se emplearon, del mismo modo v_3 puede ser coloreado con alguno de los $(k-2)$ colores disponibles, continuando de esta manera tenemos $(k-1)$ y $(k-2)$ colores para v_4 y v_5 respectivamente.

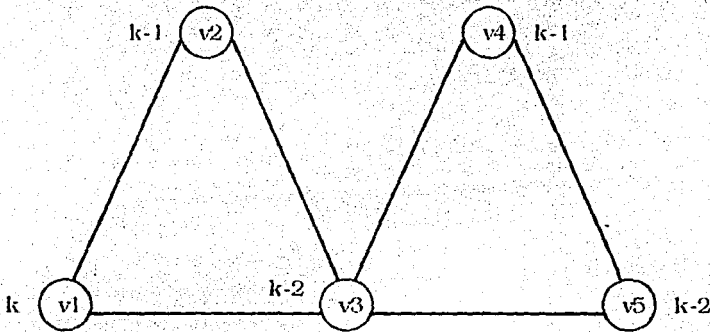


Figura 5.4

Por la regla del producto (Teoría combinatoria) tenemos $k(k-1)^2(k-2)^2$ y es inmediato que el mínimo valor de k es 3 con que $\chi^{(G)} = 3$.

El ejemplo 5.3 sugiere una técnica para gráficas de tamaño razonable pero para cualquier $k \geq 3$ el problema de determinar cuando una gráfica simple arbitraria es k -coloreable es polinomialmente equivalente a 3SAT [GJ79]. \therefore no se conoce hasta la fecha una solución eficiente, y más aún dada una gráfica simple G y un valor de k , el problema de determinar si G es k -coloreable es polinomialmente equivalente al problema del ciclo Hamiltoniano [AHU74].

Sin embargo, el problema de determinar si una gráfica simple es 2-coloreable y colorearla ya ha sido resuelto con algoritmos eficientes. Intuitivamente hablando, lo primero que debe observarse es que cualquier gráfica 2-coloreable es bipartita y esto es debido a que el conjunto V puede ser particionado en dos subconjuntos disjuntos de vértices, a decir V_1 y V_2 . De tal forma que cualquier arista de la gráfica va de uno de estos conjuntos al otro, con lo que es posible colorear los vértices de V_1 de un color y los vértices de V_2 de otro color. De hecho también es cierto que si una gráfica es bipartita entonces es 2-coloreable y resulta muy sencillo caracterizar la familia de gráficas simples que son 2-coloreables.

Proposición 5.11: *Dada una gráfica simple G , la gráfica es 2-colorable si y sólo si no contiene ningún ciclo con un número impar de aristas.*

Demostración:

Debe ser claro que si G es 2-coloreable entonces G es bipartita con lo que cualquier ciclo en la gráfica contiene un número par de aristas.

Recíprocamente, sea que todos los ciclos de la gráfica contienen un número par de aristas, podemos entonces seleccionar un vértice $v \in V$ cualquiera de la gráfica y proceder a definir los subconjuntos de vértices $V_1 = \{z \mid \delta(v, z) \text{ es par}\}$ y $V_2 = V - V_1$ donde evidentemente los subconjuntos son disjuntos y $V_1 \cup V_2 = V$. Si coloreamos los vértices de uno de estos subconjuntos de un color y los del otro subconjunto de un color distinto tenemos un colorco para la gráfica. Para demostrarlo supongamos que $u, w \in V_1$ y que la arista $e = (u, w) \in E$, de este modo tenemos un ciclo C de v a v . Dicho ciclo va de v a u por la ruta más corta, continúa por la arista $e = (u, w)$ y finalmente se recorre el camino más corto de w a v . Si E_1 es el conjunto de aristas que están en los caminos más cortos de v a u y de w a v es directo que la contribución de las aristas de E_1 a la longitud del ciclo C es un número par y se sigue que entonces debe existir al menos un ciclo con un número impar de aristas, lo que contradice la hipótesis.

Lo que prosigue es obtener una cota máxima para el valor de $\chi^{(G)}$ y es en este sentido que se desarrollan las demostraciones subsecuentes.

Proposición 5.12: Para cualquier gráfica G con $\eta \geq 3$ vértices tal que G es un ciclo simple o una gráfica completa, se cumple con que $\forall v \in V, G \setminus \{v\}$ es conexa y $\forall u, v \in V / e = (u, v) \notin E, G \setminus \{u, v\}$ es desconexa.

Demostración:

i) Sea G una K_η con $\eta \geq 3$, entonces no existen $u, v \in V / e = (u, v) \notin E$. Por otra parte $\forall v \in V, K_{\eta-1} = G \setminus \{v\}$ y es directo que es conexa.

ii) Si G es un ciclo simple con $\eta \geq 3$ tenemos que si $\eta = 3$ el caso es idéntico que para K_3 . De otra forma, es decir que si $\eta > 3$ dado que la gráfica es un ciclo simple entonces $\forall u, v \in V \exists$ únicamente un par de caminos simples P y P' entre ellos, y si $V(P)$ y $V(P')$ son los vértices de tales caminos, entonces $V(P) \cap V(P') = \{u, v\}$ así es que dado algún vértice $w \in V / w \neq u$ y $w \neq v$ es fácil ver que $w \in V(P)$ o $w \in V(P')$ pero no a ambos, con lo que en $G \setminus \{w\}$ es conexa. Si consideramos cualquier par de vértices $u, v \in V / e = (u, v) \notin E$ del razonamiento anterior tenemos que hay en la gráfica un par de caminos P y P' entre ellos, y dado que tales vértices no son adyacentes es directo que $|V(P) - \{u, v\}| > 0$ y $|V(P') - \{u, v\}| > 0$ con lo que deben existir al menos un par de vértices distintos $w \in V(P) / w \neq u$ y $w \neq v, w' \in V(P') / w' \neq u$ y $w' \neq v$, así es que en $G \setminus \{u, v\}$ no existe al menos un camino entre w y w' , y se concluye que $G \setminus \{u, v\}$ es desconexa. ■

Proposición 5.13: Dada una gráfica simple G con $\eta \geq 3$ en la que $\forall v \in V G \setminus \{v\}$ es conexa y $\forall u, v \in V / e = (u, v) \notin E G \setminus \{u, v\}$ desconexa, entonces G es un ciclo simple o es una gráfica completa.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Demostración:

Sea que G no es una gráfica completa, en la que $G \setminus \{u\}$ siempre es conexa pero dados 3 vértices distintos $u, v, w \in V / (w, u), (w, v) \in E$ y $(u, v) \notin E$ entonces $G \setminus \{u, v\}$ es disconexa, y sean V_1 y V_2 los subconjuntos de vértices disconexos el uno del otro en $G \setminus \{u, v\}$ como se esquematiza en la figura 5.5.

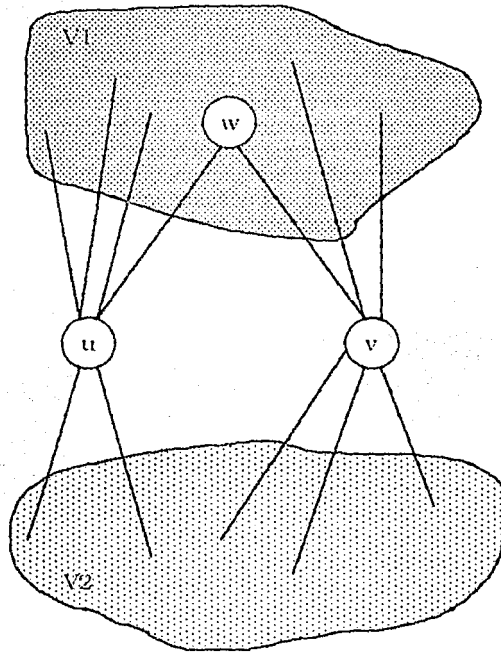


Figura 5.5

Supongamos que $w \in V_1$ y que $V_1 \neq \{w\}$. Si $x, y \in V / (u, x), (u, y) \in E$ y $(x, y) \notin E$ por las condiciones iniciales dadas $G \setminus \{x, y\}$ es disconexa, y el vértice u está en alguna de sus componentes. Si $z \in V$ es un vértice en otra componente, entonces dado que $G \setminus \{x\}$ y $G \setminus \{y\}$ son conexas, pero $G \setminus \{x, y\}$ no lo es, tenemos que todos los caminos de u a z pasan por x o por y y necesariamente, pero no por ambos.

Si $z \in V_1$ como en la figura 5.6, entonces un camino de u a z que no pasa por x , puede ir de u a w , después de w a v y finalmente de v a z , con lo que tampoco pasaría por y , y es claro que $z \notin V_1$. Procediendo de manera similar podemos demostrar que $z \notin V_2$, concluyendo que $V_1 = \{w\}$ y que $d(w) = 2$ y más aún $d(u) = d(v) = 2$ y es cierto que G es un ciclo simple.

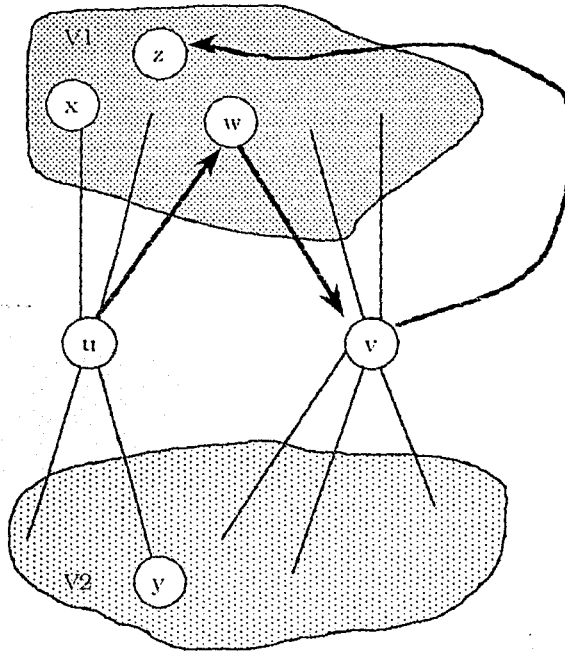


Figura 5.6

Sea ahora que G no es un ciclo simple, y que $G \setminus \{u\}$ es conexa pero dados $u, v, w \in V / (w, u), (w, v) \in E$ y $(u, v) \notin E$, $G \setminus \{u, v\}$ es desconexa.

Del razonamiento anterior tenemos que si $G \setminus \{u, v\}$ es desconexa entonces G es un ciclo simple a menos que $\forall u, v \in V$ se cumple con que $e = (u, v) \in E$ en cuyo caso G es una gráfica completa.

■

Lema 5.1: Sea $G = (V, E)$ una gráfica conexa con $m = \max\{d(v) \mid v \in V\}$ entonces, si G es una gráfica completa o si es un ciclo simple con un número impar de aristas: $\chi^{(G)} = m + 1$, y en todos los demás casos $\chi^{(G)} \leq m$.

Demstración

Si la gráfica es una completa o un ciclo simple con un número impar de aristas el resultado es directo de las proposiciones 5.9 y 5.10.

Supongamos que G es una gráfica conexa, que no es una gráfica completa, ni un ciclo simple con un número impar de aristas. Por inducción sobre $|V|$.

Paso base: Cuando $|V| = 3$ es fácil ver que $m = 2$ y $\chi^{(G)} = 2$.

Paso de inducción:

HIPOTESIS: $\forall 3 \leq j \leq |V| - 1$ el lema se cumple.

Si G tiene $\eta = |V|$ vértices y si existe un $v \in V$ tal que $G \setminus \{v\}$ es desconexa podemos separar la gráfica en sus diferentes componetes conexas G_i de manera obvia como se muestra en la figura 5.7.

Por hipótesis cada G_i puede ser coloreada con m colores de tal forma que el vértice v sea del mismo color, así al unir dichas componentes se obtiene un coloreo de G en los requeridos m colores.

Por otra parte si $G \setminus \{v\}$ es conexa dado que la gráfica no es completa, ni un ciclo simple con un número impar de aristas, por la proposición 5.12

tenemos que deben $\exists u, v, w \in V / (w, v), (w, u) \in E$ y $(u, v) \notin E$ con $G \setminus \{u, v\}$ conexa.

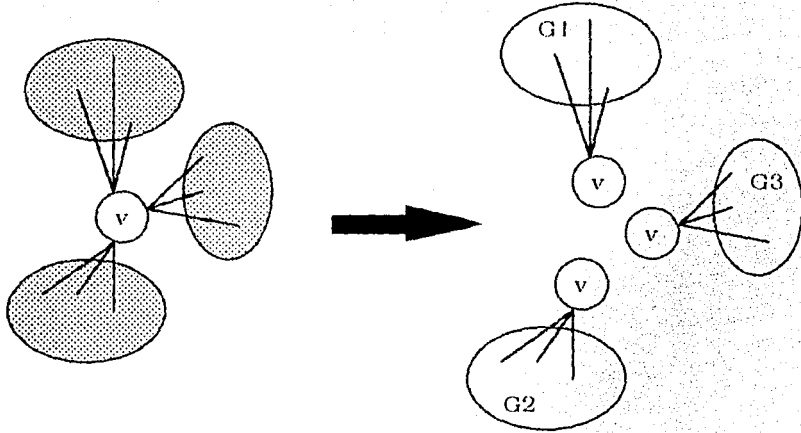


Figura 5.7

Así es pues que $G' = G \setminus \{u, v\}$ es conexa, y podemos etiquetar los vértices de G como $v_1 = u, v_2 = v$ y $v_3, v_4, \dots, v_\eta = w$ en orden decreciente respecto a la distancia mínima a partir de w en G' , con respecto a la evidente propiedad de que $\forall v_i / 1 \leq i < \eta, v_i$ es adyacente a un vértice con etiqueta mayor. Así para v_1 y v_2 ambos son adyacentes a v_η , y $\forall i > 2$ es posible considerar el camino más corto de v_i a w como en la figura 5.8, y dicho camino debe ser de la forma $P = v_i, v_j, \dots, w$ en donde $e = (v_i, v_j) \in E$ con lo que $\delta(w, v_i) = \delta(w, v_j) + 1$ y por lo tanto $j > i$.

Sean los m colores c_1, c_2, \dots, c_m y asignemos a dichos colores a los vértices de G en el orden v_1, v_2, \dots, v_η de tal forma que en cada ocasión el color usado para algún vértice v_i es el c_j con la menor j que no se ha empleado en ningún vértice adyacente a v_i .

Entonces v_1 y v_2 estarán coloreados con c_1 , y $\forall v_i / 3 \leq i < \eta$ al momento de ser coloreado v_i éste es adyacente a al menos un vértice v_j con $j > i$ y que por lo tanto no está coloreado. Debido a que el máximo grado de un

vértice es m es directo que v_i es a lo más adyacente a $m-1$ vértices con un color asignado, y por lo tanto siempre existe al menos un color disponible que asignarle.

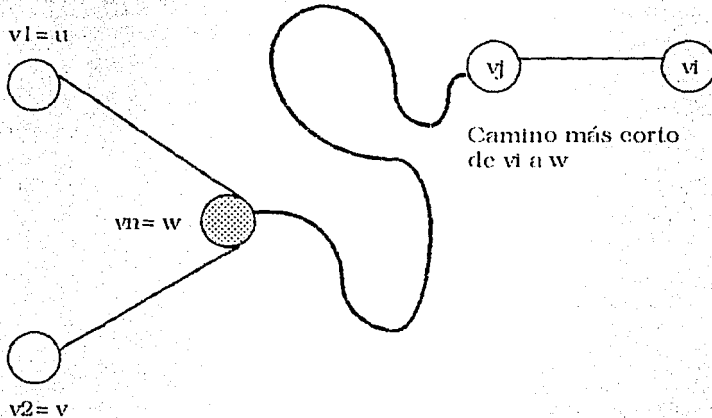


Figura 5.8

Finalmente como v_j es adyacente a $v_1 = u$ y $v_2 = v$ ambos con color c_1 , entonces al menos hay un color disponible para v_j y G entonces está coloreada con los m colores requeridos.

[B93] ■

5.4 POLINOMIOS CROMÁTICOS, ORIENTACIONES ACÍCLICAS Y MARCAJES VIVOS-SEGUROS

En el capítulo 2 (en particular en el lema 2.1) se prueba que cualquier marcaje es vivo si y sólo si para todo circuito dirigido C_i de una gráfica dada, se cumple con que $\langle M/C_i \rangle > 0$. En ese mismo capítulo (lema 2.2) se demuestra que un marcaje que cumple con la condición anterior en el que $\forall e \in E$ hay al menos un circuito C en el que la arista participa con $\langle M/C \rangle = 1$ además de ser vivo es seguro (vivo-seguro). Corresponde ahora analizar la relación que hay entre los marcajes vivos-seguros para una MGS y las orientaciones acíclicas para su G_2 asociada.

Definición 5.11: La gráfica orientada \vec{G} de una MGS (G, M) , es una digráfica en donde $V(\vec{G}) = V(G)$ y $E(\vec{G}) = \{e \in E(G) \mid \mu(e) > 0\}$.

Definición 5.12: Una orientación de una gráfica es la asignación de direcciones a cada una de sus aristas $e = (u, v)$ denotada como $e = (u, v)$ o como $e = (v, u)$ según sea el caso. Una orientación de una gráfica se dice que es acíclica si no contiene circuitos dirigidos.

Proposición 5.13: Si $\gamma = (G, M)$ es una MGS en donde M es vivo-seguro y si $\vec{\gamma}$ es la digráfica orientada a partir de dicho marcaje, entonces $\vec{\gamma}$ es una orientación acíclica de la gráfica subyacente simple G_{Σ} asociada con la MGS.

Demostración:

Supongamos lo contrario, ésto es que $\vec{\gamma}$ es cíclica, entonces existe al menos un circuito dirigido en $\vec{\gamma}$ digamos $C_a = v_1 \rightarrow v_2, v_2 \rightarrow v_3, \dots, v_k \rightarrow v_1$ pero ello implica por el lema 4.3 que en γ el circuito $C_b = v_1 \rightarrow v_k, \dots, v_3 \rightarrow v_2, v_2 \rightarrow v_1$ es tal que $\langle M / C_b \rangle = 0$ con lo que M es un marcaje vivo-seguro !!.



Si denotamos $\beta(G)$ como el total de orientaciones acíclicas de una gráfica G dada, podemos proceder a demostrar la siguiente:

Proposición 5.14: Dada una MGS y su G_{Σ} asociada entonces $\alpha(MGS) = \beta(G_{\Sigma})$.

Demostración:

1) Supongamos que $\alpha(MGS) > \beta(G_{\Sigma})$, con lo que debe haber al menos un marcaje vivo-seguro para la MGS que no puede ser representado por alguna orientación acíclica de la G_{Σ} . Sea M tal marcaje, entonces si obtenemos una gráfica dirigida G a partir de la MGS primero mediante

la eliminación de las aristas que no contienen fichas en M y segundo por medio de la eliminación de fichas de las aristas con éstas, es directo que G es una orientación acíclica de G_{Σ} y que $\alpha(MGS) > \beta(G_{\Sigma})$ es una evidente contradicción.

ii) Sea pues que $\alpha(MGS) < \beta(G_{\Sigma})$ entonces debe haber una orientación β_1 de la G_{Σ} que no corresponde a un marcaje vivo-seguro de la MGS. Sin embargo si tomamos cada arista orientada $e_i = (u, v)$ de la G_{Σ} con β_1 y hacemos $\mu(e_i) = 1$ en la MGS obtenemos un marcaje M que cumple con que para todo circuito dirigido C de la MGS, $\langle M | C \rangle > 0$ (ya que de otra forma β_1 sería una orientación cíclica). Además como entre cada par de aristas antiparalelas una contiene una marca y la otra no (ya que es imposible que en una orientación una misma arista este dirigida en ambas direcciones) tenemos que $\alpha(MGS) < \beta(G_{\Sigma})$!.

■

De la proposición 5.14 podemos concluir que el problema de hallar el número de marcajes vivos-seguros para una MGS dada es el mismo que el de hallar el total de orientaciones acíclicas para su G_{Σ} asociada, y en particular el polinomio cromático de una gráfica es primordial en este sentido.

Definición 5.13: El polinomio cromático de una gráfica G representado como $\rho_G(k)$ denota el número de k -coloreos para G .

La información que proporciona $\rho_G(k)$ se refiere en términos generales a que dado un conjunto específico de colores c_1, c_2, \dots, c_k de cuantas diferentes maneras es posible colorear los vértices de la gráfica empleando únicamente dichos colores.

Ejemplo 5.4

Sea la gráfica ilustrada en la figura 5.9. ¿ De cuántas maneras es posible colorear la gráfica empleando para ello solamente k colores ?.

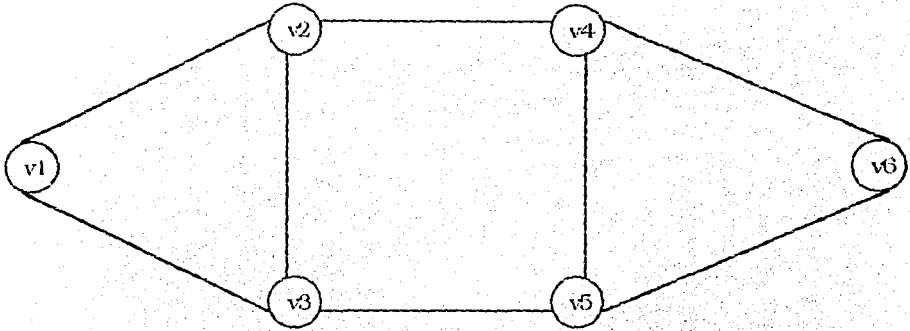


Figura 5.9

Para encontrar una función de $\rho_G(k)$ vale la pena distinguir un par de casos, el primero ocurre cuando v_2 y v_5 son del mismo color, y el segundo se refiere a aquellos coloreos en los que dichos vértices son de distinto color, así podemos totalizar el conteo de cada uno de estos casos.

Si v_2 y v_5 son de distinto color podemos entonces obtener una nueva gráfica G_1 a partir de la original adicionando la arista $e = (v_2, v_5)$ y es fácil ver que el coloreo de G_1 con k colores es igual al de la gráfica original con v_2 y v_5 de distinto color. Si procedemos como se muestra en la figura 5.10 tenemos que $\rho_{G_1}(k) = k(k-1)(k-2)^4$.

Si v_2 y v_5 son del mismo color podemos obtener una gráfica G_2 a partir de la gráfica original G , mezclando dichos vértices de tal forma que en G_2 son el mismo vértice que es adyacente a cualquier vértice vecino de v_2 o v_5 en G . Dicho proceso es mostrado en la figura 5.11.

De este modo, también es fácil ver que el coloreo de G_2 con k colores es igual al coloreo de G con v_2 y v_5 del mismo color, y procediendo del mismo modo que con G_1 tal y como se presenta en la figura 5.12, $\rho_{G_2}(k) = k(k-1)^2(k-2)^2$.

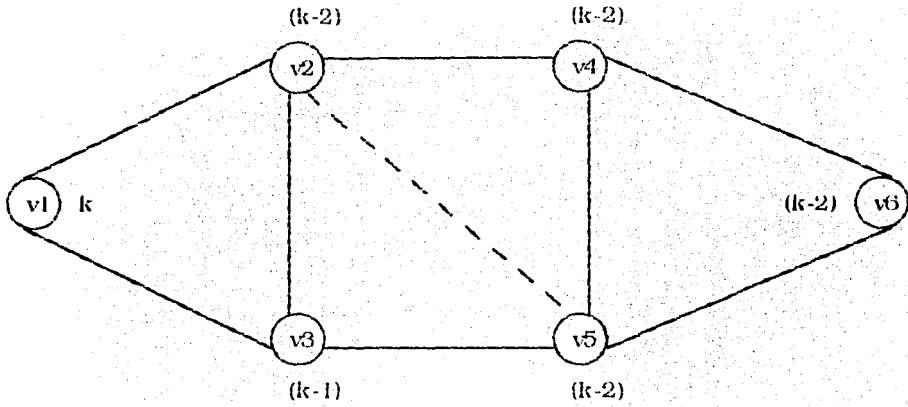


Figura 5.10

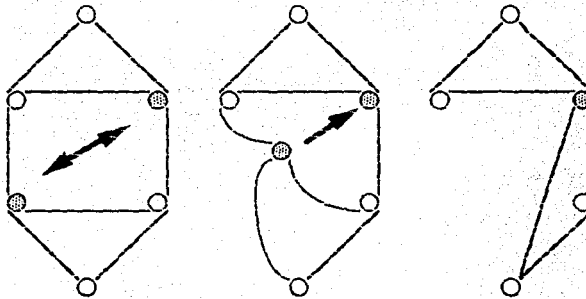


Figura 5.11

Entonces como $\rho_G(k) = \rho_{G_1}(k) + \rho_{G_2}(k)$ tenemos que:

$$\rho_G(k) = k(k-1)(k-2)^4 + k(k-1)^2(k-2)^2$$

$$\rho_G(k) = k(k-1)^2(k^3 - 3k + 3)$$

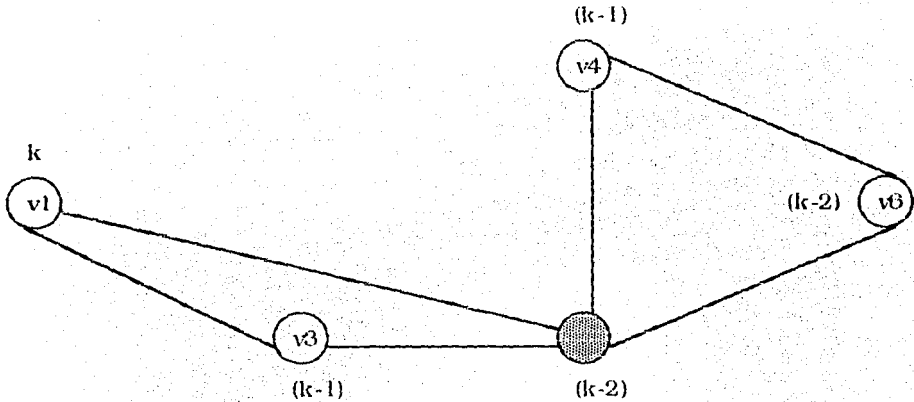


Figura 5.12

Y $\rho_G(k)$ es el total de formas en que se puede colorear G con k colores.

Proposición 5.15: Para toda gráfica G , $\rho_G(k)$ es un polinomio en k .

Demostración:

Sea ϕ el número de aristas que le faltan a G para ser una gráfica completa con $|V(G)|$ vértices. Por inducción sobre ϕ .

Paso base: Si $\phi = 0$ entonces la gráfica es completa y cada vértice debe ser de diferente color con lo que:

$$\rho_G(k) = k(k-1)(k-2)\dots(k-|V|+1)$$

que es un polinomio en k .

Paso de inducción:

HIPOTESIS: $\rho_G(j)$ es un polinomio en j , $\forall 0 \leq j \leq \phi - 1$.

Para ϕ tenemos que $\phi > j \geq 0$ con lo que $\phi > 0$, y por lo tanto al menos hace falta una arista $e = (u, v)$ para que la gráfica sea completa. Sea G_1 la gráfica obtenida a partir de G añadiendo dicha arista y sea G_2 la gráfica obtenida a partir de G mediante la unión de los vértices u y v (como en el ejemplo 5.4) en un sólo vértice, de tal forma que este último es adyacente a todos los nodos vecinos de u o v en la gráfica original G .

Claramente

$$\rho_G(k) = \text{número de maneras de colorear } G \text{ con } u \text{ y } v \text{ de distinto color} \\ + \\ \text{número de maneras de colorear } G \text{ con } u \text{ y } v \text{ del mismo color}$$

Entonces $\rho_G(k) = \rho_{G_1}(k) + \rho_{G_2}(k)$, y debido a que G_1 y G_2 tienen un número menor a ϕ de aristas necesarias para ser gráficas completas por hipótesis $\rho_{G_1}(k)$ y $\rho_{G_2}(k)$ son polinomios en k y \therefore su suma también es un polinomio en k .

[B93] ■

Lema 5.2: Para toda gráfica simple $G = (V, E)$ con polinomio cromático ρ_G el total de orientaciones acíclicas de G es exactamente $\beta(G) = (-1)^{|V|} \rho_G(-1)$.

Prueba:

Por inducción sobre el número de aristas ϕ que le faltan a G para ser una gráfica completa.

Paso base: Si la gráfica es completa ($\phi = 0$) por las proposiciones 5.3 y 5.14 $\beta(G) = |V|!$ y claramente $\rho_G(k) = k(k-1)(k-2)\dots(k-|V|+1)$ en donde hay un par de casos.

i) Si $|V|$ es par entonces $(-1)^{|V|} = 1$ y $\rho_G(-1) = (-1)(-2)\dots(-|V|) = |V|!$.

ii) Si $|V|$ es impar entonces $(-1)^{|V|} = -1$ y $\rho_G(-1) = (-1)(-2)\dots(-|V|) = -|V|!$ con que $(-1)^{|V|} \rho_G(-1) = |V|!$.

Paso de inducción:

HIPOTESIS: $\forall 0 \leq j \leq \phi - 1$ el lema se cumple.

Para $\phi > 0$ debe haber al menos una arista $e = (u, v)$ que no está en la gráfica para que ésta sea completa, y sean G_1 y G_2 las gráficas obtenidas a partir de G por medio de la adición de $e = (u, v)$ para la primera y mediante la unión de u y v en un sólo vértice que es adyacente a todos los vértices vecinos de u o v en la gráfica original (como en la proposición 5.15) para la segunda.

Podemos ver que $\beta(G_1)$ puede ser de dos tipos. El primer tipo se refiere a aquellas orientaciones en las que al invertir la dirección de la arista $e = (u, v)$ se obtiene otra orientación acíclica para la gráfica. Evidentemente el segundo tipo corresponde a aquellas orientaciones en las que si se invierte la dirección asignada a la arista en cuestión se obtiene una orientación cíclica. Si $\beta_1(G_1)$ y $\beta_2(G_1)$ son el total de orientaciones acíclicas para el primero y segundo de los tipos respectivamente, entonces $\beta(G_1) = \beta_1(G_1) + \beta_2(G_1)$.

Por otra parte, si en cada una de las $\beta(G_2)$ orientaciones acíclicas para G_2 separamos los vértices u y v en un procedimiento inverso al que se empleó para unir los vértices y si adicionamos la arista $e = (u, v)$ podemos asignar cualquier dirección a la arista y la orientación resultante es acíclica. Para demostrar este hecho supongamos lo contrario, y sin pérdida de generalidad asignemos la dirección $e = (u, v)$ con la que se forma un circuito $C = u \xrightarrow{e} v, v \xrightarrow{e_1} w, \dots, w' \xrightarrow{e_2} u$ pero los vértices w y w' son adyacentes a u y v con lo que en G_2 al momento de asignar direcciones a las aristas y dado que u y v forman un sólo vértice, a decir h entonces debe haber un circuito $C_h = h \xrightarrow{e} w, \dots, w' \xrightarrow{e_2} h$ y podemos concluir que $\beta_1(G_1) = 2\beta(G_2)$.

Finalmente, hay que notar que cada orientación acíclica de G es tal que si consideremos la adición de la arista $e = (u, v)$ entonces hay un par de casos:

i) La asignación de una dirección a dicha arista puede producir una orientación acíclica, entonces se puede considerar la dirección contraria y si está también produce un circuito es directo que la orientación de G es cíclica.

ii) La asignación de cualquiera de las direcciones a la arista no produce un circuito, y esto es debido a que todas las aristas incidentes a u y v en G están orientadas de salida o de entrada en su conjunto a ambos vértices.

Evidentemente, el primer caso corresponde a la mitad de las orientaciones acíclicas de $\beta_1(G_1)$ y el segundo es exactamente igual a $\beta_2(G_1)$ con lo que:

$$\beta(G) = \frac{1}{2}\beta_1(G_1) + \beta_2(G_1)$$

es decir que

$$\beta(G) = \beta_1(G_1) - \frac{1}{2}\beta_1(G_1) + \beta_2(G_1)$$

pero como

$$\beta(G_1) = \beta_1(G_1) + \beta_2(G_1) \quad \text{y} \quad \beta(G_2) = \frac{1}{2}\beta_1(G_1)$$

entonces

$$\beta(G) = \beta(G_1) - \beta(G_2)$$

De la forma en que fueron construidas las gráficas y por hipótesis tenemos que

$$\beta(G_1) = (-1)^{|M|} \rho_{G_1}(-1) \quad \text{y} \quad \beta(G_2) = (-1)^{|M|-1} \rho_{G_2}(-1)$$

entonces

$$\beta(G) = (-1)^{|V|} [\rho_{G_1}(-1) + \rho_{G_2}(-1)]$$

Y de la demostración de proposición 5.15 sabemos que

$$\rho_G(k) = \rho_{G_1}(k) + \rho_{G_2}(k)$$

y por lo tanto

$$\beta(G) = (-1)^{|V|} \rho_G(-1)$$

[S73] ■

Corolario 5.3: Dada una MGS cualquiera, $\alpha(MGS) = (-1)^{|V|} \rho_{G_\Sigma}(-1)$ donde G_Σ es la gráfica subyacente simple asociada con la MGS.

Demostración:

Consecuencia directa de la proposición 5.14 y del lema 5.2.

■

Finalmente, es claro que el número de rondas posibles de asignar (libres de interbloqueo y de espera infinita) a una calendarización para un SD modelado con una gráfica $G = (V, E)$, es $(-1)^{|V|} \rho_G(-1)$ donde $\rho_G(k)$ es el polinomio cromático de la gráfica.

Conclusiones

El problema de encontrar un marcaje vivo para una MG se resuelve mediante un algoritmo que consiste en poner una ficha en cada una de las aristas.

Por otra parte el problema de que dada una MG cualquiera, encontrar un marcaje vivo-seguro para tal gráfica, es resuelto por medio de una técnica sugerida por F. Commoner et al. [CHEP71]. Esta técnica consiste en que para cada arista se encuentra el circuito con menor conteo de fichas en el que dicha arista participa. Después a través de una secuencia legal de disparos esas fichas son llevadas a la arista en cuestión, y dejando únicamente una ficha en la arista se continúa con otra arista. Continuando así, finalmente se termina de revisar a todas y cada una de las aristas. La complejidad computacional de esta solución es $O(|V|^2|E|)$, empleando una variación del algoritmo de Dijkstra para encontrar el circuito con menor número de fichas.

Uno de los resultados del análisis del algoritmo para obtener marcajes vivos-seguros es que una MG puede tener un marcaje vivo-seguro si y sólo si la digráfica G de la MG es fuertemente conexa.

Una gráfica marcada simétrica MGS es una MG en la que para todo par de vértices distintos ocurre que existe un par de aristas antiparalelas entre ellos o bien ambos vértices no son adyacentes, y adicionalmente no existen bucles. Una de las aplicaciones inmediatas de este modelo consiste en el estudio de calendarizaciones de sistemas distribuidos SD . Una calendarización de un SD consiste de una secuencia de rondas (subconjuntos de unidades computacionales no adyacentes) que tienen permiso de usar las líneas de comunicación en ese momento.

La implicación teórica que tienen los marcajes vivos-seguros respecto a las calendarizaciones, tiene que ver directamente con la obtención de rondas tales que no exista interbloqueo, ni espera infinita (factores ambos relacionados con el funcionamiento correcto y las propiedades deseables en una calendarización).

Un algoritmo para obtener marcajes vivos-seguros de menor complejidad (que el algoritmo general propuesto por F. Commoner) es estudiado, con la restricción de que únicamente es aplicable a *MGS*. La idea del algoritmo consiste en ir revisando cada uno de los vértices y para cada una de sus aristas de salida poner una o ninguna ficha dependiendo de si dicha arista va a un vértice que no se ha visitado o se ha visitado con anterioridad. La complejidad de este algoritmo es lineal en el número de aristas, es decir que la complejidad es $O(E)$.

Finalmente el problema de calcular el total de marcajes vivos-seguros posibles para una *MGS* es estudiado y la solución es expresada en términos del polinomio cromático de la gráfica subyacente simple de la *MGS*, tal resultado es $(-1)^{|M|} \rho_G(-1)$, que es igual al resultado de R. P. Stanley [573] respecto al número de orientaciones acíclicas de una gráfica.

Las posibles direcciones de investigación que pueden continuarse son en particular dos. La primera de ellas se refiere a buscar un método que permita reducir una *MG* a una *MGS*, para a partir de esta, calcular el número de marcajes vivos-seguros de la *MG* o bien emplear el algoritmo de menor complejidad para obtener un marcaje vivo-seguro y regresar esta posteriormente a su estado original. La segunda dirección se refiere a calcular el total de marcajes vivos-seguros alcanzables desde otro marcaje también vivo-seguro.

Bibliografía

- [A85] B. Awerbuch (1985), "**Complexity of Network Synchronization**", Journal of the ACM, Vol 34, No 4, pp. 804-823.
- [AF73] T. Agerwala, M. Flynn (1973) "**Comments on capabilities, limitations and correctness of Petri Nets**", Proc. 1st Annual Symposium on Computer Architecture (ACM), pp. 81-86.
- [AHU74] A. Aho, J. Hopcroft, J. Ullman (1974), **The Design and Analysis of Computer Algorithms**, Addison-Wesley.
- [BG89] V.C. Barbosa, E. Gafni (1989) "**Concurrency in Heavily Loaded Neighborhood-Constrained Systems**", ACM Transactions on Programming Languages and Systems, Vol 11, No 4, pp 562-584.
- [B93] Bryant, V. (1993), **Aspects of Combinatorics A Wide-Ranging Introduction**, Cambridge University Press.
- [CHEP71] F. Commoner, A.W. Holt, S. Even, A. Pnueli (1971) "**Marked Directed Graphs**", Journal of Computer and System Sciences, Vol 5, No 5, pp. 511-522.
- [E79] S. Even (1979) **Graph Algorithms**, Computer Science Press.
- [EY83] T. Etzion, M. Yoeli (1983) "**Super-Nets and their Hierarchy**", Theoretical Computer Science, Vol 23, pp. 243-272.
- [GJ79] M. Garey, D. Johnson (1979), **Computers and Intractability: A Guide to the Theory of NP-Completeness**, W. H. Freeman, New York.

- [MMZ94] Y. Malka, S. Moran, S. Zaks (1994) "**A Lower Bound on the Period Length of a Distributed Scheduler**", Algorithmica, No 10, pp. 383-398.
- [R94] S. Rajsbaum (1994) "**Upper and Lower Bounds for Stochastic Marked Graphs**", Information Processing Letters 49, pp 292-295.
- [R85] W. Reising (1985), **Petri Nets: an Introduction**, Springer verlag.
- [S73] R. P. Stanley (1973) "**Acyclic Orientations of Graphs**", Discrete Mathematics, Vol 5, pp. 171-178.
- [S74] R. P. Stanley (1974) "**Combinatorial Reciprocity Theorems**", Advances in Mathematics, Vol 14, pp. 194-253.
- [TM87] J.P. Tremblay, R. Manohar (1987), **Discrete Mathematical Structures with Applications to Computer Science**, McGraw-Hill.
- [T87] M. Townsend (1987), **Discrete Mathematics: Applied Combinatorics and Graph Theory**, Benjamin-Cummings.
- [W85] R. J. Wilson (1985), **Introduction to graph theory**, Longman.
- [YB77] M. Yoeli, Z. Barzilai (1977) "**Behavioural descriptions of communication switching systems using extended Petri Nets**", Digital Processes, Vol 3, pp. 307-320.